1 2 9 0

UNIVERSIDADE Ɖ COIMBRA

João António Moreira Sobral

Implementation of Human-in-the-Loop Models in IoT Environments

The Vitoria and POC Use Cases

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Comunicações, Serviços e Infraestruturas, orientada pelo Professor Doutor Fernando Boavida e coorientador Professor Doutor Jorge Sá Silva, e apresentada à Faculdade de Ciências e Tecnologia, Departamento de Engenharia Informática, da Universidade de Coimbra.

Setembro de 2021

Faculty of Sciences and Technology

Department of Informatics Engineering

Implementation of Human-in-the-Loop Models in IoT Environments

The Vitoria and POC Use Cases

João António Moreira Sobral

Dissertation in the context of the Master in Informatics Engineering, Specialization in Communications, Services and Infrastructures guided by Professor Doutor Fernando Boavida and Professor Doutor Jorge Sá Silva and presented to the Faculty of Sciences and Technology Department of Informatics Engineering.

September 2021

1 2 9 0

UNIVERSIDADE Ð COIMBRA

This page was intentionally left blank.

Dissertation in the context of the Master in Informatics Engineering, Specialization in Communications, Services and Infrastructures guided by Professor Doutor Fernando Boavida and Professor Doutor Jorge Sá Silva and presented to the Faculty of Sciences and Technology Department of Informatics Engineering.

João António Moreira Sobral
University of Coimbra Department of Informatics Engineering

September, 2021

This page was intentionally left blank.

Abstract

Along with the growth in size, complexity and functionality, informatic systems are becoming increasingly more intelligent and interconnected. In line with this trend, the Internet of Things is being used more efficiently to support new types of systems. The development of "Human-in-the-Loop" systems in Internet of Things environments is still relatively recent and, as such, there are few works, and theses carried out on the subject. Through state-of-the-art research and detailed overview of Human-in-the-Loop models in Internet of Things Environments, this thesis is an effort to extend the field's knowledge with the implementation of two Human-in-the-Loop systems in IoT environments applied to medical situations. The first project (Vitoria) was meant to collect metrics and predictive techniques for people's behaviour patterns during and after a pandemic. The second project (POC) had the objective to allow the investigation of the correlation, overlapping, and distinction between autism spectrum disorder and obsessive-compulsion disorder, through passive and active data collection. Both projects follow a development process characteristic of Informatic projects that is specified in detail in this thesis, and both go through the development of systems for mobile platforms, although through different development processes. The system of the first project is based on Android platform and the system of second project is based on Xamarin platform. This thesis also includes a software library with the objective to allow any developer to implement the main features of a Human-in-the-Loop application in any software application based on Xamarin platform.

Keywords

Internet of Things; Human-in-the-Loop; Mobile Data; Passive Data; Medical Situations; Health Services; Cross-Platform.

This page was intentionally left blank.

Resumo

Com o crescimento em tamanho, complexidade e funcionalidade, os sistemas informáticos estão cada vez mais inteligentes e interconectados. Em linha com essa tendência, a Internet das Coisas é usada de forma mais eficiente para oferecer suporte a novos tipos de sistemas. O desenvolvimento de sistemas "Humanos em Ciclo" (tradução livre para "Human-in-the-Loop") em ambientes de Internet das Coisas ainda é relativamente recente e, como tal são poucos os trabalhos e teses realizados sobre o tema. Por meio de investigação de ponta e uma visão geral detalhada de modelos "Humanos em Ciclo" em ambientes de Internet das Coisas, esta tese é um esforço para estender o conhecimento do campo com a implementação de dois sistemas "Humanos em Ciclo" em ambientes IoT aplicados à área médica. O primeiro projeto (Vitoria) foi concebido para coletar métricas e criar técnicas de previsão para os padrões de comportamento das pessoas durante e após uma pandemia. O segundo projeto (POC) teve como objetivo permitir a investigação da correlação, sobreposição e distinção entre o transtorno do espectro do autismo e o transtorno obsessivo-compulsivo, por meio de recolha passiva e ativa de dados. Ambos os projetos seguem um processo de desenvolvimento característico de projetos informáticos que é detalhadamente especificado nesta tese, e ambos passam pelo desenvolvimento de sistemas para plataformas moveis, embora por processos de desenvolvimento diferentes. O sistema do primeiro projeto é baseado na plataforma Android e o sistema do segundo projeto é baseado na plataforma Xamarin. Esta tese também inclui uma biblioteca de software com o objetivo de permitir a qualquer programador implementar as principais funcionalidades de um sistema Humanos em Ciclo em qualquer aplicação de software baseada na plataforma Xamarin.

Palavras-Chave

Internet das coisas; Humanos em Ciclo; Dados Mobile; Dados Passiva; Medicina; Serviços de Saúde; Multi-Plataforma.

This page was intentionally left blank.

Acknowledgements

Having reached the end of this stage of my academic journey, I would like to express the my sincere thanks to all who made possible to carry out this dissertation.
To Professor Fernando Boavida, the superviser, and Professor Jorge Sá Silva, the co-supervisor, for having received me in their team. For all the availability, monitoring and attention given throughout the journey, and for all they taught me.
To my team colleagues, specially to Marcelo Fernandes for his friendliness and good disposition, for all his help in carrying out and planning the experiences and knowledge imparted on Human in the Loop systems, software application development and overall team work basics, even though the current situation of covid-19 pandemic. To all elements of the Department of Informatics Engineering, for their support, for the sympathy and accompaniment along this path.
To my classmates and friends, for their constant support, for getting me animated in the most difficult stages and for all the unique moments we've been through together. I wish them all the luck for their personal and academic future.
Last but not least, to my parents and siblings, for their unconditional love, for the patience and for having always been with me on this long journey, ultimately they are the reason behind my conquers.

This page was intentionally left blank.

Funding

The research presented in this thesis was supported by the EEA Grants Portugal 2014-2021 - Bilateral Initiative number 63 "PrivacyCoLab" and by the Project - "iFriend – Supervisión inteligente del estado de salud en personas mayores con insuficiencia renal mediante dispositivos inalámbricos" of Fundación CSIC, Interreg Portugal-Espanha.

Iceland Liechtenstein Norway grants

EEA Grants **Portugal**

Mecanismo Financeiro do Espaço Económico Europeu

European Economic Area Financial Mechanism

Unidade Nacional de Gestão

National Focal Point

This page was intentionally left blank.

Contents

Acronyms

API Application Programming Interface. 6, 10, 11, 13, 15, 22, 23, 26

APK Android Package. 9

APP Software Application. xxi, 1, 18–22, 24–27, 38, 48, 49, 54

BAML Binary Application Markup file. 13

BLE Bluetooth Low Energy. 50

CIBIT Coimbra Institute for Biomedical Imaging and Translational Research. 32

CLI Common Language Infrastructure. 12

CPS Cyber-Physical System. xix, 7, 8

DDoS Distributed Denial-of-Service. 6

DEI Department of Informatics Engineering. 23, 37

DGS Direção-Geral da Saúde. 18, 24, 25

DLL Dynamic link library. xvii, 14

DPO Data Protection Officers. 29

DSR Daily Self Report. xix, 33, 35, 36, 40, 43–45

FCTUC Faculdade de Ciencias e Tecnologia da Universidade de Coimbra. 32

FDA Food Drug Administration. 6

FMUC Faculdade de Medicina da Universidade de Coimbra. 32

GPS Global Positioning System. 18, 24, 49

HitL Human-in-the-Loop. 1, 7, 55

HitLCPS Human-in-the-loop Cyber-Physical System. 4, 7

HTTP Hypertext Transfer Protocol. 10, 11

ICNAS Instituto de Ciências Nucleares Aplicadas à Saúde. 32

IDE Integrated Development Environments. 9

IoMT Internet of Medical Things. 6

IoT Internet of Things. 1, 4–6, 22, 55

IPC Inter Process Communication. 9, 11

JSON JavaScript Object Notation. 14, 22, 24

M2M Machine to Machine communication. 5

MVVM Model-view-view-model. 13

NFC Near Field Communication. 5

OCD Obsessive Compulsive Disorder. 1, 31, 32, 38, 48

OS Operating System. 9

PAM Privileged Access Management. 6

PO Product Owner. 2

POC Perturbação Obsessivo-Compulsiva. xiii, xxi, 1, 3, 4, 31, 33, 35–37, 39, 41, 43, 45, 47, 49, 51, 53, 56

QR Quick Response. 32, 36, 38

RFID Radio frequency identification. 5

SDK Software Development Kit. 9, 10, 13

SM Scrum Master. 2

SSID Service Set Identifier. 50, 52, 53

UI User Interface. 9, 11, 12, 14, 49

UWP Universal Windows Platform. 15

V2V Vehicle-to-Vehicle communication. 5

VLAN Virtual Local Area Network. 6

VM Virtual Machine. 10

WPF Windows Presentation Foundation. 12, 13

XAML Extensible Application Markup Language. 13

Glossary

bug A software bug is an error, flaw or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways. 4

entity An entity is an object that exists in a database. An entity can be a single thing, person, place, or object. Data can be stored about such entities. 53

library A software library is a suite of data and programming code that is used to develop software programs and applications. It is designed to assist both the programmer and the programming language compiler in building and executing software. 1, 38, 55, 56

mobile broadband Mobile broadband is the marketing term for wireless Internet access delivered through cellular towers to computers and other digital devices using portable modems. 18, 24

NGSI NGSI is an information model and API for publishing, querying and subscribing to context information. It is meant to facilitate the open exchange and sharing of structured information between different stakeholders. It is used across application domains such as Smart Cities, Smart Industry, Smart Agriculture, and more generally for the Internet of Things and Cyber-Physical Systems. 22

NuGet Package A NuGet package is a single ZIP file with the .nupkg extension that contains compiled code (Dynamic link librarys (DLLs)), other files related to that code, and a descriptive manifest that includes information like the package's version number. 14, 15

predictive models Predictive models use known results to develop (or train) a model that can be used to predict values for different or new data. The modeling results in predictions that represent a probability of the target variable (for example, revenue) based on estimated significance from a set of input variables. xvii, 18

predictive techniques Used to create predictive models use known results to develop (or train) a model that can be used to predict values for different or new data. The modeling results in predictions that represent a probability of the target variable (for example, revenue) based on estimated significance from a set of input variables. 1

programmatically Programmatically is used to refer to tasks that can be done in an automated way (multiple purposes), especially as opposed to tasks that have to be done manually (single purpose). 37, 45, 46

Scrum Scrum is a project management framework, from organization to agile development of complex and adaptive products with the highest possible value. 2

system Information system, an integrated set of components for collecting, storing, and processing data and for providing information, knowledge, and digital products. 1

ZIP ZIP is a common file format that's used to compress one or more files together into a single location. 14

List of Figures

This page was intentionally left blank.

List of Tables

This page was intentionally left blank.

Chapter 1

Introduction

1.1 Motivation

We are currently facing a pandemic situation that affects everyone's life. This is of great importance worldwide, and as a result, there is a need to improve health services. Hence, many projects are being developed for this type of scenarios, in many different situations, for example, software to monitor patients who need personalized treatment for important psychiatric diseases, or software to monitor people during a pandemic situation.[31]

1.2 Objectives

The main objective for this work is the study, implementation and evaluation of two different systems of Human-in-the-Loop (HitL) in Internet of Things (IoT) environments and a generic library that facilitates the development of future similar systems. The first project is the Vitoria project, which aims to collect metrics and predictive techniques to forecast people's behaviour patterns, during and after a pandemic situation.
The second project is the POC project, which intends to collect clinical variables on obsessive-compulsive symptoms (obsessions, compulsions, triggering factors and interference) and associated symptoms (depressive and anxiety symptoms) to future investigation of the correlation, overlapping, and distinction between autism spectrum disorder and Obsessive-Compulsion Disorder. This project also intends to create an online forum for patients with Obsessive Compulsive Disorder (OCD), which allows for discussion between patients and the sharing of psychoeducational topics by health professionals and/or patients. Also to create a helpline where health professionals answer clinic-related questions submitted by patients with OCD. And to create a digital psychotherapeutic exercise program like an exposure-response prevention therapy. Since both projects share a common principle of collecting data passively, we developed a fully customizable library, the Socialite Sensors Library. It allows the developer to implement the main features of a HitL system on any APP without having to code everything from the start. The library is able to satisfy the need to collect, store and send data to a server of preference, and is compatible with the most common mobile platforms.
With this work we intent to share an analysis of the state of the art related to the subject of IoT systems and HitL models in the medical area, as well as clarifying some concepts about these matters and discuss the methodologies behind the making of systems that implement HitL models.

1.3 Methodology

During the first semester, we used the Scrum methodology with Sprints, that allow us to have agile project changes and rapidly adapted the software development. This methodology relies on a self-organizing, cross-functional team, so that there is no overall team leader who decides which person will do which tasks or how a problem will be solved. Those are issues that are decided by the team as a whole, which is especially useful for developers with a high degree of self-motivation.
This agile development Scrum teams relay on two specific task managers. The first is the Scrum Master (SM), who can be thought of as a couch for the team, helping team members using Scrum process to perform at highest level. The other one is the Product Owner (PO) whose role is to represents the business, customers, or users, and that guides the team towards building the right product.
This methodology was especially useful because it allowed us to solve the problems as they were presented. Throughout the projects we kept doing sprints planning meetings on Mondays and Fridays every week. So, every next Sprint (meeting), a commitment to a set of features was done by each developer. The meeting on Mondays was mainly a debrief, organized only within the developers, primarily focused on helping each other and solving issues that developers could have had previously and could not solve by themselves. The meetings on Fridays were mainly focused on presenting the work done through the week to the rest of the team, interchanging feedback of the ongoing projects. This feedback loop within Scrum software development sometimes resulted in changes to the freshly implemented functionality, or sometimes in revising or adding features to the product backlog.
The following Gantt chart in figure 1.1 shows all the main tasks done throughout the semester, as well as tasks that started after the initial plan, tasks that took longer than expected, and tasks that took less time to complete than what was originally planned.

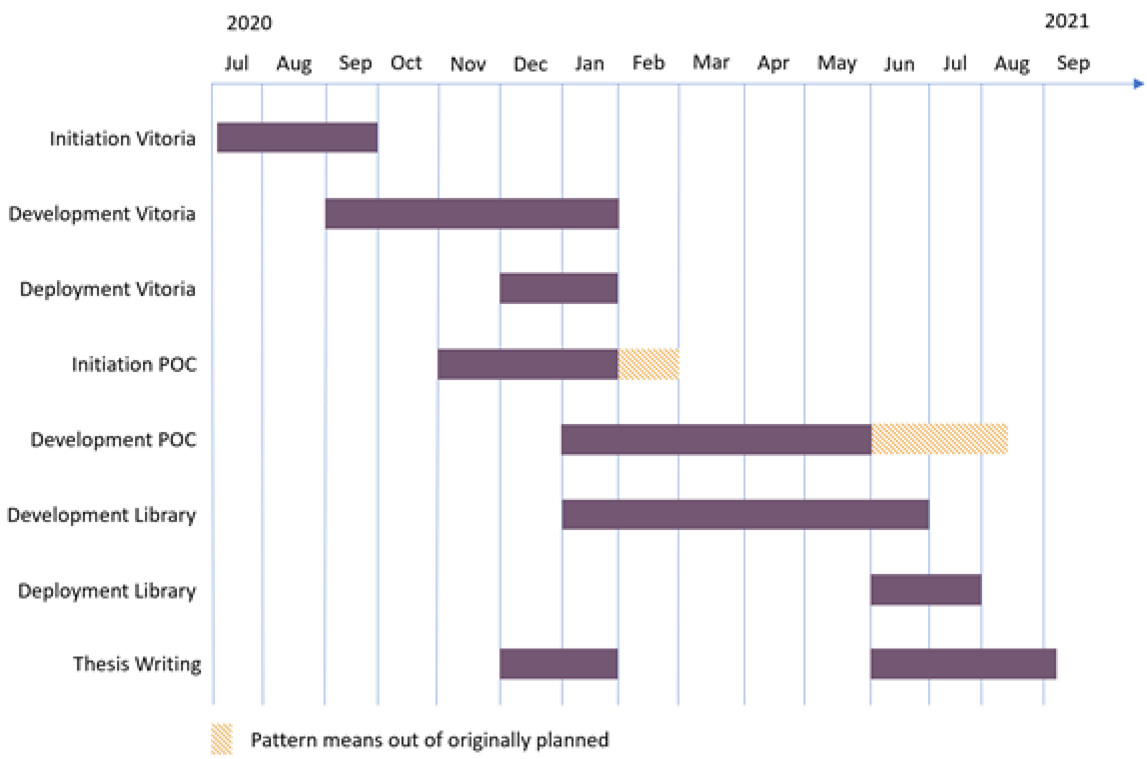


Figure 1.1: Gantt chart of the internship - planned vs accomplished

This thesis started in 2020 July, by studying the technologies required for the development of Vitoria project, based on Android programming Language. During that time, we were assigned a learning course which took approximately one month to finish alongside with the analysis of the state of the art for the project Vitoria. Then, we started to have weekly sprints, which on each week, each one of the developers were assigned a task to complete and to present the next week. Initially the easier tasks were assigned to the developers with less experience in that specific programming language. The last tasks of the initiation phase were drawing the mock-ups for the application and collecting the functional requirements.
During the phase of development of the project Vitoria, a wide nature of tasks was assigned, the author was assigned to work initially on the development of smartwatch wear system, which required programming android for wearOS. It took 2 months to finish the application for the smartwatch. A small testing phase was done after completing the smartwatch application where some parameters were changed to better optimize the battery life while not compromising on data collection. During that time, the author was assigned tasks to code on the main Android system, which were mainly related to passive data collection, testing various configurations, coding form screens and managing the Local Database.
The development phase ended early in 2021 which was within schedule. We started testing early, but most of it was done in the month of December 2020, where sometimes some changes to the system were made, mostly to calibrate the information gathering and to reduce battery level consumption of the host devices. The phase of deployment consisted of the creation of a document, which serves as user guide, which can be found as attachment to this work. Also, on December, the author started preparing the thesis for the intermediate presentation, which took 2 months while doing other tasks.
Alongside with the development of Vitoria, we studying the technologies involved in the next project, Project POC. The programming language is Xamarin. We started by gathering information of the state of the art and by collecting the functional requirements. Besides having had two meeting with the clients early January, many changes to the functional requirements were made throughout the development phase. With the documentation given by the client, we also created the mock-ups used to start up the development phase.
The development phase of the project POC started on January 2021, and unlike the previous project this one had to be coded from the ground-up, since none of the screens were already created. Initially the project was planned to be developed on its entirety alone, however it was forked into 2 main tasks, a library aimed to be implemented on the project itself and on future projects of the same nature, and the application itself which includes screen navigations and local database. Hence simultaneously we were working on both the application and the back-end library. The month of June 2021 was mainly dedicated to testing the library for its parameters considering battery usage and bandwidth used to upload its data to the server. The deployment of the library consisted of the preparation to be used on any Xamarin system, in which functions to fully use the library were implemented as well as considerations on coding level to fit different platforms.
By June the client requested a change in the design of the application, which delayed our expectations regarding the development phase of the system POC itself. Most of the screens and navigation were reused, however the extended time required was roughly 2 additional months. Lastly, the months of July and August were dedicated to finishing the work regarding the changes in design of the application for POC and the writing of this thesis.

1.4 Document Structure

This document will cover the internship's work corresponding to the Master degree and is organized and divided into 5 chapters. The first chapter is a brief introduction to the subject of study, describing its context, the motivations to the projects and its objectives. The second chapter is the state of the art, by presenting an overview of IoT including practical use cases and today's challenges. It also presents an overview of Human-in-the-loop Cyber-Physical System (HitLCPS), how this type of systems works and is operated. The third part of this chapter is dedicated to the presentation of methods for data collection, and considerations regarding human behaviour while collecting data. Part of this chapter is dedicated to the background, where it gives a brief explanation of the technologies used through the projects.

The third chapter addresses the first project, Vitoria, starting with a brief description of its context, objectives and its collaborators, and is followed by its functional and non-functional requirements, which serve as project functions that must be performed, presenting inputs, behaviors, and outputs. Also in this chapter is shown the risk analysis, pointing which precautions strategies were made, and the project architecture, where we break down all the aspects of the system functionalities. The next part of this chapter is dedicated to the implementation details, where we present the difficulties we had during the internship, as well as some strategies to mitigates some problems and the process of developing the project. Finishing the chapter is the testing phase, where we present the tests made to reduce the number of bugs as much as possible

The fourth chapter addressed the second project, POC, it starts with a description, followed by the requirements. This project differs from the previous with the library Socialite Sensors implementation, where we present its requirements to use, its functionalities, architecture, sensors available, how it works in background, and the implementation challenges. To finish this chapter we present the tests done to reduce the number of bug as much as possible.

The fifth chapter is dedicated to the conclusions that were achieved, presenting the contributions of this work, a recap of the challenges that were faced, and a description of possible future work.

Chapter 2

State-of-the-art

2.1 Internet of Things

Mobile development is undoubtedly one of the most actively expanding sectors in modern informatics. Generally, the mobile applications market is dominated by social media apps and gaming apps. Due so, various companies use mobile applications for expanding their markets and improving their costumer engagement. The growth of the use of smart wearable devices like hololens, smart clothing and smartwatches also indicate an upcoming transformation in computing, allowing more and better passive sensory data to be collected.

The Internet of Things (IoT) is one of the most addressed subjects among researchers in informatics, from academic to commercial organizations. It is considered to be the next evolutionary stage of the Internet. Since the early stages, the internet has shifted from a set of computers communicating with each other to billions of computational devices and cell phones connected all over the world.

We are moving towards a phase where all the devices in our environment are connected to the Internet and can communicate with each other with minimum Human effort. IoT describes the network of physical objects that are embedded with sensors and software. They are globally connected in an intelligent network infrastructure that manages every object. It allows the interconnection of devices that, in turn, provide new applications and services which can improve human lives. The term itself has been known since 1982, but it is getting more attention lately due to the advancement of wireless technologies. The IoT empowers substantial objects to see, hear, think, and perform jobs by having them communicate with each other and sharing information synchronously. The IoT transforms conventional objects to smart ones by manipulating its underlying technologies such as omnipresent and pervasive computing, embedded devices, communication technologies, sensor networks, protocols and applications.

Communication technologies are a vital component of the IoT scenarios. At first, Radio frequency identification (RFID) seemed to be mandatory for IoT devices, but the advancement brought some new technologies alongside, like the Near Field Communication (NFC), Machine to Machine communication (M2M), and Vehicle-to-Vehicle communication (V2V), which can be used to implement the modern idea of IoT. These new technologies can bring easiness and comfort to the lives of the users by adopting new solutions. In addition, IoT has great impact on domestic sphere, such as assisted living, smart houses, smart cars and smart cities. In the business sector, IoT has a noticeable advancement in manufacturing and services, bringing more production and superior quality. Thus, the cost to develop these systems of IoT is decreasing, making them more accessible to the industry and the common user. As a result, the worldwide adoption is smooth but involves lots of issues, that need to be solved before its major acceptance.

The major issue with IoT is the security. Some other problems are related to scalability and addressing issues. Surely businesses can reap impressive benefits from the IoT ecosystem, but more IoT devices and a more complex ecosystem also means increased security vulnerabilities from the edge to the cloud. Hard-coded and embedded credentials are a danger for IoT systems and as much hazardous for IoT devices. Many devices use default passwords, and that is a windfall for hackers to attack. The Mirai malware is a good example of such an attack, it infected IoT devices from routers to video cameras and video recorders by successfully attempting to log in using a table of 61 common hard-coded default usernames and passwords. The malware created a mesh of interconnected 400,000 devices, each of which was running one or more automated tasks (botnet). One of those tasks was a Distributed Denial-of-Service (DDoS) with an aggregated bandwidth of 1Terabit per second, taking down Amazon Web Services and its clients in September 2016. Also in 2017, a similar attack named Reaper came to light. Around 30,000 devices were found to have been compromised launching DDoS attacks. Interestingly these botnets created by malwares may have created a new market where criminals can rent out botnets to attempt to take down websites that they disagree with. In 2020 a list of Telnet default credentials for 515,000 servers, routers and IoT devices was published. Nowadays, Mozi, a Mirai-type variant, has been the most active botnet since 2019. Knowing little to none about this botnet, Mozi's infrastructure seems to be operated mostly in China, and Japan is the most targeted country according to Security Intelligence and IBM MMS (22 April 2021). The IoT devices should be flexible and have secure default settings and optional mechanisms like password complexity, expiration account lock-out, one time password. Privileged Access Management (PAM) for all devices is essential for slashing IoT security and ensuring that IoT networks cannot be hacked.[31][21]

[15] IoT products are developed with ease of use and connectivity in mind. At the time of purchased, a device may be secure, but become vulnerable when hackers find a new security breach/issue. Satori is the name of another malware like Mirai, this malware does not spread via credential guessing but has been found to target known vulnerabilities in specific models of Wi-Fi routers. Some countries have already incorporated some cybersecurity law regarding unique passwords, regular security updates, and vulnerability disclosure, in particular the state of California and Oregon in the United States.[20]

All the IoT devices process and communicate data, so they need apps, services or protocols for communicating, thus many IoT vulnerabilities originate from insecure interfaces. They are related to web, Application Programming Interface (API), cloud, and mobile interfaces that can compromise the device and its data if used malignly. This includes lack or insufficient device authentication and authorization and weak encryption or none.

An act that became public law on US on December 2020, the IoT cybersecurity Improvement Act, and the cybersecurity Act (Regulation 2019/881 of 17 April 2019) that came into force on 27 June 2019 and became law in European Union and the UK was an important step in cybersecurity.[25][27] A study published in July 2020 analysed over 5 million IoT, Internet of Medical Things (IoMT), and unmanaged connected devices in healthcare, retail, and manufacturing, revealing numerous vulnerabilities and risks across a wide variety of devices, and reported interesting facts: a) up to 15% of devices were unknown or unauthorized; b) 5 to 19% were using unsupported legacy operating systems; c) 49% of IT teams were guessing or had tinkered with their existing IT solutions to get visibility; b) 51% of them were unaware of what types of smart objects were active in their network; d) 75% of deployments had Virtual Local Area Network (VLAN) violations; e) 86% of healthcare deployments included more than ten Food Drug Administration (FDA) recalled devices; f) 95% of healthcare networks integrated Amazon Alexa and Echo devices alongside hospital surveillance equipment. Furthermore, some high-end medical machineries (Magnetic Resonance Imaging and Computed Tomography) were discovered running social media platforms.[30][12]

2.2 Human-in-the-Loop Cyber-Physical Systems

Human-in-the-Loop (HitL) computing is an area that is also having increasingly importance in research due to many safety-critical features of many systems. HitL is defined as a computation model that requires human interaction. In this type of models, a human is always part of the process and consequently influences the outcome in such a way that is difficult if not impossible to reproduce it exactly every time.
Cyber-Physical System (CPS) are a specific type of systems that integrate computation and physical processes, where computational systems monitor and control the physical processes. Those in turn affect the computational processes in a feedback loop. This type of systems is already being developed and implemented in automotive industry, healthcare devices, military, robotic and transport systems, building and environmental control. The main goal to design a CPS system is to consider the human feedback interaction with the HitL model, with predefined roles. As said by the renowned computer scientist Mark Weiser, the ultimate form of computers may be an extension of our subconscious. The ideal computer would be capable of truly understand our unconscious intentions, desires and actions. Instead of Humans having to adapt to technology and learn how to use it, it would be the technology adapting to Humans' habits and uniqueness avoiding us to hurdle through unintuitive menus, errors and incompatibilities.[18][7]
There are three different parts that can be identified in this type of systems: a) the Physical Environment, which corresponds to the system components that are not electronic nor digital networks but can be Humans, biological, chemical or mechanical processes; b) the Computer Platforms, that are computers, sensors, or actuators (there can be more than one of these platforms); c) the Network Fabric, which Interconnects all computational platforms. A CPS operation can be abstracted as a loop composed of the following actions: a) the Physical Environment that provides data to the computational platform; b) the Sensors that analyse and send data to a control component; c) the Control component decides which command to send to the actuators, and d) the actuator's actions can provoke changes in the physical environment state. The next figure shows a visual representation of all operations of a generic CPS.

There are two variants of the feedback loop: the open-loop, and the closed loop. In the open loop, the sensors capture the physical environment data but do not trigger any actions that change the physical environment state. This type of systems generally is more focused on only acquiring data with no necessary response. An example of a CPS closed loop is a system that tracks human sleep status. In the closed loop system, physical environment data is provided to the computational platform and then it can trigger the actuator's actions that in turn will change the physical environment state. An example of a closed loop system is an application of a smart thermostat, that detects occupancy and sleep patterns to control heating ventilation and air conditioning, to reduce energy consumption of a building.[23] So, the important question is where and how do the Humans take part inside this feedback loop? Human-in-the-loop Cyber-Physical Systems (HitLCPSs) are instances of CPSs, and Humans can be present in any phase of the loop. In the control phase, Humans can define parameters in the control algorithm and the system operates within the established boundaries. In the physical environment phase, Humans can be

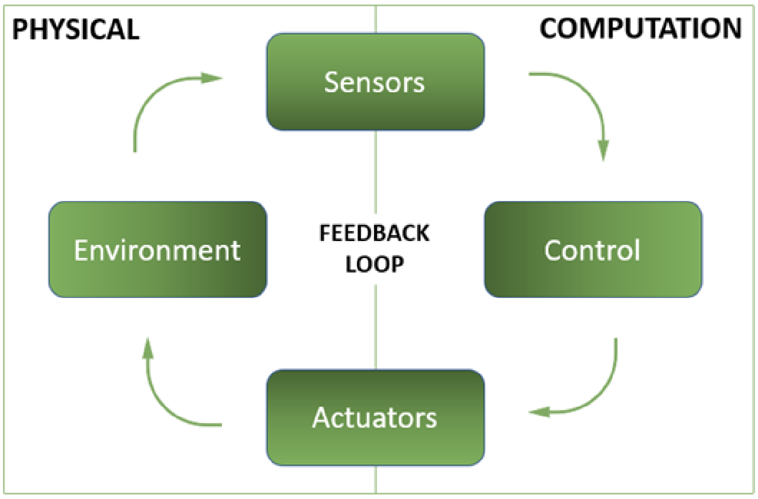


Figure 2.1: Illustrative CPS feedback loop adapted from [19].

the source for the sensors feeding data to be analysed, and they can also be the target of the actuator mechanisms or can affect in some way the physical environment state. Humans can also participate in the actuators phase, where advanced systems like artificial intelligence or high-end mechanical robots cannot yet replace a human on a task. For example, on the sensor phase, it is a valuable solution having Humans performing sensor tasks due to their reliability when measuring. Having humans perform sensor tasks on the environment where Humans are part of, is important, since for example, a human might not be accurate on temperature reading, but can definitely check if ambient temperature reading is correct.[17][10]

2.3 Big data analysis and human behaviour

Today, businesses and organizations are connected to their customers, customers, users, employees, suppliers, and sometimes even their competitors. Data can tell a story about any of these relationships, and with that information, organizations can improve nearly every aspect of their operations.While data can be valuable, too much information is difficult to control and wrong data is useless. The right method of data collection can mean the difference between useful insights and time-wasting disorientation.
While many fields have benefited greatly from the collection and analysis of big data, some fields, like medical and in particularly psychology are still lagging. [4] showed an example on how the large collection data sets helped researchers solve difficult problems in astronomy that weren't possible in the past. Interestingly, the slow process of applying big data to psychology reflects history of the development of sciences, since astronomy and other sciences are much older than the experimental psychology (which emerged in the 19th century).
It is important to note that human behavior and health problems are quite complex. For example, an Alzheimer's disease, which is a most common neurodegenerative disease in old age [6][28], is associated with several genetic factors, nutritional, cognitive and neural changes that go back centuries of variables. Standard statistical methods used in most empirical methods studies are equipped to diagnose and understand Alzheimer's diseases. Big data methods will allow us to select the most important features that differentiate AD patients from healthy ones, as this will allow physicians and neurologists to test only these variables in clinical practice.

Since human behavior is extremely complex, it's no surprise that many discoveries existing in the field of psychology and medicine are conflicting. This is perhaps due to the existence of several factors that affect human behavior, as well as the simplicity of theories used to explain human behavior. However, most psychological experiments focus primarily on measuring 2 to 7 variables. Most standard statistical methods cannot handle data sets with a large number of variables. Also, many of the small data sets cannot answer questions about causality. To do this, the clusters often collect large sets of longitudinal data.
We can argue that the more data we collect, the better our understanding of human behavior will be. Instead of trusting theory-based methods, as is often the case in psychology studies, analysis from big data can lead to discovery and allow new "theories" arise directly from the data. In addition, big data methods can provide unexpected results on participant sub-types, as well as allow a better understand of the nature of human behavior.[24]

2.4 Technologies used

2.4.1 Android

Android Operating System (OS) is recognized as the most widely used, popular and user-friendly mobile platform. Android is an open-source operating system that runs on a Linux kernel. Android applications are developed using java language. Google has its own Software Development Kit (SDK) which enables these java codes to control devices like mobiles phones, wearables, TVs, etc. The android application development provides a flexible platform for developers where they can use both java Integrated Development Environments (IDE) and android java libraries. The android operating system gained popularity among developers for its customizable nature.[13]
Android was launched in November 2007, it is free and open-source software, also known as , which is primarily licensed under the Apache License. About 70 percent of Android smartphones run Google's ecosystem; some with vendor-customized user interface and software suite, such as TouchWiz and later One User Interface (UI) by Samsung, and HTC Sense. Other Android ecosystems and forks include Fire OS (developed by Amazon) or LineageOS. However, the "Android" name and logo are trademarks of Google which impose standards to restrict "uncertified" devices outside their ecosystem to use Android branding. The source code has been used to develop variants of Android on a range of other electronics, such as game consoles, digital cameras, portable media players, PCs and others, each with a specialized user interface. Some well-known derivatives include Android TV for televisions and Wear OS for wearables, both developed by Google. Software packages on Android, which use the Android Package (APK) format, are generally distributed through proprietary application stores like Google Play Store, Samsung Galaxy Store, Huawei AppGallery, Cafe Bazaar, and GetJar, or open-source platforms like Aptoide or F-Droid.
It has been the best-selling OS worldwide on smartphones since 2011 and on tablets since 2013. As of May 2021, it has over three billion monthly active users, the largest installed base of any operating system, and as of January 2021, the Google Play Store features over 3 million apps. The current stable version is Android 11, released on September 8, 2020.
Android users have the authority to accept or deny permissions of an installed application, however, android allows Inter Process Communication (IPC) which makes the device vulnerable to security threats of privacy. One of the main features of this operating system is the architecture being quite the same as that of personal computers, therefore, security issues can be fixed in similar ways.

The android OS architecture is divided in 3 layers: a) an Application Layer, where the android application components get executed, and the end-users can make use of application that are compiled and executed using Dalvik Virtual Machine; b) the App Framework Layer, where developers get access to the core application services, like windows manager, resources and package managers. It's in this layer that developers can customize their application in order to edit the system structure of their applications so that they can make use of various services by different API libraries; c) the Android Runtime Layer and Linux Kernel, which is where the android runtime layer is primarily focused on the running state of the process, and every program has its own specific Dalvik Virtual Machine (VM) environment.[5]
Android kernel is similar to the kernel of a desktop running linux. It is an open environment providing access to the kernel system. It is mainly associated with management utilities, like power and memory control, and security and network management. Android applications are developed using java language using SDKs that provide the application interfaces. The set of services provided by the android development platform includes: a) a set of customizable and extensible views, like buttons, textboxes, browser and lists. b) Content providers that are used to enabling inter-process communications and are responsible for applications retrieving and sharing data with other applications; c) resource manager that is responsible for enabling the application to access resources like strings and layout files; d) application notifications that are handled by a customizable notification manager, from permission requests to user notifications; e) the activity manager that manages and analyses every lifecycle phase of an application.[26]

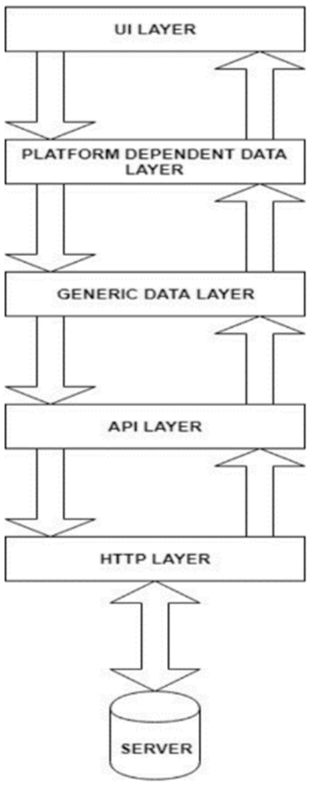


Figure 2.2: Layered Architecture for android application adapted from [29].

The android application development can be divided in 5 different layers. The lowest layer being the Hypertext Transfer Protocol (HTTP) layer which is where the communication

with the server occurs, so HTTP requests are sent and received. The second is the API layer which is responsible to parse server responses and to formulate queries. Those queries can then be sent to the generic data layer where the important field from the response are extracted. In addition, functionalities such as caching, validations and data management can also be implemented in the generic data layer. The next layer is called platform dependent because collected data is stored in several classes like adapters or lists depending on the platform they are going to be used. The highest being the UI layer which is responsible for the interactions with the user via views and layouts.[29]
Mobile apps are not like desktop apps. Desktops have copious amounts of resources such as screen real estate, memory, storage space, and a connected power supply, mobile devices do not. These constraints force mobile apps to behave differently. For example, the small screen on a mobile device typically means that only one app (i.e. Activity) is visible at a time. Other Activities are moved to the background and pushed into a suspended state where they cannot perform any work. However, just because an Android application is in the background does not mean that it is impossible for app to keep working.
Android applications are made up of at least one of the following four primary components: Activities, Broadcast Receivers, Content Providers, and Services. Activities are the cornerstone of many great Android applications because they provide the UI that allows a user to interact with the application. However, when it comes to performing concurrent or background work, Activities are not always the best choice.
The primary mechanism for background work in Android is the service. An Android service is a component that is designed to do some work without a user interface. A service might download a file, play music, or apply a filter to an image. Services can also be used for IPC between Android applications. For example one Android app might use the music player service that is from another app or an app might expose data (such as a person's contact information) to other apps via a service.
Services, and their ability to perform background work, are crucial to providing a smooth and fluid user interface. All Android applications have a main thread (also known as a UI thread) on which the Activities are run. To keep the device responsive, Android must be able to update the user interface at the rate of 60 frames per second. If an Android app performs too much work on the main thread, then Android will drop frames, which in turn causes the UI to appear jerky (also sometimes referred to as janky). This means that any work performed on the UI thread should complete in the time span between two frames, approximately 16 milliseconds (1 second every 60 frames). To address this concern, a developer may use threads in an Activity to perform some work that would block the UI. However, this could cause problems. It is very possible that Android will destroy and recreate the multiple instances of the Activity. However, Android will not automatically destroy the threads, which could result in memory leaks.

2.4.2 Xamarin

One way to build apps for different mobile platforms is to create customized versions of apps for each platform, like a separate version of the app for Android and iOS. However, this approach leads to multiple versions of the app's base code, which are difficult to maintain and evolve over time. Therefore, developers are increasingly adopting cross-platform mobile app development frameworks. [9]
In computer programming, a software framework is an abstraction in which software, providing generic functionality, can be selectively changed by additional user-written code, thus providing application-specific software. It provides a standard way to build and deploy applications and is a universal, reusable software environment that provides particular functionality as part of a larger software platform to facilitate the development of software

applications, products and solutions.
A cross-platform framework can be divided into 2 major groups, the web-based frameworks and the native frameworks. Web-based frameworks allow developers to build mobile apps using popular languages used to build web applications, such as HTML5, Javascript, and CSS. Popular examples are Adobe PhoneGap/Cordova [1], Sencha [2] and IBM MobileFirst [3]. The problem with these languages is that they do not contain primitives to allow apps to access resources on the phone, like sensors, camera or address books. But the use of non-native libraries, make these frameworks not suited for high-performance apps, such as games, animations or applications that require many readings of primitive interfaces.
On the other side, native frameworks, address the above challenges, and generally support a home platform and one or more target platforms. Developers build mobile apps as they normally would for the home platform, and the framework builds apps automatically for the target platforms as well. Examples of this type of framework is Xamarin.[29][16][14]
The most used cross platform frameworks are Flutter, React Native and Xamarin. Xamarin is the most used among companies and good examples are its use by UPS, BBC Good Food, The World Bank. Even though React Native is more talked-about, Xamarin executes the code faster, have a smoother UI and it's more advanced version Xamarin.Forms allows compiling iOS code on Windows (via the Hot Restart feature). In addition, Microsoft's documentation is always at the top. However, since Xamarin is not as popular as Flutter or React Native, finding such experienced developers might be challenging to overcome coding difficulties/challenges.
React Native is the most popular among developers outside big companies, new versions of this tool are released every few weeks, so the community around react native keeps it up to date with mobile OS updates and fixes critical issues. Despite being a web based cross platform framework, community support is huge, and as a result, react native based applications are almost identical in performance, look and feel with native ones. Also because of its nature, libraries in React Native extend its potential and allows the developer to deploy the code as a web application as well. React Native uses javascript, which has been on the market for many years, so adapting is easier, and the learning curve is lower. Overall React Native apps could be beneficial for applications that are visually heavy and require active user input and might not be best choice for performance-oriented systems.
About the flutter, the main reason it stands out among the alternatives is its approach to the user interface development. An application created in Flutter will look the same on any platform, due to the graphical engine it uses. Another thing to notice is the plugins and platform channels support OS-level features without extra hassle. Finally, there's Flutter for Web, allowing to deploy apps to the web as well, and Fuchsia OS, that is still under development but will allow using Flutter for IoT purposes. However, since it's the "youngest" tool of mentioned above, it has some immaturity problems, like insufficient stability and documentation issues.
Xamarin is a Microsoft-owned San Francisco-based software company founded in May 2011 by the engineers that created Mono (a free and open-source .NET Framework-compatible software framework), Xamarin.Android (formerly Mono for Android) and Xamarin.iOS (formerly MonoTouch), which are cross-platform implementations of the Common Language Infrastructure (CLI) and Common Language Specifications named Microsoft .NET. With a C#-shared codebase, developers can use Xamarin tools to write native Android, iOS, and Windows apps with native user interfaces and share code across multiple platforms, including Windows, macOS, and Linux. According to Xamarin, over 1.4 million developers were using Xamarin's products in 120 countries around the world as of April 2017. Xamarin.Forms was introduced on May 28, 2014, and allows one to use portable controls subsets that are mapped to native controls of Android, iOS and Windows Phone. It is also possible to target other platforms such as Tizen (by Samsung), GTK (Linux), Win-

dows Presentation Foundation (WPF) and macOS. It uses Extensible Application Markup Language (XAML) which is a declarative XML-based language developed by Microsoft that is used for initializing structured values and objects. A XAML file can be compiled into a Binary Application Markup file (BAML) file, which may be inserted as a resource into a .NET Framework assembly. At run-time, the framework engine extracts the BAML file from assembly resources, parses it, and creates a corresponding WPF visual tree or workflow. Since Xamarin's home platform is windows phone, developers build apps not only using C#, but also with the API of Windows Phone SDK.

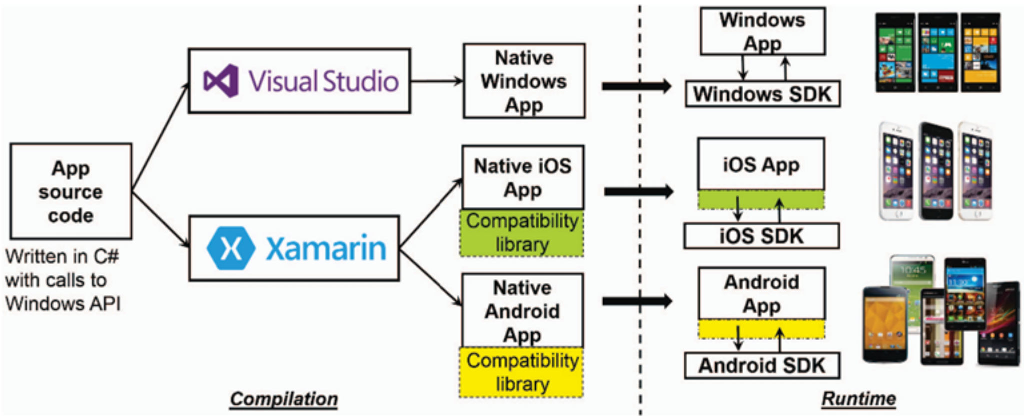


Figure 2.3: Overall operation of cross-platform Xamarin development framework adapted from [9].

The figure 2.3 represents a typical example of the workflow of a general cross-platform framework, with Xamarin as a concrete example. Developers build apps as they would for the Windows Phone, in C# using calls to the API of the Windows Phone SDK. This code then can be compiled to Windows phone with Visual Studio toolchain. Xamarin allows developers to use the same source code to build native Android or iOS apps, with libraries that translate Windows SDK API calls in the code to the relevant API calls of the underlying Android or iOS SDKs. Model-view-view-model (MVVM) is the key feature of Xamarin which makes it different from the others. Xamarin Architecture can be divided in six application layers, which are shown in the next figure.[33]

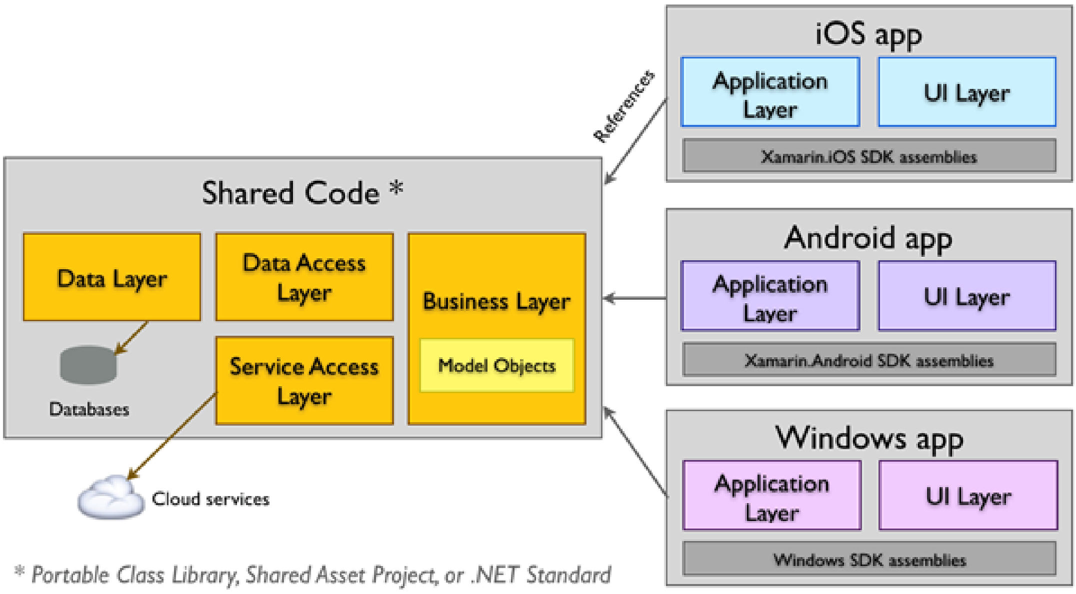


Figure 2.4: Layer architecture of Xamarin Platform adapted from [34].

The Data Layer is where all the data is saved within mobile devices with the help of SQLite database, persisting even if the power is off. The Data Access Layer is used to connect with the database to perform database operations like insert, delete, update and select elements. The Business Layer is where the business logic is applied to perform some tasks, contains business entity definitions (the Models) and the business logic. The Service Access Layer is used when some service is required, for example transferring data from one technology to another which can be developed in JavaScript Object Notation (JSON). The use of this layer can avail other services like cloud services etc. The Application Layer contains code that is typically platform specific or code that is specific to the application. The UI Layer contains screens, widgets and the controllers that manage them. A good test of whether to place code in the application layer versus the UI layer is to determine whether the class has any actual display controls or whether it could be shared between multiple screen or devices with different shapes and forms.

2.4.3 Xamarin Library

An essential tool for any modern development platform is a mechanism through which developers can create, share, and consume useful code. Often, this code is bundled into "packages" that compile code (such as Dynamic link librarys (DLLs)) along with other content needed in projects that consume these packages. For .NET (including Xamarin.Forms), Microsoft's supported mechanism for sharing code is called NuGet, which defines how packages for .NET are created, hosted and consumed, and provided as tools for each of these functions.
Simply put, a NuGet Package is a single ZIP file with a .nupkg extension that contains compiled code (DLLs), other files related to that code, and a descriptive manifest that includes information such as the package's version number. Developers with code to share create packages and publish them to a public or private host. Package packages get these packages from suitable hosts, add them to their projects, and then call the functionality of a package in their project code. NuGet itself handles all the details in between.
In its role of public host, NuGet itself maintains the central repository of over 100,000

unique packages on nuget.org. These packages are employed by millions of .NET / .NET Core developers every day. NuGet also lets you host packages privately in the cloud (such as Azure DevOps), on a private network, or even just on your local file system. By doing this, these packages are only available to developers who have access to the host, giving you the ability to make packages available to a specific group of consumers. Options are explained in Hosting your own NuGet feeds. Through the configuration options, you can also control exactly which hosts can be accessed from any computer, thus ensuring that packages are selected from specific sources rather than a public repository like nuget.org.
Whatever its nature, a host serves as the connection point between the creators and consumers of the package. Creators create useful NuGet Packages and publish them to a host. They perceive useful packages and packages on hosts, downloading and including these packages in their projects. Once installed in a project, as package API are available to the rest of the project's code.

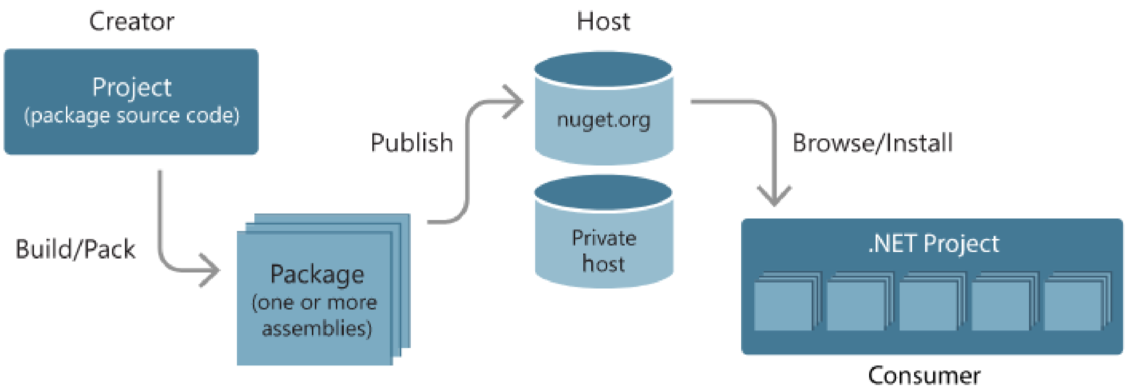


Figure 2.5: The flow of packages between creators, hosts, and consumers.

A "compatible" package means that it contains assemblies created for at least one target .NET framework that is compatible with the target framework of the consuming project. Developers can create framework-specific packages, such as Universal Windows Platform (UWP) controls, or they can offer a broader range of targets. To maximize package compatibility, .NET Standard developers target what all .NET and .NET Core projects can consume. This is the most efficient way for both creators and consumers, as a single package (usually containing a single set) works for all consumer projects.
Package developers who use APIs to make .NET Standard, on the other hand, create separate assemblies for the different target frameworks they support and include all those assemblies in the same package (which is called "multi-targeting"). When a consumer installs this package, NuGet extracts only the project-based sets. This minimizes the package footprint on the final application and/or assemblies produced by that project. The multi-target pack is obviously more difficult for its creator to maintain.[22]

This page was intentionally left blank.

Chapter 3

Vitoria project

3.1 Context

Vitoria is a project included inside a bigger project, the FCT Resilience 4 Covid-19 project, led by the UCP (Universidade Católica Portuguesa). Its partners are the UC (University of Coimbra), OPP (Ordem dos Psicólogos Portugueses) and DGS (Direção-Geral da Saúde). The Vitoria project aims to find metrics and predictive techniques for people's behaviour patterns during and after a pandemic situation. Nowadays, the creation of vaccines for COVID-19 is a widely discussed thematic, and que question is: how will citizens respond if there is no definitive vaccine? Since human behaviour is the most effective mechanism for social control of the pandemic in the absence of a vaccine, predictive models will be created based on human (psychosocial + "smart") and epidemiological data, that will allow the creation of new strategies and resources for mobilization and social resilience in different future scenarios of COVID-19.

The project promoting team, together with the Directorate-General for Health, created a multimethod approach to psychosocial monitoring of risk perception by COVID-19 in Portugal. The method DeCodeR approach [11] allows the quantitative-qualitative analysis of comments from citizens to publications about COVID-19 on the Facebook® social network of DGS and social communication. Since 26 of January 2020, 57,001 comments in the Requirements categories (Danger, Effort, Uncertainty) and Resources (Knowledge and skills, Provisions, Support external) have been collected. In each 4-day cycle, the threat level ratio is calculated based on the Biopsychosocial Threat-Challenge Model [8]: Threat = Requirements / Resources. This ratio was considered indicator of the perception of risk given that the "potential threat" (dread) is the more consistent predictor of risk perception, in the psychometric paradigm of risk [32]. This has allowed DGS to create risk and crisis communication and social mobilization resources based on evidence of changes in risk perception during pandemic.

This is a longitudinal study, with data collected by survey and smartphone, which will allow the identification of predictors of protective behaviours against COVID-19, as well as risk profiles based on sociodemographic and psychosocial variables.

3.2 Collaborations

Vitoria is a project included inside a bigger project, the FCT Resilience 4 Covid-19 project, led by the UCP (Universidade Católica Portuguesa). Its partners are the UC (University of

Coimbra), OPP (Ordem dos Psicólogos Portugueses) and DGS (Direção-Geral da Saúde). The use of several teams from partner entities INSA (Instituto Nacional de Saúde Doutor Ricardo Jorge), OPP and UC made possible to integrate data from multiple sources: extracted, epidemiological data (e.g. daily number of cases), and data collected by social sensors (3 collection slots spaced 1.5 months apart), complying with the RGDP (General Data Protection Regulation): a) by smartphone - mental health indicators and impact of confinement (e.g. stress; anxiety), data aggregated by artificial intelligence (location, physical activity, sleep patterns), sociability (calls / sms, use of social networks) and others; b) by survey and interview - psychosocial indicators of resilience, behaviours of risk, knowledge, confidence in the sources of information, perception of risks, requirements and resources. Hence using time series charts and multilevel analysis aggregating epidemiological, psychosocial data and "intelligent" data, will allow the creation of predictive models of behaviour for future scenarios.

3.3 Functional Requirements

These functional requirements define the project functions that must be performed, in which a function is an input, behavior, and output. With this, we intend to present all the functionalities of the system, such as presenting a tool that allows the verification of the good functioning of the system.

- Login page:
  - Login with facebook, the user will be asked to login with his/her Facebook account, this process becomes automatic if the user has already Facebook installed. If so, it is only required to choose the correct account.
  - Accept permission, the user will be asked to accept the use of sensors, like Global Positioning System (GPS) to detect in which place he/she is in, the sensor of activity to detect movements like walking or using transportation and the scheduler to detect alarms for notifications. The user will be asked to accept the monitoring over other Software Applications (APPs)s.
- Sign-up configuration:
  - Select home Wi-Fi SSID, where the user can either click on the icon of Wi-Fi or write the name of the home network on the textbox aside.
  - Selection of the study case, the user is asked to select the study in which is participating
  - Constrain data usage, the user can constrain the system communication with the server by toggling a switch. Either the systems sends data to the server by Wi-Fi only or by any means possible including mobile broadband.
- Main page:
  - Direção-Geral da Saúde (DGS) information, the user will be able to access the web page of DGS website by clicking on the DGS logo.
  - Information text, there is also a text box with information about the project Vitoria that the user can read, that also has information on the people that helped to develop the project.

  - Navigation, from the main page the user can access the forms, about transportation, about proximity, about app finality, about sleep, the screen about past forms and the settings of the system.
  - System language, the user can change the system language between English and Portuguese.

- Past Forms:
  - This form is triggered by a notification at 4 pm every day, only if there are any forms that have been discarded in the past 24 hours.
  - In this screen, the user can access and answer forms that have been discarded in the last 24 hours.

- System settings:
  - View ID, the user can check his/her ID that should be unique.
  - Home Wi-Fi SSID, the user can modify the home Wi-Fi name previously setup in the sign-up configuration page.
  - Case study, the user can modify the case study previously setup in the sign-up configuration page.
  - Constrain data usage, the user can constrain the system communication with the server by toggling a switch. Either the systems sends data to the server by Wi-Fi only or by any means possible including mobile communication, previously configured in the sign-up configuration page.
  - System language, the user can change the system language between English and Portuguese, previously setup in the sign-up configuration page.

- Sleep form:
  - This form is triggered by a notification, every day at 8:00.
  - Lay time: the user can define the hours at which he went to bed.
  - Wake time: the user can define the hours at which he woke up.
  - The user can then rate the sleep quality on a scale from 0 to 5, between "Very bad" and "Very good".

- Proximity form:
  - This form is triggered when the system detects that many people are around the user.
  - The user should indicate the number of people who are less than 2 meters from him and optionally include in the reply a description of the place or occasion.

- App finality form:
  - This form is triggered by a notification every day at 2 pm.
  - In this screen, the user can evaluate the use of his/her APPss in the last 24 hours. In this form it is supposed to associate each of the most used APPss to the purposes of communication, leisure, research, and work.

- Transportation form:
  - This form is triggered by a notification when the system detects activity of transportation by vehicle.

  - The user can select which vehicle he was in, either, own vehicle, others vehicle, bus, ship, train, or taxi.
  - The user can select the number of people he was with in the same vehicle.

- Emotional form:
  - This form is triggered by a random hour notification every day. If the user does not respond, he will have another opportunity to respond during the rest of the day.
  - This notification is not saved on the past forms screen.
  - The user should choose the images, that best suit his/her emotional state.

- Wear system screen:
  - The user can enable the access to the sensors.
  - The system for the watch only has a screen where the user can check the heart rate value, the number of steps recently monitored and is current activity.

3.4 Non-Functional Requirements

The non-functional requirements should be able to cover all the remaining requirements which are not covered by the functional requirements. They describe how the system Vitoria behave and what limits its functionalities:

- The APP should be active 24 hours and not consume more than 50% of user's device battery life (either on smartphone or smartwatch).
- The user should at least receive 3 notifications per day.
- The smartphone system should load in less than 5 seconds.
- The server should be able to support and gather at least 350 sets of data each day.

3.5 Risk Analysis

Building and maintaining software is a risky activity, where extra costs, delays, or the inability to complete goals can have serious consequences. Larger risks that can undermine long-term projects require immediate attention. Opportunities to collect data are somewhat rare, so, our major concern for the Vitoria Project is that the tests in the real world run smoothly. We evaluated the risks associated with the project and divided the major risks in 3 types: high, medium, and low, and associate a probability to each risk.

3.5.1 Major Risks

- Estimation and scheduling: The nature of this project requires the tasks and time estimations to be strict due to the case studies in the real world, which may change the schedule. This risk is detected when deadlines are not met. This type of risk is considered to be of high impact on the project, but of low probability to happen.

- Sudden growth in requirements: as the project progresses, issues that are not identified earlier can create a last-minute hurdle to meet the deadlines. This risk is detected when more than one task requires developments each week for consecutive weeks, or inconsistency with the time spent on a task in relation to expectations This type of risk is considered to be of medium impact on the project but have a low probability to happen.

- Productivity Issues: since this project is being developed by a team, where each one has a role, if an element has any problem which hinders his/her performance, this can impact the overall development of the project and make it difficult to meet deadlines. This type of risk is considered to be of medium impact on the project and has a medium probability to happen.

- Server problems: This project relies on being centralized around a server to collect data. This is a crucial component to the correct functioning of the project. It may happen that the server has any problem and is unable to collect more data. This type of risk is considered to be of high impact on the project, but with a low probability to happen.

- Compatibility errors: smartphone firmware versions may update/change in the middle of development. This can cause compatibility errors. This risk is detected when development time is spent on system updates and management of development software. This type of risk is considered to be of medium impact on the project and medium probability to happen.

3.5.2 Mitigation Strategies

- Estimation and scheduling: a way to avoid this risk is to use an evasion strategy. To ensure that all members of the group learn and gain experience in the technologies of the environment before the APP development phase with the implementation of personal training, related research, and online courses, will make the development phase run smoother.

- Sudden growth in requirements: a way to minimize the cost of this risk is to use a minimization strategy. To perform a planning and division of tasks considering a worst-case scenario according to the availability of the elements, minimizing the impact of this risk, even if it implies a load greater than another element with greater availability or replacement of an element task with another.

- Productivity Issues: a way to avoid this is to use an evasion strategy. By planning and dividing the tasks considering a worst-case scenario to minimize the impact of this risk, even if it implies a greater load than other risk.

- Server problems: a way to avoid this risk is to use an evasion strategy as to updated backups constantly, if the server has some major problem that compromises its function, a backup image is reinstalled, losing minor to no data.

- Compatibility errors: A way to avoid this is to use an avoidance strategy. This strategy required the definition of alternatives of the tools as the versions of the system roll out from the beginning of the development phase.

3.5.3 Success Criteria

The quality of the project Vitoria will be characterized by its simplicity in use and efficacy on the data collection. To prove the quality of our program, we will count on the participation of our testers and customers.
The most important external quality attribute identified for this project is that the program must be as simple and intuitive as possible so that it will be possible to respond to all forms. In order to test this objective, we performed tests, pointing out the interface problems and rating the problems according to level of severity and impact.
About the internal quality, we considered maintainability to be the key attribute. The code produced was structured, so that each module can be developed individually, in order to facilitate the aggregation of modules during the development phase further isolating possible errors. The written code was also legible so that anyone can modify it if necessary. This was achieved by commenting every section of the code, and by writing down every change during the development phase for the version control. The fact that we present the work done by each element at the end of the week to the group, also allows each of us to express opinions that often lead to quality improvement.

3.6 Project Architecture

The project Vitoria was originated form the project ISABELA, which aimed to monitor a student's life and their environment using physical sensors and virtual sensors. The Vitoria project shares the same system architecture with different objectives. Socialite Platform, is a server capable of receiving IoT data from different sensors, store it, represent it and delivery it to big data platforms and third-party APPs. Currently the devices used to collect data are smartphones and smartwatches.
The platform used was FIWARE, an open-source platform, it provides a rather simple yet powerful set of APIs that ease the development of Smart APPss in multiple vertical sectors. Building around the FIWARE Context Broker, complementary FIWARE Generic Enablers are available. It enables interfacing with IoTs, robots and third-party systems to capture updates on context information and translating required actuations. It also enables the ability to process, analyse and visualize context information by implementing expected smart behaviours of APPs and/or assisting end users in making smart decisions. The Orion context broker generic Enabler currently provides FIWARE NGSI v2 API, which is a Restful API enabling to perform updates, queries or subscribe to changes on context information. The Cygnus Generic Enabler brings the means for managing the history of context that is created as a stream of data which can be injected into multiple data sinks, like mongoDB. The IDAS generic enabler offers a wide range of IoT agents to simplify the interface with devices using popular protocols like LWM2M over JSON. The CKAN extensions Generic Enabler grins a number of add-ons enabling to extend current capabilities of CKAN Open Data publication platform to allow publication of datasets matching right-time context data and the assignment of access terms and policies to those datasets.

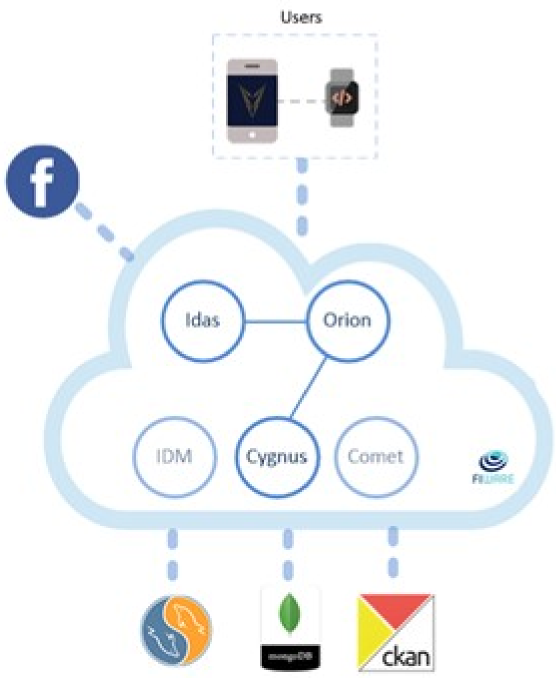


Figure 3.1: Project Vitoria System architecture.

3.7 Software Used

This section presents the software used both during the development phase and during the testing phase.

Table 3.1: Project Vitoria Software Used.

Software	Details	Version
Android Studio	IDE, the most used to develop the Android apps	4.2.2
WearOS by Google App	App that allows the development and debugging of wearable devices running WearOS	2.27
Postman	Software App that allows the development of API schemas and collections, by testing its usage outside the development environment. Was used to test how well the system was collecting data as well as third party APIs like the georeferenced API	7. 33.0

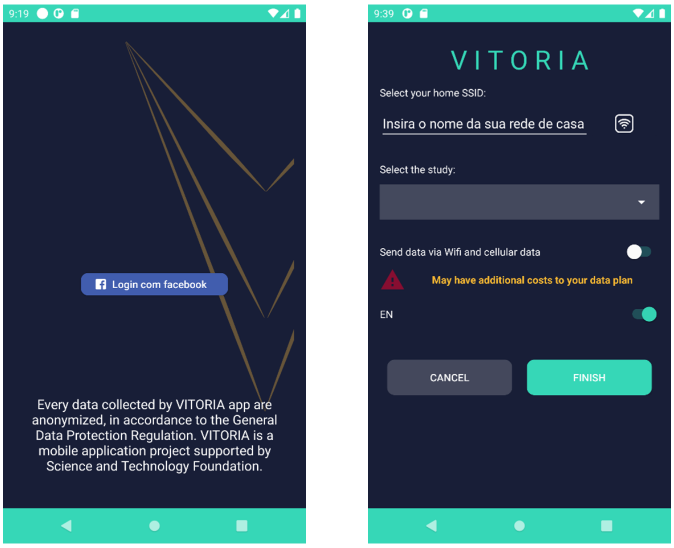
3.8 Implementation Details

During the implementation phase, each element of development was assigned a branch on the Department of Informatics Engineering (DEI) Git server. Weekly, each feature was tested individually and merged with the main branch.
When we started with the development phase, there was a concern with elements with less

knowledge regarding the technologies in use, and the tasks assigned were sometimes the same among some members. But during the development phase the tasks assigned were different between elements.
The author started with tasks more focused on developing the system for the smartwatch, which helped me getting used to the new technology, because the system for the smartwatch is simpler than the mobile system. Since the project was an ongoing development, the system for the smartwatch was already structured. Initially we implemented new features for data gathering from sensors like heart rate, gyroscope, accelerometer, light sensor and barometer, and the author's tasks of this nature were focused on the smartwatch, in order to read/collect as many sensors as possible. A more advanced task assigned to the author was related to the communication between the smartwatch and the smartphone, since this is an important bond, because the data collected from the wearable is sent to the smartphone which in turn is sent to the server.
Along with the development of the forms, it was necessary to implement corresponding notifications such as the alarms that trigger the notification. That is, each form has a notification and a corresponding alarm. In addition to these implementations, it was also necessary to prepare how information is passed to the server. We pass the data collected to JSON, which is then sent and recognized on the Fiware server. All forms have this type of properties, from the creation of the notification, the structuring of storing the data that goes through the definition of objects with rules defined to be similar between forms and thus the process of sending the data to the server is generalized, and the creation of the screen itself with the respective navigations. The author was assigned the tasks of transport form, emotional form and screen forms about the unread forms. Later in the development, new features were implemented, as was the case with location detection, the author was given the task of adding the functionality to detect the county in which the user is located and from the DGS data to recognize the risk associated with the county in question. Considering the information privacy laws, the information related to this type of data was encrypted with the user's private key and only the name of the county name and the corresponding associated risk of Covid-19 infection is sent.
The next set of figures show screenshots taken from the project Vitoria for both the smartphone system and the smartwatch system, where one screenshot was taken for each functional requirement.

3.8.1 Sign up Config

The welcome pages only appear during the first use of the APP. the user can choose to login with a facebook account. If the Facebook APP is installed on the same device, the login process is simpler, you just need to choose the account you want to login and allow the use of the Facebook account. The APP uses several sensors, the GPS is used at the district level, the physical activity to detect movements on transport, the calendar for notifications alarms.
Then the user is presented with a series of permissions popups in which the user is obliged to accept if he wanted to continue using the APP. The second screen is an initial configuration in which the user fills in the spaces on the home network and the study that is part of it.
The user can also choose to send data to the server via mobile broadband or just over the WiFi network. Finally, the selection of the APP's language, between English or Portuguese.

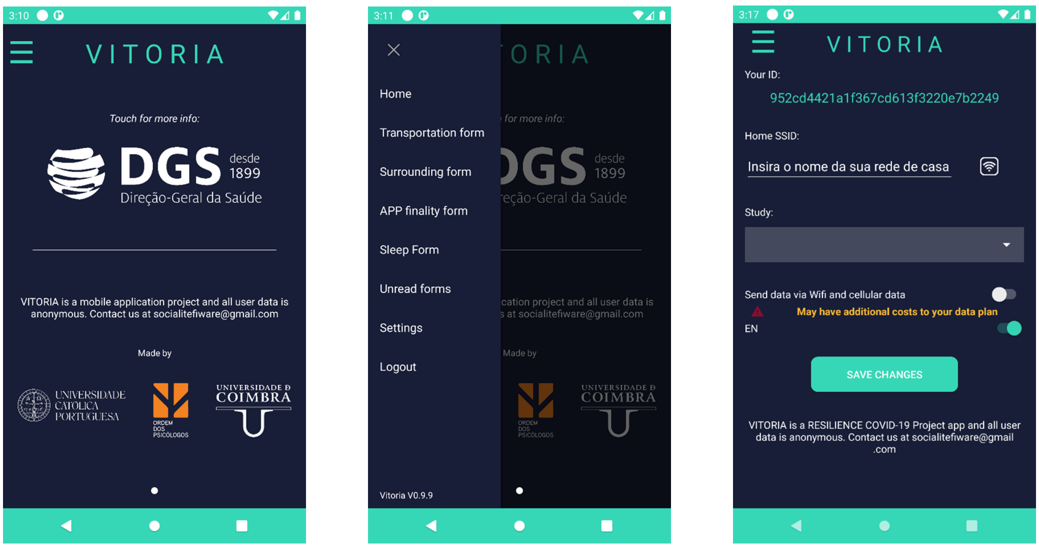


(a) Sign Up with Facebook Page (b) Sign Up configuration

Figure 3.2: Vitoria Welcome Pages.

3.8.2 Main Menu

On the Main page, when it is the first time, a new notification show up in the notification bar, a sign that the APP is working correctly. The DGS or General Directorate of Health in English is a clickable icon so that the user can get more information about the status of COVID-19 in Portugal, when clicked the user is navigated to a web browser of choice to the www.dgs.pt URL. In addition, the user can use a side menu to navigate between the Forms Pages or the Settings Page. The Settings Page serves as an edit Page to the initial configuration during the sign up configuration, but also shows up the user identification number, which is an unique number created with an hash function.

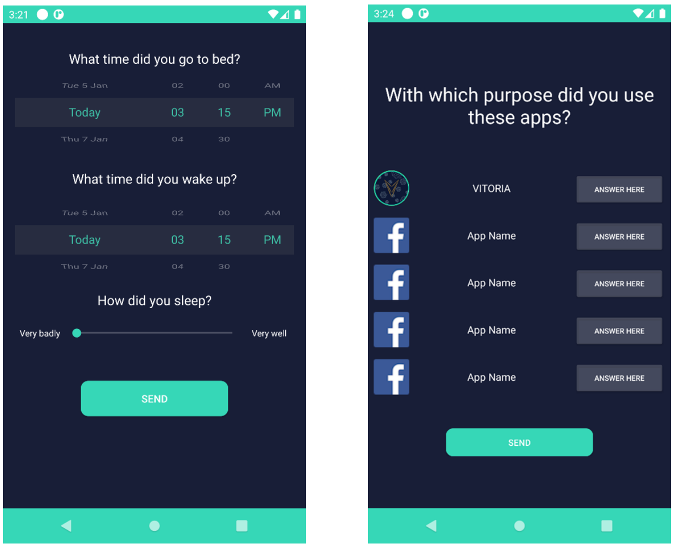


(a) Main Menu Page (b) Drawer Navigation (c) Sleep Form

Figure 3.3: Vitoria Main Menu.

3.8.3 Forms

The APP finality Form asks the user to associate each of the most used APPss with the purposes of: communication, leisure, research and work, and the user can choose more than one option. Every day at 2 pm a notification is launched to evaluate the usage of APPs in the last 24 hours. This was achieved by accessing Android protected data about APP usage times hence the usage access permission that was requested in the first usage. APPss that you have already responded to will be marked as "answered!".

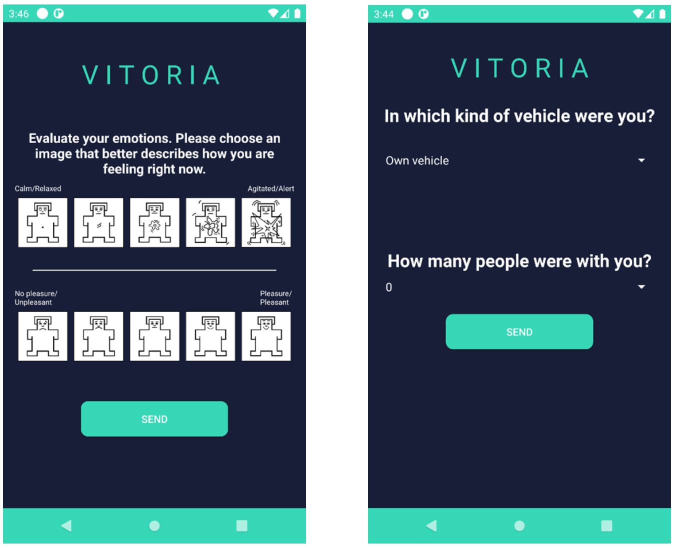


(a) Sleep Form (b) App Finality Form

Figure 3.4: Vitoria Forms Sleep and App Finality Form.

Regarding the Emotional Form, this can only be answered through a notification, never being part of the list of past forms, and this notification is launched randomly during the day. The user is asked to choose two images that best suit his mood at the moment. This has been achieved through image buttons that disable the other images when they are pressed.

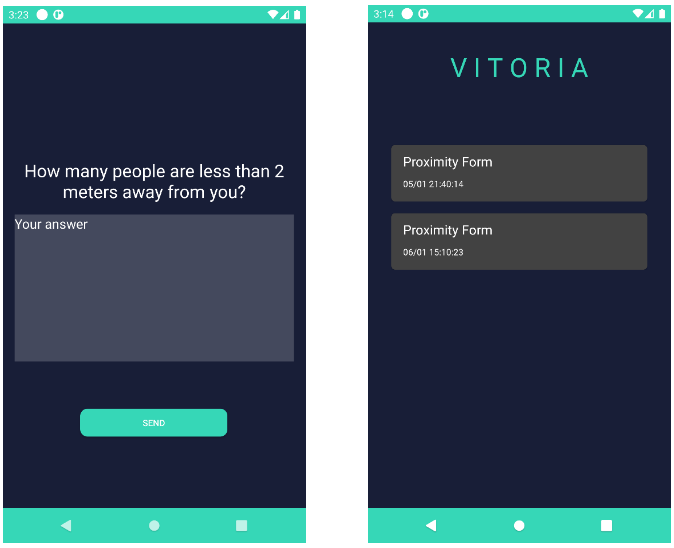
As for the Transport Form, when the application detects that the user is on the move, a notification is launched to respond to the form. This is achieved through the google API on physical activity, this way it is possible with some degree of uncertainty if the user is running, walking, stationary, or in a transport vehicle. Therefore, when movement is detected via the vehicle, we launch such notification. In this form you must indicate what type of vehicle you were in and how many people were with you. This screen has 2 dropdown spiners, which allow quick response within a limited set of responses. For the first spinner that corresponds to the type of transport, we accept your own vehicle, the vehicle of friends/colleagues, bus, boat, taxi/TVDE, Metro/Train/Tram. Since for the following spinner the answers vary according to the first spinner, but they are answers between values, for example for the boat the answers can be from 0 to 10, from 11 to 50, or more than 50.



(a) Emotional Form (b) Transportation Form

Figure 3.5: Vitoria Forms Emotional and Transportation Form.

The past forms screen displays the list of forms that were left unanswered in the last 24 hours. When selecting a form from the list, the user can respond immediately and the form leaves the list. Regarding the Proximity Form, when the APP detects that there are many people around you using the bluetooth sensor, it launches a notification to respond to the form. To answer correctly, the user must indicate the number of people who are less than 2 meters away from him. You can also include in the response a description of the place or occasion, as it is an open response.



(a) App Finality Form (b) Past Forms

Figure 3.6: Vitoria Forms Surroundings and Past Forms.

The following image shows the possible notifications to receive, although a very unlikely scenario of receiving all notifications at the same time, we only present them for the purposes of illustration.

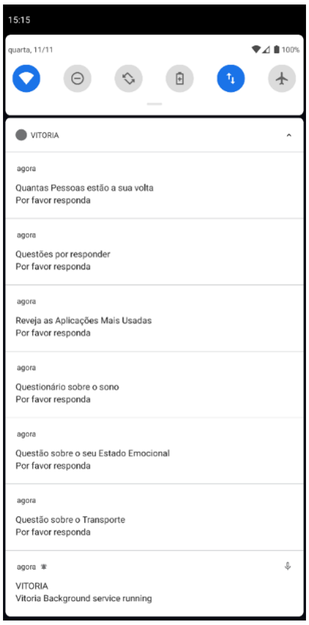


Figure 3.7: Vitoria All Notifications.

Regarding the smartwatch application, this serves in addition to collecting information passively, it also presents some metrics, such as the average heartbeat for 30 seconds and the steps taken so far. The application for the smartwatch was developed in order to minimize interaction with the user.

Figure 3.8: Vitoria Wear Device Main Page.

3.9 Testing

We tested the system continuously since the early stages. Every new functionality was tested individually by each development member, before merging it to the main branch, to avoid malfunctions as much as possible.
The real-world test case is a longitudinal, non-clinical study that will take place wherever the participant is, as long as there is an Internet connection. The study population will consist of 350 participants representing the Portuguese population by gender, age (ie, 18-19, 20-29; 30-39; 40-49; 50+) and region (North, Center, South, Islands), enrolled in the CLSBE-UCP (Católica Lisbon School of Business & Economics - University Católica Portuguese) Online Study Panel (paid) + scholarship for participants in scientific studies COVID-19 (volunteers), of which only 38 will participate in the study carried out with data collection from the Smartphone.

The data collection instruments, in the survey, scales will be used to measure behaviours, capacities (e.g., knowledge), motivations (e.g., self-efficacy), opportunities (e.g., risk exposure) and control variables. Instructions related to data protection were given by the Data Protection Officers (DPO).
Having two samplings of results, a smaller one obtained by collecting data from the Vitoria system and the other from traditional surveys, at the time of writing this thesis, the Catholic university group is still processing the crossing of data from both samples.

This page was intentionally left blank.

Chapter 4

Perturbação Obsessivo-Compulsiva (POC) project

4.1 Context

POC project refers to Obsessive Compulsive Disorder (OCD) in Portuguese, since it is developed by Portuguese institutions the project name was kept in portuguese. The approach to the treatment of diseases can follow a sequential or stratified scheme. In the sequential scheme, an initial standard therapy is instituted, with a progressive and sequential escalation in non-responders. In the stratified approach, the patient starts the therapeutic scheme according to their individual profile and the specifics of their disease. Despite being more effective, the stratified approach is practically not used in psychiatry. The operationalization of the stratified approach in psychiatry depends on the identification and characterization of subtypes within mental illnesses and the identification of markers of severity and response to therapy. Other areas of medicine overcome this problem with the use of complementary diagnostic tests (eg, identification of the severity of diabetes through fasting blood glucose levels and blood glucose levels in the last three months; cancer staging through its spread identified through of imaging studies). In psychiatry, despite the efforts invested by several clinicians and researchers over the years, the proposals for solving this problem have been unsatisfactory. Digital phenotyping techniques, which are beginning to emerge, seem to be promising tools that could revolutionize the performance of diagnostic (and therapeutic) procedures. This is true for medicine in general, but especially, due to the aforementioned gap, in psychiatry. Digital phenotyping encompasses the collection of data – behavioral, psychometric, neuropsychological, neurophysiological – in real time, actively or passively, through electronic devices – mobile phone, microphone, movement sensors and smart watches.

OCD is characterized by the appearance of intrusive, aversive, and anxiety-producing thoughts/mental images (obsessions), which are transiently alleviated by performing repetitive behaviors/mental rituals (compulsions). The time lost with obsessions and compulsions generates great interference (ex.: loss of several hours during the day), reducing the functionality of these people in all areas of their lives (ex.: relational, professional level), and impairing achievement of its potential. As for the treatment of OCD, generally speaking, a sequential approach is used. That is, after the diagnosis is made, a therapy with proven efficacy in reducing obsessive-compulsive symptoms is instituted. Its effectiveness is then evaluated over time. Depending on the outcome of these re-evaluations, the clinician chooses to maintain the "standard therapy" (serotonin reuptake inhibitor antidepressants

+ exposure-response prevention therapy) or add/change the regimen to progressively more specific and/or invasive therapies (eg antipsychotics, deep brain stimulation While cases that respond immediately to standard therapy have a good prognosis, in resistant cases it can be difficult to define the profile of the disease and the appropriate treatment.These cases are associated with a worse prognosis , greater chronicity of the disease and poorer functionality of patients (eg, ability to work) The institution of early and effective treatment is one of the main factors of good prognosis.
Therefore, we consider that the characterization of subtypes within the OCD is necessary to improve the care provided to these people. The daily collection of clinical data from people with OCD, in an approximately naturalistic format, through a mobile application, which was named MyPOC, has the potential to aid in the identification of discrete subtypes within OCD.
The main objective is to collect clinical variables on obsessive-compulsive symptoms (obsessions, compulsions, triggering factors and interference) and associated symptoms (depressive and anxiety symptoms) to future investigation of the correlation, overlapping, and distinction between autism spectrum disorder and Obsessive-Compulsion Disorder. The secondary objectives are to create an online forum for patients with OCD, which allows for discussion between patients and the sharing of psychoeducational topics by health professionals and/or patients. Also to create a helpline where health professionals answer clinic-related questions submitted by patients with OCD. And to create a digital psychotherapeutic exercise program like an exposure-response prevention therapy.

4.2 Collaborations

This project is coordinated and applied on the ground by a team from the Institute of Medical Psychology at Faculdade de Medicina da Universidade de Coimbra (FMUC) and Coimbra Institute for Biomedical Imaging and Translational Research (CIBIT)/Instituto de Ciências Nucleares Aplicadas à Saúde (ICNAS), with the development of the software carried out by a team of researchers from Faculdade de Ciencias e Tecnologia da Universidade de Coimbra (FCTUC).

4.3 Functional Requirements

These functional requirements define the project functions that must be performed, in which a function is an input, behavior, and output. With this, we intend to present all the functionalities of the system, such as presenting a tool that allows the verification of the good functioning of the system.

- Sign Up, that only shows once:
  - Code Authentication, where the code must be unique, the user will be able to insert the code either by typing or using Quick Response (QR) code reader. That code must be matched with the code stored in the data base. And every data gathered by the application must match the code. This functionality is of high priority.
  - Sign up information, where the information of the first login is stored locally. At each login, the personal information if verified, if there is already information

stored locally, the user is redirected to the main page, otherwise, the user is redirected to the sign up page. This functionality is of high priority.

  - Disclaimer is a text box with a notice about personal data gathering and protection. This functionality is of high priority.
  - Form with six questions about the appearance of the patient's disease, where each question accepts open text as a response. The questions are: at what age did the obsessions appeared? Describe the obsession. Is the obsession related to an event or a situation? At what age did the first compulsions start to appear? Describe the compulsion. Is Its appearance related to an event or situation? The screen will be composed of 6 text boxes for user input and 6 labels to display each question at least. The words obsession and compulsion have to be linked to their definition, so the user can search their definition by clicking on them. This functionality is of medium priority.
  - Define reminder notification, where the user will be able to specify at which time the notification to the Daily Self Report (DSR) will pop up, the default is 19:00h. There will be a time spinner where the user will be able to slide time and choose its preferred time. This functionality is of low priority.

- DSR:
  - First question is a multiple-choice question, it includes. The question is which type of obsession do you have in your mind today? The user can only choose one response, and has 7 possibilities: doubt, afraid of something bad to happen, obscene sexual thoughts, obscene religious thoughts, the need for things to be in a specific way (e.g., symmetric things, even numbers or matching), fear of becoming contaminated, or other. This functionality is of high priority.
  - Second question, where user is asked to classify the obsessions about their severity. There are 5 different sub questions:
  - The first sub question is about how much time he occupies with obsessive thoughts and feelings. The user can answer none, less than 1hour, between 1-3 hours, 3 to 8 hours, 8 to 12 hours, more than 12 hours or constantly or almost every time. This functionality is of high priority;
  - The second sub question is about how much consecutive time the user stayed completely free of obsessive thoughts and feelings. The user can answer always (no obsession), more than 8 consecutive hours, between 3 to 8 consecutive hours, 1 to 3 consecutive hours, 1 hour or less than 1 consecutive minute. This functionality is of high priority;
  - The third sub question is about his control degree about the obsessive thoughts. At which point can the user stop thinking about obsessive thoughts. The user can answer complete control, a lot of control, moderate control, some control, minimum control, no control. This functionality is of high priority;
  - The fourth sub question is about what degree of anguish did his obsessive thoughts cause. The user can answer none, mild, moderate, severe, very severe, extreme (are disabling thoughts). This functionality is of high priority;
  - The fifth sub question is about what extent his obsessive thoughts interfered with his social, school, or professional functioning, or did he avoided doing something, going anywhere, or being with someone because of his obsession. The user can answer none or no avoidance, slight or minimal avoidance, moderate or some avoidance, severe or many avoidances, very severe or frequent number of restricted spaces, extreme or can be confined at home. This functionality is of high priority;

- Third question asks which type of compulsion he felt compelled to perform. The user will be able to choose one response between 7 options: check or ask something; do mental rituals to prevent something bad from happening; perform non-mental or uncovered rituals to prevent something bad form happening; organize things to look really good (e.g., aligned, symmetrical or with matching colors); avoid places or people so that thoughts do not appear; wash or clean; other. This functionality is of high priority.
- Fourth question asks the user to classify the severity of his compulsions according to the sub questions. There are 5 sub questions:
  * The first sub question is about how much time he occupies doing compulsive behaviors. The user can answer none, less than 1hour, between 1-3 hours, 3 to 8 hours, 8 to 12 hours, more than 12 hours or constantly or almost every time. This functionality is of high priority;
  * The second sub question is his/her degree of effort to resist compulsive behaviors. The user can answer that he always made and effort or no effort required, that he tried to resist most of the time, that he made a moderate effort to resist, that he made some effort to resist, that he gave in to most compulsions, with some reluctance, that he gave in completely to all compulsions. This functionality is of high priority;
  * The third sub question is about his degree of control over compulsive behaviors, where the user can answer complete control, too much control, moderate control, some control, minimum control, or no control. This functionality is of high priority;
  * The fourth sub question is about his degree of discomfort if prevented from doing compulsive behaviors, where the user can answer none, mild, moderate, severe, very severe, or extreme (feeling overwhelming anxiety). This functionality is of high priority;
  * The fifth sub question is about what extent his obsessive behaviors interfered with his social, school, or professional functioning, or did he avoided doing something, going anywhere, or being with someone because of his obsession. The user can answer none or no avoidance, slight or minimal avoidance, moderate or some avoidance, severe or many avoidances, very severe or frequent number of restricted spaces, extreme or can be confined at home. This functionality is of high priority;
- Fifth question asks how his experience was for a day, according to his obsessions and compulsions. It is a multiple-choice question, where the use can choose 4 different options, definitely excessive and absurd (nothing bad would happen), excessive, reasonable, totally reasonable (something bad could happen). This functionality is of high priority;
- Sixth question about his degree of anxiety, where the user will be presented 3 sub questions:
  * The first sub question is to classify his level of anxiety associated with obsessive-compulsive symptoms. This question requires an open answer. This functionality is of high priority;
  * The second question is to describe the factors (eg situations, objects, places, people) that caused his symptoms. This question also requires an open answer. This functionality is of high priority;
  * The third question asks what he did to get rid of thoughts/behaviors. This question is also an open answer. This functionality is of high priority;

- Normal notification for the DSR:
  - The notification will show up in the notification bar of the user's device. When clicked it will open the self-report screen. This functionality is of high priority.
  - The trigger can be chosen by the user during the sign up or in the settings or else it will be defined by default at 19:00 of each day. This functionality is of low priority.
  - This notification will also show if the user chooses to delay the response of his DSR. This functionality is of high priority.
- Notification for the DSR at 19h:
  - This notification will show up if the user did not complete his DSR and its 19:00 as trigger. This functionality is of high priority.
  - If the user discards this notification, another will pop up and ask him when to respond again. The user will be able to answer 1,2,3,4 hours or not again. If he chooses an hour the normal notification will pop up for the self-report. This functionality is of high priority.
- Choose the next reminder for the DSR:
  - This screen allows the user to choose the next reminder for his next notification to answer the self-report. The user will be asked when he wants to answer the DSR. Where he can choose to answer in the moment, answer in an hour, in 2 hours, in 3 hours, in 4 hours, not again. This functionality is of high priority.
- Main Activity:
  - This is the main page of the system. The user will be able to navigate to the self-report by clicking on a button and do an additional DSR form on that day. This functionality is of medium priority.
  - The user will be able to access external links about their disease, either the definition of obsessions, compulsions or POC. This functionality is of low priority.
  - The user will be able to navigate to other menus using a lateral drawer and clicking on the desired menu. This functionality is of medium priority.
- Weekly neuro-cognitive evaluation notification:
  - The user will be able to define when he wants to do the evaluation. The user will only be able to this evaluation once a week. This functionality is of not mandatory.
- Weekly neuro-cognitive evaluation:
  - The user will be able to define when he wants to be notified to do the evaluation. The user will only be able to this evaluation once a week. This functionality is of not mandatory.
  - The user will participate in a neuro-cognitive activity one a week. This includes a variable mini game, in which the user has to react to some indications. The game registers the user's reaction time in milliseconds, registers the good and wrong responses and the stop/signal delay. This functionality is of not mandatory.
- Monitoring (personal area):

- In this screen the user will be able to review his information. The screen will display graphs about his responses related to his obsession, compulsion, anxiety, and other variables. This functionality is of low priority.

- System configurations:
  - In this screen the user will be able to select the language of the system. This functionality is of low priority.
  - The user will also be able to visualize the unique code, mainly because it will be necessary to share it with the doctor during a medical appointment. The code will be displayed either in QR code or text. This functionality is of high priority.
  - The user will also be able to switch the theme of the system, between dark and light. This functionality is of low priority.
  - The user will be able to change the time at which he wants to be notified to answer the DSR form. This functionality is of low priority.
  - The user will be able to change the time and day at which he/she wants to be notified to do the weekly neuro-cognitive evaluation. This functionality is not mandatory.

The next table summarizes the functional requirements according to its importance nature, in which the rating can be “Must have”, “Should have” or “Could have”.

Table 4.1: Project POC functional requirements nature.

Sign Up	Code Authentication	Must have
	Save sign up information	Must have
	Disclaimer	Must have
	Form with 6 questions	Should have
	Notification Reminder	Could have
Daily Self Report	First Question Group	Must have
	Second Question Group	Must have
	Third Question Group	Must have
	Fourth Question Group	Must have
	Fifth Question Group	Must have
	Sixth Question Group	Must have
Normal notification for the DSR	Push Notification	Should have
Notification for the DSR at 19h	Push Notification	Must have
Choose the next reminder for the DSR	Choose the next reminder	Must have
Main Activity	Navigation	Should have
	External link navigation	Should have
Neuro-cognitive evaluation notification	Push Notification	Could have
Neuro-cognitive evaluation	Neuro-cognitive activity	Could have
Monitoring (personal area)	Graph Display	Could have
System configurations	Language Selection	Could have
	Code visualization	Must have
	Theme switch	Could have
	Self-report notification time	Could have
	Neuro-cognitive activity time	Could have

4.4 Non-Functional Requirements

For this project, non-functional requirements are requirements for the environment that the application will be run in, such as operating system, or details concerning the hardware.

- Internet Connection
- A mobile device running Windows Phone 7 or higher, Android 9 or higher or iOS 11 or higher

4.5 Software Used

Table 4.2: Project POC Software Used.

Software	Details	Version
IntelliJ Rider	IDE, friendlier environment, the most used to develop the apps	211.7442.29
Visual Studio Community 2019 for Windows	IDE, specially used for debugging remote control options when deploying on iOS simulator	16.10.3
Visual Studio 2019 for Mac	IDE necessary to deploy Xamarin apps on iOS devices	8.10
.NET Core	.NET Core SDK and Runtime for building .net core applications	5.0.4
Xcode on macOS	IDE, necessary to deploy iOS simulators and to connect physical iOS devices for debugging purposes	12.5.1
Xamarin Forms	An open-source cross-platform framework for building native UIs for iOS, Android, and Windows from a single, shared C# codebase.	5.0
Xamarin Android	Xamarin SDK for developing Android mobile applications	11.3.0.4
Xamarin iOS & Mac	Xamarin DSK for developing iOS and Mac mobile applications	14.10.0.14
Mono	An open-source implementation of Microsoft's .NET framework based on the European Computers Manufacturers Association standards for C# and the Common Language Runtime	6.12.0.125

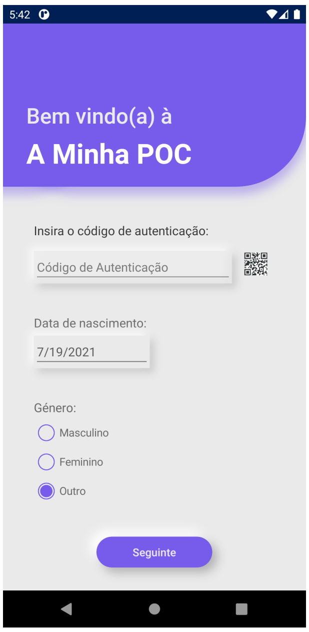
4.6 Implementation Details

In this section we present the implemented screens with the corresponding functionalities as well as how some of the screen were implemented programmatically. In the next subsections the screenshots presented are mostly from the android platform for ways to simplify.
Similar to the last project, we developed this project using DEI Git Server version control, that is, each element was assigned a branch. At the end of each week this branch was

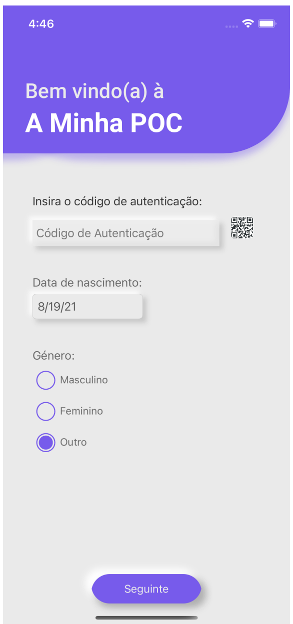
evaluated by the group, fixed bugs in existing ones, improved in certain aspects, and then merged with the main development branch. All screens were developed by the author as he was the only programmer in the graphical development part of the application.

4.6.1 Welcome Pages

Welcome pages are only displayed during the first use of the APP. To find out if it is the first use or not, a Boolean variable is stored in the preferences, as soon as the user completes the welcome pages and reaches the main menu this variable becomes false and from then on whenever the application is opened, the main menu is the first page that is displayed. The next two screenshots correspond to the first page during first use on Android and iOS. The code is the same and although the platforms are completely different, both in terms of operating system and resolution proportions, the results are very similar, practically the same.
On this page, the user can enter the authentication code that will be assigned in the future during the first face-to-face consultation, the date of birth, and gender. To read the authentication code with the phone's built-in camera, the user can click the button that looks like a QR code and point the camera at the code. This was achieved from the ZXing.Net.Xamarin.Forms Nuget Package library that provides QR code scanning for both platforms.
Throughout the application it will be possible to check the insertion of shadows to highlight elements such as text insertion, popups, buttons or other relevant design elements. Although some views such as Pancake View or any View itself have the property of showing shadows, these shadows were quite simple, and with little customization. We chose to use the Sharpnado.Shadows library, which allows the insertion of more than one shadow in any view, being able to assign color, blur radius, opacity and offset.



(a) Android

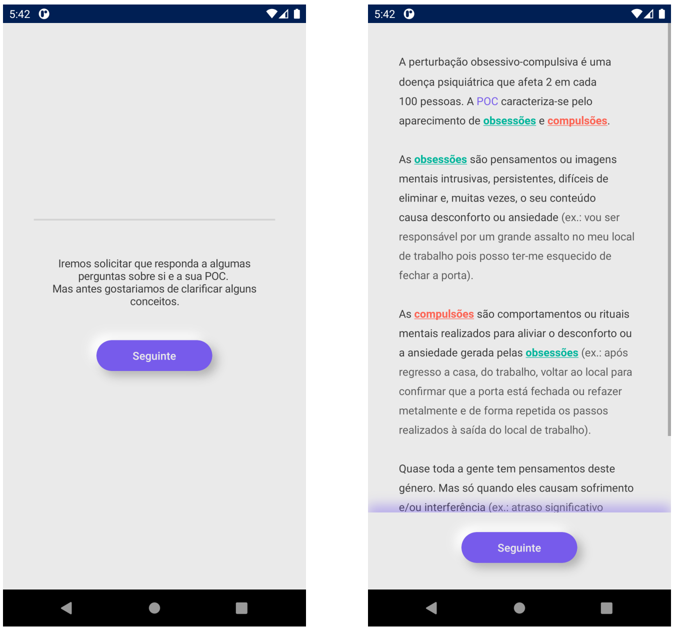


(b) iOS

Figure 4.1: MyPOC Welcome Pages First Page.

On the third page, an informative text is presented to the user in order to make known what the application is about and some definitions about OCD, obsessions and compul-

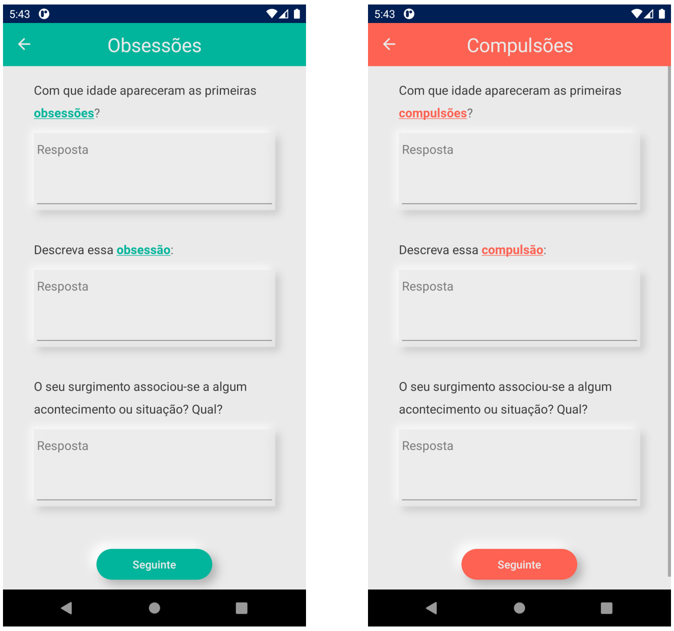
sions. The view used to embed the text was a Scroll View.
Throughout the entire application, whenever obsession or compulsion is written, the user can click on those words and informational popups will appear. This was a long process, achieved through the use of controls. A control serves as an object that when used in modelview code, the same control can have different outputs.
We start by creating a control with a label, that is, whenever during the development of the application we need to present a small text instead of using a normal label view, we use the previously created control that extends a label. Therefore, this text that we want to present can be customized in any way, whether in color, size, any textual change, among others. In the C# code responsible for the control object, some parameters were defined statically, those that are not changed between different texts, such as font family, font size, line spacing and alignment vertically and horizontally.
To be able to change the text in just one word, it was necessary to fragment the initial label view, this is done through a cycle that goes through the intended text and fragments the final text when it finds the word pretended to be changed, which in this case is obsession, compulsion and POC, changing the color, and font attributes (although POC is not clickable, its just the color that is changed).



(a) Second Page (b) Third Page

Figure 4.2: MyPOC Welcome Pages Second and Third Page.

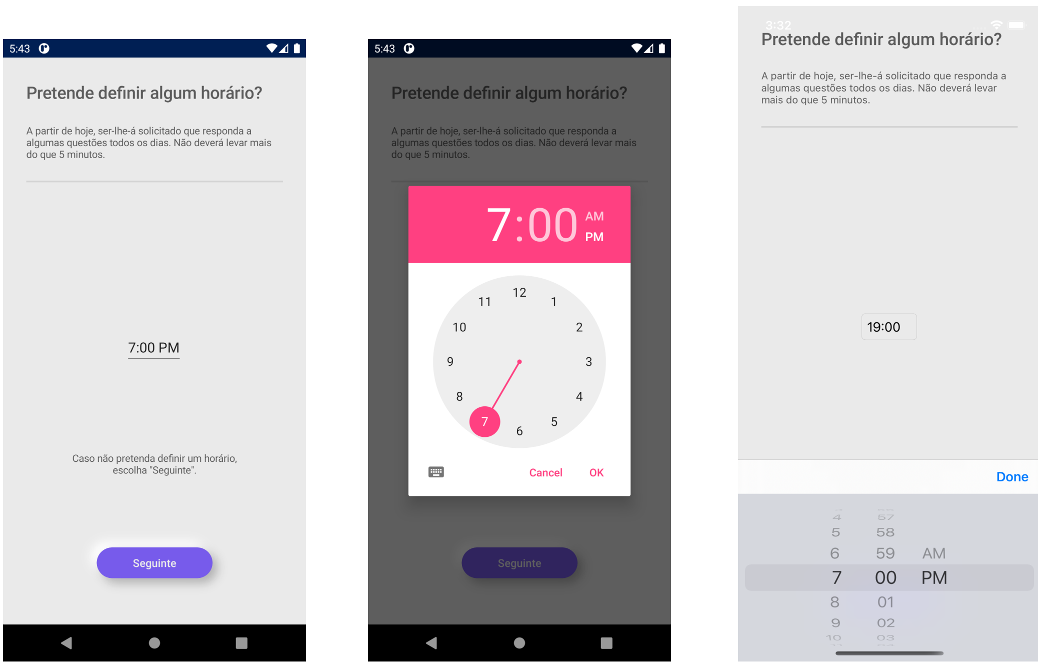
The next two pages feature text entries that allow the user to answer specific questions regarding the first compulsions and the first obsessions. They are made up of the same control object presented above in order to have the functionality to click on keywords plus an entry that allows the insertion of plain text text with multiple lines.



(a) Fourth Page (b) Fifth Page

Figure 4.3: MyPOC Welcome Pages Initial Questions.

The last page of the welcome pages allows the user to change the notification time for a daily reminder to complete the DSR. By default the time is 19:00, but if the user clicks on the time, a system time picker is displayed, and for iOS it is different from Android, as we can see in the following images.

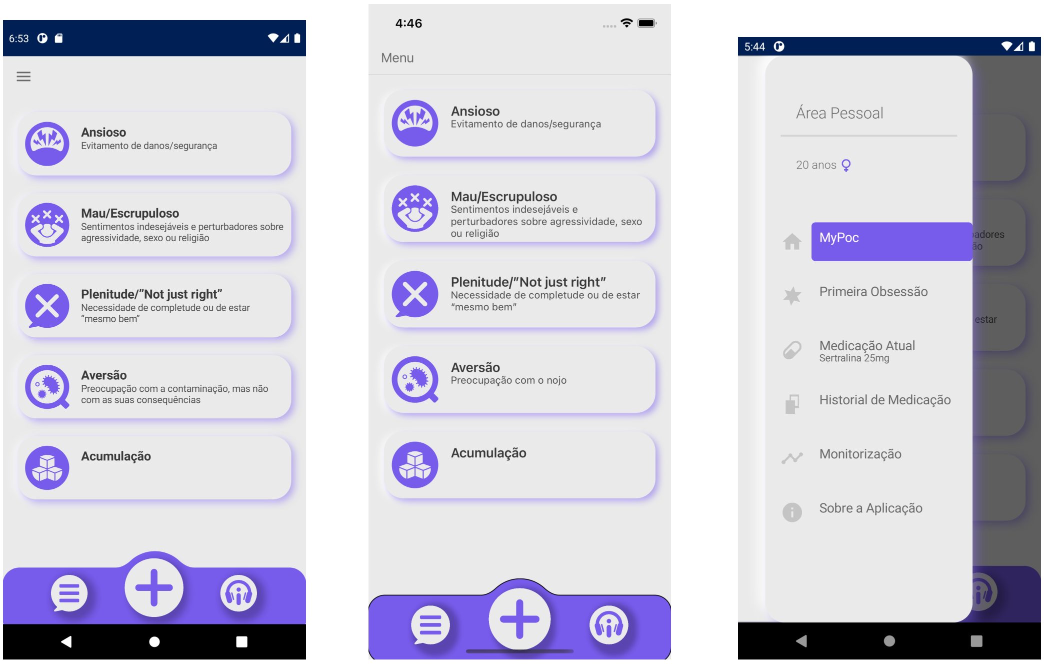


(a) Android Sixth Page (b) Android Time Picker (c) iOS Time Picker

Figure 4.4: MyPOC Welcome Pages Last Page.

4.6.2 Main Menu Page

The main menu page follows the MVVM approach. The list of 5 elements that we can see in the following figures, are part of a list of objects called ObservableCollection found in the file "ViewModel", and each entry is a "Model" object, which has: a tile string, a subtitle string, and image source path, a description string, and at most 4 popup descriptions used for the psychoeducation page. Then the List of elements is then drawn on the screen by the "View" file with a ListView. Each element of this list is constituted by a Frame View with shadows and appropriate spacing. being that the same template is used for all elements of the list, giving a uniform style even though some elements have less content.
The drawer was achieved when the main page is a page of type MaterDetailPage. In this way Xamarin creates a page template prepared to display a side drawer with a swipe command to the right or by clicking in the upper left corner of the screen. The next step was to customize the View responsible for the drawer, in which we once again inserted a ListView based on a template following the MVVM model. In this case, each entry in the list has a specific property that lets you leave the option color selected. When an element is clicked, the list is updated on that property and a new Navigation page is pushed onto the stack.

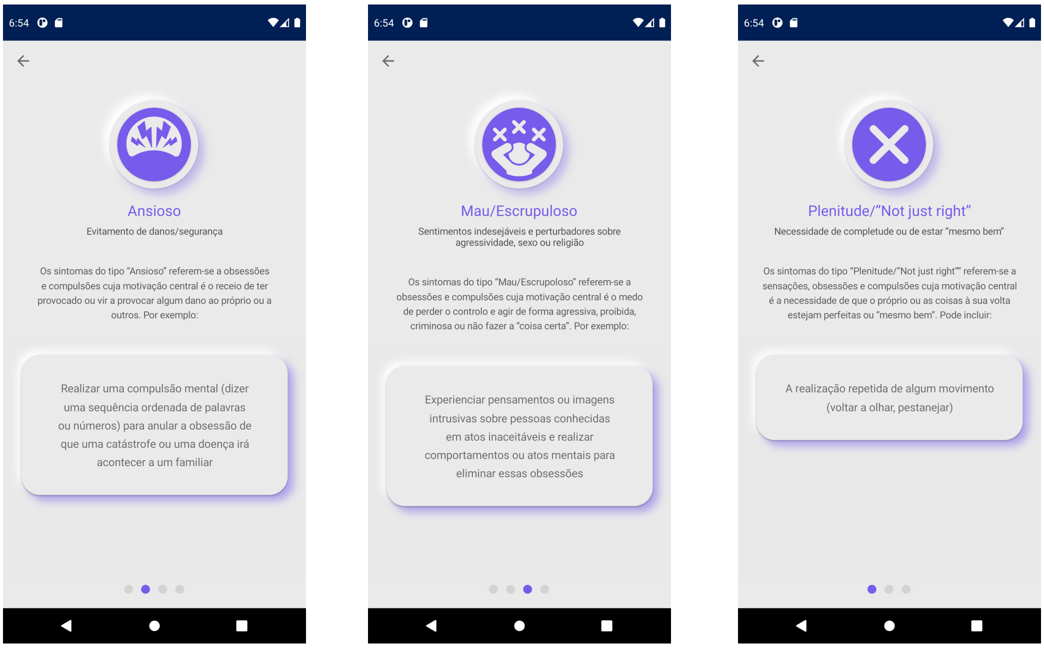


(a) Android (b) iOS (c) Drawer Navigation Android

Figure 4.5: MyPOC Main Menu Page.

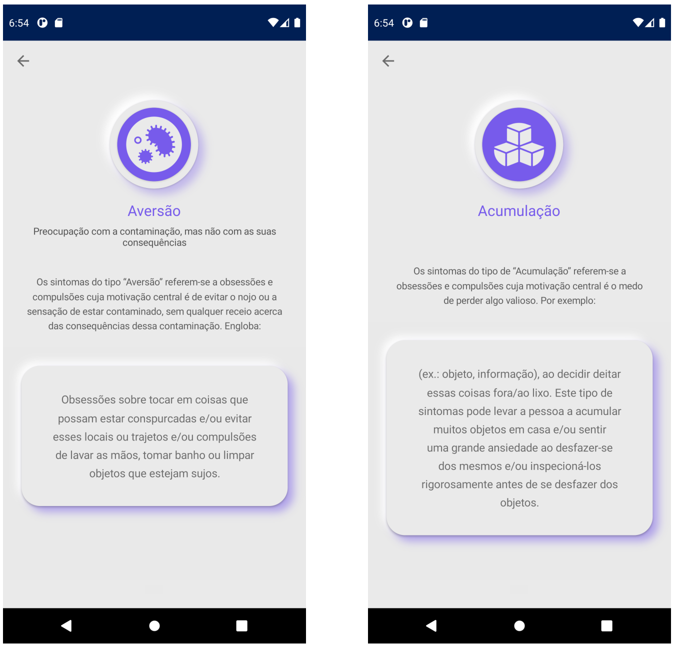
4.6.3 Psychoeducation Pages

Psychoeducation pages are informative pages about types of psychoeducation. There are 5 pages: anxiety, scrupulousness, fullness, aversion and accumulation. These pages were made in a modular way, thus being the same template server for all pages. Each page takes advantage of the control to draw custom text in a Stack Layout View. The bottom of the page is a Carrousel View, allowing the user to swipe to the left or right and read the following description, which in this case are examples of the psychoeducational problem in question.



(a) Anxiety Page (b) Scrupulousness Page (c) Fullness Page

Figure 4.6: MyPOC Psychoeducation Pages.



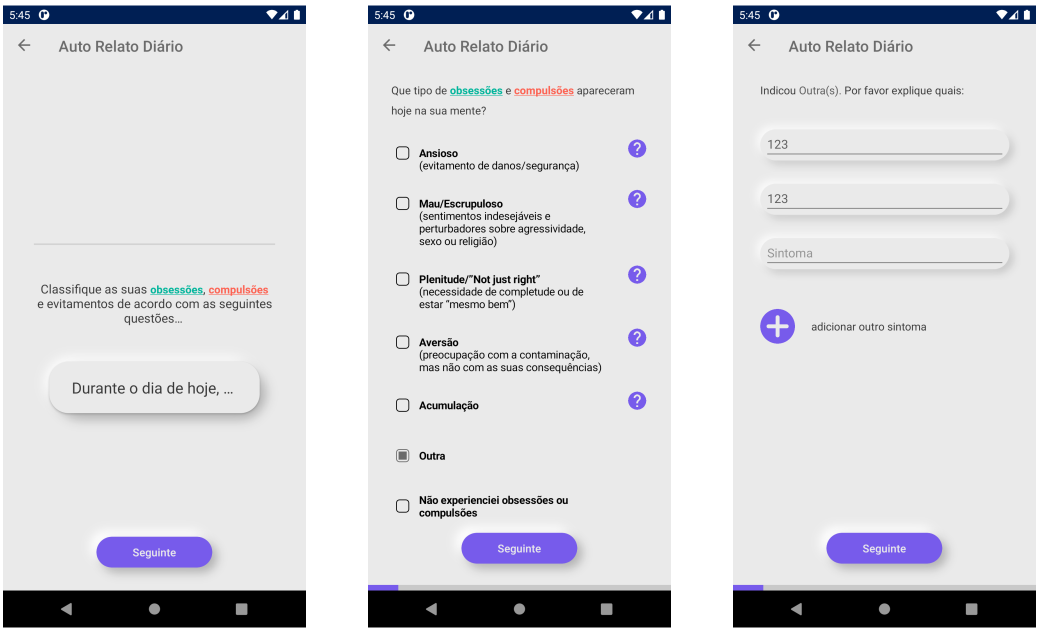
(a) Aversion Page (b) Accumulation Page

Figure 4.7: MyPOC Psychoeducation Pages.

4.6.4 Daily Self Report Pages

The DSR can be accessed through the notification or through the main menu and can be done as many times as the user sees fit. On this first page, the frame appears with an opacity gradient, to draw attention. This was done with the function FateTo() that is available to Views from that Sharpnado Library, the same we used to draw all the shadows.

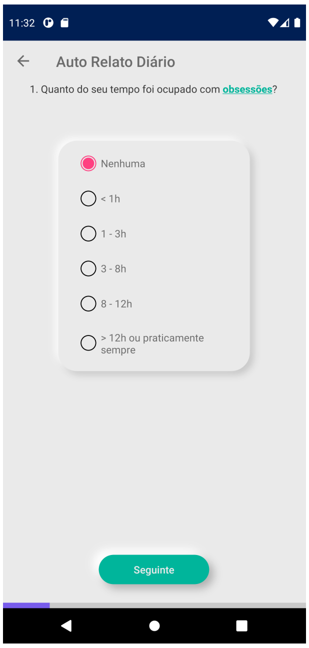
On the second page, the user is supposed to choose which symptoms he has. This is a list of checkboxes and it is possible to choose more than one option. Xamarin's default checkboxes are very basic, allowing virtually no design changes. As such, we use the InputKit library to design the CheckBoxes. If the "Other" option is chosen, by clicking on the continue button, the user is presented with a new screen where he can enter what kind of symptoms he has more. Furthermore, by clicking on the question buttons on the right side of the screen, the user is redirected to the psychoeducation pages presented above. In the lower part of the screen, there is a progress bar, which remains until the end of the DSR, in which the user can know how long it will take to finish the report.



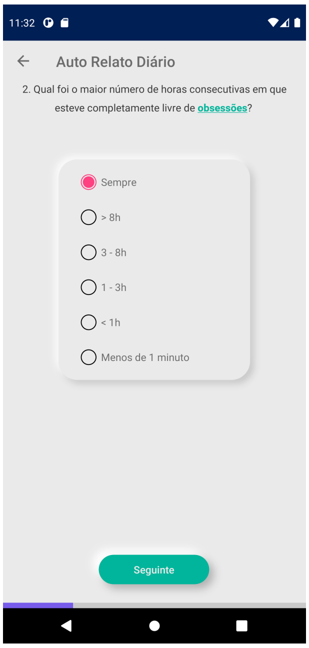
(a) First Page (b) Second Page (c) symptoms Page

Figure 4.8: MyPOC DSR Initial Pages on Android.

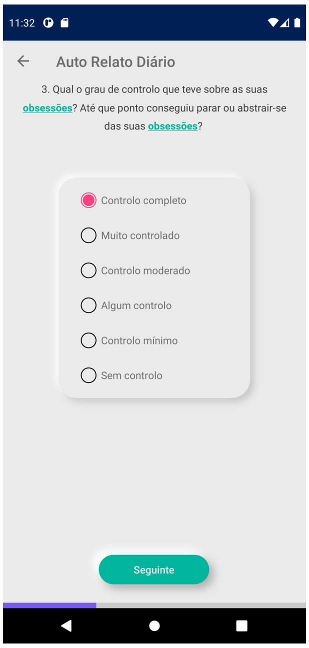
The same principle was applied in the following pages, a general template was created and was used repeatedly throughout the form as practically all pages are similar, just passing the same ViewModel between pages to stay in the same context. This way when the user reaches the end, the answers are all saved in a list and it is easy to save in a database or send to a server.



(a) First Question

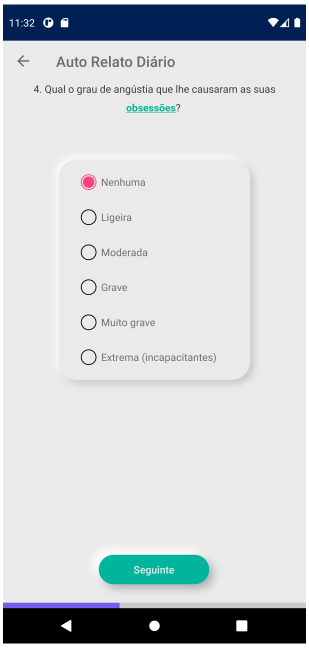


(b) Second Question

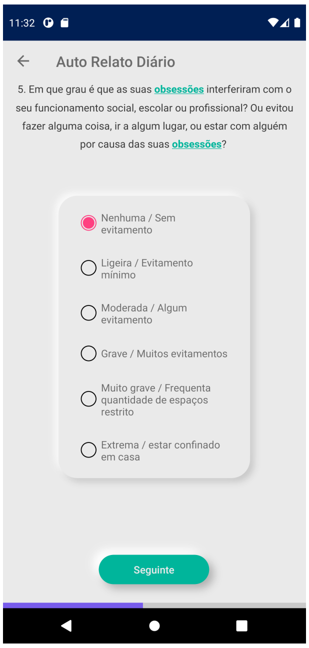


(c) Third Question

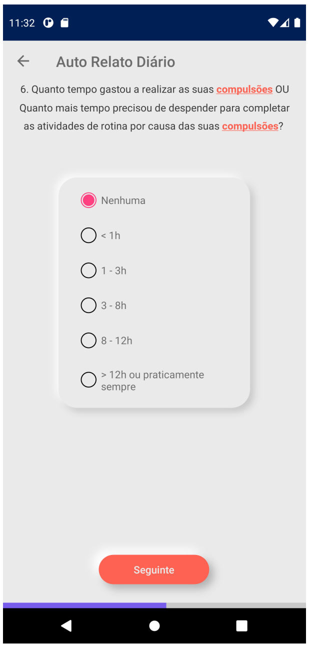
Figure 4.9: MyPOC DSR Questions on Android.



(a) Fourth Question

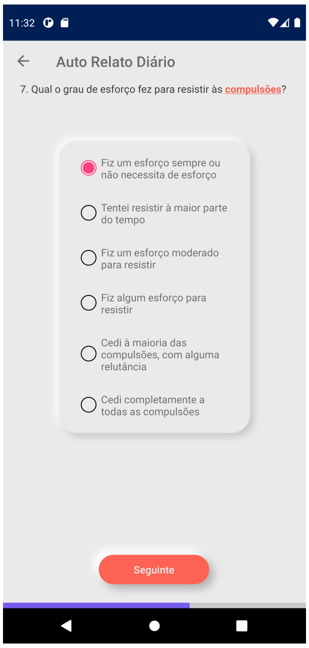


(b) Fifth Question

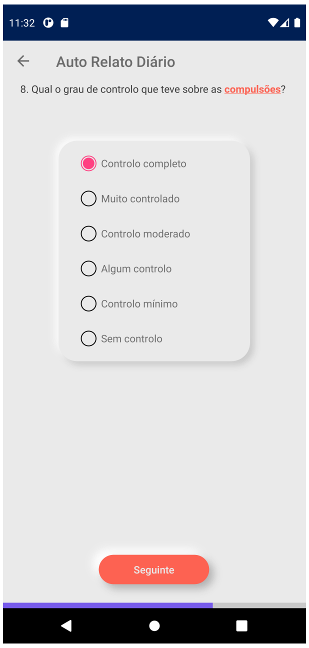


(c) Sixth Question

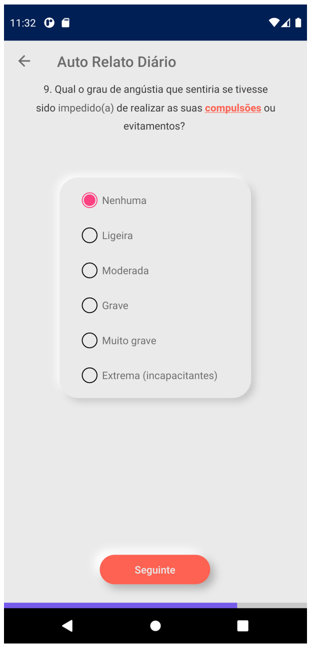
Figure 4.10: MyPOC DSR Questions on Android.



(a) Seventh Question

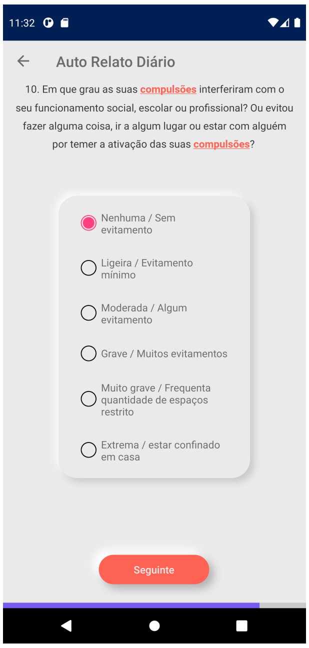


(b) Eight Question

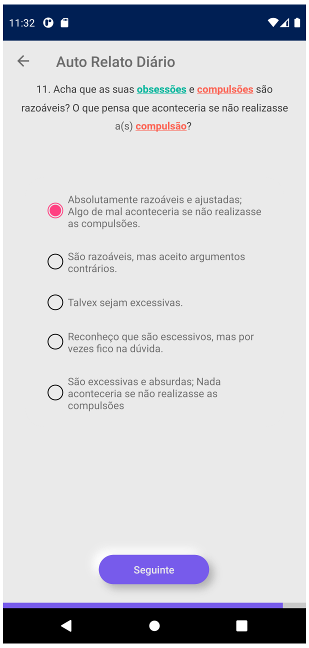


(c) Ninth Question

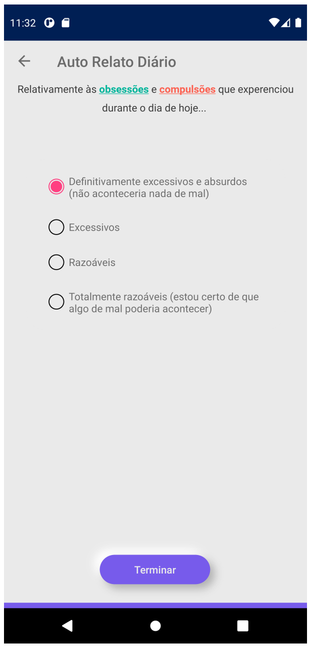
Figure 4.11: MyPOC DSR Questions on Android.



(a) Tenth Question



(b) Eleventh Question



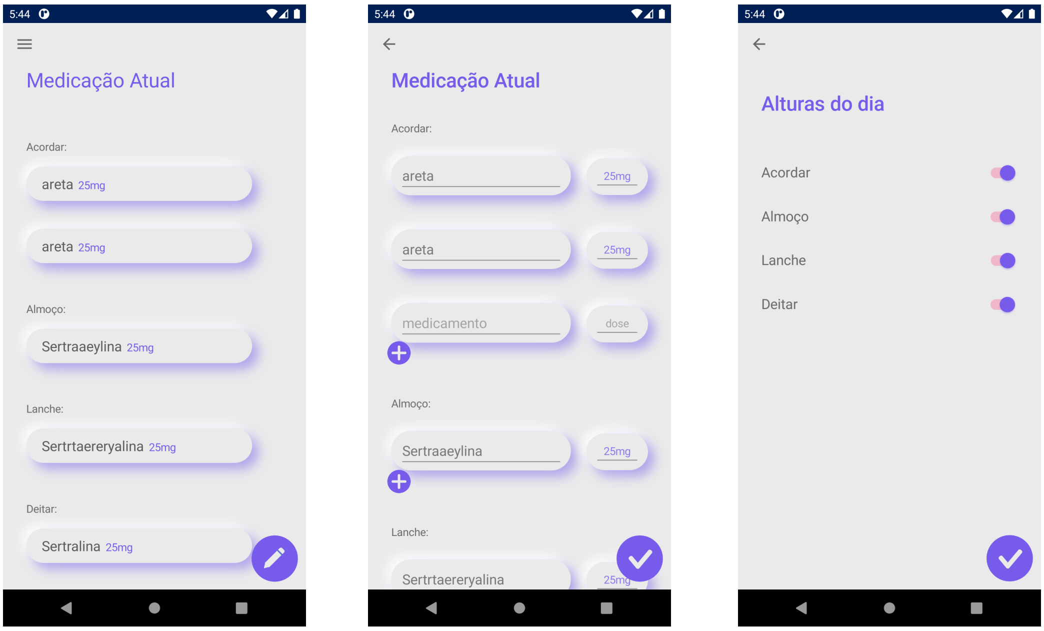
(c) Twelfth and Last Question

Figure 4.12: MyPOC DSR Questions on Android.

4.6.5 Medication Pages

On the current medication pages, the user can: enter new medications, remove past medications, rename current ones, and change amounts. These pages follow the MVVM approach, but this time the View is designed programmatically, since if the user does not have any medication for example at lunch we want to hide the lunch title and corresponding list.

Therefore, the main page has 4 ListViews, one for each part of the day, and each list is made up of an object that has the name of the drug and the amount assigned. On the edit page (on the second), the user can add new drugs by clicking on the More button, to remove elements from the list, just leave the space empty and click on the button in the lower left corner to confirm the changes. All these screens share the same ViewModel which makes it easy to maneuver information between pages.



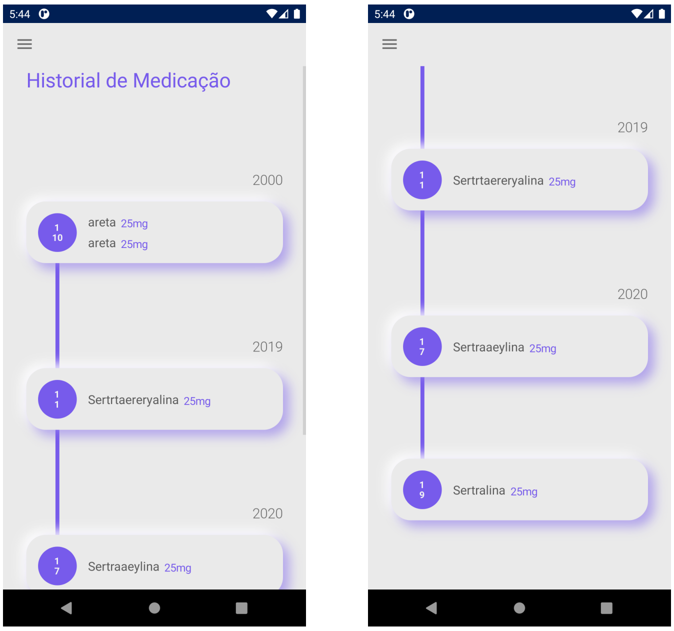
(a) Current Medication Page

(b) Edit Medication Page

(c) Medication Parts of the Day selection Page

Figure 4.13: MyPOC Current Medication Pages on Android<.

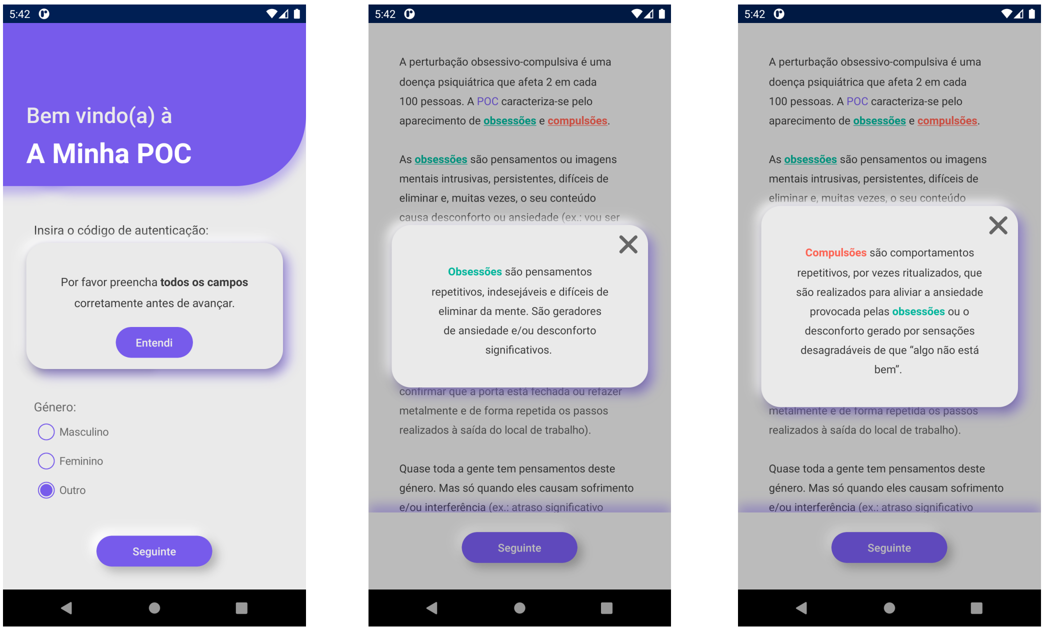
Past Medication pages allow the user to see what they have taken in the past. This page was reached through a relative screen in GridLayout. It is programmatically designed as a new ListView is created and drawn whenever there is a drug in a different year. At the beginning of each list, a label with the year is drawn in the upper left corner of the list that defines the beginning of the drug list. This page becomes a bit more complex than the others because in addition to all these lists, within each element of the lists separated by years, it is still necessary to present all the medications taken by the user in that year, separated by the day on which a certain medication was started. The vertical bar is drawn with a BoxView that starts at the coordinates of the first element of all and ends with the last element of all.



(a) Top of the Page (b) Bottom of the Page

Figure 4.14: MyPOC Medication History Page on Android.

4.6.6 Popup Pages



(a) Popup to prevent incorrect IDs (b) Popup information of Obsession (c) Popup information of Compulsion

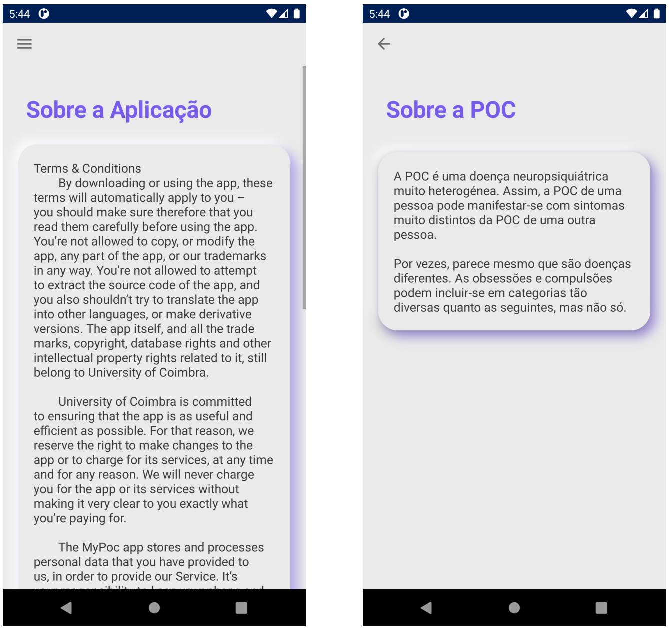
Figure 4.15: MyPOC Popups on Android.

To draw pop ups in all the platforms, the Rg.Plugins.Popup.Pages.PopupPage library was used. The process of drawing elements on the screen was the same as for a normal Page, with a transparent or semi-black background color. The popup return commands were placed in _taskCompletionSource.SetResult, sometimes True or False or any other value.

In this case, for this project, only simple Yes or No commands were needed. Incorrect ID popup is another example of a popup, this appears if the user tries to enter an incorrect ID on the first page of the welcome pages.

4.6.7 Other Pages

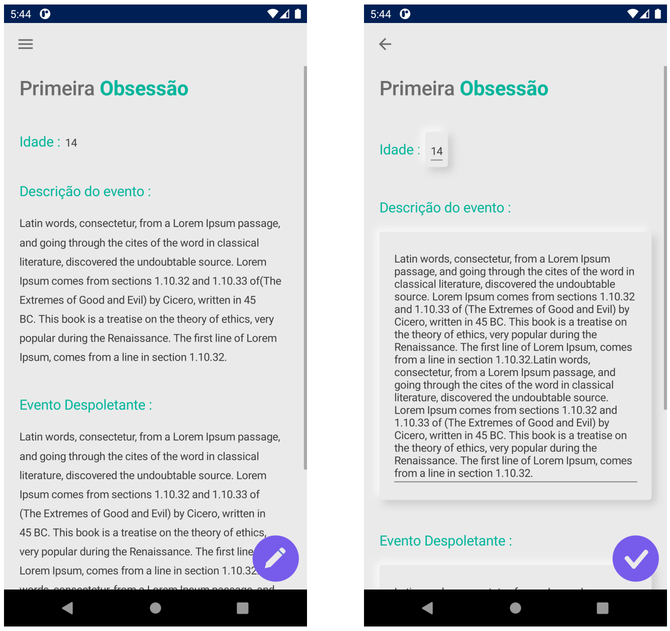
The next two pages are simple as they are for informational purposes only, being made with only one View. The OCD page can be accessed from the Home page in the lower left button, and the application page can be accessed from the navigation drawer.



(a) About the APP Page (b) About the OCD Page

Figure 4.16: MyPOC Other Pages on Android.

The first obsession page can be accessed through the navigation drawer. On this page the user is presented with information about his first obsession which corresponds to the answers he gave during his first use on the Welcome Pages. The edit page is for editing the information presented on the previous page.



(a) First Obsession Information Page (b) First Obsession Edit Page

Figure 4.17: MyPOC First Obsession Pages on Android.

4.7 Library Socialite Sensors

Human in the Loop applications are characterized by a series of features that are usually common among all. This mostly involves collecting information passively, there is no need for a UI, this library aims at the development phase of applications for Human in the Loop systems making it more accessible, faster and less expensive. This library aims to deal with the fundamental functions of a system of this type, and be able to be implemented in any app in a modular way.
The software used in the development process was the same as the software used for the POC project development process hence this section is included in the same chapter.

4.7.1 Library Functional Requirements

This library was developed for use in Xamarin.Forms 5.0 and is ready to be deployed on Android and iOS platforms.

When this library is first used in a mobile APP it will request certain permissions to access sensors and local data. In order to meets its functional requirements it is obligatory to accept all the requested permissions.

- This library should be able to initialize a local database, and save temporarily data until sent to the server.
- This library should be able to capture GPS location on Android and iOS both when its host APP is in background or in foreground.

- This library should be able to read data from motion sensors: Gyroscope, Accelerometer, Gravity and Attitude. If a specific device does not have the indicated sensor the library should only read the ones that do exist without compromising data from other sensors.
- This libray should be able to read values from the proximity sensor.
- This library should be able to detect nearby Bluetooth Low Energy (BLE) and Bluetooth devices. On iOS the library will only be able to detect BLE devices.
- This library should be able to scan nearby WI-FI networks, and identify The Service Set Identifier (SSID) of the connected WI-FI network. On iOS the library wont be able to scan for nearby networks.
- This library should be able to identify activities, such as: Walking, Running, Driving, Moving, Stationary, or unknown.
- This library should be able to identify metrics related to sleep patterns by the user.
- This library should be able to count steps made by the user, serve as pedometer.
- This library should be able to detect light intensity and measure it, exclusively to Android platform.
- This library should be able to identify the next set alarm and measure the time difference, exclusively to Android Platform.
- This library should be able to detect the level of decibels the environment reproduces during a short period of time.
- This library should be able to be initialized with any previously selected sensors and stop them in any order.
- This library should be able to be initialized with a limited selection of sensors within the possibilities. And it must be able to, during the course of the main process, stop reading any sensor. This process must run in the background and must not stop until the application is finished, that is, if the application is in the background or in foreground, the process of this library must be active.

4.7.2 Library Architecture

As explained in the Xamarin instructional chapters, a Xamarin application is divided into 3 parts: The part that has the common code across platforms, the part that has the Android-specific code, and the part that has the iOS-specific code. Since the development is for a library, the code is all placed in the common part.
The basic principle of the library is to work as a service, that is, a consumer of the library to use the functionalities, it calls the library object and executes methods inside that object. For this to happen, the library presents an interface with methods that we want to provide, such as: a method to start the service, another to stop or pause. Furthermore, these methods can be called with specific parameters to perform specific functions, for example if a library user decides that he wants to initialize the service with only the activity sensors active, that user can call the same method of initializing the service with parameters choosing the sensors you want active. This same interface inherits two other interfaces, one for Android and one for iOS, due to some sensors requiring platform-specific code.

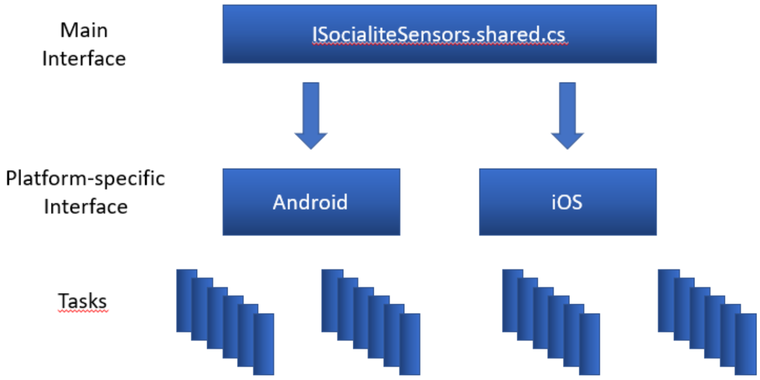


Figure 4.18: Socialite Sensors Library Interface hierarchy

So if a method is called in the main interface, automatically the android and iOS interfaces will execute the called method depending on the target platform.

4.7.3 Library Implementation Details

The development phase of the library was programmed in a team, the author was responsible for all functionalities related to the iOS platform. To start the library service it is needed 3 input parameters:

- A name for the database, which is a String type parameter, and will be used to identify the database, if there is already a database with that name, the content is added, otherwise a new database is created with that name.
- A list of sensors, in which the sensors that the user intends to use are defined, as well as the parameters associated with those sensors. And each sensor is characterized by:
  - The type of sensor, which defines the sensor in question, this type belongs to an enumeration with all types of sensors available, in this way the type is unique and always different. It is expressed in a String Value.
  - Sampling time, which defines the duration of data collection, it defines the continuous time the sensor has to be active to collect data. It is expressed in milliseconds, a Double Value.
  - The work cycle, which defines the time interval that the sensor is stopped between sampling times, that is, it can be understood as a timeout between work cycles. And it is expressed in milliseconds, also a Double Value.
  - The maximum array size, which defines the maximum array size in which each sensor value is stored. For example, for sensors where the pooling rate is based on time rather than triggering events, the array can get very large in a short amount of time while the date is not dumped into the server.

  Some sensors do not need these parameters, but the simple fact that they are assigned parameters does not imply a malfunction of the sensor, the parameters are simply discarded. In the case of the proximity sensor, only the max array size parameter is used, because this sensor only indicates true or false when triggered, when an

occurrence of proximity of the sensor is true, the time is coupled to the value and then is added to the array.
In the case of the activity sensor, whether it is the activity itself, sleep analysis or step counting (pedometer), these sensors correspond to the same category, while on Android it is used a service for counting steps, or recognizing the activity, or for sleep pattern recognition, on iOS the approach is different. iOS has its own activity information database constantly collecting information from the moment the device has the health app installed, which includes a wide variety of content, where most of the content has to be entered manually, however sleep analysis, activity and pedometer are auto-filled metrics. The procedure to collect this type of data is to "ask" for a certificate to the apple developer program that allows us to read this type of data. This type of data cannot be read in a simulator, and to use an app of this style, the certificate must be present on the device, on the macOS machine responsible for implementing the APP, on the Apple account that connects both devices and declared in the file Entitlements.plis. This certificate was the HealthKit, but it was not the only one, for the recognition of the WI-FI SSID, it was also necessary to generate a certificate in the same way being the Network-Info.

- A list with parameters related to the GPS sensor, since for this specific sensor it is necessary a set of different parameters, for a complete customization of it. The parameters are:
  - The polling rate, in milliseconds, corresponds to the time interval that the location is updated.
  - The fastest interval parameter serves to return the location if the location is collapsed more scene than normal, giving the option to preview the previous interval parameter. The value is in milliseconds.
  - The maximum waiting time corresponds to the time that the system waits for the location, if it doesn't get any until it, the system stops trying, serving as a timeout. The value is in milliseconds.
  - The Smallest displacement parameter indicates the accuracy of the sensor in meters. Bigger values imply a larger uncertainty radius.
  - The last parameter is of priority, this parameter is only applied to the Android platform, since this energy consumption factor is managed by the iOS system autonomously. This parameter inside the accuracy of the sensor, allowing for a balance between the time needed to retrieve the position and the energy consumption for that retrieval. In battery saving scenarios the position retrieval time takes longer, and vice versa. There are 4 priority levels:
    * PRIORITY_BALANCED_POWER_ACCURACY - Used to request "block" level accuracy. Block level accuracy is considered to be about 100 meter accuracy.
    * PRIORITY_HIGH_ACCURACY - Used to request the most accurate locations available.
    * PRIORITY_LOW_POWER - Used to request "city" level accuracy. City level accuracy is considered to be about 10km accuracy. Using a coarse accuracy such as this often consumes less power.
    * PRIORITY_NO_POWER - Used to request the best accuracy possible with zero additional power consumption. However, no locations will be returned unless a different client has requested location updates in which case this request will act as a passive listener to those locations.

The collected data is sent to the server in the presence of an internet connection, from json packaged. Structurally the information is collected for two different entities: the more general entity and the sleep analysis only entity. The general entity, called Smartphone-Data, has: user ID in String value, user operating system in String value, stores activity in Stirng value, gps location in json parsed to String value, distance to home in value Double, the time it was created in Universal Time Coordinated (UTC) timestamp and printed as a String value, the step count as an integer, the gravity, accelerometer, attitude, and gyroscope arrays created in json and then each parsed to String value, the light sensor value, in which the maximum and average value between sends is recorded, the Wi-Fi SSID in String value, the scanned bluetooth device array created in json and parsed to String value, the array of proximity, and the sound values, in which the maximum, minimum and average value between sends is registered, and the user's current battery level.

4.7.4 Library Collection and Storage of Data and Confidentiality

The data collected through the application to a database, is only accessed by researchers that have access. Only participants who agree and sign the informed consent will be included in the study. Data will confidentiality be ensured by assigning a code number to each participant that will appear in the informed consent and associated in the database.

4.7.5 Library Background Processing

Background processing or backgrounding is the process of letting applications perform tasks in the background while another application is running in the foreground. Backgrounding in mobile applications is fundamentally different than the traditional concept of multitasking on the desktop. Desktop machines have a variety of resources available to an application, including screen real estate, power, and memory. Applications are able to run side-by-side and remain performant and usable. On a mobile device, resources are much more limited. It is difficult to show more than one application on a small screen, and running several applications at full speed would drain the battery. Backgrounding is a constant compromise between giving applications the resources to run the background tasks they require to perform well, and keeping the foregrounded application and the device responsive. Both iOS and Android have provisions for backgrounding, but they handle it in very different ways.
On android to deploy a process in the background it is necessary to use processes called Services, which allow work to be done without an active user interface. A service notification was used, which allowed to display information on the notification bar, even if the Android application is not in the foreground. It is even possible for a notification to provide actions for the user, and complete tasks, which ultimately is what we were after.
In iOS, backgrounding is recognized as an application state, and apps are moved in and out of the background state depending on the behavior of the app and the user. iOS also offers several options for wiring an app to run in the background, including asking the OS for time to complete an important task, operating as a type of known background-necessary application, and refreshing an application's content at designated intervals. Among all the background modes, the one we used for the library was the location updates mode which not only allows you to do location updates but also allows you to perform other tasks that are not too demanding. The OS is prepared to kill background processes that prevent the application's responsiveness in the foreground. So it's never out of the question for the library process to be killed because of the operating system if the device is completely crammed with high priority foreground processes, for both iOS and Android.

4.7.6 Library Testing

In order to test this library, we adopted an independent procedure, thus unit tests were created, to verify if the functionalities correspond to the requirements. Focusing on the POC project with the integration of the library, during the library development process several parameter settings for sensor reading were tested and taking into account the battery drain, the amount of data acquired over time and the variety of devices, we arrived at the following values that are in the following table. These values may not be suitable for all APP, but they serve as a reference.

Table 4.3: MyPOC APP Library Parameters.

Sensor	Parameter	Value
Bluetooth Low Energy	Sampling Time	300,000ms
Bluetooth Low Energy	Duty Cycle Time	10,000ms
Bluetooth Low Energy	Max Array	100
Bluetooth	Sampling Time	300,000ms
Bluetooth	Duty Cycle Time	10,000ms
Bluetooth	Max Array	100
WI-FI	Sampling Time	1,800,000ms
WI-FI	Duty Cycle Time	10,000ms
WI-FI	Max Array	100
Bluetooth	Sampling Time	300,000ms
Bluetooth	Duty Cycle Time	10,000ms
Bluetooth	Max Array	100
Gravity	Sampling Time	300,000ms
Gravity	Duty Cycle Time	10,000ms
Gravity	Max Array	100
Gyroscope	Sampling Time	300,000ms
Gyroscope	Duty Cycle Time	10,000ms
Gyroscope	Max Array	100
Attitude	Sampling Time	300,000ms
Attitude	Duty Cycle Time	10,000ms
Attitude	Max Array	100
Accelerometer	Sampling Time	300,000ms
Accelerometer	Duty Cicle Time	10,000ms
Accelerometer	Max Array	100

4.8 POC Testing

As the App was ready before September, the testing phase will take place during the month of September and October, and the medicine group is responsible for implementing the field tests.

Chapter 5

Conclusion and Future Work

The development of HitL systems in IoT environments is still relatively recent, and as such there are few works and theses carried out on this theme. This thesis serves to fortify this branch by giving introspective knowledge about the development of applications for mobile platforms in this theme of HitL.
Despite the difficult pandemic scenario in which we live, we managed to achieve significant goals: we completed the development of two projects within a year. Finishing the Vitoria project on time was a very important goal since it is a project that has partnerships with important entities such as the Portuguese Catholic University, the University of Coimbra, the Order of Portuguese Psychologists, and the Directorate-General for Health, which focus mainly on this covid-19 pandemic scenario that we live since 2020.
The second project POC opened the way to the development of a library that is definitely an asset to HitL systems, making the development of future applications more accessible. Although both projects have similarities, during the development phase the approach was totally different, which is another positive point to guide in the right direction possible applications of this type that may be developed in the future, covering multiple development scenarios. Surprisingly the process of learning a new language was smooth, coming from Android to Xamarin. The final product of both projects meets the requirements to the detail and even in the case of the second project the application works incredibly well for both platforms which indicates that Xamarin is a good cross-platform tool for development for scenarios identical to this thesis. The process of creating a library at first seemed difficult and unusual but with the help of the research team it became an accessible and interesting process.
From the first project that focuses more on actively collecting data with the collection of answers from Forms, until the second project that focuses on a psychiatric environment, always taking into account the human reaction during the process of answering the Forms. The care that had to be taken into account encompasses everything from the color palette used in the applications taking into account the target audience, to the average time required for a user to respond to a form and the reaction induced during the reading of the Form's questions (hence the use of image responses for emotional Forms).The Human reaction to any type of answer is never the same, and even the repetitiveness of Forms and the time to be answered have to be factors to consider in a system of this type in which it is necessary to actively collect data.
For future work, this thesis can serve as a basis for the use of applications that influence decisions made by governments on ways to control a pandemic situations, especially the Vitoria project that will help make decisions on the basis of the behavior of citizens. Personalized disease treatment can be revolutionary when we talk about rare diseases with little supporting information or for diseases with different behaviors between patients. The

second project POC focuses on this strand of future work regarding psychiatry, and possibly the same approach can be used for other types of treatments. When it comes to systems that are still scarce, the possibilities for future work are almost unlimited. The library developed focuses specifically on future work, opening up horizons for the development of systems of this type.

References

[1] Adobe phonegap. URL https://phonegap.com/.

[2] Sencha. URL https://www.sencha.com/.

[3] Ibm platform. URL https://mobilefirstplatform.ibmcloud.com/.

[4] O. Azmak, H. Bayer, A. Caplin, M. Chun, P. Glimcher, S. Koonin, and A. Patrinos. Using big data to understand the human condition: The kavli human project. *Big Data*, 3(3):173–188, 2015. doi: 10.1089/big.2015.0012. URL https://doi.org/10.1089/big.2015.0012. PMID: 26487987.

[5] M. Backes, S. Bugiel, E. Derr, P. McDaniel, D. Octeau, and S. Weisgerber. On demystifying the android application framework: Re-visiting android permission specification analysis. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 1101–1118, Austin, TX, Aug. 2016. USENIX Association. ISBN 978-1-931971-32-4. URL https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/backes_android.

[6] C. Ballard, S. Gauthier, A. Corbett, C. Brayne, D. Aarsland, and E. Jones. Alzheimer's disease. *The Lancet*, 377(9770):1019 – 1031, Mar. 2011. ISSN 0140-6736. doi: 10.1016/S0140-6736(10)61349-9. Copyright © 2011 Elsevier Ltd. All rights reserved.

[7] S. Bhardwaj and A. Kole. Review and study of internet of things: It's the future. In *2016 International Conference on Intelligent Control Power and Instrumentation (ICICPI)*, pages 47–50, 2016. doi: 10.1109/ICICPI.2016.7859671.

[8] J. Blascovich, W. Mendes, S. Hunter, B. Lickel, and N. Kowai-Bell. Perceiver threat in social interactions with stigmatized others. *Journal of personality and social psychology*, 80:253–67, 03 2001. doi: 10.1037/0022-3514.80.2.253.

[9] N. Boushehrinejadmoradi, V. Ganapathy, S. Nagarakatte, and L. Iftode. Testing cross-platform mobile app development frameworks (t). In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 441–451, 2015. doi: 10.1109/ASE.2015.21.

[10] Cognito. What is human in the loop machine learning: Why and how used in ai? 2020. URL https://www.cogitotech.com/blog/what-is-human-in-the-loop-machine-learning-why-how-hitl-used-ai/.

[11] S. Domingos, R. Gaspar, H. Fonseca, and J. Marôco. Decoder framework: data collection and coding for demands and resources appraisal in extreme weather events (decoder framework: recogida de datos y codificación de exigencias y evaluación de recursos en fenómenos climáticos extremos). *PsyEcology*, 11(1):90–103, 2020. doi:

10.1080/21711976.2019.1643988. URL https://doi.org/10.1080/21711976.2019.1643988.

[12] M. T. Gardner, C. Beard, and D. Medhi. Using seirs epidemic models for iot botnets attacks. In *DRCN 2017 - Design of Reliable Communication Networks; 13th International Conference*, pages 1–8, 2017. ISBN 978-3-8007-4383-4.

[13] G. Goggin. Google phone rising: The android and the politics of open source. *Continuum*, 26(5):741–752, 2012. doi: 10.1080/10304312.2012.706462. URL https://doi.org/10.1080/10304312.2012.706462.

[14] H. Heitkötter, T. A. Majchrzak, and H. Kuchen. Cross-platform model-driven development of mobile applications with md². In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, page 526–533, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450316569. doi: 10.1145/2480362.2480464. URL https://doi.org/10.1145/2480362.2480464.

[15] N. Koroniotis, N. Moustafa, and E. Sitnikova. Forensics and deep learning mechanisms for botnets in internet of things: A survey of challenges and solutions. *IEEE Access*, 7:61764–61785, 2019. doi: 10.1109/ACCESS.2019.2916717.

[16] I. Kurtev, J. Bézivin, F. Jouault, and P. Valduriez. Model-based dsl frameworks. In *Companion to the 21st ACM SIGPLAN Symposium on Object-Oriented Programming Systems, Languages, and Applications*, OOPSLA '06, page 602–616, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 159593491X. doi: 10.1145/1176617.1176632. URL https://doi.org/10.1145/1176617.1176632.

[17] I. Lage, A. S. Ross, B. Kim, S. J. Gershman, and F. Doshi-Velez. Human-in-the-loop interpretability prior. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 10180–10189, Red Hook, NY, USA, 2018. Curran Associates Inc.

[18] E. A. Lee. Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369, 2008. doi: 10.1109/ISORC.2008.25.

[19] S. A. Lee, E. A., and Seshia. Introduction to embedded systems. a cyber-physical systems approach, second edition. MIT Press, 2017. ISBN 978-0-262-53381-2.

[20] C. Legislative. Legislative counsel's digest, sb-327 information privacy: connected devices. 2018. URL https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB327.

[21] D. McMillen. Internet of threats: Iot botnets drive surge in network attacks. 2021. URL https://securityintelligence.com/posts/internet-of-threats-iot-botnets-network-attacks/.

[22] Microsoft. Microsoft xamarin.forms documentation. 2021.

[23] J. Miguel, R. Araújo, and P. Cambeiro. The human in the loop in cyber-physical systems: the case of building automation. 2019. URL http://hdl.handle.net/10362/96662.

[24] Moustafa, Ahmed, Diallo, Thierno, Amoroso, Nicola, Zaki, Nazar, Hassan, Mubashir, Alashwal, and Hany. Applying big data methods to understanding human behavior and health. *Frontiers in Computational Neuroscience*, 12, 10 2018. doi: 10.3389/fncom.2018.00084.

[25] U. G. P. Office. H.r.1668 - internet of things cybersecurity improvement act of 2020. 2020. URL `https://www.congress.gov/bill/116th-congress/house-bill/1668/text/pl`.

[26] J. Park, H. Chun, and S. Jung. Api and permission-based classification system for android malware analysis. In *2018 International Conference on Information Networking (ICOIN)*, pages 930–935, 2018. doi: 10.1109/ICOIN.2018.8343260.

[27] E. PARLIAMENT. Regulation (eu) 2019/881 of the european parliament and of the council. 2019. URL `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32019R0881&from=EN%22%20%5Ct%20%22_blank`.

[28] D. R.S., G. D.S., F. M.R., S. Y., M. M., and M. J. Efficacy and safety of donepezil 23 mg versus donepezil 10 mg for moderate-to-severe alzheimer's disease: A subgroup analysis in patients already taking or not taking concomitant memantine. 33:164 – 173, 2012. doi: 10.1159/000338236.

[29] A. Sarkar, A. Goyal, D. Hicks, D. Sarkar, and S. Hazra. Android application development: A brief overview of android platforms and evolution of security systems. In *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 73–79, 2019. doi: 10.1109/I-SMAC47947.2019.9032440.

[30] H. N. Security. Threat highlight: Analysis of 5+ million unmanaged, iot and iomt devices. 2020. URL `https://www.helpnetsecurity.com/2020/07/24/analysis-of-5-million-unmanaged-iot-and-iomt-devices/`.

[31] S. H. Shah and I. Yaqoob. A survey: Internet of things (iot) technologies, applications and challenges. In *2016 IEEE Smart Energy Grid Engineering (SEGE)*, pages 381–385, 2016. doi: 10.1109/SEGE.2016.7589556.

[32] P. Slovic. Perception of risk. *Science*, 236(4799):280–285, 1987. ISSN 0036-8075. doi: 10.1126/science.3563507. URL `https://science.sciencemag.org/content/236/4799/280`.

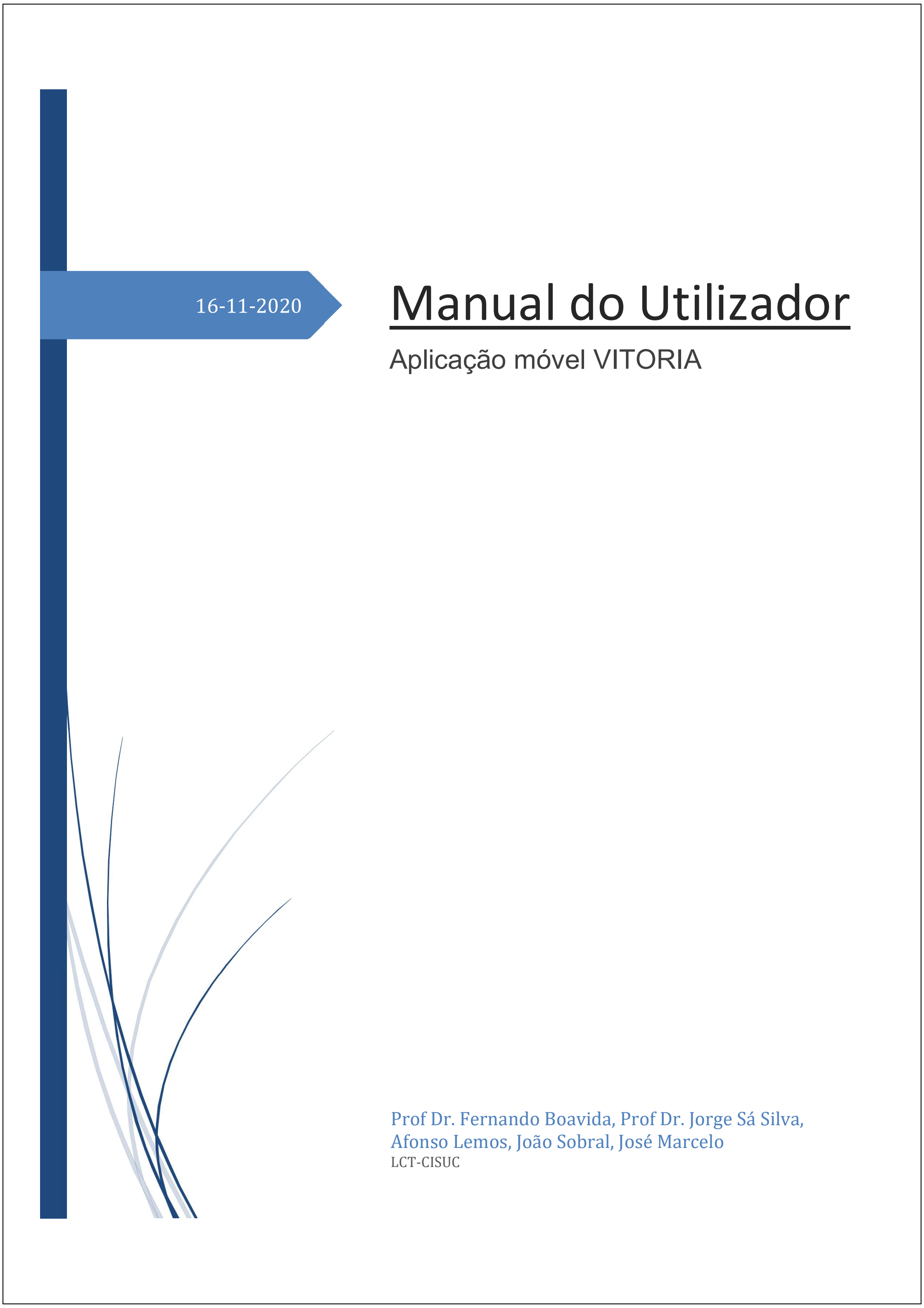
[33] K. Vishal and A. S. Kushwaha. Mobile application development research based on xamarin platform. In *2018 4th International Conference on Computing Sciences (ICCS)*, pages 115–118, 2018. doi: 10.1109/ICCS.2018.00027.

[34] K. Vishal and A. S. Kushwaha. Mobile application development research based on xamarin platform. In *2018 4th International Conference on Computing Sciences (ICCS)*, pages 115–118, 2018. doi: 10.1109/ICCS.2018.00027.

This page was intentionally left blank.

Appendix A

Vitoria User Manual



16-11-2020

Manual do Utilizador

Aplicação móvel VITORIA

Prof Dr. Fernando Boavida, Prof Dr. Jorge Sá Silva,
Afonso Lemos, João Sobral, José Marcelo
LCT-CISUC

Conteúdo

1 Introdução

A aplicação Vitória insere-se no contexto do projeto Resillience 4 Covid-19, que tem como objetivo encontrar métricas e técnicas preditivas dos padrões de comportamento das pessoas durante e após uma pandemia. Um cenário amplamente discutido prevê a criação de vacina para a COVID-19. Mas como responderão os cidadãos noutros cenários, e.g. se não existir vacina? Sendo o comportamento humano o mais eficaz mecanismo de controlo social da pandemia na ausência de vacina, serão criados modelos preditores deste baseados em dados de sensores humanos (psicossociais + dados "inteligentes") e epidemiológicos, e propostas estratégias e recursos de mobilização e resiliência social para diferentes cenários futuros. Este é um estudo longitudinal, com dados recolhidos por inquérito e por smartphone, que permitirão identificar preditores de comportamentos de proteção face à COVID19, bem como, perfis de risco com base em variáveis sociodemográficas e psicossociais.

Trata-se de um, estudo longitudinal não clínico que decorrerá em qualquer lugar onde o participante se encontre, desde que exista ligação a Internet. A população do estudo será constituída por 350 participantes representativos da população portuguesa por género, idade (i.e., 18-19, 20-29; 30-39; 40-49; 50+) e região (Norte, Centro, AML, Sul, Ilhas), inscritos no Painel de Estudos Online da CLSBE-UCP (remunerados) + bolsa de participantes em estudos científicos COVID-19 (voluntários), dos quais apenas 38 participarão no estudo realizado com recolha de dados do Smartphone.

Instrumentos de recolha de dados: No inquérito serão usadas escalas de medida de: comportamentos, capacidades (e.g. conhecimento), motivações (e.g. autoeficácia), oportunidades (e.g. exposição ao risco) e variáveis controlo. Procedimentos: Serão integrados dados já extraídos das redes sociais com dados epidemiológicos disponíveis publicamente, e dados recolhidos por sensores sociais via inquéritos (3 vagas de recolha com espaçamento de 1.5 meses) e por um sistema constituído por smartphone e smartwatch (extração diária).

2

2 Login

2.1 Permissões

Na primeira vez que abre a aplicação é requerido que efetue o login com a sua conta do Facebook. Quando se seleciona o botão de login são pedidas permissões de acesso de uso, como se apresenta na Figura 1.

Adicionalmente, são pedidas permissões para utilização dos sensores de atividade física, do GPS, do calendário, do acesso ao sistema de ficheiros e do microfone. Na Figura 2 são apresentados os passos necessários para o correto uso e configuração da aplicação, onde os retângulos a vermelho correspondem às seleções do utilizador.

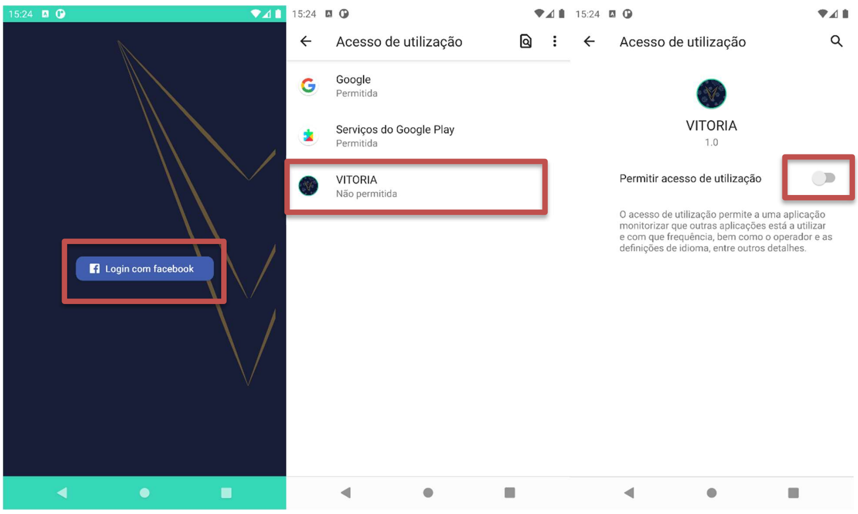


Figura 1 Permissão para acesso ao histórico de aplicações.

NOTA: Só depois de ter concedido as permissões necessárias para o correto funcionamento da aplicação, é que pode avançar para o próximo passo e fazer login pelo Facebook.

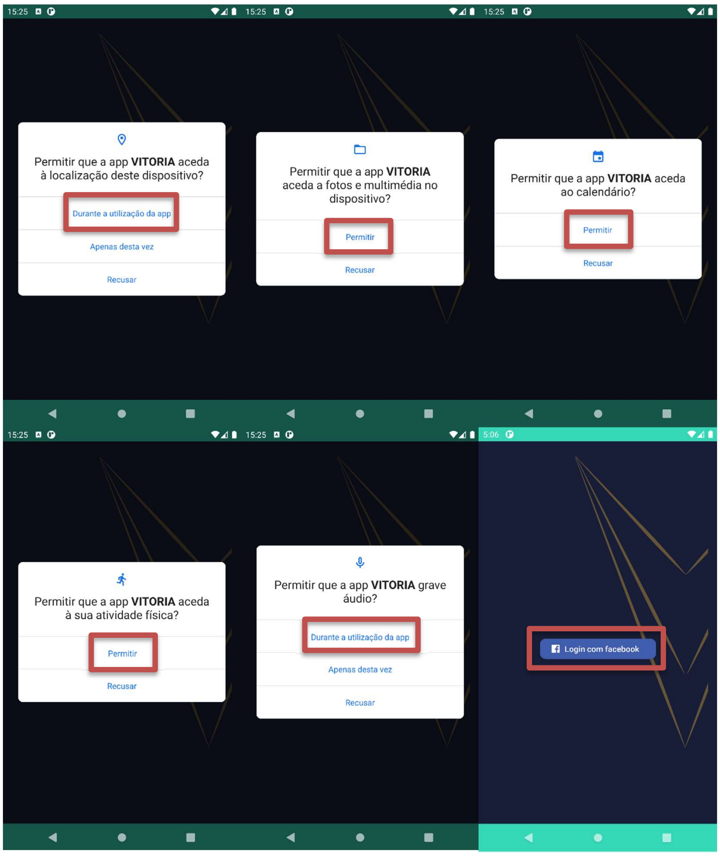


Figura 2 Permissões de utilização dos sensores e de captura dos vários dados.

A aplicação recorre a vários sensores, o de gps é usado a nível de distrito, o de atividade física para detetar movimentos sobre os transportes, o calendário para os alarmes das notificações

2.2 Login com Facebook

O passo seguinte é aceitar os cookies do Facebook. É importante salientar que o processo de login do Facebook é inteiramente controlado pelo Facebook e a aplicação não irá ter acesso a nenhuma das suas informações pessoais. Deve, então, inserir as suas credenciais para iniciar sessão, selecionar “log in” e, de seguida, “continuar” como é demonstrado na Figura 3.

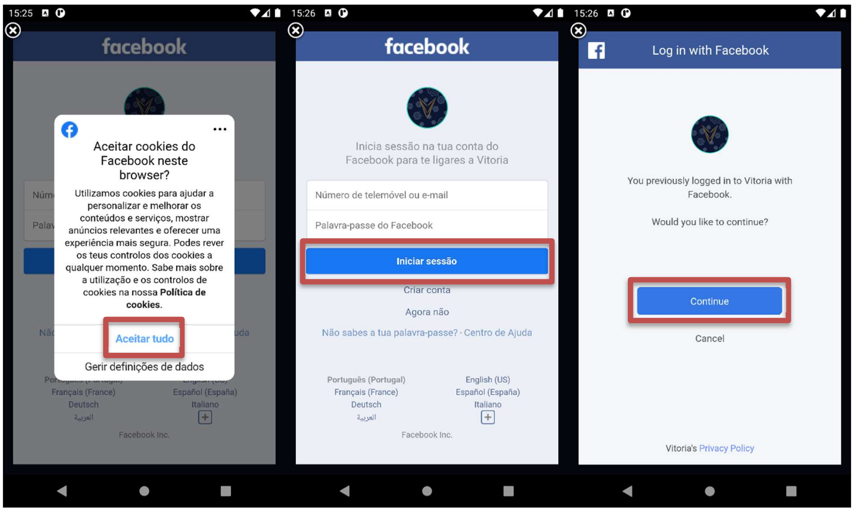


Figura 3 Login com o Facebook.

3 Preferências iniciais

Após a conclusão do login é apresentado um ecrã onde pode configurar algumas das suas preferências. Como é apresentado na Figura 4, é solicitado o nome da rede de casa (pode escrever manualmente ou selecionar o ícone com o símbolo de WiFi para identificar as redes na periferia e selecionar a rede de casa) e o tipo de estudo - nesta opção deve escolher UCL. Pode optar por enviar dados para o servidor a partir de dados móveis ou só através da rede WiFi. Por fim pode escolher a linguagem da aplicação, entre inglês ou português.

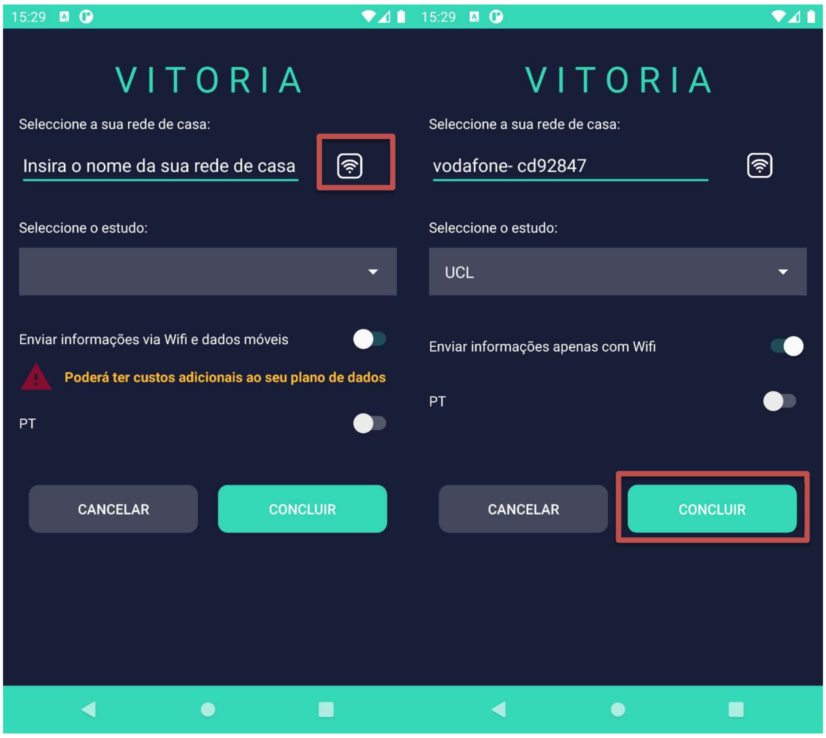


Figura 4 Configuração Inicial/ Preferências do utilizador.

Nenhuma destas preferências é permanente, poderá sempre alterar cada uma no painel de definições da aplicação - ver secção 7 Definições. Selecione Concluir/Finish para continuar.

NOTA: Se escolher dados móveis poderá implicar a ter custos adicionais de acordo com o seu plano de dados.

6

4 Página Principal

Na Página principal, quando entra pela primeira vez irá aparecer uma nova notificação, é sinal de que a aplicação está a funcionar corretamente. Pode tocar no ícone da DGS para obter mais informações sobre o estado da COVID-19 em Portugal. (*www.dgs.pt*)

Figura 5 Página Principal

7

5 Menu de navegação

Deslize com o dedo da esquerda para a direita ou toque no ícone das 3 linhas na margem esquerda superior para puxar o Menu de Navegação, apresentado na Figura 6. Pode responder manualmente ao Formulário de Sono, à Finalidade da App, ao Formulário de Proximidade e ao Formulário de Transporte (ver secção 8 Formulários). Pode rever os Formulários que deixou por responder durante as últimas 24 horas. (ver secção 6 Formulários por responder).

Também pode navegar para o menu de Definições onde pode alterar as suas preferências que configurou na secção de 3 Preferências iniciais (ver secção 7 Definições).

Pode também fazer *Logout* para desassociar a sua conta de Facebook. Se quiser voltar a fazer login, será necessário repetir os passos mencionados na secção 2.2 Login com Facebook.

.

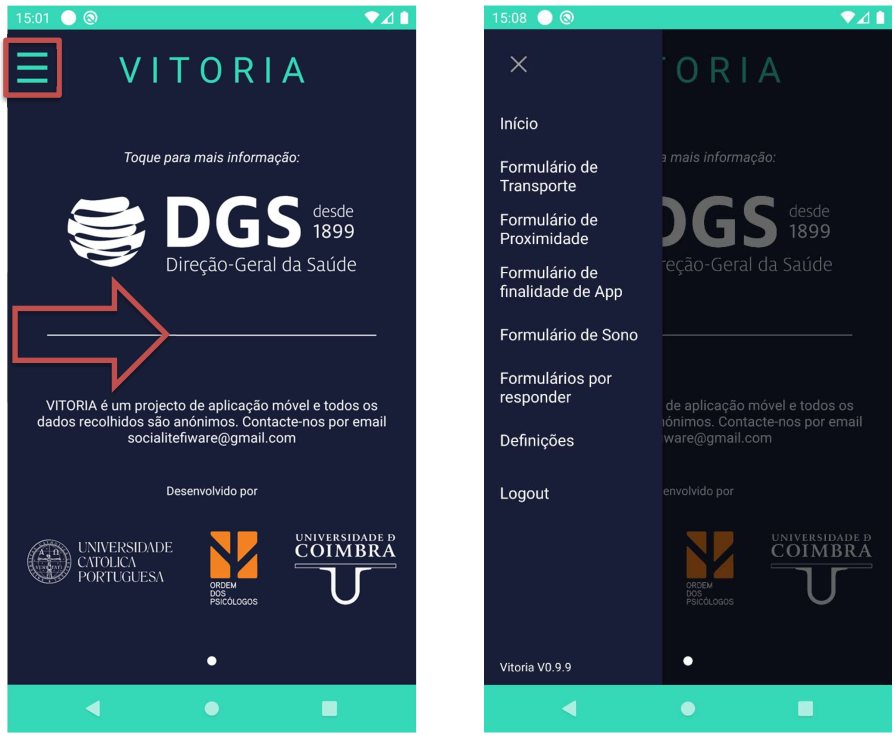


Figura 6 Menu de Navegação.

8


9

6 Formulários por responder

Esta página apresenta a lista de formulários que ficaram por responderem nas últimas 24 horas. Ao selecionar um formulário da lista, pode responder de imediato e o formulário desaparecerá da lista.

Figura 7 Formulários por responder

10

7 Definições

Neste ecrã o utilizador pode alterar as preferências da app que foram definidas na secção de 3 Preferências iniciais. Caso lhe seja pedido para colocar o id de associação da app do smartphone deve copiar os últimos 5 dígitos do seu id, que pode ser visto na Figura 8.

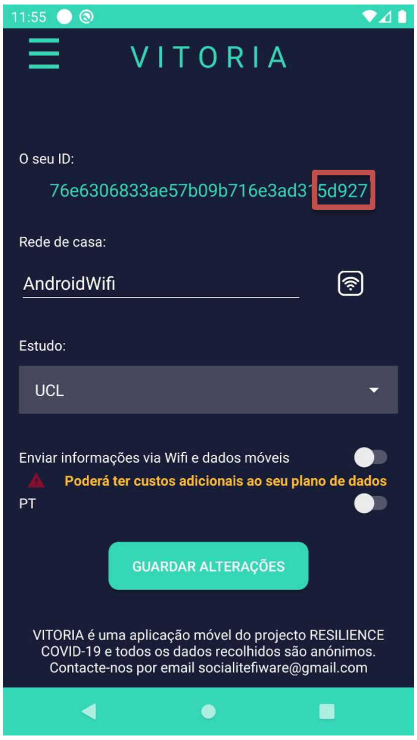


Figura 8 Definições das preferências

8 Formulários

8.1 Formulário de Sono

Todos os dias às 8 horas receberá uma notificação para responder ao formulário da Figura 9. Deve indicar aproximadamente a que horas se deitou, a que horas acordou e classificar a sua qualidade do sono numa escala de 5 entre *"Muito mal"* e *"Muito bem"*.

Figura 9 Formulário de Sono

12

8.2 Formulário de Proximidades

Quando a aplicação deteta que estão muitas pessoas à sua volta, lança uma notificação para responder ao formulário que pode ser visto na Figura 10. Deve indicar o número de pessoas que estão a menos de 2 metros de si. Pode também incluir na resposta a descrição do local ou da ocasião (p.e. *"5 pessoas, num autocarro")*.

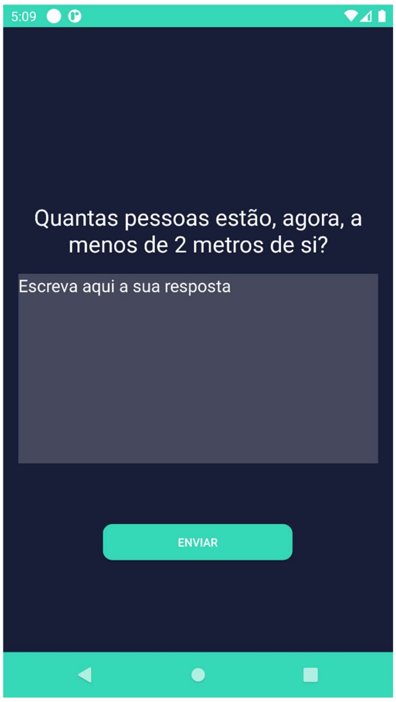


Figura 10 Formulário de proximidade

8.3 Formulário de Finalidades das apps

Todos os dias às 14 horas é lançada uma notificação para avaliar o uso das suas aplicações nas últimas 24 horas. Neste formulário é suposto associar cada uma das aplicações mais utilizadas às finalidades de: comunicação, lazer, pesquisa e trabalho. Tal como pode ver pela Figura 11 pode selecionar mais do que uma opção por aplicação. As aplicações a que já respondeu iram ficar marcadas como *"respondido!"*.

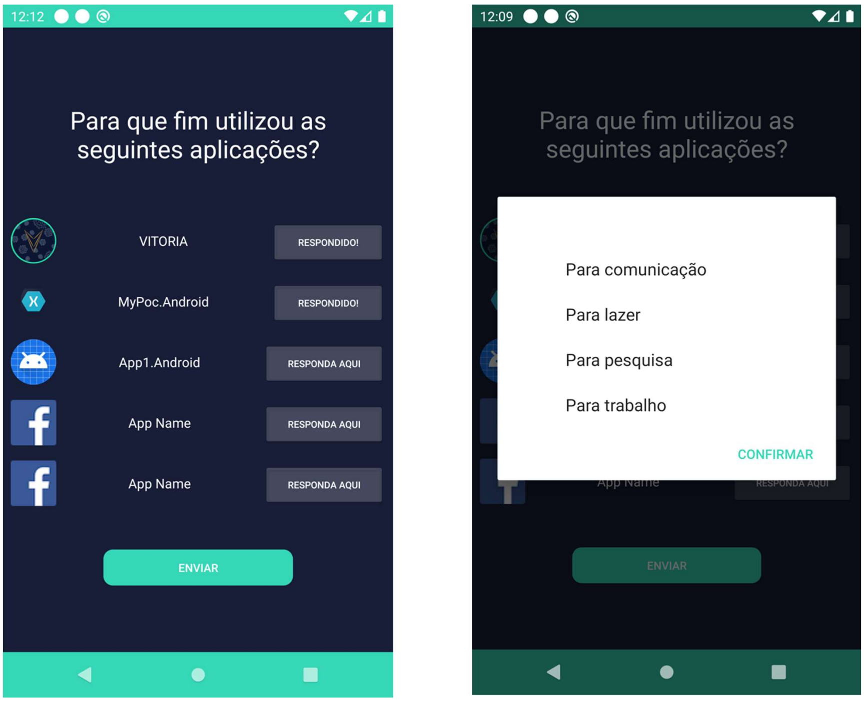


Figura 11 Formulário de finalidade das aplicações mais usadas

8.4 Formulário do Transporte

No caso de aplicação detetar que se encontra em movimento, é lançada uma notificação para responder ao formulário apresentado na Figura 12. Neste formulário deve indicar em que tipo de veículo esteve e quantas pessoas estavam consigo.

Figura 12 Formulário de transporte

15

8.5 Formulário Emocional

Aleatoriamente durante o dia receberá uma notificação para responder ao formulário da Figura 13. Deve escolher as imagens que mais se adequam ao seu estado de espírito no momento em que responde. Este é o único formulário que não poderá ser respondido manualmente.

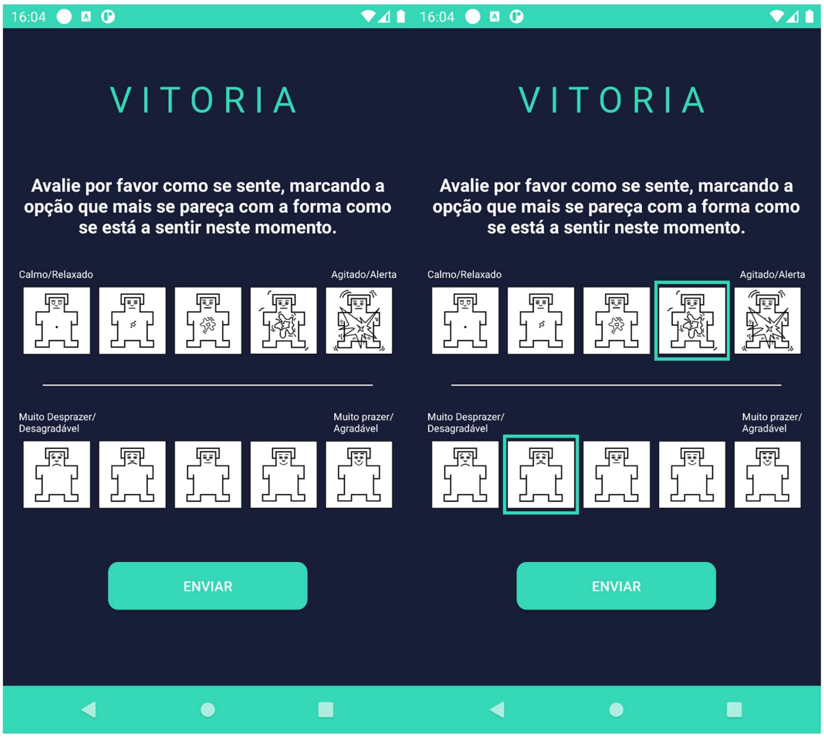


Figura 13 Formulário emocional

16

9 Notificações

A aplicação foi desenvolvida para notificar o utilizador sempre que for preciso responder a um Formulário. Durante o dia aparecerão notificações - tocar nelas permite abrir o Formulário para responder de imediato. Na Figura 14 estão apresentadas as notificações todas juntas.

Assim, todos os dias:

- às 8h é lançada uma notificação para responder ao Formulário do Sono (ver Figura 9);
- às 14h é lançada uma notificação para responder ao Formulário de Finalidade das Aplicações mais utilizadas (ver Figura 11);
- A notificação do Formulário Emocional não tem hora predefinida, se o utilizador não responder pela notificação no momento, terá outra oportunidade durante o resto do dia para responder (ver Figura 13)
- A notificação de Transporte é lançada automaticamente sempre que a aplicação deteta que o utilizador viajou (ver Figura 12);
- A notificação de Proximidade é lançada sempre que a aplicação pensa que existe um número acrescido de pessoas à sua volta (ver Figura 10);
- adicionalmente a aplicação irá ainda notifica-lo, uma vez por dia, se existir algum formulário por responder durante as últimas 24 horas.

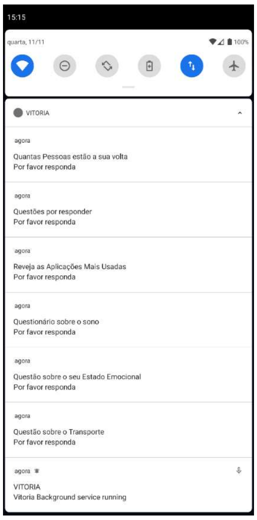


Figura 14 Notificações

17

10 Relógio

Durante a primeira utilização é pedido ao utilizador que permita o acesso aos sensores do relógio.

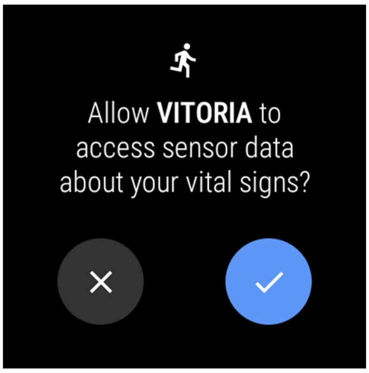


Figura 15 Permissão de sensores

A aplicação para o relógio foi desenvolvida de modo a minimizar a interação com o utilizador, deste modo, a aplicação só possui um ecrã onde o utilizador pode verificar o valor de batimento cardíaco e o número de passos recentemente monitorizados.

Figura 16 Aplicação wear VITORIA

18

This page was intentionally left blank.