UNIVERSITAT DE BARCELONA

Final Degree Project

Biomedical Engineering Degree

Development of an Android mobile application to monitor and control daily health of users

Barcelona, 23rd January 2023
Author: Beatriz Bachs
Director/s: Dr. Jordi Colomer Farrarons

ABSTRACT

Non-communicable diseases (NCDs), also known as chronic or long-term diseases, are the result of a combination of genetic, physiological, environmental and behavioural factors that, once initiated, can last a lifetime. The main types are cardiovascular diseases, cancers, chronic respiratory diseases (such as chronic obstructive pulmonary disease, COPD and asthma) and diabetes.

In this context, the Agencia de Salud Pública de Cataluña (ASPCAT) initiated several cancer prevention and health promotion programmes some years ago, which are promoted through primary health care.

In this sense, and with the intention of promoting prevention and facilitating the daily lives of these patients, the mobile application Applud has been created. This application aims to be a tool that helps middle-aged people to keep track of their health, both for patients with chronic diseases and for the healthy population at risk of developing one of these diseases.

Table of Contents

1. Introduction

Non-communicable diseases (NCDs), also known as chronic or long-term diseases, are the result of a combination of genetic, physiological, environmental and behavioural factors that, once initiated, can last a lifetime. The main types are cardiovascular diseases, cancers, chronic respiratory diseases (such as chronic obstructive pulmonary disease, COPD and asthma) and diabetes.

As indicated by Instituto Nacional de Estadística (INE)[i], 55% of deaths in 2020 were due to chronic diseases (non-communicable diseases), specifically of the circulatory system, tumours and diseases of the respiratory system.

In Catalonia, 36’2% of the population suffers from some long-term illness or health problem[ii]. It is clear that the presence of chronic health problems increases with age. From the age of 55 onwards, approximately 60%[iii] of the Catalan population in each age range, both women and men, suffer from some illness of this type. This graph is shown below.

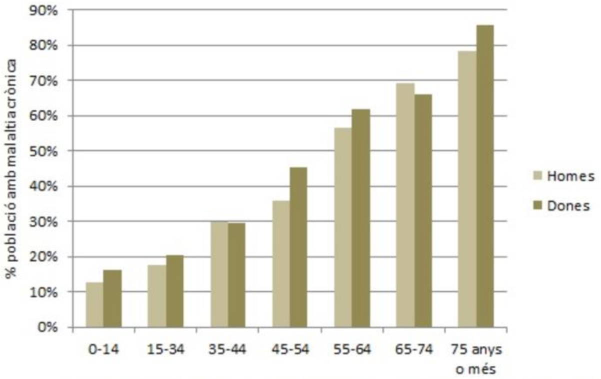


Figure 1. Prevalence of disease or chronic health problem by age and sex

Given the high cost of NCDs and their population-wide impact, an integrated preventive approach to all their determinants or risk factors is particularly important. Evidence has shown that disease prevention and health promotion strategies are highly effective in reducing the incidence of NCDs.

In this context, the Agencia de Salud Pública de Cataluña (ASPCAT) initiated several cancer prevention and health promotion programmes some years ago, which are promoted through primary health care. The programmes are focused on encouraging people to do sport, helping them to stop smoking, controlling the amount of alcohol they drink and promoting a balanced diet. In this way, the main risk factors are avoided: overweight, poor diet, sedentary lifestyle, alcoholism and tobacco consumption.

In this sense, and with the intention of promoting prevention and facilitating the daily lives of these patients, the mobile application Applud has been created. This application aims to be a tool that helps middle-aged people to keep track of their health, both for patients with chronic diseases and for the healthy population at risk of developing one of these diseases.

The application focuses on people between 50-75 years of age because it is at this age that people start to attend health centres more frequently and have the first stable contacts with professionals. They start to have their blood tests, cholesterol and weight checked and the years of sedentary life take their toll.

The application also aims to create synergies and facilitate communication between health professionals and users and make health care more accessible and closer to them. It is for this reason that one of the main functionalities of the application will be to allow access to both the public and private healthcare system. It is also for this reason that it has been decided to develop a mobile application and not a web application.

1.1 Objectives

The aim of this final degree project is, therefore, to carry out a study on the functional and economic viability of Applud. To this end, the following objectives have been defined:

- Identifying the daily needs of people suffering from chronic illnesses
- Conduct market research to detect unmet needs of target users
- Selecting those needs that the Applud mobile application can meet
- Design a prototype aimed at a good user experience
- Develop the front end of the prototype for mobile phones with Android operating system and Spanish language

In order to achieve the ultimate goal of the work, which is to develop the prototype of a mobile application, the process that is followed in any mobile application development will be followed.

First of all, the strategy to be followed will be defined by carrying out a situation and market analysis. These two aspects are dealt with in sections “2. Theoretical background” and “3. Market analysis”, respectively. The system requirements and specifications will then be defined in section “4. Conceptual engineering”. In this section the design of the application will also be carried out. Finally, and only after all the above steps have been taken, the implementation will start.

Every time a new functionality is developed, the entire application is tested. This process is repeated iteratively until all the functionalities are finally implemented and the application can go to market.

In this work, only the first iteration of the process will be carried out, as the objective is to study the functional and economic viability of the application and not to publish the application on the market for free use by users.

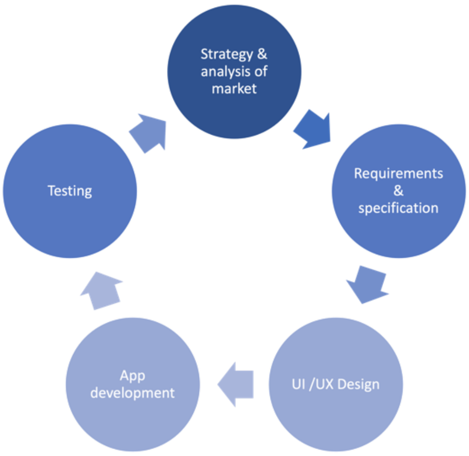


Figure 2. Development phases of a mobile application

2. Theoretical background

2.1 State of the art

An app, which is short for "application," is a type of software that can be installed and run on a computer, tablet, smartphone or other electronic devices. An app most frequently refers to a mobile application or a piece of software that is installed and used on a computer. Most apps have a specific and narrow function.

2.1.1 Types of app

There are a few different types of apps available for download by businesses and individuals alike. The following are the most recognized apps in use today[iv]:

- Web-based: A web-based app is an application that requires internet access for complete use. These types of apps are coded in JavaScript, HTML5 or CSS. A web-based app typically requires a much smaller amount of memory space in a user's device because the databases are stored on the Internet server. Examples of web-based apps include Netflix, Google Docs and Dropbox.

- Native app: Apps that are created for a certain mobile platform are known as native applications. For example, an app that is made for an iPhone will only be usable on Apple devices and would not work on other types of mobile phones, such as Android. These types of apps are primarily used to provide the highest performance on a particular mobile operating system.

- Hybrid app: A hybrid app is an app that is made to support both native and web-based technologies and is a combination of both web-based and native applications. These apps are easier and quicker to create and only use a single code base that can be integrated across various platforms. However, it's important to note that hybrid apps often have a lower performance rate than native or web-based apps.

2.1.2 Mobile apps

When developing an application, it is important to be clear on which device it will be used and how users will access it. This will make the approach in one case or another different[v]. For example, if a web application is to be developed, a different language is used than if a mobile application is to be developed; one is accessed from a browser, which executes HTML code, and the other uses Java, Kotlin or Swift, among others[vi]. The operating system is also another factor to consider.

2.1.2.1 Operating systems

The operating system (OS) allows users to perform the basic functions of a computer[vii]. The OS manages all software and peripheral hardware and accesses the central processing unit (CPU) for memory or storage purposes. It also makes it possible for a system to simultaneously run applications. All PCs, laptops, tablets, smartphones, and servers require an OS. Developers may use specific operating systems that are more conducive to programming and application development, while the average employee will likely use a proprietary system for more common, everyday usage. There are different operating systems depending on whether the device is a mobile phone-tablet or computer[viii].

The most common operating systems that exist for computers or tablets are Windows, Apple OS, and Linux[ix].

Mobile phones also have an operating system[x]. This software allows mobile devices like phones, tablets and other smart devices like wearable technology to run applications and other programs. Most mobile operating systems only work on specific hardware. The most popular operating systems used by most mobile phones are iOS and Android, marketed by Apple and Google respectively[xi]. iOS is only used on Apple devices, but Android is an open-source operating system provided by Google to most hardware manufacturers, such as Samsung, Motorola, Xiaomi, Huawei and others[xii].

However, since the US government vetoed Huawei in 2019, Huawei can use Android as an operating system, as it is open-source, but it cannot use Google's services. This is why Huawei developed and updated its own operating system, called Harmony, which is used by most Asian smartphones, and launched several of its own apps to replace those marketed by Google.

It is worth noting that most manufacturers use Android as their OS, but make modifications to the code to adapt it to their needs. The phones manufactured by Google, the Pixels, are the only ones that use pure Android as their OS.

Although Android is the most widely used OS for all mobile phones, many manufacturers have also developed their own OS that they release on a few devices. This is the case of Samsung with Tizen, Windows with Microsoft or Nokia with Symbian OS.

2.1.2.2 Programming languages for mobile app development

Mobile applications are programmed using one language or another depending on the operating system of the device.

For iOS devices, including smartphones, iPads and tv, the two main languages used are Objective-C and Swift[xiii]. Each of them has both advantages and disadvantages, suitable for certain tasks. All of them relate to object-oriented programming.

- Objective-C: Objective-C is a superset of the C programming language, which means that it includes all of the features of C, plus some additional ones — such as dynamic typing, method overloading, data encapsulation, and automatic memory management. Objective-C is easy to learn for developers who are already familiar with C or C++, because it's built on top of the C language. However, differs in syntax from other programming languages, so it is pretty hard for beginners to learn.

- Swift: Swift is the modern best programming language for iOS that incorporates the best C code and Objective-C languages and is used a lot for today's software projects. To inheritance, polymorphism, encapsulation, and abstraction, many modern functions have been added: closures, generics, and others. The programming process for iOS has become more flexible and so much fun. One advantage it has is that it is a compiled language, so it is fast and efficient.

In the case of android, the most used languages are[xiv]:

- Java: Java was the official language for Android App Development, now it is replaced by Kotlin, and consequently, it is the most used language as well. Many of the apps in the Play Store are built with Java, and it is also the most supported language by Google. In addition to all this, Java has a great online community for support in case of any problems.

However, Java is a complicated language for a beginner to use as it contains complex topics like constructors, null pointer exceptions, concurrency, checked exceptions, etc. Also, The Android Software Development Kit (SDK) increases the complexity to a new level, but once you have learned it, you have the basics of object-oriented programming, but once learned it, you have the basics of object-oriented programming.

- Kotlin: Kotlin is the official language for Android App Development declared by Google in 2019[xv]. Kotlin is a cross-platform programming language that may be used as an alternative to Java for Android App Development. It has also introduced as a secondary "official" Java language in 2017. Kotlin can interoperate with Java and it runs on the Java Virtual Machine. The only sizable difference is that Kotlin removes the superfluous features of Java such as null pointer exceptions. It also removes the necessity of ending every line with a semicolon. In short, Kotlin is much simpler for beginners to try as compared to Java and it can also be used as an "entry point" for Android App Development.

  However, this language is not very widespread, as Java is a language that is used in many programming environments not only for mobile application development, and Kotlin is only used in mobile application development for Android.

2.1.3 Android app development

To develop a mobile application for Android, as in the case of this project, it is simply necessary to have knowledge of one of the programming languages explained above and to know how to use the Android Studio IDE[xvi].

Android Studio[xvii] is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux-based operating systems. It is used as the primary IDE for developing native Android applications. Prior to its release in 2014, Eclipse Android Development Tools were used to develop android applications. Although this IDE is the best-known IDE for development, you can still use Eclipse or Netbeans, for example.

There are many ways in which Android Studio differs from other IDEs and development tools. Some of these differences are quite subtle, such as the way support libraries are installed, and others, for instance, the build process and the UI design, are profoundly different.

Before taking a closer look at the IDE itself, it is a good idea to first understand what some of these important differences are and why it is the first choice of programmers when it comes to development. The major ones are listed here[xviii]:

- UI development: The most significant difference between Studio and other IDEs is its layout editor, which is far superior to any of its rivals, offering text, design, and blueprint views, and most importantly, constraint layout tools for every activity or fragment, an easy-to-use theme and style editors, and a drag-and-drop design function. The layout editor also provides many tools unavailable elsewhere, such as a comprehensive preview function for viewing layouts on a multitude of devices and simple-to-use theme and translation editors.

- Project structure: Although the underlying directory structure remains the same, the way Android Studio organizes each project differs considerably from its predecessors. Rather than using workspaces as in Eclipse, Studio employs modules that can more easily be worked on together without having to switch workspaces.

- Code completion and refactoring: The way that Android Studio intelligently completes code as you type makes it a delight to use. It regularly anticipates what you are about to type, and often a whole line of code can be entered with no more than two or three keystrokes. Refactoring too, is easier and more far-reaching than alternative IDEs, such as Eclipse and NetBeans. Almost anything can be renamed, from local variables to entire packages.

- Emulation: Android Studio comes equipped with a flexible virtual device editor, allowing developers to create device emulators to model any number of real-world devices. These emulators are highly customizable, both in terms of form factor and hardware configurations, and virtual devices can be downloaded from many manufacturers. This way you can test what you have developed without having to buy a physical device, and it also allows you to test it on different devices.

- Build tools: Android Studio employs the Gradle build system, which performs the same functions as the Apache Ant system that many Java developers will be familiar with. It does, however, offer a lot more flexibility and allows for customized builds, enabling developers to create APKs that can be uploaded to TestFlight, or to produce demo versions of an app, with ease. It is also the Gradle system that allows for the modular nature. Rather than each library or a third-party SDK being compiled as a JAR file, Studio builds each of these using Gradle.

Another very useful tool used by all software developers, whether they work on the backend or the frontend, is Git[xix]. By far, Git is the most widely used modern version control system in the world. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development using different code "branches". Linus Torvalds created Git in 2005 for the development of the Linux kernel.

Git can be used with commands that are executed through the terminal or through graphical user interfaces (GUIs). As Git[xx] is a very powerful tool and widely used throughout the industry, several customers have created GUIs for ease of use. The most important ones are GitHub, GitLab, Sourcetree or QGit, among others.

Git[xxi] works through repositories. It can be said that a repository is a code project that one or more developers work on. Repositories are stored in the cloud (remote repository) and can be downloaded to each developer's computer (local repository).

Developers can work in parallel on the same repository by creating different branches (or copies of the repository) in which they make their changes, each time they want to develop a new feature or fix a defect. To do this, they create a local copy of the remote repository on their machine. Once the development is done, they add their changes to the remote repository so that the whole team can have access.

The following image shows a schematic diagram that illustrates the above. The main Git commands can be found in the annex.

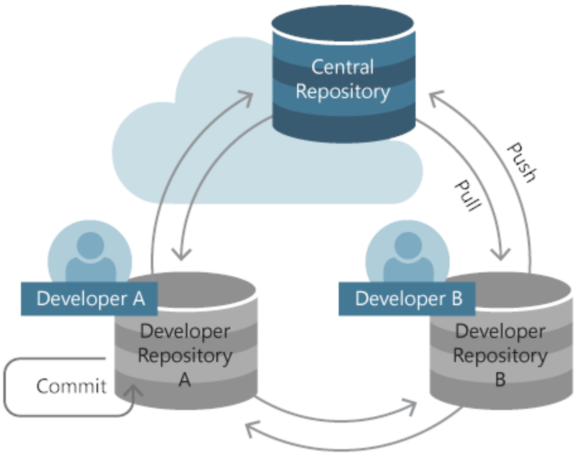


Figure 3. Remote repository and local Git repositories

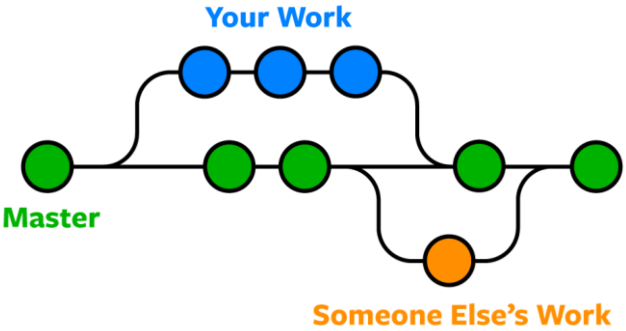


Figure 4. How Git version control works through the branches that are created

2.2 State of the situation

In this section we have consulted the studies carried out by the Catalan Public Health System to present the state of the situation. The information published by this entity is very extensive and has a globalised vision, as it takes into account the population of the entire Catalan territory and its professionals.

In December 2021, the Catalan Public Health System published the health plan for the years 2021-2025[xxii]. This document details the objectives and strategies for improving the system in this autonomous community after an extensive study of the situation.

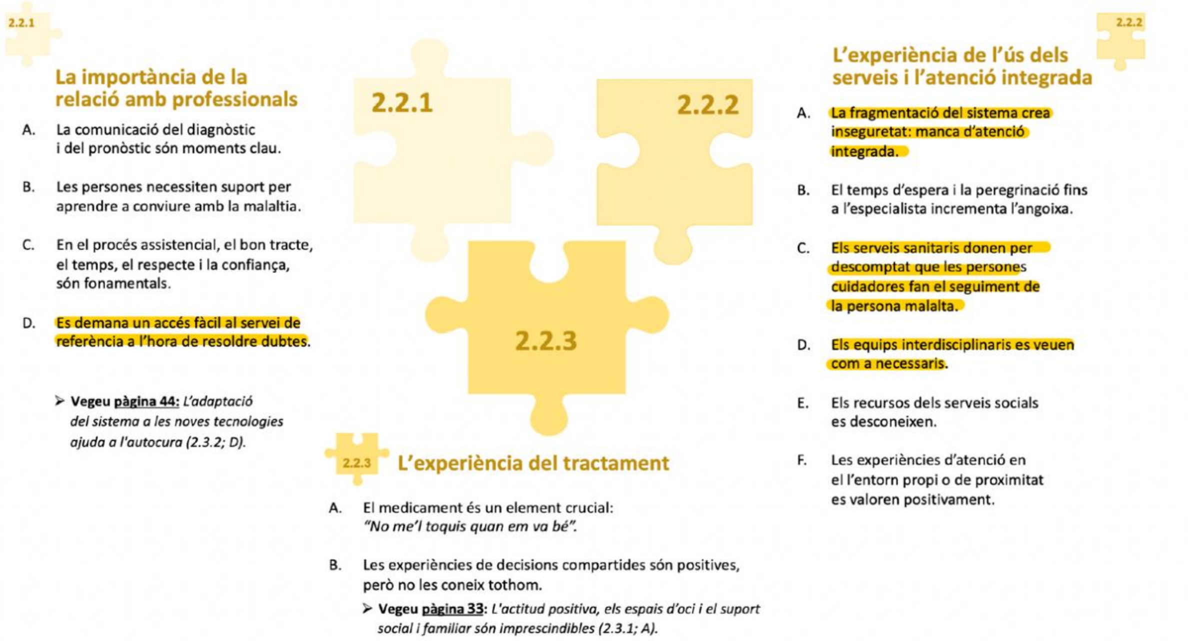
This document indicates that the main cause of death in women are diseases of the circulatory system and in men are tumours[xxiii]. The second cause in both cases is COVID-19 followed by other chronic diseases.

This is why two of its nine health priorities are aspects that have to do with chronicity and are as follows:

- Lifestyles and health risk factors
- Chronicity, complexity and fragility

In another document entitled *"La visió de les persones sobre l'atenció a la gent gran i a la cronicitat"*[xxiv], and as a result of the line initiated by the Department of Health to promote people's participation in health policies, the vision of healthy people and people with chronic illnesses on care for the elderly and chronicity is presented.

This study involved people with complex chronic diseases, carers, people with mild or moderate chronic diseases and healthy people from different parts of Catalonia and from different social backgrounds. Several conclusions have been drawn from the work with the 96 people who participated in the study and are published in the document. The following image is taken from this document and shows a summary of the contents of the care that the people who participated in the study would like to receive.



2.2.1 La importància de la relació amb professionals

A. La comunicació del diagnòstic i del pronòstic són moments clau.

B. Les persones necessiten suport per aprendre a conviure amb la malaltia.

C. En el procés assistencial, el bon tracte, el temps, el respecte i la confiança, són fonamentals.

D. Es demana un accés fàcil al servei de referència a l'hora de resoldre dubtes.

➢ **Vegeu pàgina 44:** *L'adaptació del sistema a les noves tecnologies ajuda a l'autocura (2.3.2; D).*

2.2.2 L'experiència de l'ús dels serveis i l'atenció integrada

A. La fragmentació del sistema crea inseguretat: manca d'atenció integrada.

B. El temps d'espera i la peregrinació fins a l'especialista incrementa l'angoixa.

C. Els serveis sanitaris donen per descomptat que les persones cuidadores fan el seguiment de la persona malalta.

D. Els equips interdisciplinaris es veuen com a necessaris.

E. Els recursos dels serveis socials es desconeixen.

F. Les experiències d'atenció en el l'entorn propi o de proximitat es valoren positivament.

2.2.3 L'experiència del tractament

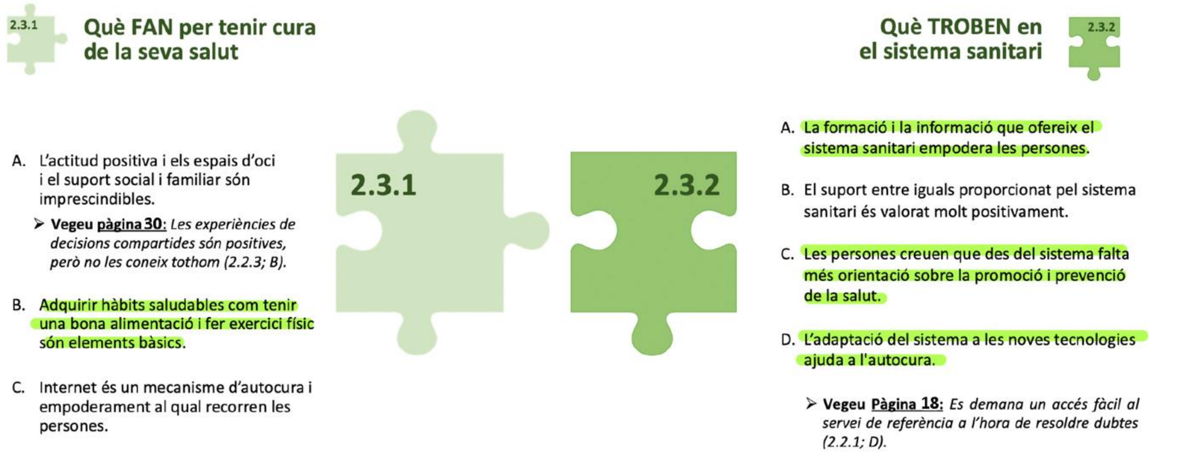
A. El medicament és un element crucial: *"No me'l toquis quan em va bé".*

B. Les experiències de decisions compartides són positives, però no les coneix tothom.

➢ **Vegeu pàgina 33:** *L'actitud positiva, els espais d'oci i el suport social i familiar són imprescindibles (2.3.1; A).*

Figure 5. Scheme summary of the contents of the care that people suffering from a chronic disease want to receive.

Elsewhere in the document, the benefits of new technologies in health care and the need for healthy habits and behaviours are discussed.



2.3.1 Què FAN per tenir cura de la seva salut

A. L'actitud positiva i els espais d'oci i el suport social i familiar són imprescindibles.

➢ **Vegeu pàgina 30:** *Les experiències de decisions compartides són positives, però no les coneix tothom (2.2.3; B).*

B. Adquirir hàbits saludables com tenir una bona alimentació i fer exercici físic són elements bàsics.

C. Internet és un mecanisme d'autocura i empoderament al qual recorren les persones.

2.3.2 Què TROBEN en el sistema sanitari

A. La formació i la informació que ofereix el sistema sanitari empodera les persones.

B. El suport entre iguals proporcionat pel sistema sanitari és valorat molt positivament.

C. Les persones creuen que des del sistema falta més orientació sobre la promoció i prevenció de la salut.

D. L'adaptació del sistema a les noves tecnologies ajuda a l'autocura.

➢ **Vegeu Pàgina 18:** *Es demana un accés fàcil al servei de referència a l'hora de resoldre dubtes (2.2.1; D).*

Figure 6. Health care outline

All this leads us to detect aspects that the Applud application can help to cover. Based on these points, the functionalities that the application will have to cover will be detailed[xxv].

Easy access to the reference service is requested when resolving queries
Fragmentation of the system creates insecurity: lack of integrated care
Health services assume that caregivers follow up with the sick person.
Healthy habits such as good nutrition and physical exercise are basic elements of a healthy lifestyle.
The training and information provided by the health system empowers individuals
People feel that the system lacks more guidance on health promotion and prevention.
Adaptation of the system to new technologies helps self-healing.

Table 1. Summary table of the unmet needs in health care for people with a chronic disease

3. Market Analysis

An important point of the project is to be aware of existing applications that provide a similar service to the one Applud intends to provide. In this way, the strengths and weaknesses of these applications can be detected, the key functionalities that Applud can bring to users can be described and what differentiates it from its competitors can be better defined.

3.1 Similar applications

Digital distribution platforms for mobile apps, such as Play Store, Apple Store, Xiaomi's GetApps or Huawei's AppGallery, offer a wide range of apps that help to monitor and promote health[xxvi]. Most of them are focused on very specific topics such as smoking cessation, diabetes or heart disease. Of these, a sample of five has been selected that explain the market situation very well[xxvii].

- La Meva Salut: It is the official application of the Catalan Health System. From it, beneficiaries of the Catalan health system can consult and download reports and test results and the medication plan that the doctor has prescribed for them. They can also carry out procedures and make queries to professionals.

- iHealth Myvitals: It is an application developed by the company iHealth that allows the user to synchronise all iHealth devices in a single application. It provides instant access to the user's measurements and allows them to position themselves according to the targets they have set.

- Social Diabetes: This is an application for self-monitoring of blood glucose levels. More oriented towards type 1 diabetes mellitus. It allows the configuration of alerts with personalised messages according to the user's needs and also allows communication with the patient's healthcare staff.

- SmartBP: It is a blood pressure monitoring application that allows the user to record, track, analyse and share blood pressure information via iPhone/iPod touch/iPad, Apple Watch devices.

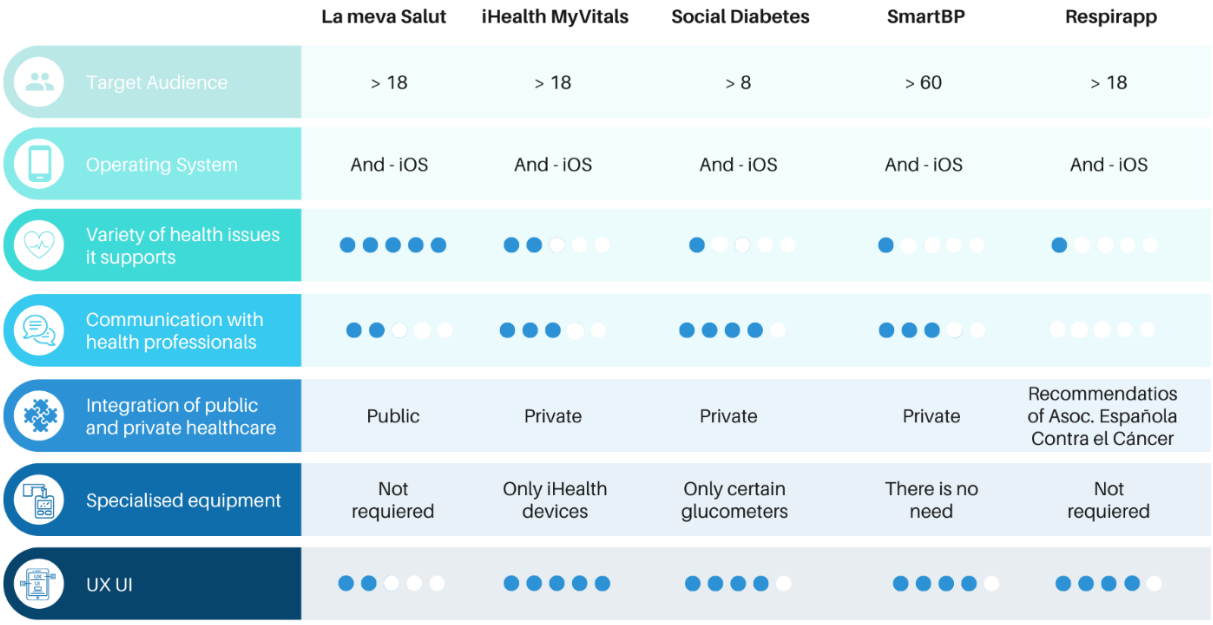
- Respirapp: It is a free application that the “Asociación Española Contra el Cáncer” launched thanks to the National Plan on Drugs. It helps users to quit smoking. It is a complete process divided into three phases that allows the user to keep a personal control of their activity, know their progress and take control in the most difficult moments.

3.2 Benchmarking

Benchmarking is a process where a company/app/project success is measured against other similar companies/apps/projects to discover if there is a gap in performance that can be closed by improving the app’s performance. Studying other companies can highlight what it takes to enhance the app’s efficiency and become a bigger player in the industry.

This section presents a competitive benchmarking. This study has been carried out with the aim of discovering the strengths of the competition and detecting market needs that Applud can satisfy.

The benchmarking has considered the variety of health issues it supports, health applications have been found that are highly specialized in one aspect, such as smoking cessation, heart disease, etc. and therefore only useful for users with very specific pathologies; whether the application enables easy and smooth communication with healthcare professionals; whether it allows contact with both private and public health care; whether a specific device is required to use the application and, of course, the user experience, i.e. whether the app is easy to use and intuitive.



	La meva Salut	iHealth MyVitals	Social Diabetes	SmartBP	Respirapp
Target Audience	> 18	> 18	> 8	> 60	> 18
Operating System	And - iOS	And - iOS	And - iOS	And - iOS	And - iOS
Variety of health issues it supports	●●●●●	●●○○○	●○○○○	●○○○○	●○○○○
Communication with health professionals	●●○○○	●●●○○	●●●●○	●●●○○	○○○○○
Integration of public and private healthcare	Public	Private	Private	Private	Recommendatios of Asoc. Española Contra el Cáncer
Specialised equipment	Not requiered	Only iHealth devices	Only certain glucometers	There is no need	Not requiered
UX UI	●●○○○	●●●●●	●●●●○	●●●●○	●●●●○

Table 2. Benchmarking of the different applications studied

- Operating system: All the applications studied are available on both iOS and Android devices. Some, such as SmartBP, even allow connection to smartwatches.

- Variety of health issues it support: On this point, all the applications except La Meva Salut receive very low scores. This is due to the fact that the apps studied are very specific in one aspect of health and have a specific functionality (diabetes, stop smoking, blood pressure). On this point, iHealth MyVitals offers a little more variety, as it allows you to connect glucometers, blood pressure monitors, scales and a smartwatch. La Meva Salut receives the highest score here, as it allows a wide variety of actions, such as consulting medical reports, the medication plan, making appointments with the public health system and contacting professionals.

- Communication with health professionals: Respirapp does not receive any score in this aspect as it does not allow any type of communication. La meva Salut allows communication but only for certain users, as it is necessary for the primary care centre to activate the service. iHealth MyVitals allows data to be shared with the professional but only through the app. SmartBP allows you to share reports of the data entered in the app with anyone as a pdf file, which can be sent by email. Social Diabetes receives the highest score here as the healthcare professional can access the user's data, track it online, adjust the guideline as they see fit and also provides a chat.

- Integration of public and private health care: La meva Salut is the official application of the public health system, so it only allows management in this area. iHealth MyVitals allows contact with professionals and access to recommendations but from the company's own staff or from those private healthcare doctors who have integrated the iHealth communication system. The same applies to Social Diabetes and SmartBP. Respirapp, although it does not have any type of communication with public or private healthcare, does receive recommendations from the association. The parameters and the method also come from this state-sponsored association.

- Specialised equipment: Both meva salut and respirapp work without devices. SamrtBP, although it is not necessary, allows the synchronization of certain specialized sphygmomanometers. Social Diabetes only works with some glucometers and the same thing happens with iHealth, which only works with the company's own glucometers and tensiometers.

- UX UI: iHealth MyVitals has an intuitive and easy-to-use design. The interface is the same for the application as for the devices that are synchronized, that is why users loyal to this brand have easy handling of all devices and it will not require a great effort to learn how to use the app, that is why it takes the highest score. Social Diabetes, SamrtBP, and Respirapp also have simple, easy-to-use designs. La meva Salut is the one who receives the lowest score in this aspect. Signing up for the application requires many steps, also, although it contains a menu where all the actions of the application are, the design is quite poor.

3.3 Applud idea

Having done this analysis and taking into account the state of situation discussed in paragraph 2 above, it is possible to better define the objective and the functionalities that Applud has to offer in order to meet the needs of the market.

The first shortcoming of the market is that there is no application in Catalonia that provides access to private and public healthcare from the same point. This is the most important feature of the Applud and the major differentiating point with respect to its competitors. It is a fact that in Catalonia the public and private systems operate completely independently, but the people who make use of these two systems are the same. One way to facilitate access to these two systems for all these users is to unify the access point.

Another shortcoming of the market is that the applications on offer are oriented towards a very specific aspect of health. There are many apps for diabetes, for stop smoking, for monitoring blood pressure or other vital signs, and then there are those specific to private hospitals for making appointments and consulting reports. With the exception of La meva Salut, there is no other app that offers all these functionalities. This is another point that Applud wants to address.

La meva Salut is an application aimed at all users of the public health system in Catalonia. This includes patients aged 18 and over. However, it is a fact that middle-aged and elderly people do not find it easy to use technology, which makes it extremely important to create user interfaces for these applications that are especially intuitive and easy to use. La meva Salut, although it does have a wide range of health actions to do, does not have a great user experience.

Moreover, this is the age at which the use of healthcare soars. More frequent visits to doctors, blood tests, cancer screening programmes, etc. start. It is therefore of vital importance that these people have easy access to all this information. On the other hand, another important point is that they can receive the recommendations of the doctors according to their age and the vaccination schedules they are due for.

These three points discussed in this section are added to the table of requirements in the previous section. Once this analysis has been carried out, we can consider the needs of chronic patients and the market as closed.

The following table shows the needs that have been discussed and the requirements or functionalities that Applud will have to offer in order to respond to these aspects:

Needs of target users	Applud Requirement/Functionality
Easy access to the reference service is requested when resolving queries	Integrated chat allowing contact with a healthcare professional
Fragmentation of the system creates insecurity: lack of integrated care	Access to all medical records, both private and public, by the user and by health professionals
Health services assume that caregivers follow up with the sick person	Easy disease monitoring from the app
Healthy habits such as good nutrition and physical exercise are basic elements of a healthy lifestyle	Publish articles promoting healthy diet and lifestyle habits, published by different professionals
The training and information provided by the health system empowers individuals	Publish outreach articles and offer training courses
People feel that the system lacks more guidance on health promotion and prevention	Publish articles promoting healthy diet and lifestyle habits and easy health monitoring through vital signs monitoring
Adaptation of the system to new technologies helps self-healing	All of Applud's functionalities respond to this need
Lack of common gateway to public and private health care systems	Access to public and private health care from the same application
Applications too specific to one type of chronic disease	All of Applud's functionalities respond to this need
Difficulty for older people to interact with technology	Design an easy and intuitive interface that offers a great user experience

Table 3. Summary table of the needs not covered by the current applications on the market and the key functionalities of Applud

4. Conceptual engineering

In this section we move on to the next step of the process, "requirements & specification", which is necessary in any IT project.

In this step, the aim is to answer what system needs to be built, what does the system need to do, and how does it need to do it? These three questions will be answered in the sections "requirements analysis", "specification" and "design" respectively.

4.1 Requirements analysis

Requirements are capabilities and conditions to which the system must conform. A prime challenge of requirements work is to find, communicate, and remember what is really needed, in a form that clearly speaks to the client and development team members. The requirements can be used to define which system needs to be built[xxviii].

Along these lines, the famous American company founded by William Hewlett and David Packard, better known as HP, published a model for classifying software quality attributes, both functional and non-functional requirements, known as the FURPS model.

In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between inputs and outputs. Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. The plan for implementing these type of requirements is detailed in the system design. On the other hand, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. The plan for implementing these type of requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

FURPS stands for[xxix]:

- Functionality: specification of behavior.
- Usability (UX): specification designed to ensure that a product is easy to use. For example, the system shall be finger-operated.
- Reliability: technical specifications. For example, The system will be available 24 hours a day, 7 days a week.
- Performance: define how well the software system accomplishes certain functions under specific conditions. Examples include the software's speed of response, throughput, execution time and storage capacity.
- Supportability: is concerned with characteristics such as maintainability, scalability of the solution. For example, the user interface will be implemented on a web browser with different languages.

The definition of these requirements is an essential point in the design and development of an application. Thanks to the market analysis and the state of the situation carried out in previous sections, Applud's functional requirements have been requested to be described.

Only the functional ones are described since, as mentioned in the introduction, the aim of the work is to develop the front-end of a mobile application. The other requirements are part of the architecture and backend of the application, which are not addressed in this project.

- The system will ask all users who are not registered on the app to register
- The system will prompt all registered users to log in at each connection
- The application's homepage will display health recommendations from health professionals
- The application will contain a side drawer menu specifying the different actions that the user can perform
- The system will allow a calendar to be displayed
- The system will allow appointments to be added to a calendar
- The system will allow the addition of reminders for an appointment
- The system will allow the insertion of readings for weight, blood pressure, oxygen saturation, temperature and heart rate
- The system will allow access to the public health system
- The system will allow access to the private health care system to which the user is affiliated
- The application will offer a chat where the user will be able to contact a nursing service
- The user may modify his or her personal data at any time
- The user will receive in the same application notices issued by the health systems to which they are affiliated about their vaccination plan or any other programme they are recommended to be enrolled in because of their age, such as cancer screening programmes

4.2 Specification: Use Case Diagram

Use cases are a technique for capturing, modelling and specifying the requirements of a system[xxx]. A use case corresponds to a set of behaviours that the system may perform in interaction with its actors, and which produces an observable result that contribute to its goals. Actors represent the role that human users or other systems have in the interaction.

Use cases are normally written in diagrammatic format. In order to draw the diagram, the following must be taken into account a few key components[xxxi]:

- System: A system is the product, service, or software under discussion. In this case, the application Applud.

- Actors: An actor is a user or anything else that exhibits behavior when interacting with the system. The actor could be another system, a piece of hardware, or an entire organization. There are four types of actors: a system under discussion, an internal actor, a primary actor, and a secondary actor. The most commonly referred to are the latter two systems. A primary actor initiates the interaction with the system, while a secondary actor may provide a service to the system.
- Scenario: A scenario is a specific sequence of actions and interactions between actors and the system under discussion; it is also called a use case instance.

- Use case: A use case describes the success and failure scenarios that can occur when the actor(s) interact with the system. In this section, the main success scenario, that is, the most desirable outcome between the actor and the system, is established. It also sets out the alternative paths, which explain what happens in case of failure or error.

Once the above components have been identified, they can be illustrated by drawing a use case diagram. An use case diagram summarize de details of the system's users (actors) and their interactions with the system. To build one, it is used a set of specialized symbols and connectors. An effective use case diagram help the development team to discuss and represent:

- Scenarios in which the system or application interacts with people, organizations, or external systems

- Goals that your system or application helps those entities (known as actors) achieve

- The scope of the system

4.2.1 Applud use case diagram

This section explains how the Applud use case diagram has been created.

- Actors: It is clear that the two main actors in our application are the user and the system itself. It is the user who, once logged into the application, will be able to interact with the system by performing different actions. The system will respond in one way or another depending on what the user asks it to do.

- Scenarios:

  - Sign up: The medical plan and the app's recommendations will depend on the age and health status of the patient, which is why it will be necessary for anyone to register by filling in the registration form in order to use the app.
  - Sign in: Any registered user must be able to log in by entering their credentials.
  - View recommendations: As mentioned above, the system will recommend a series of articles and videos to each user based on their age and health status. The user will be able to view these recommendations.
  - Edit profile: A basic action that any application allows its users to do is to edit their profile information. The user can change the email with which he/she has registered, the password, the date of birth, the name and other personal information.
  - Calendar: The application also has a calendar where the user can add the appointments he/she has made with the doctor and create reminders.
  - Request appointment: A basic action of the application is that the user will be able to make an appointment with a doctor whether he/she is in public or private health care. The system, at the backend level, will make the connection to the IT systems of the health systems.
  - Contact nurse: For all those doubts that the user may have but which do not need to be resolved in person at a consultation, the system offers the possibility of contacting a network of nurses and healthcare professionals through the application's chat. If the question requires a visit from a doctor, the professional will let the user know and ask them to make an appointment with the necessary doctor.
  - Vital constants: Another key feature of the application is that it allows the user to keep a record of all vital signs and keep track of them. This data can be sent to the doctor directly from the app, making monitoring by professionals much simpler and smoother.

The following figure shows the diagram described above:

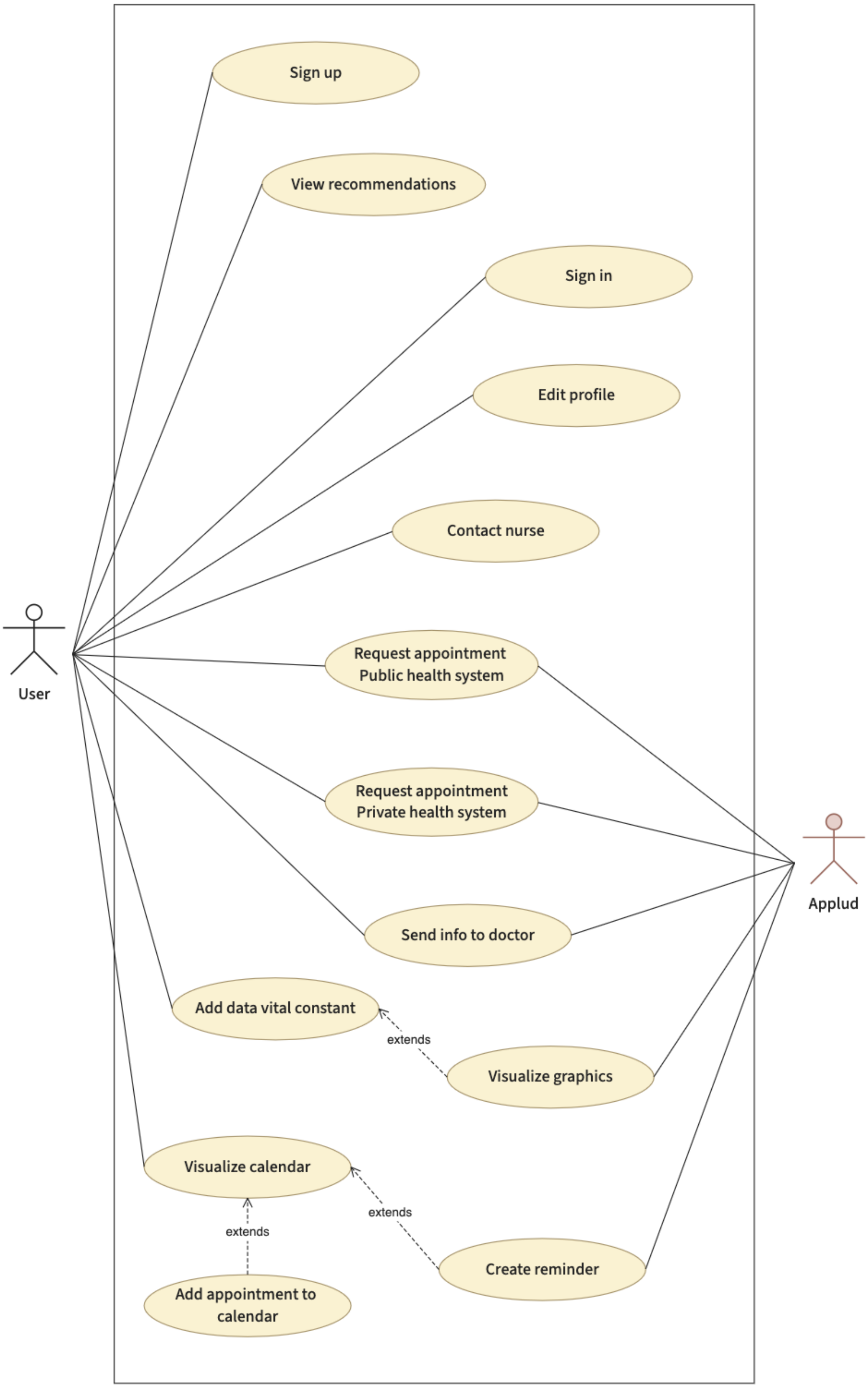


Figure 7. Applud application use case diagram

4.3 Design

The difference between good app design and a poor one is usually the quality of its user experience. Fast loading times, ease of use, and overall customer satisfaction during an interaction should be integral parts of your design. Great app design is clearly laid out, efficient to use, and aesthetically pleasing.

App design combines the user interface (UI) and user experience (UX). While UI lends itself to the overall style of the app (including the colors, fonts, and general look and feel), UX focuses on the actual functionality and usability.

A large number of users abandon an app after using it for the first time. Since users are picky about which apps they use and so quick to abandon those they don't enjoy, it's essential to invest time and effort in creating a great user experience. The better the design, the better the chance that a user will engage with it and thus keep using it.

Given its importance and having seen in section "3.3 Applud idea" that user experience is one of the priority requirements of Applud, a specific section has been devoted to this point in this final thesis.

The logo and the colours chosen for the application as well as the design of the screens are presented here.

4.3.1 Logotype and colours

Colour is used to convey a feeling or a theme. In the case of this project, the theme is health and the sensation to be transmitted is cleanliness and minimalism. For this reason, green and white have been chosen as the main colours for the application.

The logo is intended to easily identify the application with health. This is achieved with the shape, which is reminiscent of both a heart and medication pills. It is also reminiscent of a crossing of hands, which conveys the idea of companionship and teamwork. Blue tones have been chosen for the logo as this colour is attributed to harmony and hope.

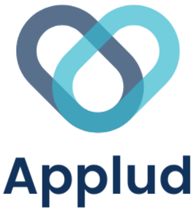


Figure 8. Applud logo

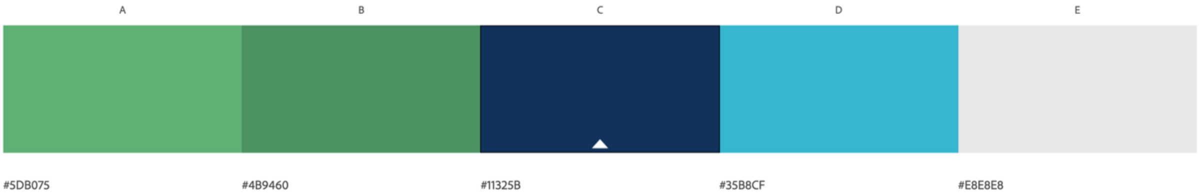
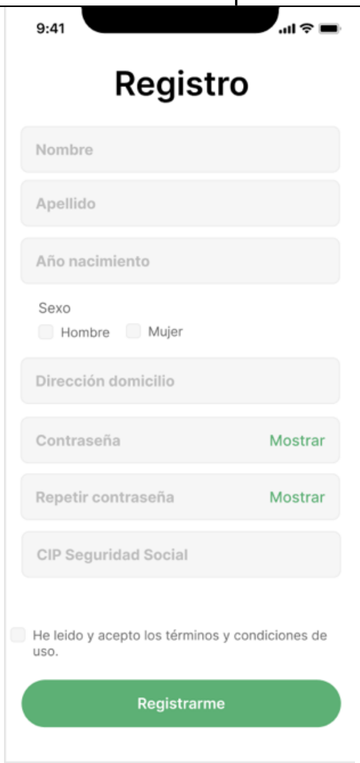


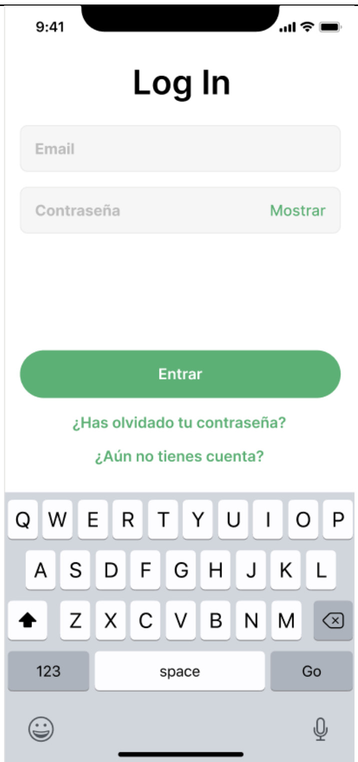
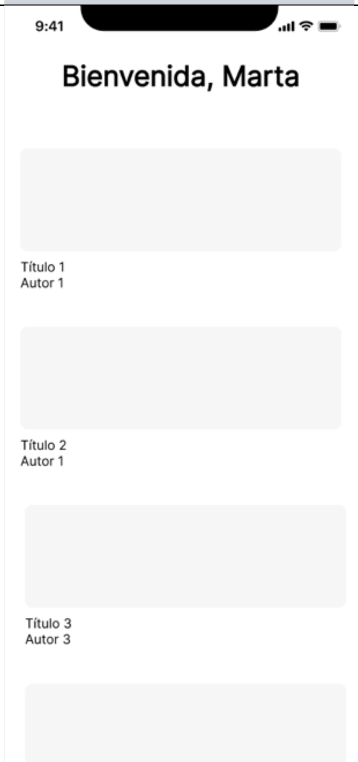
Figure 9. Applud color palette

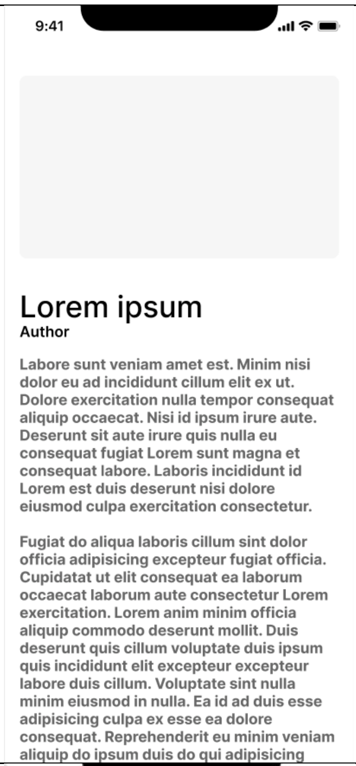
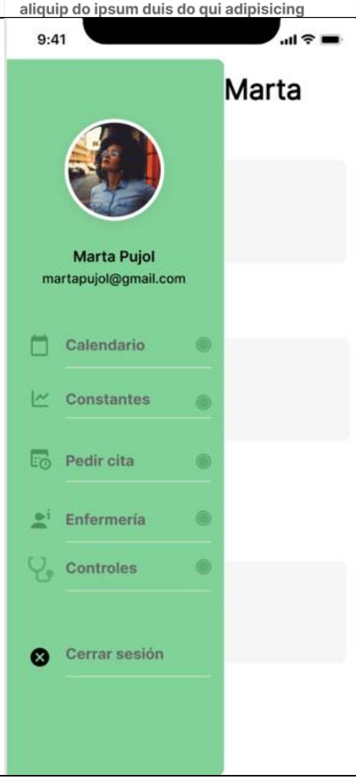
4.3.2 Mock-ups

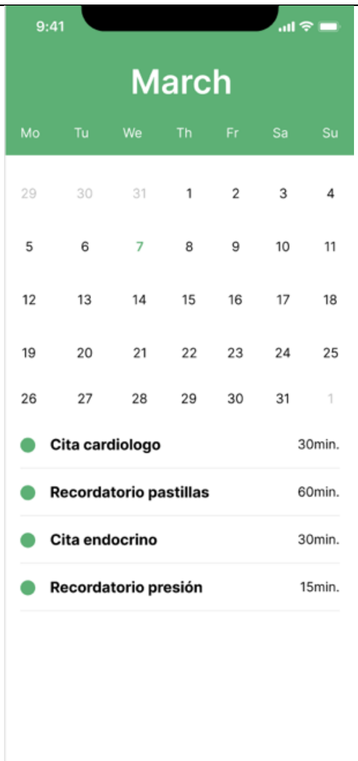
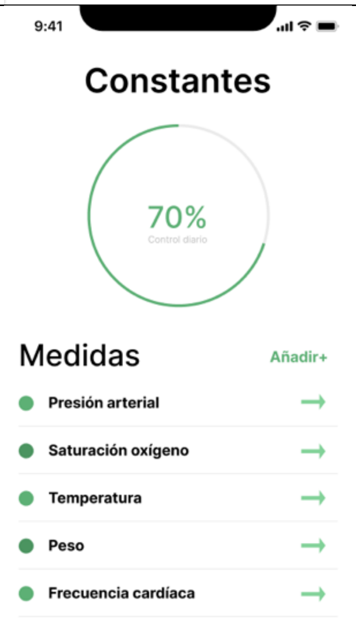
With the logo designed and the colour palette of the application chosen, the team of graphic designers would go on to design the screens of the application in Figma. Figma is a web-based graphics editor used by the vast majority of the industry. It offers several tools for designing websites and mobile applications.

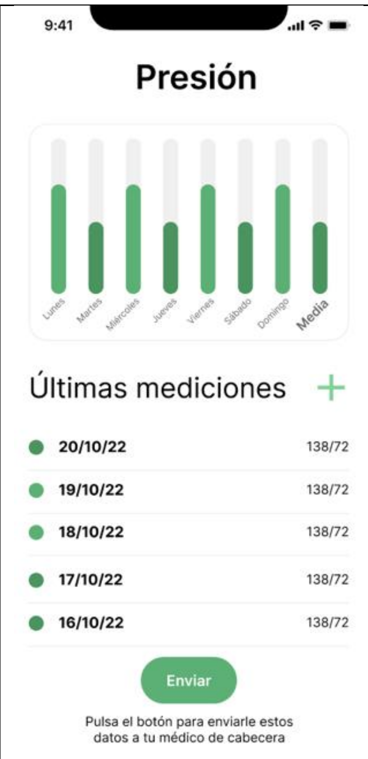
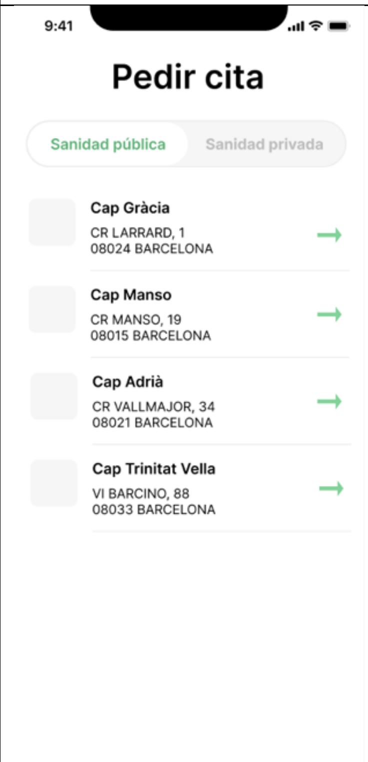
In this section, therefore, the screen design made with Figma is be presented in the following table.

Screen identifier	Mock-up	Description
Register	 9:41 Registro Nombre Apellido Año nacimiento Sexo □ Hombre □ Mujer Dirección domicilio Contraseña Mostrar Repetir contraseña Mostrar CIP Seguridad Social □ He leido y acepto los términos y condiciones de uso. Registrarme	Screen where new users register in the application. Gender and age are requested in order to recommend different items and healthy lifestyle habits more personalised to the user's situation. Also so that the controls shown by the app are according to their needs. For example, a man will not be shown the gynaecological check-ups he has pending on the control screen, but he will be shown the early detection programme for prostate cancer. An important piece of information requested from the user is their CIP, in order to make the connection with the public health system in Catalonia. The user is also asked for their policy number, if they have private insurance.

Login	 9:41 Log In Email Contraseña Mostrar Entrar ¿Has olvidado tu contraseña? ¿Aún no tienes cuenta? Q W E R T Y U I O P A S D F G H J K L Z X C V B N M 123 space Go	Application login screen. You need to be registered first. If the user or password is entered incorrectly, the system will give a warning and will not allow you to enter. This is the first screen displayed when the user open the application. If the user is not registered, he/she can navigate to the registration screen from this screen.
Home	 9:41 Bienvenida, Marta Título 1 Autor 1 Título 2 Autor 1 Título 3 Autor 3	Main screen of the application, once the user has logged in. A scrolling list of items that the system recommends to promote healthy living is displayed. For each article you can see a descriptive image, the title and the author who wrote it.

Article detail view	 9:41 Lorem ipsum Author Labore sunt veniam amet est. Minim nisi dolor eu ad incididunt cillum elit ex ut. Dolore exercitation nulla tempor consequat aliquip occaecat. Nisi id ipsum irure aute. Deserunt sit aute irure quis nulla eu consequat fugiat Lorem sunt magna et consequat labore. Laboris incididunt id Lorem est duis deserunt nisi dolore eiusmod culpa exercitation consectetur. Fugiat do aliqua laboris cillum sint dolor officia adipisicing excepteur fugiat officia. Cupidatat ut elit consequat ea laborum occaecat laborum aute consectetur Lorem exercitation. Lorem anim minim officia aliquip commodo deserunt mollit. Duis deserunt quis cillum voluptate duis ipsum quis incididunt elit excepteur excepteur labore duis cillum. Voluptate sint nulla minim eiusmod in nulla. Ea id ad duis esse adipisicing culpa ex esse ea dolore consequat. Reprehenderit eu minim veniam aliquip do ipsum duis do qui adipisicing	Article detail. Displayed when the user clicks on a specific article item. From here he/she can read it and return to the main screen.
Side menu	 9:41 Marta Marta Pujol martapujol@gmail.com Calendario Constantes Pedir cita Enfermería Controles Cerrar sesión	The side menu can be accessed from any screen of the application. Through this menu the user can navigate to any functionality offered by the application. It also allows to log out.

Calendar	 9:41 March Mo Tu We Th Fr Sa Su 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 ● Cita cardiologo 30min. ● Recordatorio pastillas 60min. ● Cita endocrino 30min. ● Recordatorio presión 15min.	From the side menu the user can access the "Calendar" screen. Here the user can save appointments and add reminders. With the calendar the user can access any day and time in the past or future. Appointments and reminders for the current day are displayed at the bottom.
Vital Signs	 9:41 Constantes 70% Control diario Medidas Añadir+ ● Presión arterial → ● Saturación oxígeno → ● Temperatura → ● Peso → ● Frecuencia cardíaca →	From the side menu the user can access the "Constants" screen. At the top, which shows a graph that reports the progress of the daily health check to be carried out by that particular user. The daily monitoring is determined by the user's doctor. At the bottom, a menu is displayed to access the vital signs that the doctor has instructed the user to check. Even if the doctor has not indicated it, the user can add a vital sign that he/she wants to track. Clicking on any of the options will take the user to the details of that particular vital constant.

Detail Vital Signs	 9:41 Presión Lunes Martes Miércoles Jueves Viernes Sábado Domingo Media Últimas mediciones + ● 20/10/22 138/72 ● 19/10/22 138/72 ● 18/10/22 138/72 ● 17/10/22 138/72 ● 16/10/22 138/72 Enviar Pulsa el botón para enviarle estos datos a tu médico de cabecera	From any of the options on the previous screen, the user can access the detail of a vital constant. A graph of the weekly monitoring of that constant is displayed at the top. At the bottom, the user can add a value manually. From this screen the data can also be sent manually to the doctor, although health professionals can automatically access the information that is updated daily.
Make an appointment	 9:41 Pedir cita Sanidad pública Sanidad privada Cap Gràcia CR LARRARD, 1 08024 BARCELONA → Cap Manso CR MANSO, 19 08015 BARCELONA → Cap Adrià CR VALLMAJOR, 34 08021 BARCELONA → Cap Trinitat Vella VI BARCINO, 88 08033 BARCELONA →	From the side menu the user can access the "Appointments" screen. On this screen the user can access the public health system of Catalonia to make an appointment. Also to his/her private insurance, if he/she have one. The corresponding Primary Care Centre will be shown, as well as the hospitals or day centres assigned to the user according to his/her health situation.

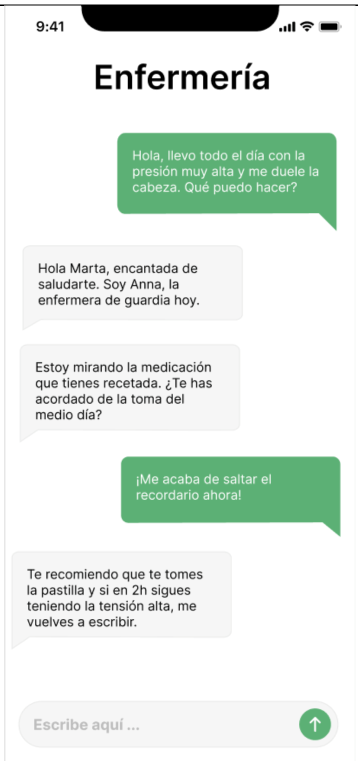
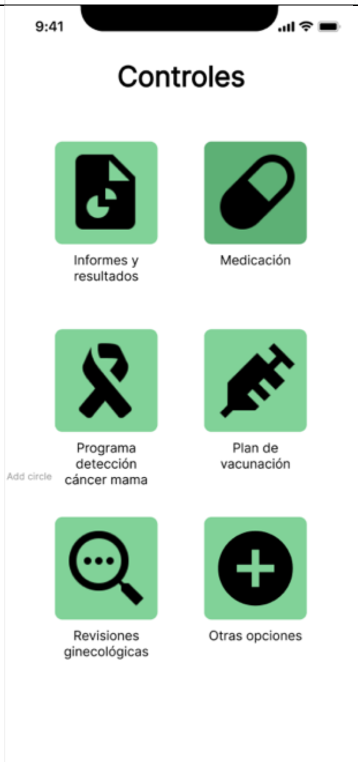
Nursing	 9:41 Enfermería Hola, llevo todo el día con la presión muy alta y me duele la cabeza. Qué puedo hacer? Hola Marta, encantada de saludarte. Soy Anna, la enfermera de guardia hoy. Estoy mirando la medicación que tienes recetada. ¿Te has acordado de la toma del medio día? ¡Me acaba de saltar el recordario ahora! Te recomiendo que te tomes la pastilla y si en 2h sigues teniendo la tensión alta, me vuelves a escribir. Escribe aquí ...	From the side menu the user can access the "Nursing" screen. From here the user can initiate a conversation with health professionals to resolve any queries he/she may have. The professional will be able to access the user's health information to guide him/her.
Checkpoints	 9:41 Controles Informes y resultados Medicación Programa detección cáncer mama Add circle Plan de vacunación Revisiones ginecológicas Otras opciones	From the side menu the user can access the "Checkpoints" screen. In this screen the user will be able to see all the personalised health options according to their situation. In the case of this example, the user is a woman of approximately 60 years of age. This is why the system shows the option "Early detection of breast cancer" and gynaecological check-ups. Also the vaccination plan specific to her situation.

Table 4. Design and description of Applud screens

5. Detail Engineering

With the design done, it is now possible to start the development of the application code. The programmers try to implement the screens that the graphic designers have designed. However, sometimes it is necessary to introduce small modifications because some points on a technical level cannot be realised.

In order to better explain to the reader how the work has been done, this section explains the visual and logical implementation of a specific screen, Vital Constant Detail, because it is the most complete and incorporates the greatest diversity of components.

It is important to mention that when developing a mobile application it is first implemented the visual part, the screen itself, in XML files and then add logic and function to those visual components. In the following, therefore, it is explained how the visual part is implemented in the corresponding files and with the appropriate language. The same for the logic.

The other files of the project can be found in the annex and in the following Github link: https://github.com/bbachs/TFG_EBM.git

5.1 Visual implementation

Android layouts are written **in** eXtensible Markup Language, also known as XML[xxxii]. Much like HTML (or HyperText *Markup Language*), XML is also a markup language. It was created as a standard way to encode data in internet-based applications. However, *unlike* HTML, XML is case-sensitive, requires each tag is closed properly, and preserves whitespace.

Android XML layouts are also part of a larger umbrella of Android files and components called resources. Resources are the additional files and static content an application needs, such as animations, color schemes, layouts, menu layouts.

Each layout file must contain one root element. Linear Layouts, Relative Layouts, Constraint Layouts and Frame Layouts may all be root elements. Other layouts may not be. All other XML elements will reside within this root object.

A View is simply an object from Android's built-in View class. It represents a rectangular area of the screen, and is responsible for displaying information or content, and event handling. Text, images, and buttons are all Views in Android.

A ViewGroup is a subclass of View, and is essentially an 'invisible container' that holds multiple Views or ViewGroups together, and defines their layout properties[xxxiii].

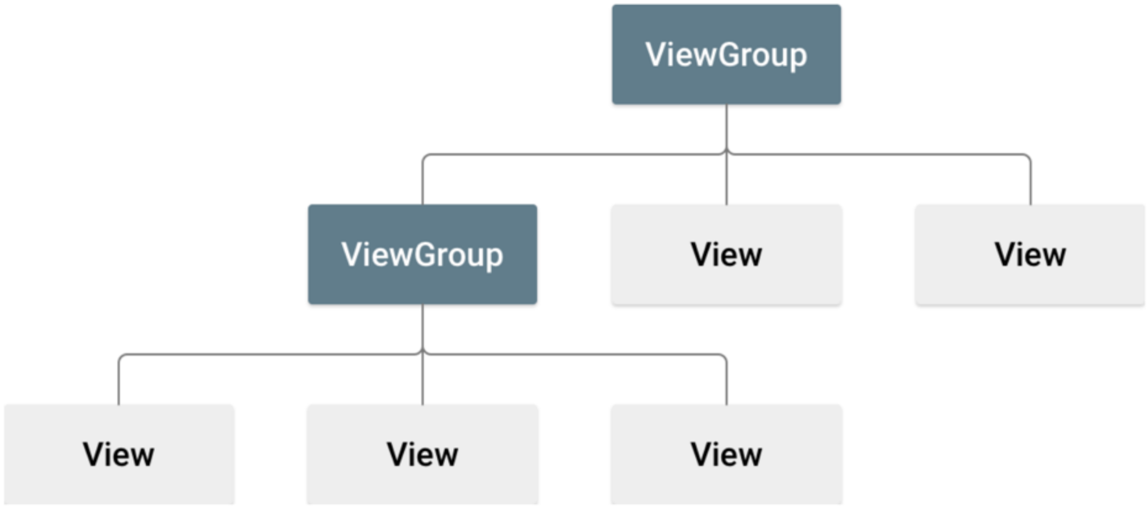
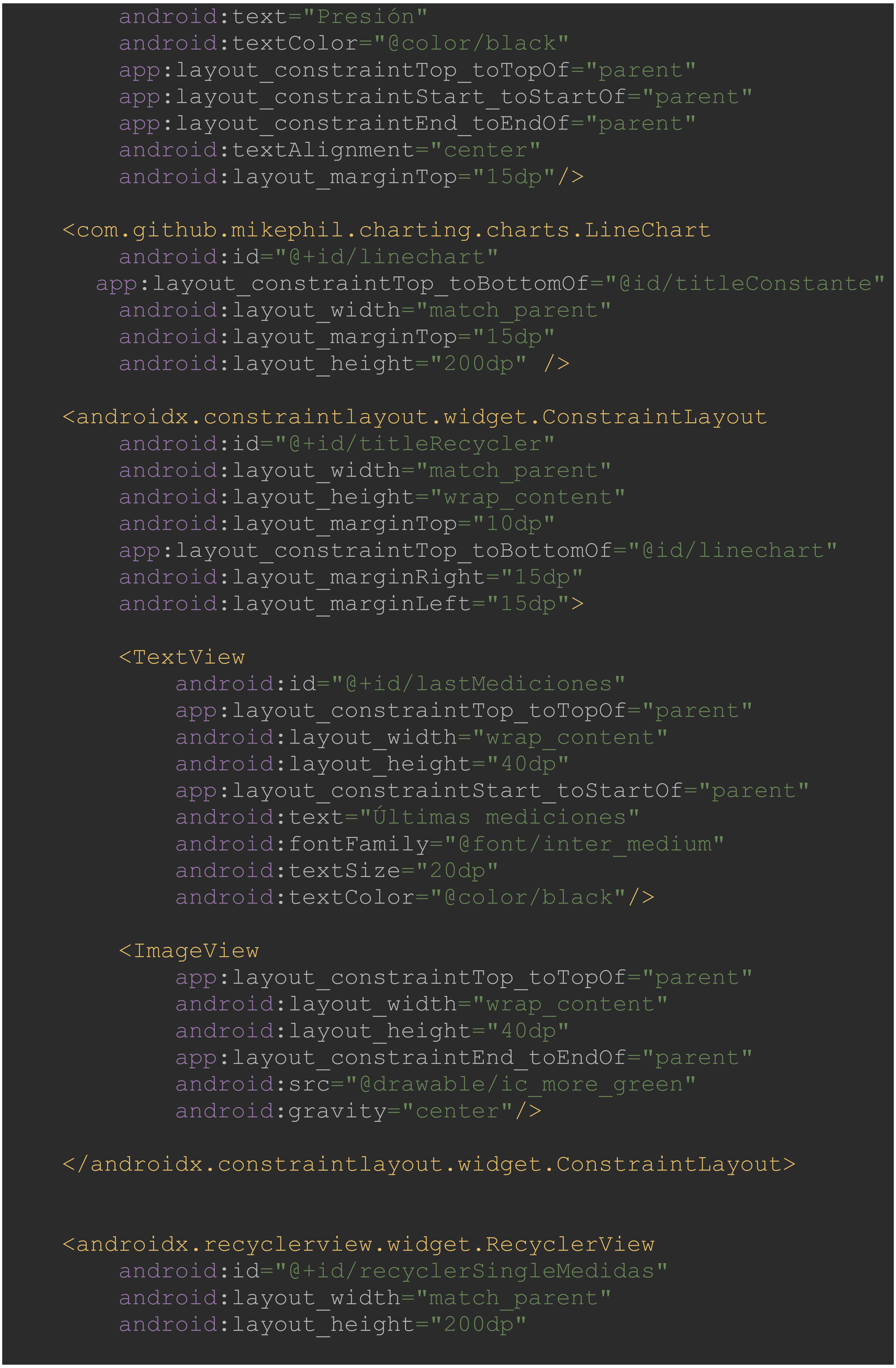


Figure 10. Illustration of a view hierarchy, which defines a UI layout

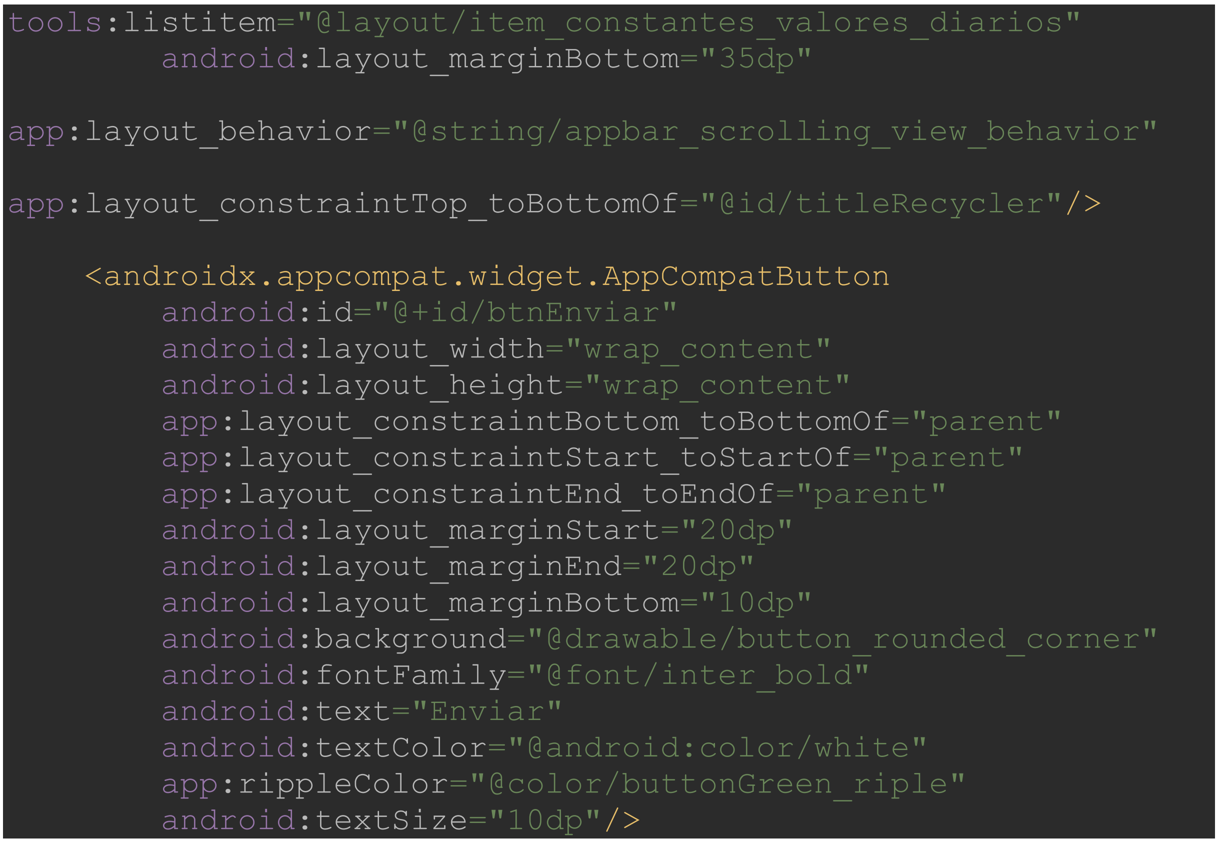
Once it is defined the root element, it can be add additional layout objects or widgets as child elements to gradually build a View hierarchy that defines the layout. The necessary properties (colour, size, position, etc.) are added to each widget to meet the design requirements.

The following lines show the XML file of the Vital Constant Detail screen. As can be seen, the root element assigned in this case is a ConstraintLayout. This is achieved with the second line of code (<androidx.constraintlayout.widget.ConstraintLayout>).

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.features.mainMenu.ui.constants.Cons
tantsFragment">

    <ImageButton
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:id="@+id/backButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/white"
        android:layout_marginTop="5dp"
        android:layout_marginLeft="8dp"
        android:src="@drawable/ic_arrow_back"/>

    <TextView
        android:id="@+id/titleConstante"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fontFamily="@font/inter_semibold"
        android:textSize="20dp"
```



```
        android:text="Presión"
        android:textColor="@color/black"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:textAlignment="center"
        android:layout_marginTop="15dp"/>

    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/linechart"
      app:layout_constraintTop_toBottomOf="@id/titleConstante"
        android:layout_width="match_parent"
        android:layout_marginTop="15dp"
        android:layout_height="200dp" />

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/titleRecycler"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        app:layout_constraintTop_toBottomOf="@id/linechart"
        android:layout_marginRight="15dp"
        android:layout_marginLeft="15dp">

        <TextView
            android:id="@+id/lastMediciones"
            app:layout_constraintTop_toTopOf="parent"
            android:layout_width="wrap_content"
            android:layout_height="40dp"
            app:layout_constraintStart_toStartOf="parent"
            android:text="Últimas mediciones"
            android:fontFamily="@font/inter_medium"
            android:textSize="20dp"
            android:textColor="@color/black"/>

        <ImageView
            app:layout_constraintTop_toTopOf="parent"
            android:layout_width="wrap_content"
            android:layout_height="40dp"
            app:layout_constraintEnd_toEndOf="parent"
            android:src="@drawable/ic_more_green"
            android:gravity="center"/>

    </androidx.constraintlayout.widget.ConstraintLayout>


    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerSingleMedidas"
        android:layout_width="match_parent"
        android:layout_height="200dp"
```



```
tools:listitem="@layout/item_constantes_valores_diarios"
        android:layout_marginBottom="35dp"

app:layout_behavior="@string/appbar_scrolling_view_behavior"

app:layout_constraintTop_toBottomOf="@id/titleRecycler"/>

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btnEnviar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="10dp"
        android:background="@drawable/button_rounded_corner"
        android:fontFamily="@font/inter_bold"
        android:text="Enviar"
        android:textColor="@android:color/white"
        app:rippleColor="@color/buttonGreen_riple"
        android:textSize="10dp"/>
```

Next, the ImageButton widget has been added. It is located at the top left of the screen and corresponds to the arrow icon to return to the previous screen:

```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintStart_toStartOf="parent"
```

It has been assigned a white background:

```
android:background="@color/white"
```

A margin has been added so that it is not stuck to the edges:

```
android:layout_marginTop="5dp"
android:layout_marginLeft="8dp"
```

And, finally, the icon of an arrow has been assigned to it:

```
android:src="@drawable/ic_arrow_back"
```

The next element is a TextView. Corresponds to the title of the screen. The font, the text size, the text to be displayed on the screen and the text colour have been set.

```
android:fontFamily="@font/inter_semibold"
android:textSize="20dp"
android:text="Presión"
android:textColor="@color/black"
```

It is positioned at the top of the screen, leaving a margin of 15dp with the top edge, and centred horizontally.

```
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"
android:textAlignment="center"
android:layout_marginTop="15dp"
```

Below the title, a chart is displayed thanks to the insertion of the com.github.mikephil. charting.charts.LineChart component. It is indicated to be positioned below the title, leaving a margin of 15dp. It is also given a maximum height of 200dp.

```
app:layout_constraintTop_toBottomOf="@id/titleConstante"
android:layout_marginTop="15dp"
android:layout_height="200dp"
```

Another ConstraintLayout is added below the title to form a component that will consist of a TextView and an ImageView. This ensures that the TextView and ImageView are always aligned, whatever the screen size on which the application will be opened.

The next component is a RecyclerView. This is a widget widely used in Android Studio to form lists of any type with N items. The items are all the same, so there is no need to create them one by one but, as the name suggests, the views are recycled. The only thing that is done from the code is to set the information of each item, but the items are not initialized from the view.

The recyclerView is given the size it is to occupy within the screen. The margin with the bottom component. It is placed below the component we have discussed in the previous paragraph and it is also told which items are going to be used.

```
tools:listitem="@layout/item_constantes_valores_diarios"
android:layout_marginBottom="35dp"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
app:layout_constraintTop_toBottomOf="@id/titleRecycler"
```

Finally, the screen contains a button of type AppCompatButton. This button is positioned at the bottom of the screen and margins are added:

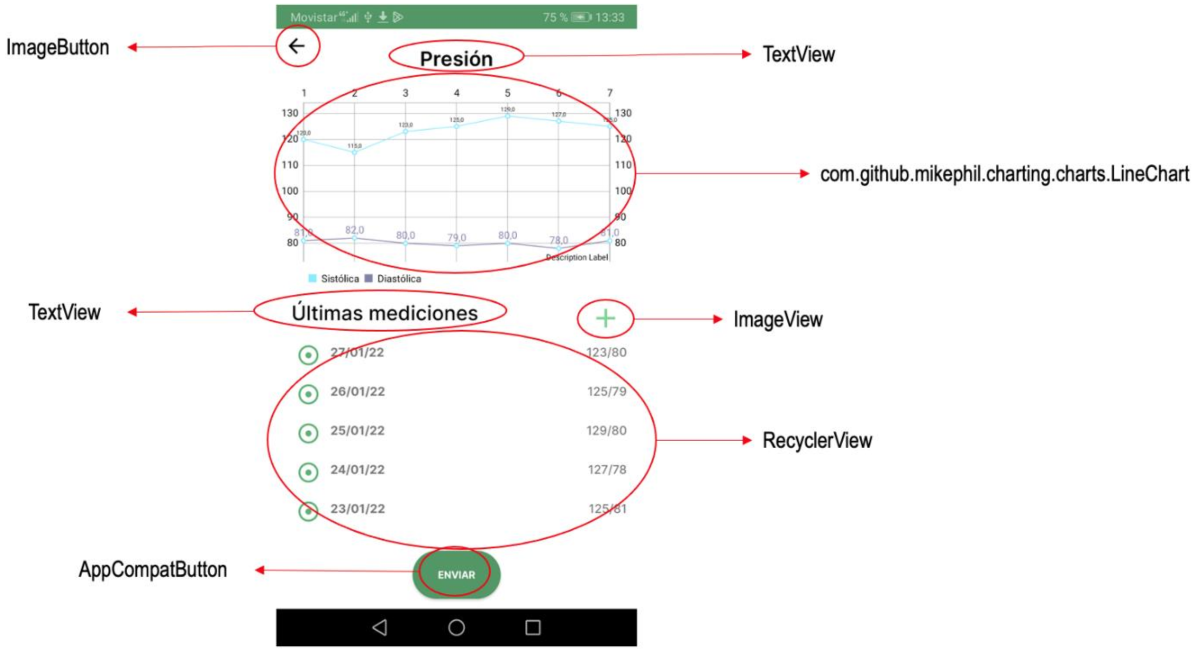
```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"
android:layout_marginStart="20dp"
android:layout_marginEnd="20dp"
android:layout_marginBottom="10dp"
```

The style to be applied (green button with rounded corners) is also indicated. As well as the text of the button, the typography and the size.

```
android:background="@drawable/button_rounded_corner"
android:fontFamily="@font/inter_bold"
```

```
android:text="Enviar"
android:textColor="@android:color/white"
android:textSize="10dp"
```

In the image below you can see a screenshot of what this screen looks like on a real Huawei phone and the components mentioned above have been identified.



Presión

Últimas mediciones

27/01/22	123/80
26/01/22	125/79
25/01/22	129/80
24/01/22	127/78
23/01/22	125/81

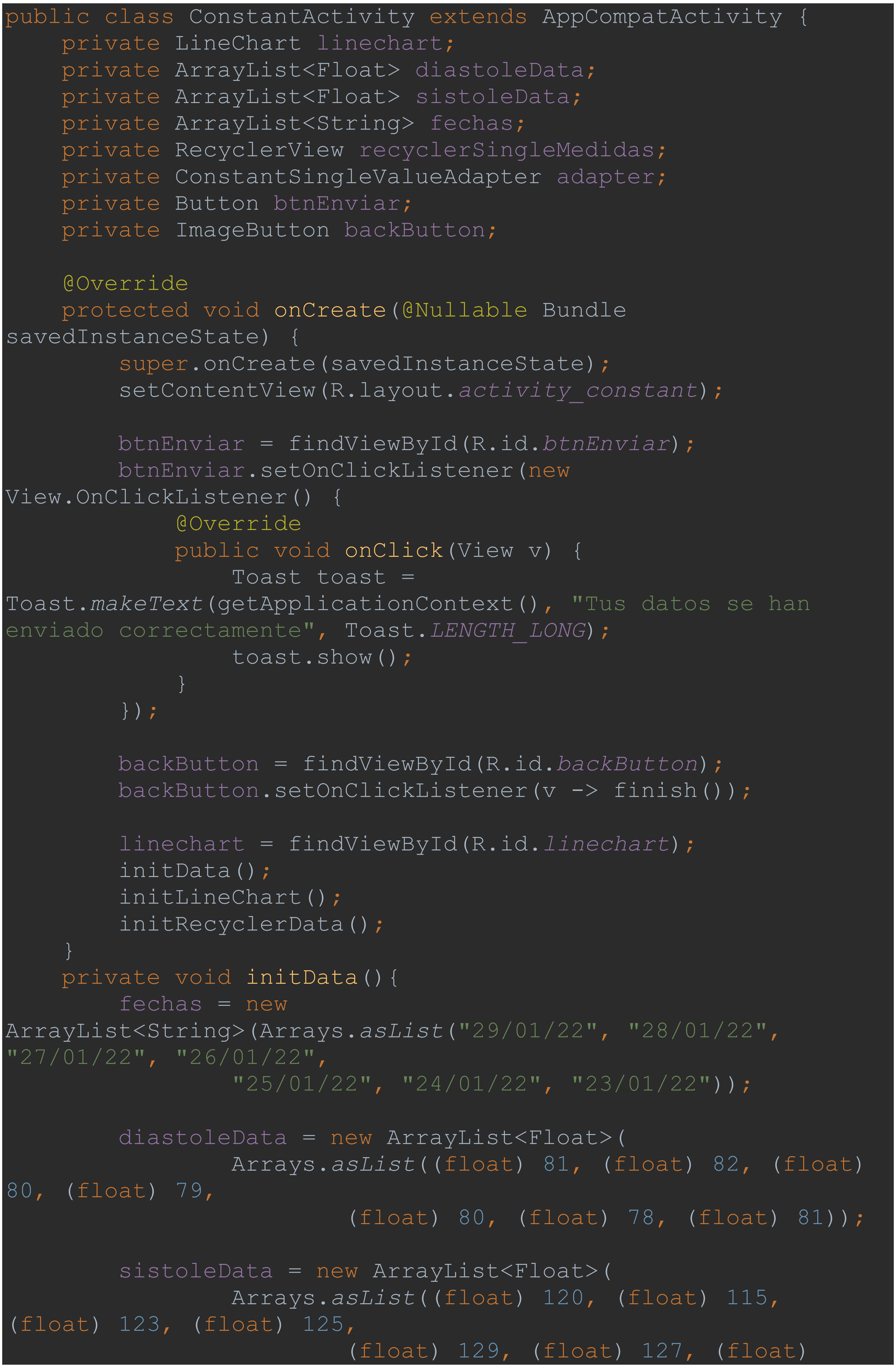
Figure 11. Diagram of the android components that make up the Vital Signs Detail screen

5.2 Logic implementation

The mobile-app experience differs from its desktop counterpart in that a user's interaction with the app doesn't always begin in the same place. Instead, the user journey often begins non-deterministically. For instance, if you open an email app from your home screen, you might see a list of emails. By contrast, if you are using a social media app that then launches your email app, you might go directly to the email app's screen for composing an email.

The Activity class is designed to facilitate this paradigm. When one app invokes another, the calling app invokes an activity in the other app, rather than the app as an atomic whole. In this way, the activity serves as the entry point for an app's interaction with the user. You implement an activity as a subclass of the Activity class.

It is a crucial component of an Android app, and the way activities are launched and put together is a fundamental part of the platform's application model. Unlike programming paradigms in which apps are launched with a main() method, the Android system initiates code in an Activity instance by invoking specific callback methods that correspond to specific stages of its lifecycle.
An activity provides the window in which the app draws its UI. This window typically fills the screen, but may be smaller than the screen and float on top of other windows. Generally, one activity implements one screen in an app. For instance, the Vital Constants Detail screen is implemented in the following Java Activity:



```
public class ConstantActivity extends AppCompatActivity {
    private LineChart linechart;
    private ArrayList<Float> diastoleData;
    private ArrayList<Float> sistoleData;
    private ArrayList<String> fechas;
    private RecyclerView recyclerSingleMedidas;
    private ConstantSingleValueAdapter adapter;
    private Button btnEnviar;
    private ImageButton backButton;

    @Override
    protected void onCreate(@Nullable Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_constant);

        btnEnviar = findViewById(R.id.btnEnviar);
        btnEnviar.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast toast =
Toast.makeText(getApplicationContext(), "Tus datos se han
enviado correctamente", Toast.LENGTH_LONG);
                toast.show();
            }
        });

        backButton = findViewById(R.id.backButton);
        backButton.setOnClickListener(v -> finish());

        linechart = findViewById(R.id.linechart);
        initData();
        initLineChart();
        initRecyclerData();
    }
    private void initData(){
        fechas = new
ArrayList<String>(Arrays.asList("29/01/22", "28/01/22",
"27/01/22", "26/01/22",
                "25/01/22", "24/01/22", "23/01/22"));

        diastoleData = new ArrayList<Float>(
                Arrays.asList((float) 81, (float) 82, (float)
80, (float) 79,
                        (float) 80, (float) 78, (float) 81));

        sistoleData = new ArrayList<Float>(
                Arrays.asList((float) 120, (float) 115,
(float) 123, (float) 125,
                        (float) 129, (float) 127, (float)
125));
    }
    private void initLineChart(){
        ArrayList<Entry> lineEntries2 = new
ArrayList<Entry>();
        ArrayList<Entry> lineEntries1 = new
ArrayList<Entry>();
        lineEntries2.add(new Entry(1, sistoleData.get(0)));
        lineEntries2.add(new Entry(2, sistoleData.get(1)));
        lineEntries2.add(new Entry(3, sistoleData.get(2)));
        lineEntries2.add(new Entry(4, sistoleData.get(3)));
        lineEntries2.add(new Entry(5, sistoleData.get(4)));
        lineEntries2.add(new Entry(6, sistoleData.get(5)));
        lineEntries2.add(new Entry(7, sistoleData.get(6)));

        lineEntries1.add(new Entry(1, diastoleData.get(0)));
        lineEntries1.add(new Entry(2, diastoleData.get(1)));
        lineEntries1.add(new Entry(3, diastoleData.get(2)));
        lineEntries1.add(new Entry(4, diastoleData.get(3)));
        lineEntries1.add(new Entry(5, diastoleData.get(4)));
        lineEntries1.add(new Entry(6, diastoleData.get(5)));
        lineEntries1.add(new Entry(7, diastoleData.get(6)));

        //Genera el nou Linechart amb les hores aparcades per
l'usuari
        LineDataSet lineDataSet2=new LineDataSet(lineEntries2,
"Sistólica");

        LineDataSet lineDataSet1=new LineDataSet(lineEntries1,
"Diastólica");
        lineDataSet1.setColor(R.color.purple_200);
        lineDataSet1.setValueTextColor(R.color.purple_200);
        lineDataSet1.setValueTextSize(10f);

        LineData ll= new LineData(lineDataSet2, lineDataSet1);
        ll.setHighlightEnabled(true);
        linechart.setData(ll);
        linechart.invalidate();
    }

    private void initRecyclerData(){
        recyclerSingleMedidas = (RecyclerView)
findViewById(R.id.recyclerSingleMedidas);
        recyclerSingleMedidas.setLayoutManager(new
LinearLayoutManager(this));
        adapter = new ConstantSingleValueAdapter(fechas,
diastoleData, sistoleData, this);
        recyclerSingleMedidas.setAdapter(adapter);
    }
}
```

As can be seen in the first lines of code, the attributes contained in this class are of the Java type, like the ArrayList<>, and Android components used in the view, in the xml discussed in the previous section.

The first method executed in any Activity is the onCreate() method. This is where we link the Actity to its XML file with:

```
setContentView(R.layout.activity_constant);
```

Then it is related the attributes of the activity to the corresponding widget in the layout with findViewById:

```
btnEnviar = findViewById(R.id.btnEnviar);
backButton = findViewById(R.id.backButton);
linechart = findViewById(R.id.Linechart);
```

Some functionalities are also added, such as the onClickListener on the "Send" button and the icon to return to the previous screen. When the “Send” button is pressed, a toast will be displayed with the message “Tus datos se han enviado correctamente”. In the case of the icon, the finish() method will be executed, which closes the current screen and returns to the previous one.

```
btnEnviar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast toast = Toast.makeText(getApplicationContext(), "Tus 
datos se han enviado correctamente", Toast.LENGTH_LONG);
        toast.show();
    }
});
backButton.setOnClickListener(v -> finish());
```

Next, the initData(), initLineChart() and initRecyclerData() methods are executed, which prepare the data to be displayed on the chart and in the RecyclerView, respectively.

6. Execution schedule

A research project requires a lot of time and dedication on the part of the author, the director, the tutor and the different supervisors who collaborate in the project. That is why it is necessary to plan the activities to be carried out, in order to try to keep to the previously established timetables and to recognize the exact point at which work is being done.

For this purpose, a WBS (Work Breakdown Structure) and its dictionary to qualify the tasks, a PERT (Program Evaluation and Review Technique) and its diagram to determine the times and a GANTT to evaluate the fulfilment of the schedules have been carried out.

6.1 WBS

As its name indicates, the Work Breakdown Structure or WBS is a technique that allows breaking down any project, in this case, the end-of-degree project, into smaller tasks to make activities more manageable and approachable. It determines and describes each part of the global project, enabling a good organization and planning of the activities involved[xxxiv].

The WBS of this project is structured into three levels. The first level encloses the entire project, in this case, the development of a mobile application for android to help prevent de development of chronic diseases and facilitate patients' day-to-day life. The second level encompasses the project's main blocks, going from the description of the context the project is framed into the delivery report. The final and third level is made up of the small tasks included in each block found in the second layer.

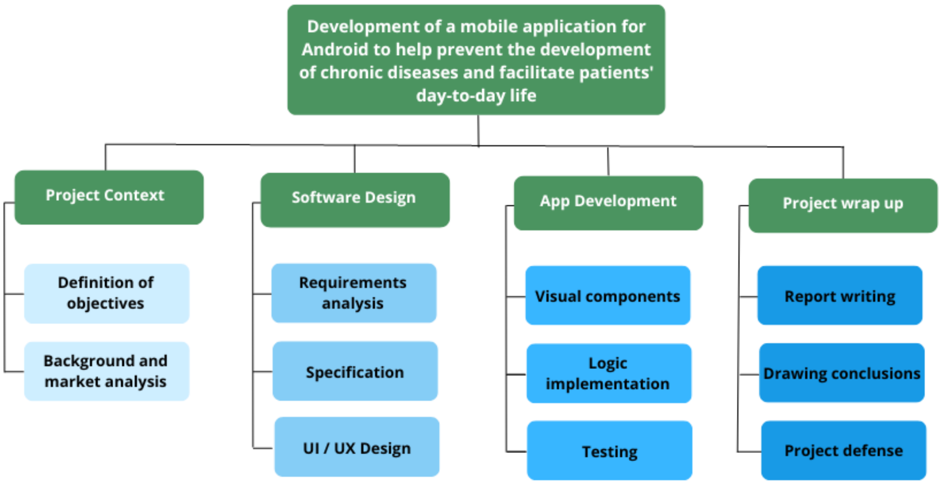


Figure 12. WBS of the project

6.2 Program Evaluation and Review Techniques (PERT)

A PERT diagram, based on making a temporary network, is designed to analyse and represent the tasks involved in completing the project. It is elaborated from a task matrix where activities defined in the Work

Breakdown Structure are assigned a duration[xxxv]. Furthermore, relationships of precedence and dependency are established between tasks.

Activity	Identifier	Duration (weeks)	Previous activities	Following activities
Definition of objectives	A	1	-	B
Background and market analysis	B	2	A	C
Requirements analysis	C	2	B	D
Specification	D	2	C	E
UI / UX Design	E	1	D	F, G
Visual components	F	3	E	H
Logic implementation	G	4	E	H
Testing	H	1	F, G	I
Drawing conclusions	I	1	H	J
Report writing	J	Whole project	I	-
Project defense	K	0,5	I, J	-

Table 5. PERT diagram table of this project

The PERT diagram, shown in Figure 31, was then developed from the matrix. In each node, two main values can be found: on the bottom left (in green), one can see the "Early time", which is the minimum time needed to reach a node and, on the bottom, right (in blue), one can see the "Late time" which is the maximum time that we can take to reach a node without the project being delayed.

Furthermore, the arrows in orange indicate the "Critical path", that is, the minimum time necessary to carry out the project, therefore, the longest time between the initial and final node.

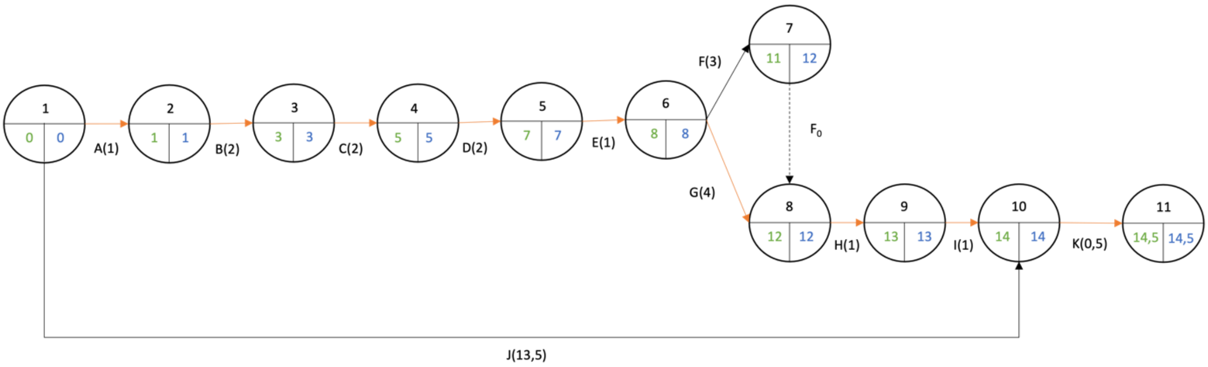
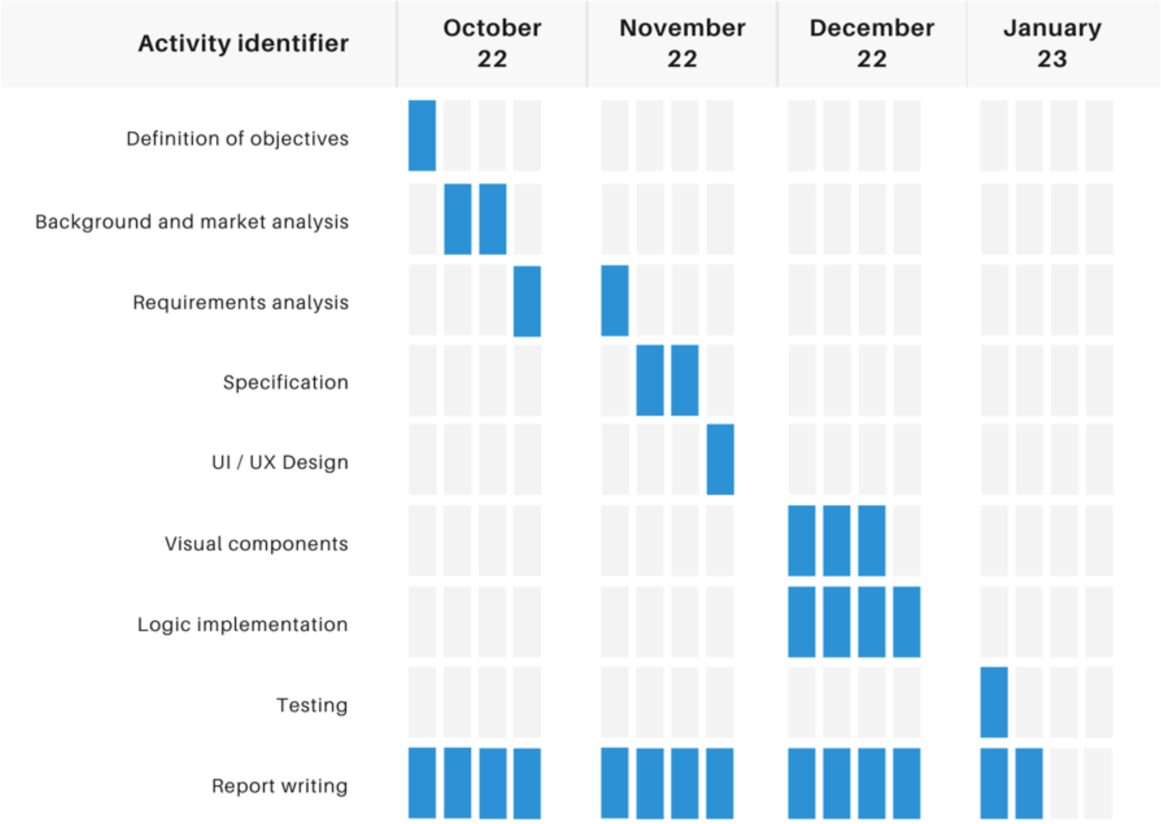


Figure 13. PERT diagram of this project

6.3 GANTT Diagram

A GANTT diagram is a practical way of displaying the project's tasks against time[xxxvi], which is weeks in the charts shown in Figure. On the left of the chart, one can see the list of the activities, and along the top, one can see the time scale. Each activity is represented by a bar, where the position and length of the bar reflect the start, duration and end of the activity.



Activity identifier	October 22				November 22				December 22				January 23			
Definition of objectives	■															
Background and market analysis		■	■													
Requirements analysis				■	■											
Specification						■	■									
UI / UX Design								■								
Visual components									■	■	■					
Logic implementation									■	■	■	■				
Testing													■			
Report writing	■	■	■	■	■	■	■	■	■	■	■	■	■	■		

Figure 14. GANTT chart of this project

7. Technical and economic feasibility

In this section, an analysis was undertaken to determine the technical feasibility aspects of the project. Furthermore, this section shows the analysis of the economic aspects of the project to have an idea of the expenses, which made the realisation of the project possible.

7.1 Technical feasibility

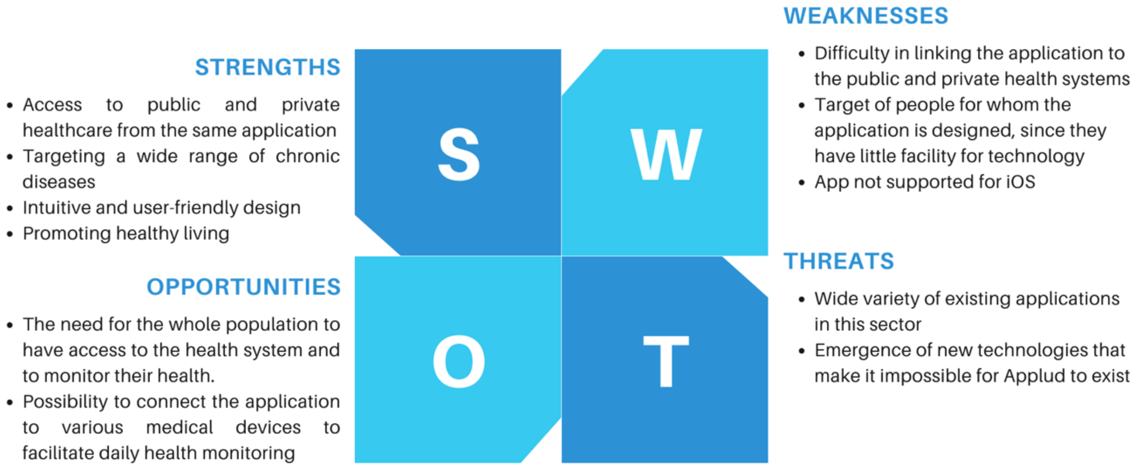
To carry out this project it is necessary to have the Android Studio IDE installed on the computer in order to develop the mobile application for Android operating systems. This IDE is completely free and anyone with a computer can have access to it.

Another tool highly recommended for its usefulness is Figma, a graphics editor used to design prototypes. It is a tool widely used by all designers, whether web or mobile. It can be accessed directly from the web and there is no need to download any software onto the computer. It offers a premium subscription, but with the free subscription you can access most of the functionalities required for a basic design, as in this case.

It can therefore be concluded with high confidence that this project is technically feasible. However, one difficulty that will arise in the development of the work is undoubtedly the difficulty of connecting the back-end of the application with the front-end.

7.1.1 SWOT analysis

A SWOT will allow the assessment of the project's Strengths, Weaknesses, Opportunities, and Threats. Carrying out an internal and external analysis provides a realistic point of view of the situation. Understanding what the project lacks, checking its best advantages, and seeing potential opportunities and threats will create complete awareness of all factors involved.



STRENGTHS
• Access to public and private healthcare from the same application
• Targeting a wide range of chronic diseases
• Intuitive and user-friendly design
• Promoting healthy living
WEAKNESSES
• Difficulty in linking the application to the public and private health systems
• Target of people for whom the application is designed, since they have little facility for technology
• App not supported for iOS
S
W
O
T
OPPORTUNITIES
• The need for the whole population to have access to the health system and to monitor their health.
• Possibility to connect the application to various medical devices to facilitate daily health monitoring
THREATS
• Wide variety of existing applications in this sector
• Emergence of new technologies that make it impossible for Applud to exist

Figure 15. SWOT analysis of Applud

- Strengths

  As discussed in the state of play, one of the drawbacks for people with a chronic illness is the fragmentation of the system, as they have to visit many different professionals who operate independently. The great differential point of this project is that it will allow access to both public and private healthcare from the same application.
  Another differential point is that it breaks with the specificity of the applications offered on the market, as it is not oriented to a type of chronic disease but aims to be a tool for all users, both healthy and those suffering from a disease. Healthy users can find in the app a tool to prevent the development of diseases and tips for health care.

  Its intuitive interface will make it easy for all users, regardless of their technological experience, to use the application. This breaks down the technological barrier that older people have when accessing the healthcare system.

- Weaknesses

  The great difficulty of this project is to connect the front-end of the application, which is what is presented in this final degree project, with the back-end that serves the app. It is necessary to integrate the technological support of the public health system in addition to the support for the private system. Apart from the technical difficulty that this entails, an added problem is the lack of cohesion that exists between the IT systems of the private centres.

  Another weakness is the users themselves, since the application is aimed at an audience that is not accustomed to the use of technology and this can hinder the success of the app.

- Opportunities

  The application has several functionalities for daily health monitoring, such as blood pressure, weight, etc. Currently these data are entered manually by the user once measured by the corresponding device. A great market opportunity is to be able to connect these devices to the app for instant communication.

- Threats

  Since health is a common good, there are already several applications offering similar services in this sector. In addition, there are many pharmaceutical companies, with extensive experience and resources, that have a presence in this field.

7.2 Economic feasibility

As mentioned above, to carry out the project it is only necessary to have a computer to download the Android Studio IDE and to have an internet connection to work with the Figma graphics editor.

No other equipment is needed to carry out the work. Therefore, the cost of the project is derived directly from the hours of work that an engineer has dedicated (16 weeks and 19h each week have been spent), in addition to the meetings that have been held with different professionals to resolve doubts that have arisen during the development of the project.

Four 1-hour meetings with the project tutor were held. Six 1.5-hour meetings were held with qualified computer engineers to resolve technical queries.
A summary of the total cost is shown in the table below.

Role	Price/hour	Hours spent	Amount
Biomedical Enginnering student	20€/h	304h	6080€
Supervisor	150€/h	4h	600€
Computer engineers	150€/h	9h	1350€
		Total cost	**8030€**

Table 6. Total project cost table

8. Conclusions and next steps

The aim of this final degree project was to carry out a study of the functional and economic viability of a mobile application aimed at health care. Specifically, the application aims to be a tool to facilitate the daily lives of people suffering from chronic diseases, as well as promoting a healthy lifestyle to help prevent them.

The main chronic diseases suffered in Catalonia are heart disease, tumours, diseases of the respiratory system and diabetes. As stated in the statistics we have presented, 60% of the population over 55 years of age suffers from one of these diseases.

Such is the impact of these diseases on society that public and private health systems are promoting various initiatives to help prevent risk factors and avoid the development of these diseases in healthy people. One example is the health plan of the Public Health System of Catalonia for the years 2021-2025. This document details, among other issues, the health priorities of the system. Two of the nine priorities refer to chronic diseases and the promotion of a healthy lifestyle.

It is for this reason that this project has been carried out and that we wanted to study how an application such as Applud can contribute to this society's concern. For this purpose, a study of the daily situation of all the people living in this context (patients, relatives and carers) has been carried out and their needs have been identified. The main disadvantages that they mention are listed in the table 3.

On the other hand, a market study has been carried out to detect which of these needs are being covered by current technology and what functionalities Applud can include. Following this analysis, it has been identified that the main functionalities to be offered by Applud are as follows:

- Access to public and private healthcare from the same application
- Facilitating health monitoring by adding vital signs data (blood pressure, weight, temperature, oxygen saturation, etc.)
- Promoting healthy lifestyle through articles and information
- Intuitive and user-friendly design, given that people accessing the application often have little experience with technology

Once these key functionalities were identified, the application was designed using Figma and the code was developed using the Android Studio IDE. At the end of development, the executable file was generated, thus fulfilling another of the project's objectives.

With this work done, it can be stated that this project is feasible and can be further developed. The next step is to develop the back-end services that will support the application and connect it to the different health IT systems.

9. References

[i] Defunciones según la causa de muerte. Instituto Nacional de Estadística. 10 November 2021. Available from: https://www.ine.es/prensa/edcm_2020.pdf

[ii] Personas con enfermedades o problemas de salud de larga duración por CCAA y periodo. Instituto Nacional de Estadística. Available from: https://www.ine.es/jaxiPx/Datos.htm?path=/t00/ICV/Graficos/dim3/l0/&file=314G1.px#!tabs-tabla

[iii] Prevalença de malaltia o problema de salud crònic. Observatori del Sistema de Salut de Catalunya. Available from:https://observatorisalut.gencat.cat/ca/detalls/article/08_IND_prevalenca_malaltia_problema_salut_cronic-00002

[iv] Kumar, V. (2016). What Is The Difference Between A Mobile App And A Web App?

[v] Randhawa, Tejinder S. Mobile Applications: Design, Development and Optimization. Cham: Springer International Publishing AG, 2022. Print.

[vi] Mroczkowska, A. (2021). What Is a Mobile App? | App Development Basics for Businesses. Droids On Roids, 1–18. Retrieved from https://www.thedroidsonroids.com/blog/what-is-a-mobile-app-app-development-basics-for-businesses

[vii] Terry, K. (2015). Prescribing mobile apps: What to consider. Medical Economics, 92(12).

[viii] Yugandhar, D., & J.Kiran Kumar. (2017). A Study on Current Mobile Operating Systems. International Journal of Scientific & Engineering Research, 8(5), 72–78.

[ix] Succi, G., & Ivanov, V. V. (2018). Comparison of mobile operating systems based on models of growth reliability of the software. Computer Research and Modeling, 10(3), 325–334. https://doi.org/10.20537/2076-7633-2018-10-3-325-334

[x] Divyap, K., & Venkata Krishnakumar, S. (2016). COMPARATIVE ANALYSIS OF SMART PHONE OPERATING SYSTEMS ANDROID, APPLE iOS AND WINDOWS. International Journal of Scientific Engineering and Applied Science, (22), 2395–3470. Retrieved from www.ijseas.com

[xi] Almisreb, A., Hadžo Mulalić, H., Mučibabić, N., & Numanović, R. (2019). A review on mobile operating systems and application development platforms. Sustainable Engineering and Innovation, 1(1), 49–56. https://doi.org/10.37868/sei.v1i1.94

[xii] R. (2018). A Comparative Analysis of mobile Operating Systems. International Journal of Computer Sciences and Engineering, 6(12), 69–73. https://doi.org/10.26438/ijcse/v6i12.6973

[xiii] González García, C., Pascual-Espada, J., Pelayo G-Bustelo, C., & Cueva-Lovelle, J. M. (2015). Swift vs. Objective-C: A New Programming Language. International Journal of Interactive Multimedia and Artificial Intelligence, 3(3), 74. https://doi.org/10.9781/ijimai.2015.3310

[xiv] Ardito, L., Coppola, R., Malnati, G., & Torchiano, M. (2020). Effectiveness of Kotlin vs. Java in android app development tasks. Information and Software Technology, 127. https://doi.org/10.1016/j.infsof.2020.106374

[xv] Góis Mateus, B., & Martinez, M. (2019). An empirical study on quality of Android applications written in Kotlin language. Empirical Software Engineering, 24(6), 3356–3393. https://doi.org/10.1007/s10664-019-09727-4

[xvi] Tewari, A., & Singh, P. (2021). Android App Development: A Review. Journal of Management and Service Science (JMSS), 1(2), 1–6. https://doi.org/10.54060/jmss/001.02.006

[xvii] Devandroid. (2019). Introducción a Android Studio. Google, LLC, 15. Retrieved from https://developer.android.com/studio/intro?hl=es-419

[xviii] Aman, S. N., Yasin, N. Z. M., & Azizan, I. (2018). Quality Factor between Visual Studio and Android Studio for Developing Video Control Based on Hand Gestures. International Journal of Trend in Scientific Research and Development, Special Issue(Special Issue-ICAEIT2017), 181–184. https://doi.org/10.31142/ijtsrd19134

[xix] Vuorre, M., & Curley, J. P. (2018). Curating Research Assets: A Tutorial on the Git Version Control System. Advances in Methods and Practices in Psychological Science, 1(2), 219–236. https://doi.org/10.1177/2515245918754826

[xx] Atlassian. (s. f.). Qué es Git | Atlassian Git Tutorial. https://www.atlassian.com/es/git/tutorials/what-is-git

[xxi] Git. (s. f.). https://git-scm.com/

[xxii] Departament de Salut de Catalunya (2021). Pla de Salut de Catalunya. https://salutweb.gencat.cat/web/.content/_departament/pla-de-salut/pla-de-salut-2021-2025/pla-salut-catalunya-2021-2025.pdf

[xxiii] Observatori del Sistema de Salut de Catalunya. Determinants socials i econòmics de la sa-Retrospectiva de trenta anys de la planifcació sanitària a Catalunya: el motor de transforlut: efectes de la crisi econòmica en la salut de la població de Catalunya. Barcelona: Agència mació del sistema de salut. Barcelona: Direcció General de Planifcació en Salut; 2020. Disde Qualitat i Avaluació Sanitàries de Catalunya; 2014. Available from:
http://observatorisalut.gencat.cat/web/.content/minisite/observatorisalut/contingutsad-
https://salutweb.gencat.cat/web/.content/_departament/pla-de-salut/Pla-de-sa-
ministratius/observatori_efectes_crisi_salut_document.pdf

[xxiv] Departament de Salut de Catalunya (2020). La visió de les persones sobre l’atenció a la gent gran i a la cronicitat. Estudi qualitatiu per construir un nou model d’atenció. Available from: https://scientiasalut.gencat.cat/handle/11351/7008

[xxv] National Voices [Internet]. London: National Voices. Principles for integrated care. Available from: https://www.nationalvoices.org.uk/publications/our-publications/principlesintegrated-care

[xxvi] Lee, M., Lee, H., Kim, Y., Kim, J., Cho, M., Jang, J., & Jang, H. (2018, December 13). Mobile app-based health promotion programs: A systematic review of the literature. International Journal of Environmental Research and Public Health. MDPI AG. https://doi.org/10.3390/ijerph15122838

[xxvii] Portenhauser, A. A., Terhorst, Y., Schultchen, D., Sander, L. B., Denkinger, M. D., Stach, M., … Messner, E. M. (2021, January 1). Mobile apps for older adults: Systematic search and evaluation within online stores. JMIR Aging. JMIR Publications Inc. https://doi.org/10.2196/23313

[xxviii] N Inukollu, V., Keshamon, D. D., Kang, T., & Inukollu, M. (2014). Factors Influncing Quality of Mobile Apps: Role of Mobile App Development Life Cycle. International Journal of Software Engineering & Applications, 5(5), 15–34. https://doi.org/10.5121/ijsea.2014.5502

[xxix] Eeles, P. (2001). What Is an Architectural Requirement? The FURPS + System for Classifying Requirements. The Rational Edge. Retrieved from http://itculiacan.webs.com/descargas/z_FURPS Requisites.pdf

[xxx] Mule, S. S., & Waykar, Y. (2015). Role of use case diagram in software development. International Journal of Management and Economics, 6. Retrieved from: https://www.researchgate.net/publication/322991847_role_of_use_case_diagram_in_software_development

[xxxi] Hutauruk, M. K. (2019). UML Diagram : Use Case Diagram. BINUS University. Retrieved from https://socs.binus.ac.id/2019/11/26/uml-diagram-use-case-diagram/

[xxxii] Vidgen, R., & Goodwin, S. (2000). XML: What is it good for? Computing and Control Engineering Journal, 11(3), 119–124. https://doi.org/10.1049/cce:20000305

[xxxiii] Diseños | Desarrolladores de Android |. (s. f.). Android Developers. https://developer.android.com/guide/topics/ui/declaring-layout?hl=es-419

[xxxiv] Work Breakdown Structure [Internet]. workbreakdownstructure.com. 2021 [cited 13 June 2021]. Available from: https://www.workbreakdownstructure.com/

[xxxv] Ultimate Guide to PERT [Internet]. ProjectManager.com. 2021 [cited 10 August 2021]. Available from: https://www.projectmanager.com/pert-chart

[xxxvi] What Is A Gantt Chart? | Definition & Examples | APM [Internet]. Apm.org.uk. 2021 [cited 8 August 2021]. Available from: https://www.apm.org.uk/resources/find-a-resource/gantt-chart/

10. Annexes

The complete project can be consulted and downloaded at https://github.com/bbachs/TFG_EBM.git

1. Logic implementation

- Register Activity

```
package com.example.features.login_enrolment;

import android.app.DatePickerDialog;
import android.app.Dialog;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.DialogFragment;

import com.example.features.mainMenu.MainMenu;
import com.example.tfg.R;

import java.util.Calendar;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class RegisterActivity extends AppCompatActivity {
    EditText name;
    EditText email;
    CheckBox terms;
    EditText date;
    EditText pwd;
    EditText pwdBis;
    Button registerBtn;
    TextView goLogin;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.register_activity);

        name = findViewById(R.id.et_name);
        email = findViewById(R.id.et_email);
        date = findViewById(R.id.et_birthdate);
        pwd = findViewById(R.id.et_password);
        pwdBis = findViewById(R.id.et_passwordBis);
        terms = findViewById(R.id.cb_terms);
        registerBtn = findViewById(R.id.btn_login);
        goLogin = findViewById(R.id.goLogin);

        goLogin.setOnClickListener(v -> go2Login());
        date.setOnClickListener(v -> showDatePickerDialog());
        registerBtn.setOnClickListener(v -> register());
    }

    private void go2Login(){
        Intent intent = new Intent(this, LoginActivity.class);
        startActivity(intent);
    }

    private void showDatePickerDialog() {
        DatePickerFragment newFragment = DatePickerFragment.newInstance(new
DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePicker datePicker, int year, int month, int day) {
                // +1 because January is zero
                final String selectedDate = day + " / " + (month+1) + " / " + year;
                date.setText(selectedDate);
            }
        });
        newFragment.show(getSupportFragmentManager(), "datePicker");
    }

    private void register(){
            String nom = name.getText().toString();
            String correu = email.getText().toString();
            String password = pwd.getText().toString();
            String confirm_password = pwdBis.getText().toString();

            if(nom.equals("") || correu.equals("") || password.equals("") ||
confirm_password.equals("")) {
                Toast.makeText(getApplicationContext(), "Nombre, correo y contraseña
requeridos", Toast.LENGTH_SHORT).show();
            } else {
                if(password.equals(confirm_password)) {
                    if (isMail(correu)){
                        if (isPasswordSegur(password)) {
                            if (terms.isChecked()) {
                                Toast.makeText(getApplicationContext(), "Registrado con
éxito", Toast.LENGTH_SHORT).show();
                                //Fem l'intent
                                Intent processar_main = new Intent(this, MainMenu.class);
                                startActivity(processar_main);
                            } else {
                                Toast.makeText(getApplicationContext(), "Tienes que aceptar
los términos y condiciones", Toast.LENGTH_SHORT).show();
                            }
                        } else {
                            Toast.makeText(getApplicationContext(),"Contraseña - mínimo 8
carácteres, 1 mayúscula, 1 minúscula, 1 número.", Toast.LENGTH_SHORT).show();
                        }
                    } else {
                        Toast.makeText(getApplicationContext(), "Correo no válido.",
Toast.LENGTH_SHORT).show();
                    }
                } else {
                    Toast.makeText(getApplicationContext(), "Las contraseñas no coinciden.",
Toast.LENGTH_SHORT).show();
                }
            }
    }

    /* Classe per a comprovar si un mail és valid (minim una majuscula,
     * format correcte, etc)
     * @param correu
     * */
    public boolean isMail(String correu) {
        Pattern pattern = Pattern.compile("^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-]+)*@"
                + "[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)*(\\.[A-Za-z]{2,})$");
        Matcher matcher = pattern.matcher(correu);
        return matcher.find();
    }

    /* Classe per a comprovar si una contrasenya és valida (format correcte, etc)
     * @param password
     * */
    public boolean isPasswordSegur(String password) {
        Pattern pattern = Pattern.compile("^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9]).{8,}$");
        Matcher matcher = pattern.matcher(password);
        return matcher.find();
    }


    public static class DatePickerFragment extends DialogFragment {

        private DatePickerDialog.OnDateSetListener listener;

        public static DatePickerFragment newInstance(DatePickerDialog.OnDateSetListener
listener) {
            DatePickerFragment fragment = new DatePickerFragment();
            fragment.setListener(listener);
            return fragment;
        }

        public void setListener(DatePickerDialog.OnDateSetListener listener) {
            this.listener = listener;
        }

        @Override
        public Dialog onCreateDialog(Bundle savedInstanceState) {
            // Use the current date as the default date in the picker
            final Calendar c = Calendar.getInstance();
            int year = c.get(Calendar.YEAR);
            int month = c.get(Calendar.MONTH);
            int day = c.get(Calendar.DAY_OF_MONTH);

            // Create a new instance of DatePickerDialog and return it
            return new DatePickerDialog(requireContext(), listener, year, month, day);
        }
    }

}
```

- Login Activity

```
package com.example.features.login_enrolment;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

import com.example.features.mainMenu.MainMenu;
import com.example.tfg.R;

import java.util.concurrent.TimeUnit;

public class LoginActivity extends AppCompatActivity {

    EditText mail;
    EditText pwd;
    Button loginBtn;
    TextView goRegister;
    ProgressBar progressBar;
    RelativeLayout screen;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.login_activity);

        screen = findViewById(R.id.screen);

        progressBar = findViewById(R.id.circular_progressbar);
        progressBar.setVisibility(View.GONE);

        mail = findViewById(R.id.et_email);
        pwd = findViewById(R.id.et_password);
        loginBtn = findViewById(R.id.btn_login);
        goRegister = findViewById(R.id.goRegister);

        goRegister.setOnClickListener(v -> go2Register());
        loginBtn.setOnClickListener(v -> go2MainMenu());
    }

    private void go2MainMenu(){
        if(mail.getText().toString().equals("Beatriz") &&
pwd.getText().toString().equals("1")){
            //progressBar.setVisibility(View.GONE);
            Intent intent = new Intent(this, MainMenu.class);
            startActivity(intent);
        }else{
            //progressBar.setVisibility(View.GONE);
            Toast.makeText(getApplicationContext(), "Usuario o contraseña incorrectos",
Toast.LENGTH_SHORT).show();
        }
    }

    private void go2Register(){
        Intent intent = new Intent(this, RegisterActivity.class);
        startActivity(intent);
    }
}
```

- MainMenuActivity.java

```
package com.example.features.mainMenu;

import android.os.Bundle;
import android.view.Menu;

import com.example.tfg.R;
import com.example.tfg.databinding.ActivityMainMenuBinding;
import com.google.android.material.navigation.NavigationView;

import androidx.core.view.MenuCompat;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.appcompat.app.AppCompatActivity;

public class MainMenu extends AppCompatActivity {

    public static String currentUser;
    private AppBarConfiguration mAppBarConfiguration;
    private ActivityMainMenuBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainMenuBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        setSupportActionBar(binding.appBarMainMenu.toolbar);
        DrawerLayout drawer = binding.drawerLayout;
        NavigationView navigationView = binding.navView;
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        mAppBarConfiguration = new AppBarConfiguration.Builder(
                R.id.nav_home, R.id.nav_calendar, R.id.nav_constants,
R.id.nav_appointments, R.id.nav_nursing,
                R.id.nav_checkPoints, R.id.nav_closeApp)
                .setOpenableLayout(drawer)
                .build();
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment_content_main_menu);
        NavigationUI.setupActionBarWithNavController(this, navController,
mAppBarConfiguration);
        NavigationUI.setupWithNavController(navigationView, navController);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main_menu, menu);
        return true;
    }

    @Override
    public boolean onSupportNavigateUp() {
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment_content_main_menu);
```

```
        return NavigationUI.navigateUp(navController, mAppBarConfiguration)
                || super.onSupportNavigateUp();
    }
}
```

- Appointments Fragment.java

```
package com.example.features.mainMenu.ui.appointments;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentStatePagerAdapter;
import androidx.lifecycle.ViewModelProvider;
import androidx.viewpager.widget.ViewPager;

import com.example.tfg.R;
import com.example.tfg.databinding.FragmentAppointmentsBinding;

public class AppointmentsFragment extends Fragment {
    private FragmentAppointmentsBinding binding;
    AppointmentsPagerAdapter appointmentsAdapter;
    ViewPager viewPager;

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {

        binding = FragmentAppointmentsBinding.inflate(inflater, container, false);
        View root = binding.getRoot();
        return root;
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        appointmentsAdapter = new AppointmentsPagerAdapter(getChildFragmentManager());
        viewPager = view.findViewById(R.id.pager);
        viewPager.setAdapter(appointmentsAdapter);
    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
```

- AppointmentsPagerFragment.java

```
package com.example.features.mainMenu.ui.appointments;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.tfg.R;

import java.util.ArrayList;

public class AppointmentPagerFragment extends Fragment {
    public static final String ARG_OBJECT = "object";

    private RecyclerView recyclerViewCaps;
    private AppointmentsRecyclerAdapter adapter;

    @Override
    public View onCreateView(LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_appointments_pager_fragment, container,
false);
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        Bundle args = getArguments();
        ArrayList<CapsModel> datos = (ArrayList<CapsModel>)
args.getSerializable(ARG_OBJECT);

        recyclerViewCaps = view.findViewById(R.id.recyclerAppointments);
        recyclerViewCaps.setLayoutManager(new LinearLayoutManager(getContext()));
        adapter = new AppointmentsRecyclerAdapter(datos);
        recyclerViewCaps.setAdapter(adapter);
    }
}
```

- AppointmentsPagerAdapter.java

```
package com.example.features.mainMenu.ui.appointments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentPagerAdapter;

import java.util.ArrayList;


public class AppointmentsPagerAdapter extends FragmentPagerAdapter {

    private ArrayList<CapsModel> privadaArray;
    private ArrayList<CapsModel> publicaArray;

    public AppointmentsPagerAdapter(FragmentManager fm) {
        super(fm);
        initData();
    }

    private void initData(){
        publicaArray = new ArrayList<CapsModel>();
        CapsModel pb1 = new CapsModel("Cap Gràcia", "CR LARRARD, 1", "08024 BARCELONA");
        CapsModel pb2 = new CapsModel("Cap Manso", "CR MANSO, 19", "08015 BARCELONA");
        CapsModel pb3 = new CapsModel("Cap Adrià", "CR VALLMAJOR, 34", "08021 BARCELONA");
        CapsModel pb4 = new CapsModel("Cap Trinitat Vella", "VI BARCINO, 88", "08033
BARCELONA");
        publicaArray.add(pb1);
        publicaArray.add(pb2);
        publicaArray.add(pb3);
        publicaArray.add(pb4);

        privadaArray = new ArrayList<CapsModel>();
        CapsModel pv1 = new CapsModel("Quirón Salud", "CR LARRARD, 1", "08024 BARCELONA");
        CapsModel pv2 = new CapsModel("Clínica Corachan", "CR MANSO, 19", "08015
BARCELONA");
        CapsModel pv3 = new CapsModel("Assistència Sanitaria", "CR VALLMAJOR, 34", "08021
BARCELONA");
        CapsModel pv4 = new CapsModel("Sanitas", "VI BARCINO, 88", "08033 BARCELONA");
        privadaArray.add(pv1);
        privadaArray.add(pv2);
        privadaArray.add(pv3);
        privadaArray.add(pv4);
    }

    @Override
    public Fragment getItem(int i) {
        Fragment fragment = new AppointmentPagerFragment();
        Bundle args = new Bundle();
        // Our object is just an integer :-P
        if(i == 0){
            //args.putInt(AppointmentPagerFragment.ARG_OBJECT, i + 10);
            args.putSerializable(AppointmentPagerFragment.ARG_OBJECT, publicaArray);
        }
        else{
            args.putSerializable(AppointmentPagerFragment.ARG_OBJECT, privadaArray);
        }

        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public int getCount() {
        return 2;
    }

    @Override
    public CharSequence getPageTitle(int position) {
        return (position == 0) ? "Sanidad pública" : "Sanidad privada";
    }
}
```

- AppointmentsRecyclerAdapter.java

```
package com.example.features.mainMenu.ui.appointments;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import com.example.tfg.R;

import java.util.ArrayList;

public class AppointmentsRecyclerAdapter extends
RecyclerView.Adapter<AppointmentsRecyclerAdapter.ViewHolder>{

    private ArrayList<CapsModel> datos;

    public AppointmentsRecyclerAdapter(ArrayList<CapsModel> datos) {
        this.datos = datos;
    }

    @Override
    public AppointmentsRecyclerAdapter.ViewHolder onCreateViewHolder(ViewGroup viewGroup,
int viewType) {
        // Create a new view.
        View v = LayoutInflater.from(viewGroup.getContext())
                .inflate(R.layout.item_appointments, viewGroup, false);

        return new AppointmentsRecyclerAdapter.ViewHolder(v);
    }

    @Override
    public void onBindViewHolder(AppointmentsRecyclerAdapter.ViewHolder viewHolder, final
int position) {
        viewHolder.nombreCap.setText(datos.get(position).getNombreCap());
        viewHolder.direccionCap.setText(datos.get(position).getDireccionCap());
        viewHolder.cpCap.setText(datos.get(position).getCpCap());
    }

    @Override
    public int getItemCount() {
        return datos.size();
    }

    public static class ViewHolder extends RecyclerView.ViewHolder {
        private final TextView nombreCap;
        private final TextView direccionCap;
        private final TextView cpCap;

        public ViewHolder(View v) {
            super(v);
            nombreCap = (TextView) v.findViewById(R.id.nombreCap);
            direccionCap = (TextView) v.findViewById(R.id.direccionCap);
            cpCap = (TextView) v.findViewById(R.id.cpCap);
        }
    }
}
```

- CapsModel.java

```
package com.example.features.mainMenu.ui.appointments;

import android.os.Parcelable;

import java.io.Serializable;

public class CapsModel implements Serializable {
    private String nombreCap;
    private String direccionCap;
    private String cpCap;

    public CapsModel(String nombre, String direccion, String cp){
        this.nombreCap = nombre;
        this.direccionCap = direccion;
        this.cpCap = cp;
    }

    public String getNombreCap() {
        return nombreCap;
    }

    public String getDireccionCap() {
        return direccionCap;
    }

    public String getCpCap() {
        return cpCap;
    }
}
```

- CalendarFragment.java

```
package com.example.features.mainMenu.ui.calendar;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.ViewModelProvider;

import com.example.tfg.databinding.FragmentCalendarBinding;

public class CalendarFragment extends Fragment {

    private FragmentCalendarBinding binding;

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {
        CalendarViewModel calendarViewModel =
                new ViewModelProvider(this).get(CalendarViewModel.class);

        binding = FragmentCalendarBinding.inflate(inflater, container, false);
        View root = binding.getRoot();

        return root;
    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
```

- CheckpointsFragment.java

```
package com.example.features.mainMenu.ui.checkPoints;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;

import com.example.tfg.R;
import com.example.tfg.databinding.FragmentCheckpointsBinding;

public class CheckPointsFragment extends Fragment {

    private FragmentCheckpointsBinding binding;
    private LinearLayout firstColumn;
    private LinearLayout secondColumn;

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {
        binding = FragmentCheckpointsBinding.inflate(inflater, container, false);
        View root = binding.getRoot();

        firstColumn = root.findViewById(R.id.firstColumn);
        firstColumn.setOnClickListener(v -> Toast.makeText(getContext(), "Esta 
funcionalidad aún no está disponible", Toast.LENGTH_SHORT).show());
        secondColumn = root.findViewById(R.id.secondColumn);
        secondColumn.setOnClickListener(v -> Toast.makeText(getContext(), "Esta 
funcionalidad aún no está disponible", Toast.LENGTH_SHORT).show());

        return root;
    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
```

- ConstantsActivity.java

```
package com.example.features.mainMenu.ui.constants;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.Toast;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.tfg.R;
import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;

import java.util.ArrayList;
import java.util.Arrays;
public class ConstantActivity extends AppCompatActivity {
    private LineChart linechart;
    private ArrayList<Float> diastoleData;
    private ArrayList<Float> sistoleData;
    private ArrayList<String> fechas;
    private RecyclerView recyclerSingleMedidas;
    private ConstantSingleValueAdapter adapter;
    private Button btnEnviar;
    private ImageButton backButton;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_constant);

        btnEnviar = findViewById(R.id.btnEnviar);
        btnEnviar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast toast = Toast.makeText(getApplicationContext(), "Tus datos se han
enviado correctamente", Toast.LENGTH_LONG);
                toast.show();
            }
        });

        backButton = findViewById(R.id.backButton);
        backButton.setOnClickListener(v -> finish());

        linechart = findViewById(R.id.linechart);
        initData();
        initLineChart();
        initRecyclerData();
    }

    private void initData(){
        fechas = new ArrayList<String>(Arrays.asList("29/01/22", "28/01/22", "27/01/22",
"26/01/22",
                "25/01/22", "24/01/22", "23/01/22"));

        diastoleData = new ArrayList<Float>(
                Arrays.asList((float) 81, (float) 82, (float) 80, (float) 79,
                        (float) 80, (float) 78, (float) 81));

        sistoleData = new ArrayList<Float>(
                Arrays.asList((float) 120, (float) 115, (float) 123, (float) 125,
                        (float) 129, (float) 127, (float) 125));
    }

    private void initLineChart(){
        ArrayList<Entry> lineEntries2 = new ArrayList<Entry>();
        ArrayList<Entry> lineEntries1 = new ArrayList<Entry>();
        lineEntries2.add(new Entry(1, sistoleData.get(0)));
        lineEntries2.add(new Entry(2, sistoleData.get(1)));
        lineEntries2.add(new Entry(3, sistoleData.get(2)));
        lineEntries2.add(new Entry(4, sistoleData.get(3)));
        lineEntries2.add(new Entry(5, sistoleData.get(4)));
        lineEntries2.add(new Entry(6, sistoleData.get(5)));
        lineEntries2.add(new Entry(7, sistoleData.get(6)));

        lineEntries1.add(new Entry(1, diastoleData.get(0)));
        lineEntries1.add(new Entry(2, diastoleData.get(1)));
        lineEntries1.add(new Entry(3, diastoleData.get(2)));
        lineEntries1.add(new Entry(4, diastoleData.get(3)));
        lineEntries1.add(new Entry(5, diastoleData.get(4)));
        lineEntries1.add(new Entry(6, diastoleData.get(5)));
        lineEntries1.add(new Entry(7, diastoleData.get(6)));

        //Genera el nou Linechart amb les hores aparcades per l'usuari
        LineDataSet lineDataSet2=new LineDataSet(lineEntries2, "Sistólica");

        LineDataSet lineDataSet1=new LineDataSet(lineEntries1, "Diastólica");
        lineDataSet1.setColor(R.color.purple_200);
        lineDataSet1.setValueTextColor(R.color.purple_200);
        lineDataSet1.setValueTextSize(10f);

        LineData ll= new LineData(lineDataSet2, lineDataSet1);
        ll.setHighlightEnabled(true);
        linechart.setData(ll);
        linechart.invalidate();
    }

    private void initRecyclerData(){
        recyclerSingleMedidas = (RecyclerView) findViewById(R.id.recyclerSingleMedidas);
        recyclerSingleMedidas.setLayoutManager(new LinearLayoutManager(this));
        adapter = new ConstantSingleValueAdapter(fechas, diastoleData, sistoleData, this);
        recyclerSingleMedidas.setAdapter(adapter);
    }


}
```

- ConstantsAdapter.java

```
package com.example.features.mainMenu.ui.constants;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import com.example.tfg.R;

import java.util.ArrayList;

public class ConstantsAdapter extends RecyclerView.Adapter<ConstantsAdapter.ViewHolder>{

    private ArrayList<String> recyclerData;
    private Context context;

    public ConstantsAdapter(ArrayList<String> recyclerData, Context context) {
        this.recyclerData = recyclerData;
        this.context = context;
    }

    @Override
    public ConstantsAdapter.ViewHolder onCreateViewHolder(ViewGroup viewGroup, int
viewType) {
        // Create a new view.
        View v = LayoutInflater.from(viewGroup.getContext())
                .inflate(R.layout.item_constants_medida, viewGroup, false);

        return new ConstantsAdapter.ViewHolder(v);
    }

    @Override
    public void onBindViewHolder(ConstantsAdapter.ViewHolder viewHolder, final int
position) {

        viewHolder.tituloConstante.setText(recyclerData.get(position));

        viewHolder.goArrow.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(context, ConstantActivity.class);
                context.startActivity(intent);
            }
        });
    }

    @Override
    public int getItemCount() {
        return recyclerData.size();
    }

    public static class ViewHolder extends RecyclerView.ViewHolder {
        private final TextView tituloConstante;
        private final ImageView goArrow;

        public ViewHolder(View v) {
            super(v);
            // Define click listener for the ViewHolder's View.
            tituloConstante = (TextView) v.findViewById(R.id.tituloConstante);
            goArrow = (ImageView) v.findViewById(R.id.goArrow);
        }
    }
}
```

- ConstantsFragment.java

```java
package com.example.features.mainMenu.ui.constants;

import android.graphics.Color;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.constraintlayout.widget.ConstraintLayout;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.features.mainMenu.ui.home.HomeAdapter;
import com.example.tfg.R;
import com.example.tfg.databinding.FragmentConstantsBinding;

import org.eazegraph.lib.charts.PieChart;
import org.eazegraph.lib.models.PieModel;

import java.util.ArrayList;

public class ConstantsFragment extends Fragment {

    private FragmentConstantsBinding binding;
    private PieChart chart;
    private RecyclerView recyclerMedidas;
    private ConstantsAdapter adapter;
    private ArrayList<String> recyclerData;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        recyclerData = new ArrayList<>();
        initRecyclerData();
    }

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {
        binding = FragmentConstantsBinding.inflate(inflater, container, false);
        View root = binding.getRoot();

        ConstraintLayout component = root.findViewById(R.id.component);
        chart = component.findViewById(R.id.pieChart);
        initPieChart();

        recyclerMedidas = (RecyclerView) root.findViewById(R.id.recyclerMedidas);
        recyclerMedidas.setLayoutManager(new LinearLayoutManager(root.getContext()));
        adapter = new ConstantsAdapter(recyclerData, getContext());
        recyclerMedidas.setAdapter(adapter);

        return root;
    }

    private void initPieChart() {
        chart.addPieSlice(new PieModel("Integer 1", 70, Color.parseColor("#FFA726")));
        chart.addPieSlice(new PieModel("Integer 1", 30, Color.parseColor("#2986F6")));
        chart.startAnimation();
    }

    private void initRecyclerData(){
        recyclerData.add("Presión arterial");
        recyclerData.add("Saturación oxígeno");
        recyclerData.add("Temperatura");
        recyclerData.add("Peso");
        recyclerData.add("Frecuencia cardíaca");
    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
```

- ConstantSingleValueAdapter.java

```
package com.example.features.mainMenu.ui.constants;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import com.example.tfg.R;

import java.util.ArrayList;

public class ConstantSingleValueAdapter extends
RecyclerView.Adapter<ConstantSingleValueAdapter.ViewHolder>{

    private ArrayList<String> fechas;
    ArrayList<Float> diastole;
    ArrayList<Float> sistole;
    private Context context;

    public ConstantSingleValueAdapter(ArrayList<String> fechas, ArrayList<Float> diastole,
                                      ArrayList<Float> sistole, Context context) {
        this.fechas = fechas;
        this.diastole = diastole;
        this.sistole = sistole;
        this.context = context;
    }

    @Override
    public ConstantSingleValueAdapter.ViewHolder onCreateViewHolder(ViewGroup viewGroup,
int viewType) {
        // Create a new view.
        View v = LayoutInflater.from(viewGroup.getContext())
                .inflate(R.layout.item_constantes_valores_diarios, viewGroup, false);

        return new ConstantSingleValueAdapter.ViewHolder(v);
    }

    @Override
    public void onBindViewHolder(ConstantSingleValueAdapter.ViewHolder viewHolder, final
int position) {
        String valorDiario = String.valueOf(Math.round(sistole.get(position)) + "/" +
String.valueOf(Math.round(diastole.get(position))));
        viewHolder.fecha.setText(this.fechas.get(position));
        viewHolder.valorDiario.setText(valorDiario);
    }

    @Override
    public int getItemCount() {
        return diastole.size();
    }

    public static class ViewHolder extends RecyclerView.ViewHolder {
        private final TextView fecha;
        private final TextView valorDiario;

        public ViewHolder(View v) {
            super(v);
            // Define click listener for the ViewHolder's View.
            fecha = (TextView) v.findViewById(R.id.tituloValorDiario);
            valorDiario = (TextView) v.findViewById(R.id.valorDiario);

        }
    }
}
```

- ArticleActivity.java

```
package com.example.features.mainMenu.ui.home;

import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import com.example.tfg.R;

public class ArticleActivity extends AppCompatActivity {

    private ImageView imageView;
    private TextView name;
    private TextView author;
    private TextView articleText;
    private ImageButton backButton;

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_article);

        imageView = findViewById(R.id.articleImage);
        name = findViewById(R.id.articleName);
        author = findViewById(R.id.articleAuthor);
        articleText = findViewById(R.id.articleText);
        backButton = findViewById(R.id.backButton);

        Bundle extras = getIntent().getExtras();
        if(extras != null){
            if(extras.containsKey("ArticleModel")){
                ArticleModel articleModel = (ArticleModel)
extras.getSerializable("ArticleModel");
                setData(articleModel);
            }

        }

        backButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

    }

    private void setData(ArticleModel articleModel){
        name.setText(articleModel.getName());
        author.setText(articleModel.getAuthor());
        articleText.setText(articleModel.getArtText());
    }
}
```

- ArticleModel.java

```
package com.example.features.mainMenu.ui.home;

import java.io.Serializable;

public class ArticleModel implements Serializable {
    private String name;
    private String author;
    private String text;
    private String imgType;

    public String getText() {
        return text;
    }
```

```java
    public void setText(String text) {
        this.text = text;
    }

    public String getImgType() {
        return imgType;
    }

    public void setImgType(String imgType) {
        this.imgType = imgType;
    }

    public ArticleModel(String name, String author){
        this.name = name;
        this.author = author;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setArtText(String text) {
        this.text = text;
    }

    public String getArtText(){
        return text;
    }
}
```

- HomeAdapter.java

```java
package com.example.features.mainMenu.ui.home;

import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;

import com.example.features.login_enrolment.LoginActivity;
import com.example.tfg.R;

import java.io.File;
import java.util.ArrayList;

public class HomeAdapter extends RecyclerView.Adapter<HomeAdapter.ViewHolder>{
    private static final String TAG = "HomeAdapter";

    protected ArrayList<ArticleModel> articlesList;
    private Context context;

    public HomeAdapter(ArrayList<ArticleModel> articlesList, Context context) {
        this.articlesList = articlesList;
        this.context = context;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup viewGroup, int viewType) {
        Log.d(TAG, "onCreateViewHolder");
        // Create a new view.
        View v = LayoutInflater.from(viewGroup.getContext())
                .inflate(R.layout.item_home_articlecard, viewGroup, false);

        return new ViewHolder(v);
    }

    @Override
    public void onBindViewHolder(ViewHolder viewHolder, final int position) {
        Log.d(TAG, "onBindViewHolder");
        Log.d(TAG, "Element " + position + " set.");

        viewHolder.authorName.setText(articlesList.get(position).getAuthor());
        viewHolder.articleName.setText(articlesList.get(position).getName());
        String imgType = articlesList.get(position).getImgType();
        /*if(imgType != null){
            int imgResource = 0;
            if(imgType.equals("1")){
                imgResource = context.getResources().getIdentifier("@drawable/img_sleep",
null, context.getPackageName());
            }
            else if(imgType.equals("2")){
                imgResource = context.getResources().getIdentifier("@drawable/img_smoke",
null, context.getPackageName());
            }

viewHolder.getImage().setImageDrawable(context.getResources().getDrawable(imgResource));
        }*/

        viewHolder.cardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(context, ArticleActivity.class);
                intent.putExtra("ArticleModel", articlesList.get(position));
                context.startActivity(intent);
            }
        });
    }

    @Override
    public int getItemCount() {
        Log.d(TAG, "getItemCount");

        return articlesList.size();
    }

    public static class ViewHolder extends RecyclerView.ViewHolder {
        private final TextView articleName;
        private final TextView authorName;
        private final ImageView image;
        private final CardView cardView;

        public ViewHolder(View v) {
            super(v);
            Log.d("ViewHolder", "constructor");
            // Define click listener for the ViewHolder's View.
            articleName = (TextView) v.findViewById(R.id.articleName);
            authorName = (TextView) v.findViewById(R.id.authorName);
            image = (ImageView) v.findViewById(R.id.cardImage);
            cardView = v.findViewById(R.id.card_view_item);
        }
    }

}
```

- HomeFragment.java

```
package com.example.features.mainMenu.ui.home;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.example.tfg.R;
import com.example.tfg.databinding.FragmentHomeBinding;

import java.util.ArrayList;

public class HomeFragment extends Fragment {

    private FragmentHomeBinding binding;

    private HomeAdapter adapter;
    private RecyclerView recycler;

    private ArrayList<ArticleModel> articlesList;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        initDataset();
    }

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {

        binding = FragmentHomeBinding.inflate(inflater, container, false);
        View root = binding.getRoot();


        recycler = (RecyclerView) root.findViewById(R.id.recyclerArticulos);
        recycler.setLayoutManager(new LinearLayoutManager(root.getContext()));
        adapter = new HomeAdapter(articlesList, getContext());
        recycler.setAdapter(adapter);
        return root;
    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }

    private void initDataset(){
        articlesList = new ArrayList<ArticleModel>();

        ArticleModel firstArticle = new ArticleModel("Alimentación saludable", "Dr. Garcia
Garcia");
        firstArticle.setArtText("La alimentación es un proceso que nos acompaña a lo largo
de la vida, a través del cual obtenemos los nutrientes y energía que permiten cubrir los
requerimientos del organismo.\n" +
                "\n" +
                "Ahora bien, el acto de comer, además de ser un proceso nutritivo, tiene
connotaciones importantes de convivencia -proporción de placer, relaciones afectivas,
identificación social y religiosa, etc.- que configuran el comportamiento alimentario y
que pueden repercutir en el estado de salud de la persona.\n" +
                "\n" +
                "Las diferentes formas de comer de cada país están definidas, entre otros
factores, por el territorio, por sus tradiciones, la historia y la economía. La comida es
sin duda un elemento de identificación cultural; por tanto, es importante mantener y
potenciar los hábitos alimentarios propios del área mediterránea, adaptados a los cambios
sociológicos y culturales que se van produciendo.\n" +
                "\n" +
                "Una alimentación saludable se define como aquella que es satisfactoria,
suficiente, completa, equilibrada, armónica, segura, adaptada al comensal y al entorno,
sostenible y asequible.\n" +
                "\n" +
                "Aunque hay muchas propuestas saludables, una de las más conocidas en
nuestro entorno es la dieta mediterránea , que se basa en el consumo mayoritario de
alimentos de origen vegetal, frescos y mínimamente procesados, como frutas frescas y
verduras del tiempo, legumbres , frutos secos, cereales integrales, aceite de oliva virgen
extra y, en menor cantidad y frecuencia, los lácteos preferentemente fermentados y el
pescado, seguido de las carnes (preferentemente blancas y magras) y los huevos. El agua es
la bebida principal y la más saludable. La dieta mediterránea también se caracteriza por
procurar las comidas familiares, la utilización de alimentos locales y estacionales y por
potenciar las actividades culinarias y gastronómicas.\n" +
                "\n" +
                "Con pequeños cambios para comer mejor, se puede seguir un patrón
alimenticio saludable y mediterráneo con un presupuesto ajustado. Puede consultar la
infografía sobre el cálculo del coste económico de una alimentación básica saludable y
sostenible que la Agencia de Salud Pública de Cataluña (ASPCAT) ha diseñado a partir de
los precios de alimentos de 5 cadenas de supermercados de Cataluña (enero-junio 2022). La
media del coste para una alimentación saludable en Cataluña, calculado para una persona
adulta, es de 6'6 €/día, lo que representa una media semanal de aproximadamente unos
46€.");
        articlesList.add(firstArticle);

        ArticleModel secondArticle = new ArticleModel("El sueño", "Dra. Pujol Pujol");
        secondArticle.setImgType("1");
        secondArticle.setArtText("El sueño es un proceso biológico complejo que ayuda a
las personas a mantenerse sanas. Se caracteriza por los cambios en la actividad cerebral,
respiración, ritmo cardíaco, temperatura corporal y otras funciones fisiológicas.\n" +
                "\n" +
                "El sueño se divide en dos estados que se alternan cada 90 minutos en un
ciclo que se repite entre 4 y 5 veces a lo largo de la noche:\n" +
                "\n" +
                "Son sin movimientos oculares rápidos (sueño no-MOR, sueño no-REM o sueño
lento, son de ondas lentas o sueño profundo). Durante esta etapa aumenta el suministro de
sangre a los músculos, la energía se restaura, los tejidos crecen y se reparan, liberando
hormonas importantes para el crecimiento y el desarrollo. Se divide en tres fases a lo
largo de las cuales se avanza hacia un sueño más profundo:\n" +
                "Fase 1. Etapa intermedia entre estar despierto y dormirse. Puede entrarse
en un sueño superficial.\n" +
                "Fase 2. Inicio de un sueño más profundo. La respiración y el ritmo
cardíaco son regulares y la temperatura corporal disminuye.\n" +
                "Fase 3. El sueño alcanza el máximo nivel de profundidad. La respiración
se ralentiza, la presión arterial disminuye y los músculos se relajan. El cuerpo libera
hormonas.\n" +
                "Son movimientos oculares rápidos (sueño MOR, sueño REM o sueño
paradójico) . El cerebro está activo y se sueña. El cuerpo permanece inmóvil y la
respiración y la frecuencia cardíaca son irregulares. En esta fase predominan los procesos
de reparación cerebral (reorganización neuronal, consolidación y almacenamiento de
recuerdos).");
        articlesList.add(secondArticle);

        ArticleModel thirdArticle = new ArticleModel("Dejar de fumar", "Dra. Fernandez
Fernandez");
        thirdArticle.setImgType("2");
        thirdArticle.setArtText("El tabaco es una  droga legal  , pero también es una
sustancia muy adictiva que tiene efectos estimulantes sobre el sistema nervioso central de
la persona fumadora y que conlleva muchos riesgos para la salud.\n" +
                "\n" +
                "Una vida saludable es una vida sin tabaco. Dejar de fumar es
probablemente lo más importante que una persona fumadora puede hacer para mejorar su
salud, independientemente de su edad o sus antecedentes como fumadora. Al dejar de fumar
mejorará su salud y la de quienes le rodean.\n" +
                "\n" +
                "Si está pensando en dejar de fumar, enhorabuena. Ha tomado una buena
decisión tanto para usted como para las personas que le rodean. Los siguientes consejos
prácticos y recursos pueden ayudarle a dejar de fumar.");
        articlesList.add(thirdArticle);
    }
}
```

2. Visual implementation

- Activity_article.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical">

    <ImageButton
```

```
        android:id="@+id/backButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/white"
        android:layout_marginTop="5dp"
        android:layout_marginLeft="8dp"
        android:src="@drawable/ic_arrow_back"/>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <androidx.cardview.widget.CardView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                app:cardCornerRadius="20dp"
                android:elevation="10dp"
                android:layout_marginStart="20dp"
                android:layout_marginEnd="20dp"
                android:layout_marginTop="10dp">

                <ImageView
                    android:id="@+id/articleImage"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:src="@drawable/img_article"
                    android:adjustViewBounds="true"
                    android:background="@drawable/image_rounded_corners"/>

            </androidx.cardview.widget.CardView>

            <TextView
                android:id="@+id/articleName"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginStart="20dp"
                android:text="Lorem Ipsum"
                android:fontFamily="@font/inter_medium"
                android:textSize="20dp"
                android:textColor="@color/black"/>

            <TextView
                android:id="@+id/articleAuthor"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginStart="20dp"
                android:text="Lorem Ipsum"
                android:fontFamily="@font/inter_medium"
                android:textSize="16dp"
                android:textColor="@color/black"/>

            <TextView
                android:id="@+id/articleText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:layout_marginStart="20dp"
                android:layout_marginEnd="20dp"
                android:layout_marginBottom="20dp"
                android:fontFamily="@font/inter"
                android:textSize="14dp"
                android:textColor="#666666"
                android:justificationMode="inter_word"
                android:text="Labore sunt veniam amet est. Minim nisi dolor eu ad
incididunt cillum elit ex ut. Dolore exercitation nulla tempor consequat aliquip occaecat.
Nisi id ipsum irure aute. Deserunt sit aute irure quis nulla eu consequat fugiat Lorem
sunt magna et consequat labore. Laboris incididunt id Lorem est duis deserunt nisi dolore
eiusmod culpa exercitation consectetur."/>
        </LinearLayout>
    </ScrollView>
</LinearLayout>
```

- Activity_constant.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.features.mainMenu.ui.constants.ConstantsFragment">

    <ImageButton
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:id="@+id/backButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/white"
        android:layout_marginTop="5dp"
        android:layout_marginLeft="8dp"
        android:src="@drawable/ic_arrow_back"/>

    <TextView
        android:id="@+id/titleConstante"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fontFamily="@font/inter_semibold"
        android:textSize="20dp"
        android:text="Presión"
        android:textColor="@color/black"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:textAlignment="center"
        android:layout_marginTop="15dp"/>

    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/linechart"
        app:layout_constraintTop_toBottomOf="@id/titleConstante"
        android:layout_width="match_parent"
        android:layout_marginTop="15dp"
        android:layout_height="200dp" />

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/titleRecycler"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        app:layout_constraintTop_toBottomOf="@id/linechart"
        android:layout_marginRight="15dp"
        android:layout_marginLeft="15dp">

        <TextView
            android:id="@+id/lastMediciones"
            app:layout_constraintTop_toTopOf="parent"
            android:layout_width="wrap_content"
            android:layout_height="40dp"
            app:layout_constraintStart_toStartOf="parent"
            android:text="Últimas mediciones"
            android:fontFamily="@font/inter_medium"
            android:textSize="20dp"
            android:textColor="@color/black"/>

        <ImageView
            app:layout_constraintTop_toTopOf="parent"
            android:layout_width="wrap_content"
            android:layout_height="40dp"
            app:layout_constraintEnd_toEndOf="parent"
            android:src="@drawable/ic_more_green"
            android:gravity="center"/>

    </androidx.constraintlayout.widget.ConstraintLayout>


    <androidx.recyclerview.widget.RecyclerView
```

```
        android:id="@+id/recyclerSingleMedidas"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        tools:listitem="@layout/item_constantes_valores_diarios"
        android:layout_marginBottom="35dp"
        app:layout_behavior="@string/appbar_scrolling_view_behavior"
        app:layout_constraintTop_toBottomOf="@id/titleRecycler"/>

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btnEnviar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="10dp"
        android:background="@drawable/button_rounded_corner"
        android:fontFamily="@font/inter_bold"
        android:text="Enviar"
        android:textColor="@android:color/white"
        app:rippleColor="@color/buttonGreen_riple"
        android:textSize="10dp"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

- Activity_main_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <include
        android:id="@+id/app_bar_main_menu"
        layout="@layout/app_bar_main_menu"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <com.google.android.material.navigation.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_main_menu"
        app:menu="@menu/activity_main_drawer"
        android:background="@color/buttonGreen_claro"
        android:itemBackground="@color/buttonGreen_riple"
        app:itemTextColor="#666666"
        app:itemIconTint="@color/menuIconGreen"/>
</androidx.drawerlayout.widget.DrawerLayout>
```

- App_bar_main_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.features.mainMenu.MainMenu">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/Theme.TFG.AppBarOverlay">
        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="@color/buttonGreen"
            app:popupTheme="@style/Theme.TFG.PopupOverlay" />

    </com.google.android.material.appbar.AppBarLayout>

    <include layout="@layout/content_main_menu" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

- Component_piechart.xml

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.features.mainMenu.ui.constants.ConstantsFragment"
    android:paddingLeft="30dp"
    android:paddingRight="30dp">


    <org.eazegraph.lib.charts.PieChart
        android:id="@+id/pieChart"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>

    <TextView
        android:id="@+id/porcentaje"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:text="70%"
        android:textAlignment="center"
        android:fontFamily="@font/inter_medium"
        android:textSize="28dp"
        android:textColor="@color/buttonGreenPrimary"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@id/porcentaje"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:text="Control diario"
        android:textAlignment="center"
        android:fontFamily="@font/inter"
        android:textSize="15dp"
        android:textColor="#BDBDBD"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

- Content_main_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/app_bar_main_menu">

    <fragment
        android:id="@+id/nav_host_fragment_content_main_menu"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- Fragment_appointments.xml

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.features.mainMenu.ui.constants.ConstantsFragment">

    <androidx.viewpager.widget.ViewPager
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/pager"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <com.google.android.material.tabs.TabLayout
            android:id="@+id/tab_layout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

    </androidx.viewpager.widget.ViewPager>

</androidx.constraintlayout.widget.ConstraintLayout>
```

- Fragment_appointments_pager_fragment.xml

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools">


    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerAppointments"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        tools:listitem="@layout/item_appointments"
        android:layout_marginStart="30dp"
        android:layout_marginEnd="30dp"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

- Fragment_calendar.xml

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.features.mainMenu.ui.calendar.CalendarFragment">

    <CalendarView
        android:id="@+id/calendar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"/>

    <LinearLayout
        android:id="@+id/citasProgramadas"
        android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:orientation="vertical"
android:layout_marginLeft="10dp"
android:layout_marginRight="10dp"
app:layout_constraintTop_toBottomOf="@id/calendar"
android:layout_marginTop="30dp">

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/cita1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <ImageView
        android:id="@+id/puntoIncial"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:layout_marginLeft="10dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_point"/>

    <TextView
        app:layout_constraintTop_toTopOf="@id/puntoIncial"
        app:layout_constraintLeft_toRightOf="@+id/puntoIncial"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:text="Cita cardiólogo"
        android:fontFamily="@font/inter_medium"
        android:textSize="16dp"/>

    <TextView
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="10:30h"
        android:fontFamily="@font/inter_medium"
        android:textSize="16dp"
        android:layout_marginRight="10dp"/>

</androidx.constraintlayout.widget.ConstraintLayout>

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/cita3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="10dp">

    <ImageView
        android:id="@+id/puntoIncial2"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:layout_marginLeft="10dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_point"/>

    <TextView
        app:layout_constraintTop_toTopOf="@id/puntoIncial2"
        app:layout_constraintLeft_toRightOf="@+id/puntoIncial2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:text="Cita endrocino"
        android:fontFamily="@font/inter_medium"
        android:textSize="16dp"/>

    <TextView
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="11:30h"
        android:fontFamily="@font/inter_medium"
```

```
                android:textSize="16dp"
                android:layout_marginRight="10dp"/>

        </androidx.constraintlayout.widget.ConstraintLayout>

        <androidx.constraintlayout.widget.ConstraintLayout
            android:id="@+id/cita2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_marginTop="10dp">

            <ImageView
                android:id="@+id/puntoIncial1"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                android:layout_marginLeft="10dp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:src="@drawable/ic_point"/>

            <TextView
                app:layout_constraintTop_toTopOf="@id/puntoIncial1"
                app:layout_constraintLeft_toRightOf="@+id/puntoIncial1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="10dp"
                android:text="Recordatorio pastillas"
                android:fontFamily="@font/inter_medium"
                android:textSize="16dp"/>

            <TextView
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="13:30h"
                android:fontFamily="@font/inter_medium"
                android:textSize="16dp"
                android:layout_marginRight="10dp"/>

        </androidx.constraintlayout.widget.ConstraintLayout>
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

- Fragment_checkpoints.xml

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.features.mainMenu.ui.constants.ConstantsFragment">


    <LinearLayout
        android:id="@+id/firstColumn"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:orientation="vertical"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:gravity="center_vertical"
        android:layout_marginStart="70dp">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:id="@+id/informesOption"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">

            <androidx.cardview.widget.CardView
                android:id="@+id/informesCard"
                android:layout_width="100dp"
                android:layout_height="100dp"
                app:cardCornerRadius="10dp"
                app:cardBackgroundColor="@color/buttonGreen_claro"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:cardElevation="6dp">

                <ImageView
                    android:id="@+id/informesIcon"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:src="@drawable/ic_informes"
                    android:layout_gravity="center"/>
            </androidx.cardview.widget.CardView>

            <TextView
                android:id="@+id/informesTitle"
                android:layout_width="80dp"
                android:layout_height="wrap_content"
                android:fontFamily="@font/inter"
                android:text="Informes y resultados"
                android:textAlignment="center"
                android:textColor="@color/black"
                android:textSize="14dp"
                android:layout_marginStart="10dp"
                android:layout_marginEnd="10dp"
                app:layout_constraintStart_toEndOf="@id/informesCard"
                app:layout_constraintStart_toStartOf="@id/informesCard"
                app:layout_constraintTop_toBottomOf="@id/informesCard" />

        </androidx.constraintlayout.widget.ConstraintLayout>

        <androidx.constraintlayout.widget.ConstraintLayout
            android:id="@+id/cancerOption"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="25dp">

            <androidx.cardview.widget.CardView
                android:id="@+id/cancerCard"
                android:layout_width="100dp"
                android:layout_height="100dp"
                app:cardCornerRadius="10dp"
                app:cardBackgroundColor="@color/buttonGreen_claro"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:cardElevation="6dp">

                <ImageView
                    android:id="@+id/cancerIcon"
                    android:layout_width="63dp"
                    android:layout_height="58dp"
                    android:layout_gravity="center"
                    android:src="@drawable/ic_cancer" />
            </androidx.cardview.widget.CardView>

            <TextView
                android:id="@+id/cancerTitle"
                android:layout_width="85dp"
                android:layout_height="wrap_content"
                android:fontFamily="@font/inter"
                android:text="Detección cáncer de mama"
                android:textAlignment="center"
                android:textColor="@color/black"
                android:textSize="14dp"
                android:layout_marginStart="10dp"
                android:layout_marginEnd="10dp"
                app:layout_constraintStart_toEndOf="@id/cancerCard"
                app:layout_constraintStart_toStartOf="@id/cancerCard"
                app:layout_constraintTop_toBottomOf="@id/cancerCard" />

        </androidx.constraintlayout.widget.ConstraintLayout>

        <androidx.constraintlayout.widget.ConstraintLayout
            android:id="@+id/ginecologoOption"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="25dp">
```

```
            <androidx.cardview.widget.CardView
                android:id="@+id/ginecologoCard"
                android:layout_width="100dp"
                android:layout_height="100dp"
                app:cardCornerRadius="10dp"
                app:cardBackgroundColor="@color/buttonGreen_claro"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:cardElevation="6dp">

                <ImageView
                    android:id="@+id/ginecologoIcon"
                    android:layout_width="63dp"
                    android:layout_height="58dp"
                    android:layout_gravity="center"
                    android:src="@drawable/ic_ginecologo" />
            </androidx.cardview.widget.CardView>

            <TextView
                android:id="@+id/ginecologoTitle"
                android:layout_width="80dp"
                android:layout_height="wrap_content"
                android:fontFamily="@font/inter"
                android:text="Revisiones ginecólogo"
                android:textAlignment="center"
                android:textColor="@color/black"
                android:textSize="14dp"
                android:layout_marginStart="10dp"
                android:layout_marginEnd="10dp"
                app:layout_constraintStart_toEndOf="@id/ginecologoCard"
                app:layout_constraintStart_toStartOf="@id/ginecologoCard"
                app:layout_constraintTop_toBottomOf="@id/ginecologoCard" />

        </androidx.constraintlayout.widget.ConstraintLayout>
    </LinearLayout>

    <LinearLayout
        android:id="@+id/secondColumn"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:orientation="vertical"
        app:layout_constraintStart_toEndOf="@id/firstColumn"
        app:layout_constraintTop_toTopOf="parent"
        android:gravity="center_vertical"
        android:layout_marginStart="50dp"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginEnd="70dp">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:id="@+id/medicacionOption"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">

            <androidx.cardview.widget.CardView
                android:id="@+id/medicacionCard"
                android:layout_width="100dp"
                android:layout_height="100dp"
                app:cardCornerRadius="10dp"
                app:cardBackgroundColor="@color/buttonGreen_claro"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:cardElevation="6dp">

                <ImageView
                    android:id="@+id/medicacionIcon"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:src="@drawable/ic_medicacion"
                    android:layout_gravity="center"/>
            </androidx.cardview.widget.CardView>

            <TextView
                android:id="@+id/medicacionTitle"
                android:layout_width="80dp"
                android:layout_height="wrap_content"
                android:fontFamily="@font/inter"
                android:text="Medicación y recetas"
                android:textAlignment="center"
                android:textColor="@color/black"
                android:textSize="14dp"
                android:layout_marginStart="10dp"
                android:layout_marginEnd="10dp"
                app:layout_constraintStart_toEndOf="@id/medicacionCard"
                app:layout_constraintStart_toStartOf="@id/medicacionCard"
                app:layout_constraintTop_toBottomOf="@id/medicacionCard" />

        </androidx.constraintlayout.widget.ConstraintLayout>

        <androidx.constraintlayout.widget.ConstraintLayout
            android:id="@+id/vacunasOption"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="25dp">

            <androidx.cardview.widget.CardView
                android:id="@+id/vacunasCard"
                android:layout_width="100dp"
                android:layout_height="100dp"
                app:cardCornerRadius="10dp"
                app:cardBackgroundColor="@color/buttonGreen_claro"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:cardElevation="6dp">

                <ImageView
                    android:id="@+id/vacunasIcon"
                    android:layout_width="63dp"
                    android:layout_height="58dp"
                    android:layout_gravity="center"
                    android:src="@drawable/ic_vacunas" />
            </androidx.cardview.widget.CardView>

            <TextView
                android:id="@+id/vacunasTitle"
                android:layout_width="80dp"
                android:layout_height="wrap_content"
                android:fontFamily="@font/inter"
                android:text="Plan de vacunación anual"
                android:textAlignment="center"
                android:textColor="@color/black"
                android:textSize="14dp"
                android:layout_marginStart="10dp"
                android:layout_marginEnd="10dp"
                app:layout_constraintStart_toEndOf="@id/vacunasCard"
                app:layout_constraintStart_toStartOf="@id/vacunasCard"
                app:layout_constraintTop_toBottomOf="@id/vacunasCard" />

        </androidx.constraintlayout.widget.ConstraintLayout>

        <androidx.constraintlayout.widget.ConstraintLayout
            android:id="@+id/otrosOption"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="25dp">

            <androidx.cardview.widget.CardView
                android:id="@+id/otrosCard"
                android:layout_width="100dp"
                android:layout_height="100dp"
                app:cardCornerRadius="10dp"
                app:cardBackgroundColor="@color/buttonGreen_claro"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:cardElevation="6dp">

                <ImageView
                    android:id="@+id/otrosIcon"
                    android:layout_width="63dp"
                    android:layout_height="58dp"
                    android:layout_gravity="center"
                    android:src="@drawable/ic_more" />
            </androidx.cardview.widget.CardView>

            <TextView
```

```
                android:id="@+id/otrosTitle"
                android:layout_width="80dp"
                android:layout_height="wrap_content"
                android:fontFamily="@font/inter"
                android:text="Otras opciones"
                android:textAlignment="center"
                android:textColor="@color/black"
                android:textSize="14dp"
                android:layout_marginStart="10dp"
                android:layout_marginEnd="10dp"
                app:layout_constraintStart_toEndOf="@id/otrosCard"
                app:layout_constraintStart_toStartOf="@id/otrosCard"
                app:layout_constraintTop_toBottomOf="@id/otrosCard" />

        </androidx.constraintlayout.widget.ConstraintLayout>


    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

- Fragment_constants.xml

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.features.mainMenu.ui.constants.ConstantsFragment">

    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:layout_marginBottom="35dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            android:paddingLeft="15dp"
            android:paddingRight="15dp">

            <include
                android:id="@+id/component"
                layout="@layout/component_piechart"
                android:layout_width="wrap_content"
                android:layout_height="250dp"
                app:layout_constraintTop_toTopOf="parent"
                android:layout_marginTop="15dp"/>

            <androidx.constraintlayout.widget.ConstraintLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content">

                <TextView
                    android:id="@+id/medidas"
                    app:layout_constraintTop_toTopOf="parent"
                    android:layout_width="wrap_content"
                    android:layout_height="40dp"
                    android:layout_marginTop="15dp"
                    app:layout_constraintStart_toStartOf="parent"
                    android:text="Medidas"
                    android:fontFamily="@font/inter_medium"
                    android:textSize="25dp"
                    android:textColor="@color/black"/>

                <TextView
                    app:layout_constraintTop_toTopOf="parent"
                    android:layout_width="wrap_content"
                    android:layout_height="40dp"
                    android:text="Añadir+"
                    app:layout_constraintEnd_toEndOf="parent"
                    android:layout_marginTop="15dp"
                    android:fontFamily="@font/inter_medium"
                    android:textColor="@color/buttonGreenPrimary"
                    android:textSize="16dp"
                    android:gravity="center"/>

            </androidx.constraintlayout.widget.ConstraintLayout>

            <androidx.recyclerview.widget.RecyclerView
                android:id="@+id/recyclerMedidas"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                app:layout_constraintTop_toBottomOf="@id/medidas"
                android:layout_marginTop="10dp"
                tools:listitem="@layout/item_constants_medida"/>

        </LinearLayout>
    </androidx.core.widget.NestedScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

- Fragment_home.xml

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.features.mainMenu.ui.home.HomeFragment">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerArticulos"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0"
        tools:listitem="@layout/item_home_articlecard" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- Login_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:id="@+id/screen">

    <TextView
        android:id="@+id/title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerVertical="true"
        android:layout_marginTop="50dp"
        android:fontFamily="@font/inter_bold"
        android:text="@string/LoginTitle"
        android:textAlignment="center"
        android:textColor="#000000"
        android:textSize="34dp" />

    <EditText
        android:id="@+id/et_email"
        style="@android:style/Widget.AutoCompleteTextView"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_below="@id/title"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:background="@drawable/edittext_rounded_grey"
        android:fontFamily="@font/inter_bold"
        android:hint="@string/e_mail"
        android:inputType="textEmailAddress"
        android:padding="15dp"
        android:textSize="15dp"
        android:layout_marginTop="40dp"/>

    <EditText
        android:id="@+id/et_password"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_centerInParent="true"
        android:layout_below="@+id/et_email"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:layout_marginTop="10dp"
        android:background="@drawable/edittext_rounded_grey"
        android:fontFamily="@font/inter_bold"
        android:hint="@string/password"
        android:inputType="textPassword"
        android:padding="15dp"
        android:textSize="15dp" />

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btn_login"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@+id/forgotPwd"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="20dp"
        android:background="@drawable/button_rounded_corner"
        android:fontFamily="@font/inter_bold"
        android:text="@string/login"
        android:textColor="@android:color/white"
        app:rippleColor="@color/buttonGreen_riple" />

    <TextView
        android:id="@+id/forgotPwd"
        style="@style/GreenTextLink"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/goRegister"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:text="@string/forgotPwd"
        android:textAlignment="center"
        android:layout_marginBottom="5dp"/>

    <TextView
        android:id="@+id/goRegister"
        style="@style/GreenTextLink"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:text="@string/goRegister"
        android:textAlignment="center"
        android:layout_marginBottom="40dp"/>

    <ProgressBar
        android:id="@+id/circular_progressbar"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:layout_marginEnd="6dp"
        android:layout_marginBottom="96dp"
        android:indeterminateDrawable="@drawable/circular_progressbar"
        android:indeterminateDuration="@android:integer/config_longAnimTime" />

</RelativeLayout>
```

- Register Layout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:id="@+id/title"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_alignParentTop="true"
                android:layout_centerVertical="true"
                android:layout_marginTop="20dp"
                android:fontFamily="@font/inter_bold"
                android:text="@string/RegisterTitle"
                android:textAlignment="center"
                android:textColor="#000000"
                android:textSize="34dp" />

            <EditText
                android:id="@+id/et_name"
                style="@android:style/Widget.AutoCompleteTextView"
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:layout_below="@id/title"
                android:layout_alignParentStart="true"
                android:layout_alignParentEnd="true"
                android:layout_marginStart="20dp"
                android:layout_marginTop="20dp"
                android:layout_marginEnd="20dp"
                android:background="@drawable/edittext_rounded_grey"
                android:fontFamily="@font/inter_bold"
                android:hint="@string/name"
                android:inputType="text"
                android:padding="15dp"
                android:textSize="15dp" />

            <EditText
                android:id="@+id/et_email"
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:layout_below="@+id/et_name"
                android:layout_alignParentStart="true"
                android:layout_alignParentEnd="true"
                android:layout_centerInParent="true"
                android:layout_marginStart="20dp"
                android:layout_marginTop="10dp"
                android:layout_marginEnd="20dp"
                android:background="@drawable/edittext_rounded_grey"
                android:fontFamily="@font/inter_bold"
                android:hint="@string/e_mail"
                android:inputType="text"
                android:padding="15dp"
                android:textSize="15dp" />

            <EditText
                android:id="@+id/et_birthdate"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_centerInParent="true"
                android:layout_below="@+id/et_email"
                android:layout_alignParentStart="true"
                android:layout_alignParentEnd="true"
                android:layout_marginStart="20dp"
                android:layout_marginEnd="20dp"
                android:layout_marginTop="10dp"
                android:background="@drawable/edittext_rounded_grey"
                android:fontFamily="@font/inter_bold"
                android:hint="@string/date"
                android:inputType="date"
                android:focusable="false"
                android:clickable="true"
                android:maxLines="1"
                android:padding="15dp"
                android:textSize="15dp" />

            <androidx.appcompat.widget.LinearLayoutCompat
                android:id="@+id/checkbox_sex"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_below="@id/et_birthdate"
                android:layout_marginStart="20dp"
                android:orientation="horizontal">

                <CheckBox
                    android:id="@+id/cb_man"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="@string/man"
                    android:fontFamily="@font/inter"/>

                <CheckBox
                    android:id="@+id/cb_woman"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="@string/woman"
                    android:fontFamily="@font/inter"/>

            </androidx.appcompat.widget.LinearLayoutCompat>

            <EditText
                android:id="@+id/et_address"
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:layout_centerInParent="true"
                android:layout_below="@+id/checkbox_sex"
                android:layout_alignParentStart="true"
                android:layout_alignParentEnd="true"
                android:layout_marginStart="20dp"
                android:layout_marginEnd="20dp"
                android:layout_marginTop="10dp"
                android:background="@drawable/edittext_rounded_grey"
                android:fontFamily="@font/inter_bold"
                android:hint="@string/address"
                android:inputType="text"
                android:padding="15dp"
                android:textSize="15dp" />

            <EditText
                android:id="@+id/et_password"
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:layout_centerInParent="true"
                android:layout_below="@+id/et_address"
                android:layout_alignParentStart="true"
                android:layout_alignParentEnd="true"
                android:layout_marginStart="20dp"
                android:layout_marginEnd="20dp"
                android:layout_marginTop="10dp"
                android:background="@drawable/edittext_rounded_grey"
                android:fontFamily="@font/inter_bold"
                android:hint="@string/password"
                android:inputType="textPassword"
```

```
                android:padding="15dp"
                android:textSize="15dp" />
            <EditText
                android:id="@+id/et_passwordBis"
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:layout_centerInParent="true"
                android:layout_below="@+id/et_password"
                android:layout_alignParentStart="true"
                android:layout_alignParentEnd="true"
                android:layout_marginStart="20dp"
                android:layout_marginEnd="20dp"
                android:layout_marginTop="10dp"
                android:background="@drawable/edittext_rounded_grey"
                android:fontFamily="@font/inter_bold"
                android:hint="@string/passwordBis"
                android:inputType="textPassword"
                android:padding="15dp"
                android:textSize="15dp" />
            <EditText
                android:id="@+id/et_seguridadSocial"
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:layout_centerInParent="true"
                android:layout_below="@+id/et_password"
                android:layout_alignParentStart="true"
                android:layout_alignParentEnd="true"
                android:layout_marginStart="20dp"
                android:layout_marginEnd="20dp"
                android:layout_marginTop="10dp"
                android:background="@drawable/edittext_rounded_grey"
                android:fontFamily="@font/inter_bold"
                android:hint="CIP Seguridad Social"
                android:inputType="textPassword"
                android:padding="15dp"
                android:textSize="15dp" />
            <CheckBox
                android:id="@+id/cb_terms"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/terms"
                android:layout_below="@+id/et_passwordBis"
                android:layout_marginStart="20dp"
                android:layout_marginEnd="20dp"
                android:layout_marginTop="40dp"/>
            <androidx.appcompat.widget.AppCompatButton
                android:id="@+id/btn_login"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_above="@+id/goLogin"
                android:layout_alignParentStart="true"
                android:layout_alignParentEnd="true"
                android:layout_marginStart="20dp"
                android:layout_marginEnd="20dp"
                android:layout_marginBottom="20dp"
                android:background="@drawable/button_rounded_corner"
                android:fontFamily="@font/inter_bold"
                android:text="@string/register"
                android:textColor="@android:color/white"
                app:rippleColor="@color/buttonGreen_riple" />
            <TextView
                android:id="@+id/goLogin"
                style="@style/GreenTextLink"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_alignParentStart="true"
                android:layout_alignParentEnd="true"
                android:text="@string/goLogin"
                android:textAlignment="center"
                android:layout_marginBottom="40dp"/>

        </LinearLayout>
    </ScrollView>
</RelativeLayout>
```