



UNIVERSIDADE D  
COIMBRA

António Moisés Tomás Dias

URSO VERDE  
A SMART CITY PLATFORM FOR  
HEALTHY AND SUSTAINABLE LIVING

Dissertation in Master's Degree in Electrical and Computer Engineering,  
supervised by Prof. Dr. Jorge Sá Silva and Prof. Dr. André Rodrigues,  
presented to the Department of Electrical and Computer Engineering of the  
Faculty of Science and Technology of the University of Coimbra.

July 2023







FCTUC FACULDADE DE CIÊNCIAS  
E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA

# URSO VERDE - A Smart City Platform for Healthy and Sustainable Living

António Moisés Tomás Dias

Coimbra, July 2023





# URSO VERDE - A Smart City Platform for Healthy and Sustainable Living

## **Supervisors:**

Prof. Doutor Jorge Miguel Sá Silva

Prof. Doutor André Miguel de Almeida Marrão Rodrigues

## **Jury:**

Prof. Doutor Carlos Alberto Henggeler de Carvalho Antunes

Prof. Doutor Jorge Miguel Sá Silva

Prof. Doutor Paulo José Monteiro Peixoto

Dissertation submitted in partial fulfillment for the degree of Master of Science in  
Electrical and Computer Engineering.

Coimbra, July 2023



# Acknowledgements

I wish to express my deepest gratitude to everyone who have made this journey possible:

Firstly, I would like to extend my heartfelt thanks to my parents and sister for their support throughout this journey. Your encouragement have been my strength and inspiration.

My sincere appreciation goes to Professor Jorge Sá Silva and Professor André Rodrigues, whose supervision and constant availability were instrumental in the realization of this project. Their insights and guidance have made this work not only possible, but also enlightening and rewarding.

I would also like to acknowledge my work colleagues, especially Marcelo Fernandes. His expertise, advice, and willingness to help me learn and explore through various technologies have been invaluable.

My thanks also extend to my course colleagues and friends, Afonso Lemos, Alexandre Pimentel, João Martins, José Pedro Silva, Marco Silva, Pedro Henriques, Rúben Bidault and Tiago Baltazar, whose companionship and camaraderie throughout this academic journey have been enriching and memorable.

To my beloved friends from Lousã, André Pinto, António Silva, Daniel Barbosa, Gabriel Lopes, João Pinto, Paulo Marques, Pedro Gil, Ricardo Serra, Tiago Casaleiro and Tomás Cravo, your faith in me and my abilities has been a constant source of motivation. Your belief that I could accomplish this has made a world of difference.

Lastly, my deepest gratitude is reserved for my girlfriend, Beatriz Castanho. Her determined support and love have been my light during the most challenging moments. She is the person who gave me countless reasons to persevere and never give up.

Thank you all for being a part of this amazing journey.

# Resumo

Com o crescimento exponencial dos dispositivos ligados à Internet das Coisas (IoT), o conceito de Cidades Inteligentes surgiu como uma abordagem transformadora da gestão urbana e da sustentabilidade. No centro desta transformação está a adoção generalizada de *smartphones*, que se tornaram ferramentas indispensáveis na nossa vida quotidiana. As aplicações móveis, em particular, revolucionaram a forma como interagimos com a tecnologia e têm o potencial de induzir mudanças comportamentais positivas. O projeto UrsoVerde é um exemplo de como a IoT e as aplicações móveis podem ser utilizadas para promover práticas sustentáveis em ambientes urbanos. Através da gamificação de actividades sustentáveis, a plataforma incentiva comportamentos como a reciclagem, a utilização de espaços verdes e a deslocação em transportes públicos. O conceito de *Human-in-the-Loop Cyber-Physical Systems* (HITLCPS) está no centro do UrsoVerde, permitindo a integração de dispositivos físicos, plataformas digitais e interação humana. Ao incorporar dispositivos IoT e ao tirar partido da presença generalizada de smartphones, o UrsoVerde cria um ecossistema dinâmico que promove a sustentabilidade e a consciência ambiental. Os utilizadores são motivados a participar em actividades sustentáveis através de um sistema baseado em recompensas, ganhando Pontos de Cidadania que podem ser trocados por vários incentivos, como eventos culturais ou descontos em transportes públicos. O sucesso do projeto UrsoVerde apoia-se na sua capacidade de fazer a ponte entre a tecnologia e o comportamento humano. Ao fornecer uma interface de fácil utilização, experiências personalizadas e feedback em tempo real, a aplicação cultiva hábitos duradouros que contribuem para um estilo de vida sustentável. Além disso, a colaboração do projeto com a Câmara Municipal de Coimbra e a Ordem dos Engenheiros da Região Centro (OERC) exemplifica a importân-

cia das parcerias e do envolvimento das partes interessadas. Em conclusão, o projeto UrsoVerde demonstra o potencial transformador da IoT, das aplicações móveis e do HITLCPS na criação de Cidades Inteligentes sustentáveis. Ao integrar a tecnologia na vida quotidiana dos indivíduos, a aplicação permite que os utilizadores façam escolhas conscientes e participem ativamente na preservação do ambiente.

**Palavras Chave:** HITLCPS; IoT; Aplicações Móveis; Cidades Inteligentes; Sustentabilidade

# Abstract

With the exponential growth of devices connected to the Internet of Things (IoT), the concept of Smart Cities has emerged as a transformative approach to urban management and sustainability. At the heart of this transformation is the widespread adoption of smartphones, which have become indispensable tools in our daily lives. Mobile applications, in particular, have revolutionized the way we interact with technology and have the potential to induce positive behavioural changes. The UrsoVerde project is an example of how IoT and mobile apps can be used to promote sustainable practices in urban environments. Through gamification of sustainable activities, the platform encourages behaviors such as recycling, using green spaces and commuting by public transport. The concept of Human-in-the-Loop Cyber-Physical Systems (HITLCPS) is at the heart of UrsoVerde, enabling the integration of physical devices, digital platforms and human interaction. By incorporating IoT devices and leveraging the widespread presence of smartphones, UrsoVerde creates a dynamic ecosystem that promotes sustainability and environmental awareness. Users are motivated to participate in sustainable activities through a reward-based system, earning Citizenship Points that can be exchanged for various incentives, such as cultural events or discounts on public transportation. The success of the UrsoVerde project rests on its ability to bridge the gap between technology and human behavior. By providing a user-friendly interface, personalized experiences and real-time feedback, the app cultivates lasting habits that contribute to a sustainable lifestyle. Furthermore, the project's collaboration with the Coimbra City Council and the Order of Engineers of the Central Region (OERC) exemplifies the importance of partnerships and stakeholder engagement. In conclusion, the UrsoVerde project demonstrates the transformative potential of IoT, mobile ap-



plications and HITLCPS in creating sustainable Smart Cities. By integrating technology into the daily lives of individuals, the app enables users to make informed choices and actively participate in preserving the environment.

**Keywords:** HITLCPS; IoT; Mobile Applications; SmartCities; Sustainability



*"I used to be and adventurer like you, then I took an arrow in the knee."*

— A random Skyrim guard, 2011



# Contents

<b>Acknowledgements</b>	<b>ii</b>
<b>Resumo</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>List of Acronyms</b>	<b>xiv</b>
<b>List of Figures</b>	<b>xvii</b>
<b>List of Tables</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Context . . . . .	1
1.2 Motivation . . . . .	2
1.3 Application Overview . . . . .	2
1.4 The Team and Partners . . . . .	3
1.5 Methodology and Work Plan . . . . .	3
1.6 Project Risks . . . . .	4
1.7 Contributions . . . . .	5
1.8 Document Structure . . . . .	6
<b>2 State of the Art</b>	<b>7</b>
2.1 Mobile Applications Promoting Sustainable Behavior . . . . .	7
2.2 Reward Systems for Behavior Change . . . . .	8
2.3 Smart Cities . . . . .	9
2.4 The Smartphone in the XXI Century . . . . .	13
2.5 Internet of Things . . . . .	15
2.6 Human-in-the-Loop Cyber-Physical Systems . . . . .	16

<b>3</b>	<b>Technologies</b>	<b>18</b>
3.1	IoT nodes . . . . .	18
3.2	QR Codes . . . . .	19
3.3	Bluetooth Low Energy . . . . .	20
3.4	FIWARE . . . . .	20
3.5	Firebase Cloud Messaging . . . . .	21
3.6	Git - Version Control System . . . . .	23
3.7	Dialogflow . . . . .	23
3.8	Xamarin . . . . .	24
3.9	ASP.NET Blazor . . . . .	25
3.10	Docker . . . . .	25
3.11	SQL . . . . .	25
3.12	Third Party Libraries . . . . .	26
<b>4</b>	<b>UrsoVerde</b>	<b>28</b>
4.1	Objectives . . . . .	28
4.2	Requirements . . . . .	31
4.2.1	Functional Requirements . . . . .	31
4.2.2	Non-functional Requirements . . . . .	39
4.3	Architecture . . . . .	40
4.3.1	Overall architecture . . . . .	40
4.3.2	Mobile application architecture . . . . .	41
<b>5</b>	<b>Urso Verde Development</b>	<b>44</b>
5.1	Partner Meetings . . . . .	44
5.1.1	Internal Weekly Meetings . . . . .	44
5.1.2	Partnership with Coimbra City Hall . . . . .	44
5.2	Cascais CityPoints Meeting . . . . .	47
5.3	Partnership with Escuela Politécnica Nacional . . . . .	48
5.4	Implementation . . . . .	49
5.4.1	Xamarin application development . . . . .	49
5.4.2	Future Internet Ware (FIWARE) Communication . . . . .	51
5.4.3	Login and accout registration . . . . .	53
5.4.4	User data . . . . .	54

5.4.5	Activity tracking . . . . .	55
5.4.6	Step Tracker . . . . .	56
5.4.7	User Objectives . . . . .	57
5.4.8	Points Transfer . . . . .	57
5.4.9	Leaderboard . . . . .	58
5.4.10	Map . . . . .	59
5.4.11	ChatBot . . . . .	59
5.4.12	Notifications . . . . .	60
5.5	User Interfaces . . . . .	61
5.5.1	App Icon and Initial Mockups . . . . .	61
5.5.2	Application Flux . . . . .	62
5.5.3	Login and Registration Page . . . . .	63
5.5.4	Shell . . . . .	64
5.5.5	Activity Page . . . . .	64
5.5.6	Map Page . . . . .	65
5.5.7	Objectives Page . . . . .	66
5.5.8	Quick Response code (QR code) Page . . . . .	67
5.5.9	ChatBot Page . . . . .	68
5.6	Tests . . . . .	68
5.6.1	Devices Used for Testing . . . . .	68
5.6.2	Performance Metrics . . . . .	69
5.6.3	Real Scenarios . . . . .	70
<b>6</b>	<b>Engenheiros Sem Fronteiras</b>	<b>71</b>
6.1	Engenheiros Pelo Mundo . . . . .	71
6.2	Engenheiros Sem Fronteiras . . . . .	73
6.2.1	Objectives . . . . .	73
6.2.2	Requirements and Functionalities . . . . .	73
6.2.3	Similar Applications . . . . .	78
<b>7</b>	<b>Conclusion</b>	<b>79</b>
7.1	Final Results and Future Work . . . . .	79
7.2	Personal Considerations . . . . .	80
<b>8</b>	<b>Bibliography</b>	<b>81</b>

- A UrsoVerde - Guião para demonstração
- B Confluence: An Integration Model for Human-in-the-Loop IoT  
Privacy-Preserving Solutions Towards Sustainability in a Smart City
- C Engenheiros Sem Fronteiras Presentation
- D LinkedIn Post



# List of Acronyms

<b>ATL</b>	Android Transport Layer
<b>API</b>	Application Programming Interface
<b>APNs</b>	Apple Push Notification service
<b>ASP</b>	Active Server Pages
<b>BATINA</b>	Best Assisting Tutor and INteractive Advisor
<b>BLE</b>	Bluetooth Low Energy
<b>BR/EDR</b>	Bluetooth Basic Rate/Enhanced Data Rate
<b>CPSs</b>	Cyber-Physical Systems
<b>CISUC</b>	Center for Informatics and Systems of the University of Coimbra
<b>DEI</b>	Departamento de Engenharia Informática
<b>DEEC</b>	Departamento de Engenharia Eletrotécnica e de Computadores
<b>EPM</b>	Engenheiros Pelo Mundo
<b>ESF</b>	Engenheiros Sem Fronteiras
<b>FCM</b>	Firebase Cloud Messaging
<b>FIWARE</b>	Future Internet Ware
<b>FR</b>	Functional Requirements
<b>GDPR</b>	General Data Protection Regulation
<b>GIS</b>	Green Information Systems
<b>HitL</b>	Human-in-the-loop

<b>HitLCPS</b>	Human-in-the-loop Cyber-Physical Systems
<b>INESC Coimbra</b>	Instituto de Engenharia de Sistemas e Computadores de Coimbra
<b>IBM</b>	International Business Machines Corporation
<b>IDE</b>	Integrated Development Environment
<b>IoT</b>	Internet of Things
<b>ISABELA</b>	IoT Student Advisor and BEst Lifestyle Analyzer
<b>ISCAC</b>	Instituto Superior de Contabilidade e Administração de Coimbra
<b>JSON</b>	JavaScript Object Notation
<b>MB</b>	Megabyte
<b>NFRs</b>	Non-Functional Requirements
<b>.NET MAUI</b>	.NET Multi-platform App UI
<b>OE</b>	Ordem dos Engenheiros
<b>OERC</b>	Ordem dos Engenheiros da Região Centro
<b>PDA</b>	Personal Digital Assistant
<b>PM</b>	Problem Makers
<b>PS</b>	Problem Solvers
<b>QR code</b>	Quick Response code
<b>RAM</b>	Random Access Memory
<b>REST</b>	REpresentational State Transfer
<b>SDK</b>	Software Development Kit
<b>SQL</b>	Structured Query Language
<b>SoC</b>	System-on-Chip
<b>UC</b>	Universidade de Coimbra
<b>UI</b>	User Interface

<b>URL</b>	Uniform Resource Locator
<b>VCS</b>	Version Control System
<b>XAML</b>	Extensible Application Markup Language

# List of Figures

- 1.1 Overview of the main UrsoVerde pages. . . . . 2
- 2.1 Smart City Components. . . . . 10
- 2.2 Growth of Smartphone Users in U.S.. . . . . 14
- 2.3 Smartphone as Portuguese users primary mobile device. . . . . 14
- 2.4 Number of IoT connected devices worldwide from 2019 to 2021, with forecasts  
from 2022 to 2030 (in billions). . . . . 16
- 2.5 The process of human-in-the-loop control. . . . . 17
- 3.1 IoT Node . . . . . 18
- 3.2 Structure of the QR code . . . . . 19
- 3.3 Firebase Architectural Overview . . . . . 22
- 4.1 User’s activities and ChatBot from the first version of the UrsoVerde project 28
- 4.2 Leaderboard and user’s points display from the first version of the UrsoVerde  
Project . . . . . 29
- 4.3 UrsoVerde platform architecture . . . . . 41
- 4.4 UrsoVerde Application Architecture Diagram . . . . . 42
- 5.1 Timeline of meetings and events with the Coimbra City Hall. . . . . 45
- 5.2 Update meeting with the Coimbra City Hall. . . . . 45
- 5.3 UrsoVerde Team at Portugal Smart Cities Summit 2022 . . . . . 46
- 5.4 Meeting with Coimbra City Hall Team. . . . . 47
- 5.5 Overall architecture of a cross-platform Xamarin application. . . . . 49
- 5.6 View (.XAML) and ViewModel (.CS) structure. . . . . 50
- 5.7 Definition of component in Extensible Application Markup Language (XAML)  
and respective output . . . . . 50
- 5.8 Button action defined on the ViewModel. . . . . 51

5.9	Example of a GET request using Postman. . . . .	52
5.10	GET request response in JSON format. . . . .	52
5.11	Generated QR code associated to a recycling bin. . . . .	56
5.12	Push Notification in iOS devices. . . . .	60
5.13	UrsoVerde logo. . . . .	61
5.14	First application Mockups. . . . .	61
5.15	Application UI Flux Diagram. . . . .	62
5.16	Login and Registration Pages . . . . .	63
5.17	Application Shell Tab Bar and Tool Bar . . . . .	64
5.18	Activity Page . . . . .	65
5.19	Map page and pin interaction . . . . .	65
5.20	Objectives and LeaderBoard Page . . . . .	66
5.21	QR code Reader Page . . . . .	67
5.22	ChatBot Page . . . . .	68
5.23	Test metrics on the Android device. . . . .	70
5.24	Battery usage metrics on the Android device. . . . .	70
6.1	Login and Map pages of the EPM application. . . . .	72
6.2	EPM application on the AppStore . . . . .	72

# List of Tables

- 1.1 Software tools used during development . . . . . 4
- 3.1 NuGet Packages used in the project . . . . . 27
- 4.1 Functional Requirements defined for the UrsoVerde platform. . . . . 31
- 4.2 Requirements parameters . . . . . 32
- 5.1 Devices used during testing and their main specifications. . . . . 69
- 6.1 ESF Functional requirements . . . . . 74



# 1 Introduction

## 1.1 Context

The worldwide landscape of Internet of Things (IoT) connected devices is evolving, setting the way for unprecedented growth and integration into our daily lives [1]. The rise of IoT technology gives a unique opportunity to use these developments to improve urban efficiency. With their extensive network of infrastructures and robust financial systems, these metropolitan environments have the potential to profit substantially from IoT adoption. This integration marks an evolutionary step forward in the way we manage our cities, transforming them into "Smart Cities [2]. Smart Cities are a new approach to urban management, infrastructure and services that reflect the constant spirit of innovation that has become a distinguishing feature of modern societies [3]. Smartphones have taken center stage at the center of this technological revolution, drastically influencing our everyday lives. These gadgets, which are packed with tremendous computational capabilities, enable an astounding array of new applications that have radically changed the way we live [4]. The influence of smartphones extends far beyond mere communication and convenience. Mobile applications, in particular, have started playing a critical role in promoting sustainable habits, influencing both individuals and organizations. They serve as platforms for environmental consciousness, driving behavioral changes that contribute to a more sustainable future [5]. This impact is amplified by carefully designed motivation and reward systems embedded within these applications, further encouraging users to adopt more sustainable practices [6]. The notion of Human-in-the-loop Cyber-Physical Systems (HitLCPS) encapsulates this unique relationship of humans, digital technology and the real environment. HitLCPS are an important evolutionary leap in today's technology world, highlighting the integration of digital and physical parts of our environments. They represent a transformation in how we engage with and affect our surroundings, laying the way for a future in which technology and people living together and interact in a seamless linked environment [7].



## 1.2 Motivation

The direction and theme of this project were inspired by the current digital transformation we are witnessing. A transformation that is not only reshaping the global technology landscape but also how we live, work and interact with the world around us. This rapid evolution has increased the significance of domains like mobile application development, IoT and HitLCPS. These areas hold potential for creating advanced, smart solutions that can contribute to the sustainability of our cities. It was in this context that I was inspired to dive into the complexities of these technologies and use them in the construction of a sustainable smart city solution for my Master's thesis. The goal was to provide a solution that may stimulate a change in habits and behaviors toward more sustainable practices, so contributing to the development of a more resilient and sustainable urban environment. We were able to define clear and practical project objectives through a collaboration with the Coimbra City Hall. These goals not only served as a foundation for the project's development, but they also played an important part of enhancing my motivation.

## 1.3 Application Overview

UrsoVerde, shown in figure 1.1, includes a sustainability-focused mobile application, compatible with both iOS and Android devices. It incentivizes environmentally friendly behaviors such as recycling, visiting green zones, using public transportation and walking. To aid these activities, the app includes a detailed map indicating locations of recycling bins, public transportation stops and green zones. Users validate their activities by scanning QR codes or interacting with IoT devices, earning 'Citizen Points' by completing weekly objectives. These points can be redeemed for rewards, such as tickets to municipal events or discounts on public transportation.



Figure 1.1: Overview of the main UrsoVerde pages.

## 1.4 The Team and Partners

The UrsoVerde platform was developed collaboratively by a team of student and professors from the Instituto de Engenharia de Sistemas e Computadores de Coimbra (INESC Coimbra) and Center for Informatics and Systems of the University of Coimbra (CISUC). Óscar Torres, a doctoral student in Sciences and Technologies of Information at the Department of Informatics Engineering (DEI), gave support in programming the IoT devices. The SOCIALITE Sensors library, ChatBot server, FIWARE back-end and Git were managed by Marcelo Fernandes, a former doctoral student in Sciences and Technologies of Information at DEI. Addressing data security and privacy, Jorge Rivadeneira, a doctoral student in Sciences and Technologies of Information at Departamento de Engenharia Informática (DEI), is currently developing advanced mechanisms that manage these challenges, using Artificial Intelligence and Blockchain technologies. As a Master's student in Electrical and Computer Engineering, I, António Moisés Dias, developed the application. The project was guided by Prof. Dr. Jorge Sá Silva, from the Department of Electrical and Computer Engineering (DEEC) and researcher of INESC Coimbra, and Prof. Dr. André Rodrigues, from Instituto Superior de Contabilidade e Administração de Coimbra (ISCAC) and researcher of CISUC.

## 1.5 Methodology and Work Plan

During the semester we used approaches consistent with the Agile methodology in the planning and execution of this project, with a particular emphasis on the concept of "working software over comprehensive documentation" [8]. One of these methods was "code reuse" [9], which includes reusing existing code from previous done projects and adapting it for a new purpose. In our case, we made use of the code from two comparable applications, IoT Student Advisor and BEst Lifestyle Analyzer (ISABELA) and Best Assisting Tutor and INteractive Advisor (BATINA), to build the UrsoVerde application. This method not only accelerated the development process, but it also improved its efficiency and success. We were able to construct a functioning and efficient application in a simplified manner by reusing and modifying code in line with the specific requirements of UrsoVerde.

We conducted weekly internal meetings that were attended by the supervisors and occasionally by other team members. In these sessions, we reviewed the tasks and objectives accomplished during the previous week and planned new tasks for the upcoming week.

Throughout the course of the project, a range of tools and technologies were essential in

supporting our work processes and communication, as shown in table 1.1.

Table 1.1: Software tools used during development

Tool	Purpose
Skype	Remote meetings
Trello	Managing and tracking implementation of features
JetBrains Rider	Integrated Development Environment (IDE) used
Android Studio	Running the Android device emulator
Xcode	Running the iOS device emulator
Postman	Testing HTTP requests to the FIWARE backend
Slack	Main team communication platform
Adobe Illustrator	Creating app icons and mockups
Visual Studio for Mac 2022	Providing tools necessary for Xamarin App Development

## 1.6 Project Risks

The development of this project did not come without potential risks that needed to be taken into consideration. These risks were:

- **Compatibility Errors:** Smartphone firmware versions might update or change during the development phase of the application. These updates could potentially lead to compatibility issues. This risk is usually detected when development time is unexpectedly spent on system updates and management of development software.
- **Server Problems:** The project is heavily reliant on a server to store and collect data, which is a critical aspect of its functioning. There's a potential risk that the server may encounter issues.
- **Damage or Theft of IoT Devices:** The IoT devices used in the project are supposed to be set on public zones, making them vulnerable to damage or theft.
- **QR Code Damage or Theft:** Similarly, the QR codes used for validating user activities could be damaged or stolen.
- **Exploitation of Citizen Points Mechanism:** There's a risk that users could find ways to cheat the system and earn Citizen Points dishonestly.

Mitigation strategies to prevent this issues are:

- **Server Problems:** To mitigate this risk, you could have a backup server in place. Regular data backups should be scheduled and the server's health status should be monitored continuously.
- **Compatibility Errors:** Regular testing on different versions of firmware can help identify any compatibility issues early.
- **Damage or Theft of IoT Devices:** Regular maintenance and inspection of the devices can help identify and rectify any damages early. Additionally, we could consider implementing security measures to prevent theft, such as securely mounting the devices in public areas using strong enclosures.
- **QR Code Damage or Theft:** Printing QR codes on durable materials can help reduce the risk of damage. Also, they can be displayed on a screen on a public transport, discarding the need to print these codes.
- **Exploitation of Citizen Points Mechanism:** Regular audits of the points system and monitoring for unusual user behavior can help identify potential abuse early. Also, we build the system to award only one point per challenge completed.

## 1.7 Contributions

The contributions of this thesis are as follows:

- Provides a review of literature related to smart cities, IoT, mobile application development and HitLCPS, serving as a valuable resource for other researchers interested in these areas.
- Presents a comprehensive framework for designing and developing mobile applications aimed at sustainable practices within smart cities, offering a blueprint for the key features and functionalities that such applications should possess.
- The prototype developed from this project was used by the Coimbra City Hall as part of their application for the "Bairros Digitais Comerciais" initiative [10].
- An under revision paper, titled "Confluence: An Integration Model for Human-in-the-Loop IoT Privacy-Preserving Solutions Towards Sustainability in a Smart City", to

which I have significantly contributed as a co-author, contains a section (appendix B) dedicated to the UrsoVerde platform.

## 1.8 Document Structure

In this document we begin with an "Introduction" chapter that provides context, motivations, methodology and contributions of this project. Then, we explore into the "State of the Art" chapter, discussing the conceptual foundations of the study. The third chapter, "Technologies and Protocols", details the fundamental tools that were used in this research. The "UrsoVerde" chapter describes the application's goals, features and architecture. In the "Development" chapter we detail the development process, how features were implemented and how the User Interface (UI) was designed. Also, in the "Engenheiros Sem Fronteiras" chapter we describe two parallel projects that are being developed simultaneously, these projects will be combined into a bigger project that expands the capabilities of the UrsoVerde platform, in which users receive Citizen Points in exchange for performing volunteer engineering work. The last chapter, "Conclusions", presents the conclusions and future work of this project.

## 2 State of the Art

In the following section, we will discuss some of the conceptualizations that served as the base to develop this project as well as other applications and initiatives with related goals. The core subjects of our research span multiple important areas: the societal benefits of sustainable behavior, the use of mobile applications to promote sustainable behavior, the influence of reward systems on behavior change, the impact of smart city technologies in driving and reinforcing these sustainable behaviors, the importance of smartphones in the XXI century, IoT and HitLCPS. Each of these themes is highly complex and could make up an extensive chapter. Thus, in each section, it was decided to present a short summary to frame the present thesis.

### 2.1 Mobile Applications Promoting Sustainable Behavior

The importance of sustainable behaviors is made evident by the complex connection between human behavior and the overall condition of our environment. In their 2009 article [11], "Encouraging pro-environmental behavior: An integrative review and research agenda", Steg and Velek offer an in-depth review of the value and potential of environmental psychology in comprehending and promoting pro-environmental behavior that need to be evolved, an in-depth examination of the main variables influencing this behavior, the creation and implementation of initiatives to change behavior and the evaluation of the results of these interventions. This strategy provides a planned and methodical way to encourage sustainable behaviour, significantly advancing environmental preservation. The study reveals how sustainable practices have larger societal effects, such as improved community health, biodiversity enhancement and financial saving on waste management, in addition to environmental benefits. As a result, the importance of sustainable acts for society's well-being cannot be overstated, which is why mobile applications that support these actions have recently been

developed [11].

A variety of studies have focused on the role that mobile applications might engage in promoting sustainable lifestyles. The authors of the paper "Green by app: The contribution of mobile applications to environmental sustainability" explore how mobile applications could be beneficial to promote environmental conscious activities. The study analyzes appropriate contexts for private utilization of sustainable mobile apps by conducting an extensive analysis of apps accessible on the Google Play Store [12]. Additionally, it clarifies the connections between user objectives and applications while examining the motivational and acceptance-related elements that influence user engagement. The article also assesses the performance of current Green Information Systems (GIS) roles in the business domain, providing useful insights for upcoming mobile applications intended to promote sustainable actions [13].

The research, titled "Exploring the Use of Mobile Apps for Fostering Sustainability-oriented Corporate Culture: A Qualitative Analysis" shifts the focus to the commercial field, emphasizing the critical role of IS in fostering corporate sustainability and pro-environmental behavior. The researchers investigate the potential of mobile apps to establish an industry culture focused on sustainability. The study presents a framework for designing, executing and monitoring targeted app usage through a qualitative research methodology incorporating an examination of four corporate apps created to encourage sustainable behavior. The paper goes on to address app meta-requirements, design principles and implementation standards, widening the scope of this research [5].

In conclusion, these studies highlight the crucial significance of mobile applications in promoting sustainable habits on both individual and organizational levels. They do, however, emphasise the need for additional study, robust evaluation procedures and effective use of persuasive technologies to ensure that these applications can impact meaningful and long-term behavioral changes.

## 2.2 Reward Systems for Behavior Change

Motivation and reward methods are critical in the field of behavioral changes. An investigation titled "Motivation and reward mechanisms in health behavior change processes" explores the complexities of these mechanisms, emphasizing the role of non-cognitive or implicit aspects of behavior, such as emotions and positive results. Their research underlines that being aware of these mechanisms and reward-motivation systems is critical to the efficacy of lifestyle modification procedures, particularly those aimed at promoting environmentally

conscious habits. These findings highlight the critical role that motivation and reward processes can play in guiding behavioral change, providing valuable insights and direction for future programs targeted at enhancing sustainability through behavior modification [6].

Notably, incentives often result in better results than penalties. While punishment encourages obedience and the cessation of unwanted conduct, it also creates negative connections with painful memories and fear. Rewards, on the other hand, promote learning and internalization, leading to long-lasting improvements [14, 15].

## 2.3 Smart Cities

Smart Cities represent a revolutionary approach to urban management, services and infrastructure, reflecting the spirit of innovation that characterizes modern society [3]. At their core, Smart Cities rely mostly on infrastructures, the foundational elements that bring the city together. However, these infrastructures do not exist in isolation, they are embedded and driven by technology, the critical enabler that allows the city to move from a traditional to a smart model [16].

A smart city infrastructure is a complex system that integrates four important components: data collection, data transmission, data storage and data analysis. The first step in this digital ecosystem is data collecting, which is supported by numerous applications, resulting in the growth of sensor technology across several sectors. These sensors are critical for gathering useful and timely data from the city's environment. Following data collection, the information is transmitted. This process is made possible by a variety of means, including city-wide Wi-Fi networks, developments in 4G and 5G technology and local networks capable of supporting both local and worldwide data transmission. The third consideration is data storage in cloud-based systems. These systems apply a variety of organizing methods to arrange and make data compliant for the following phase, data analysis. This analysis extracts actionable patterns and conclusions from the collected data to guide informed decision-making. The depth of analysis can range from simple decision-making and data aggregation to more advanced processes that use statistical approaches, machine learning and deep learning algorithms in real time, made possible by the cloud's processing capabilities. As a result, a smart city effectively acts as a vast and integrated data flow network, opening the way for efficient and intelligent urban administration [17, 18].



A smart city is made up of several components which are illustrated in 2.1.

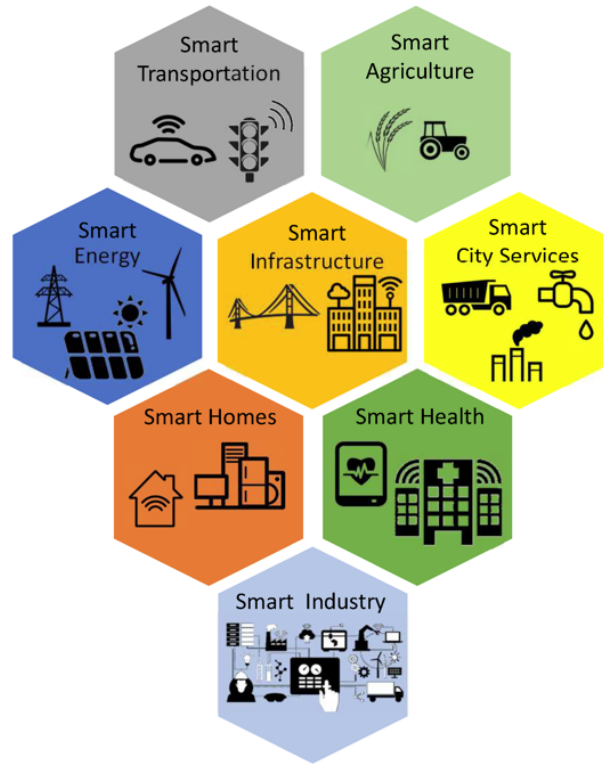


Figure 2.1: Smart City Components. From: SYED, Abbas et al.: IoT in Smart Cities: A Survey of Technologies, Practices and Challenges, Smart Cities, 2021, 4, 429-475. [17].

- Smart Transportation: Discusses how IoT has increased communication between cars, infrastructure and people, resulting in better traffic control and public transportation scheduling. The information gathered by these technologies is also used in mapping and navigation apps [17, 19].
- Smart Agriculture: Crop monitoring and disease detection may be considerably improved with the use of embedded sensors and precision agriculture, consequently boosting agricultural decision-making processes [17, 20, 21].
- Smart Energy: Discusses the transition of traditional power generating systems to smart grids. These grids give real-time data on power usage and integrate various energy sources, resulting in improved power management and continuous supply [17, 22].
- Smart Infrastructure: Explores the application of IoT in structural health monitoring, which enables predictive maintenance of city infrastructure such as bridges, roads and buildings, maintaining their optimal performance [17, 23, 24].

- Smart City Services: Highlighting IoT's role in enhancing city services such as water delivery, waste management and environmental control. Sensor-equipped waste bins and water quality monitoring systems are being considered to be major factors to making cities smarter [17, 25, 26, 27].
- Smart Homes: Using numerous sensors installed in residences, significant data may be obtained to improve quality of life and overall efficiency. These sensors might include environmental sensors, motion trackers and power/energy usage monitors [17].
- Smart Health: Dedicated to the use of technology to enhance healthcare services. Telemedicine and AI-assisted diagnostics make healthcare more accessible, lowering the demand on healthcare institutions [17, 28, 29].
- Smart Industry: Discusses the significance of IoT and AI in transforming businesses, streamlining production processes and improving worker safety [30, 31, 32].

The concept of a smart city originally emerged in 1993, when Singapore declared itself to be a "intelligent city" [33]. The notion of a "digital city" originated between 2000 and 2010, it was closely tied to the concept of a smart city. However, it wasn't until 2010 that interest in smart cities began to increase and the first scenarios of true smart cities started to appear [34]. As we explore the field of technology-enhanced urban development, it is important to explore practical implementations. In the following paragraphs we will look at examples of urban areas from across the world that have adopted smart city solutions.

- Santander: The SmartSantander project in Santander, Spain, built a large IoT infrastructure for research and experimentation [35]. This includes sensors and devices spread across the city, managed using the FIWARE platform [36]. The project allows for real-time access to city-wide data for a variety of applications including environmental monitoring, parking management, traffic intensity monitoring, efficient irrigation, waste management and participatory sensing [34, 37, 38].
- Singapore: The Smart Nation and Digital Governance Group in Singapore is in charge of a variety of smart city projects [39]. The government has established six smart city visions focused on urban transportation, sensor platforms, digital identification, life moments, e-payments and operations development. The city has adopted multiple smart city services, including public transportation enhancement, health record management and life event management, through the use of a policy infrastructure to enable an innovative smart city ecosystem [34].

- Atlanta: Atlanta’s smart city strategy prioritizes transportation, public safety and sustainability. The North Avenue Smart Corridor [40] for mobility and safety, SoundThinking (formerly ShotSpotter) [41] for gunshot detection, the ATL311 [42] and NotifyATL [43] applications for citizen participation are all part of the city’s data-centric model [34]. These projects are supported by the CityIQ platform [44] .
- Amsterdam: Amsterdam’s smart city strategy is a bottom-up, public-private cooperation focusing on citizen-centric solutions. Open data sharing, MyNeighbour App for community engagement, Wyzer App for crowd control, river plastic collection, social media data analysis and an IoT Living Lab are among the projects developed by the city [34, 45, 46].
- Cascais: Cascais CityPoints is a Portuguese smartphone application that promotes good citizenship and rewards residents who actively contribute to local sustainability. The software allows users to earn points by doing predetermined tasks and then redeem those points for product coupons or services from local partners. Recycling, engaging in local cultural activities, volunteering and taking public transportation are examples of actions that earn points in the CityPoints program. The application includes a map with activities for collecting points. Users search for QR codes related to civic activity across the city and points are awarded for each QR code gathered [47]. Cascais is the first urban area to use this application, but it is intended that other municipalities join the system, which will allow to leverage synergies and expand the network of opportunities for citizens [48].

The numerous scenarios shows that there is no one-size-fits-all solution for building a smart city. Instead, effective changes rely on the interaction between technological resources and the unique urban context, which includes citizens needs and the vision of the city’s management. These cities offer not just a picture of current technological achievements, but also a view into the future of urban living. As smart city technologies advance, the potential for improving sustainability, efficiency and human well-being remains enormous. Smart cities are ushering in a more connected, sustainable and resilient future, welcoming in an exciting period for urban growth.

## 2.4 The Smartphone in the XXI Century

Smartphones stand as a testament to the marvels of modern technology, having an enormous effect on our daily lives. These powerful computing devices support an extensive number of new apps that have altered the way we live [4]. According to Agger, smartphones have the potential to "change everything". These technologies have already disrupted our traditional modes of communication. Where previously we relied on landline phones, smartphones have joined the picture, delivering profound changes in only a few years [49].

This era of change began in 1909, with Nikola Tesla's vision of an integrated communication and computer device [50]. Since then, a series of technical advances has culminated in today's smartphones, the enormity of which cannot be overstated [51]. Smartphone use has risen over the past years, changing the social opportunities and constraints that technology offers to our lives [52, 53]. In 1993, International Business Machines Corporation's (IBM) Simon launched the smartphone revolution as the first device that could be called a "smartphone" [54]. However it is important to note that when it was released, it was described as a Personal Digital Assistant (PDA) with communication capabilities via a cellular connection. It was the result of IBM's desire to design a "computer that could fit in your shirt pocket" [55]. The launch of the BlackBerry [56] smartphone, which had web browsing, a camera and Internet connectivity, propelled the movement even more. Apple [57] made a huge market debut in 2007 with the iPhone [58], which was quickly followed by the introduction of Google's Android operating system by the end of the same year [59]. Notably, the first versions of smartphones were mostly used as business tools and were too expensive for the general public due to the slow and expensive technology that were available at the time. However, as technology has advanced and become more accessible, smartphone use has expanded to a broader audience [52].

It comes as no surprise that smartphone usage is constantly increasing in a time when they are so embedded in our lives. Figure 2.2 shows that smartphone adoption in the United States is rising in the last eight years and reached 81.6% in 2023, implying that around 270 million Americans own a smartphone. Globally, there are over 6.92 billion smartphone users, accounting for approximately 86.29% of the global population [60].

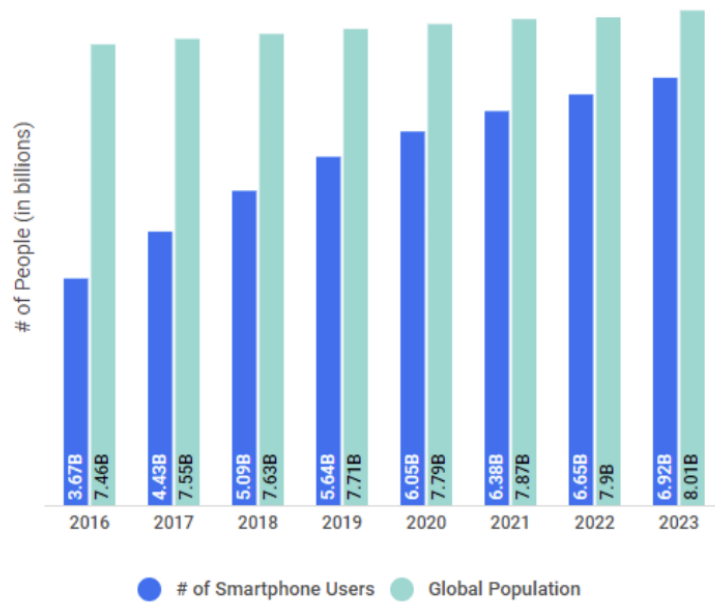


Figure 2.2: Growth of Smartphone Users in U.S.. From: FLYNN, Jack: 20 Vital Smartphone usage statistics: Facts, data and trends on mobile use in the U.S., 2023 [60].

Moving across the Atlantic to Portugal, the Target Group Index (TGI) [61] research by Marktest [62] found that 7.2 million individuals, or 84.2% of citizens aged 15 and above, reported using smartphones as their primary mobile device. This percentage has risen from 27.0% in 2012, demonstrating that smartphone usage in Portugal has more than tripled over between 2012 and 2020, as shown in 2.3 [63].

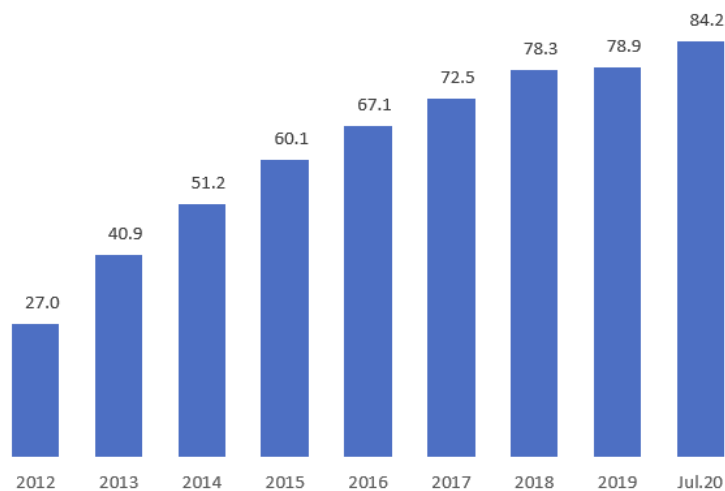


Figure 2.3: Smartphone as Portuguese users primary mobile device. From: Marktest, TGI , 2020[63].

Smartphone users are rising as the smartphone plays an important role on our daily basis [64]. The introduction of this devices altered people’s access to information. Aside from call-

ing and texting, people use their smartphones for a number of activities, including streaming music, online shopping and banking, watching movies, playing games and exchanging images [65].

## 2.5 Internet of Things

The expression "IoT" refers to a vision in which common objects, devices and sensors go beyond their innate roles to include network connectivity and computing power [66]. While there is no commonly recognized definition, the essential premise is that these objects can produce, share and consume data with little to no human interaction. Although the concept of combining computers, sensors and networks to control objects has been around for some time, the mainstream adoption of IoT has been accelerated by a number of technical advances. These include universal connection, widespread use of IP-base networking, device downsizing and the proliferation of cloud computing [66].

The main goal of the IoT is to provide a network architecture that includes compatible communication protocols and applications. This degree of association allows for the seamless integration and communication of numerous physical and virtual sensors, personal computers, smart devices and everyday items at any time and over any network [67]. This integration and interoperability demonstrate the IoT's transformation potential, enabling new levels of automation and data interchange across a wide range of application [66, 67].

The worldwide landscape of IoT-connected devices is set to grow significantly. According to Lionel Vailshery's research, as shown in figure 2.4, the number of IoT devices is expect to nearly triple, rising from 9.7 billion in 2020 to an estimated 29 billion by 2030 [1].

Applying IoT technology to improve the efficiency of cities, with their complex network of infrastructures, sociocultural and financial systems, can be an evolutionary step. Current city management is frequently challenged by bureaucratic and communication inefficiencies. However, IoT adoption can overcome these gaps by seamlessly merging the digital and physical worlds, resulting in smart, linked ecosystems. As a result, city services may be more successfully regulated, transforming cities into innovative arenas for both implementation and research of IoT technologies. Ultimately, IoT has the potential to revolutionize urban management, leading to smarter and more efficient cities [2].

In conclusion, despite its early stages, IoT is set to promote new solutions across a wide range of sectors. According to future forecasts, intelligent IoT-based applications will give rise to smarter homes, offices, transit systems, hospitals and corporations. These advanced

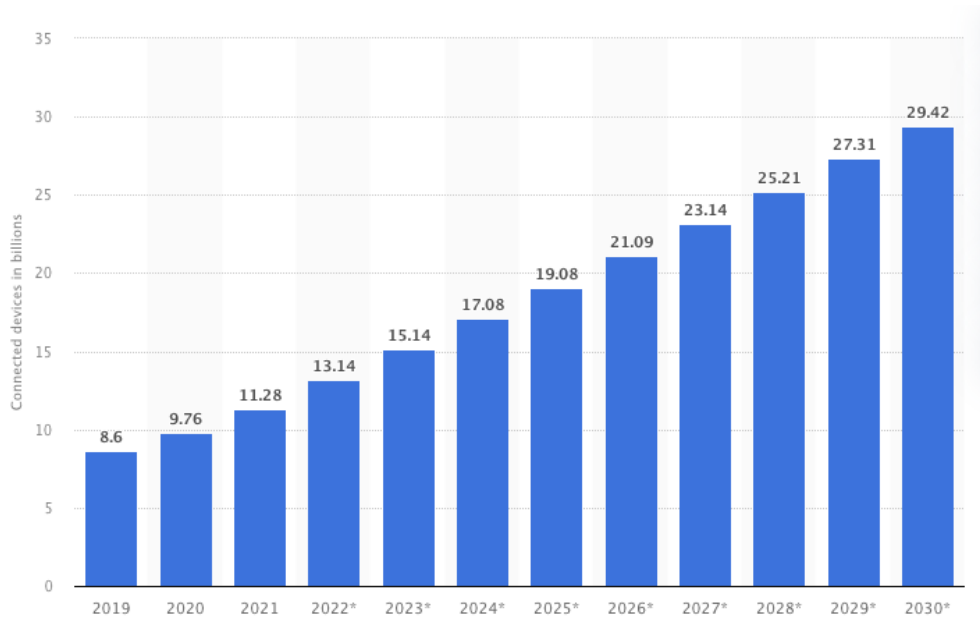


Figure 2.4: Number of IoT connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030 (in billions) From: VAILSHERY, Lionel, Number of internet of things (iot) connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030 [1].

approaches are designed to improve efficiency, increase convenience and provide higher degrees of automation, pushing us further into the age of digital transformation. As we move forward, the integration of IoT with other developing technologies will push the boundaries of what is possible, sending us on a path towards a more connected future [68].

## 2.6 Human-in-the-Loop Cyber-Physical Systems

Cyber-Physical Systems (CPSs) represent the evolution of our current technological landscape's convergence between digital and physical aspects. This notion originates from the vision of a IoT in which common items "tagged" with radio-frequency identification might be read, identified and managed by computers [69]. CPSs developed from this basis. CPSs combine elements of robotics, wireless sensor networks, mobile computing and IoT to create highly monitored, easily controlled and adaptable environments [7].

When these devices equipped with sensors begin to monitor and analyze human behavior, individuals become integrated into the loop of these CPSs. HitLCPS therefore constitute CPSs that actively incorporate human interaction into their design and operation, resulting in a symbiotic relationship between the system's technology and human aspects [7]. This was one of the main concepts that drove this thesis, in which the group has been working

on.

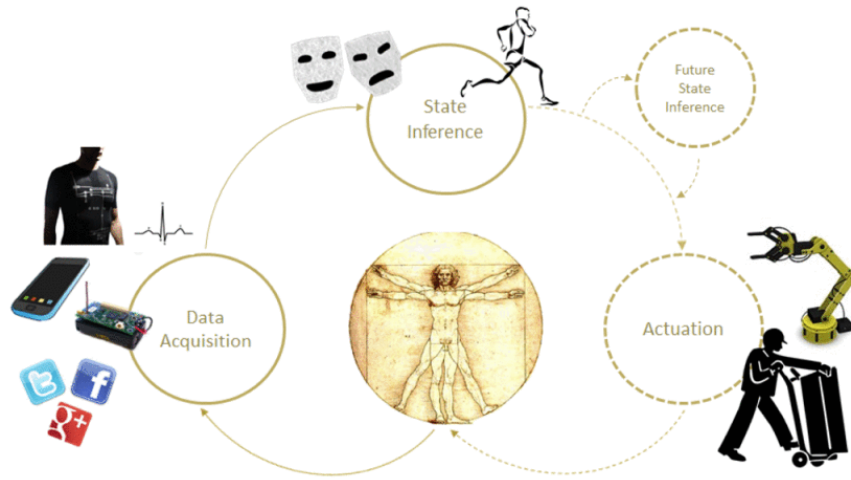


Figure 2.5: The process of human-in-the-loop control. From: NUNES, David Sousa et al.: A survey on human-in-the-loop applications towards an internet of all, *IEEE Communications Surveys Tutorials*, 2015, 944-965 [7].

Figure 2.5 presents the various processes associated with Human-in-the-loop (HitL). Initially, sensor data related to a human individual is acquired. This information is analyzed to determine the person's status, their physical and/or mental health. Some systems may also forecast future states based on present and historical data. The system may perform certain actions based on identified situations. "Open-loop" systems have no effect on the system as a whole, their findings are merely informative, with no direct actuation, "closed-loop" systems work directly on the environment or the human to impact the loop and achieve a desired state [7].

The combination of HitL approaches with future iterations of CPSs and IoT signifies an advancement in the field. This convergence presents the possibility to develop highly responsive, configurable ecosystems in which people, computers and robotics coexist together [7].



# 3 Technologies

The third chapter, "Technologies and Protocols" provides a comprehensive exploration of the state-of-the-art technologies and protocols that serve as foundation for the current research, each chosen for its individual capabilities and potential contributions.

## 3.1 IoT nodes

As illustrated in figure 3.1, Pycom FiPy serves as our project's IoT nodes. This device is characterized by its versatility and adaptability, equipped with a micro-controller that can be programmed using Micropython [70]. The diversity of its connectivity options which includes LoRaWAN, SigFox, Wi-Fi, Bluetooth and LTE, contributes to the development of heterogeneous systems with diverse communication capabilities [71].



Figure 3.1: IoT Node

Internally, Pycom devices are powered by the ESP32 System-on-Chip (SoC), a design choice that prioritizes ultra-low power consumption. This SoC is equipped with 4 Megabyte (MB) of Random Access Memory (RAM), 8MB Flash Memory and a variety of peripherals ideal for IoT applications [72]. Pycom's firmware offers access to a collection of micro controller features including interrupts, timers, analog-to-digital converters, digital-to-analog

converters and general-purpose input/output pins to further increase their usefulness. These capabilities boost Pycom devices adaptability, making them a solid alternative for a variety of IoT solutions [71].

## 3.2 QR Codes

A QR code is a form of matrix bar code or two-dimensional code developed by Denso Corporation, Japan in 1994. The QR code has become a standard as its architecture enables the contents of the code to be decoded at high speed, earning the term "Quick Response" [73].

QR codes, which are made up of black modules arranged in a square pattern on a white background, are designed to store a range of data types, including text, Uniform Resource Locator (URL) and other sorts of data. The QR code's simplicity masks its capacity for large amounts of data storage, with the maximum version of the QR code capable of storing up to 7089 numeric data points or 4296 alphanumeric data points [73, 74].

The structure of the QR code comprises an encoding region and function patterns, as shown in figure 3.2. Function patterns, which include finder, separator, timing and alignment patterns, are critical in setting the code's location, size and inclination. The finder patterns, which are situated at three corners of the symbol, are specifically designed to assist in quickly detecting the QR code [73].

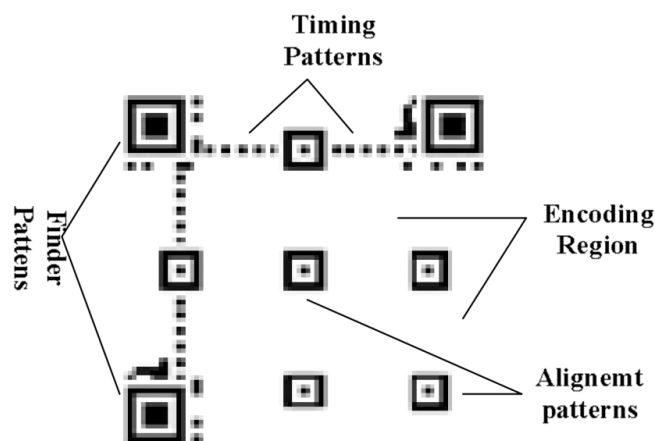


Figure 3.2: Structure of the QR code [73]

QR codes have grown in popularity due to their adaptability and the widespread usage of smartphones with built-in cameras. They are used in a wide range of applications, from

production to logistics and sales [73, 74].

### 3.3 Bluetooth Low Energy

Bluetooth Low Energy (BLE) emerges as a pivotal technology in the wireless communications landscape, due to its energy efficiency, versatility and extensive feature set [75]. Bluetooth Classic, formerly referred to as Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR), is a wireless audio streaming protocol that primarily supports point-to-point device connections [76]. BLE, on the other hand, extends connection capabilities.

BLE, which is designed for low-power operation, is an ideal alternative for devices with small low capacity batteries. This feature enables the creation of compact, power-efficient gadgets that can operate for years without the need for battery replacement. Furthermore, BLE's low power consumption does not limit its capacity to support a wide range of communication topologies. BLE may support mesh networks in addition to point-to-point and broadcast communication, allowing the establishment of large, dependable device networks [76].

A vital component of BLE functionality is its Advertising feature. This element serves as the foundation for beacon technology, in which tiny devices (called beacons) broadcast information to their surroundings, on a periodic basis. A network of beacons, when combined with a suitable smartphone app, may provide a variety of services. These include indoor localization, proximity-triggered advertising and mobile payment systems, highlighting BLE potential in modern wireless communications [77].

### 3.4 FIWARE

FIWARE is an advanced European collection of specifications designed to provide a Application Programming Interface (API) that ease the creation of smart applications across multiple vertical sectors. It stands as an open, public and royalty-free set of tools for innovative developers and businesses [36, 78].

The FIWARE platform relies on the FIWARE Context Broker and a set of open-source Generic Enablers that provide extensive functionality for a wide range of applications [79]:

- Core Context Management: This tool suite stores and manipulates context data in order to prepare it for subsequent processing and applications. It serves as the foundation for the FIWARE ecosystem's handling of context data.

- **Interfacing with IoT, Robots and Third-Party Systems:** These Generic Enablers are intended to collect context information updates from a number of sources, including IoT devices, robots and other systems. They also translate actuation needs, making interactions between different components simpler.
- **Context Information Processing, Analysis and Visualization:** This set of tools implements the smart behavior expected of applications, analyzing and visualizing context data in a useful format. They assist end-users in making informed decisions, leveraging the power of smart applications.
- **Context Data/API Management, Security, Publication and Monetization:** This aspect of FIWARE deals with managing access to and control over context data. It includes tools for secure data handling, publishing context data and providing opportunities for monetization. It ensures that context data can be securely and effectively utilized.
- **Deployment Tools:** Docker images and Helm-chart recipes are provided for each Generic Enabler to aid in the deployment of FIWARE applications.

Each of these components adds to the FIWARE platform's robustness, ensuring that it can serve as a stable foundation for developing smart applications across numerous sectors[36, 79].

## 3.5 Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) is a flexible, cross-platform messaging framework that enables reliable and cost-free transmission of messages. FCM's major role is to enable client app notifications, such as alerting the app about new email or data ready for synchronization, which are critical for user re-engagement and retention [80].

FCM allows to send two types of messages: notification messages and data messages. The FCM Software Development Kit (SDK) handles notification messages, also known as "display messages" automatically. They have a defined set of user-visible keys and an optional data payload. Data messages, on the other hand, include only user-defined custom key-value pairs and are controlled by the client app [81].

The efficacy of FCM is dependent on a number of components functioning together to build, transmit and receive messages [82]. As shown in figure 3.3, these elements are as follows:

1. Message Building Tools: These are used to create message requests. A tool like the Notifications composer provides a user-friendly interface for this task. For more complex needs, these requests can be created in an environment that supports the Firebase Admin SDK or the FCM server protocol, such as Cloud Functions for Firebase, App Engine, or a custom app server.
2. FCM Back-end: This is the system responsible for accepting message requests, distributing messages and generating relevant information such as the message ID.
3. Platform-Level Transport Layer: This is the mechanism responsible for delivering the message to the intended device and applying any necessary adjustments. It is made up of several layers for various devices:
  - Android Transport Layer (ATL) for Android devices.
  - Apple Push Notification service (APNs) for Apple devices.
  - Web push protocol for web apps.
4. FCM SDK on User Device: This is the FCM component on the user's device that displays or manages the notification, depending on the app's state and rules [82].

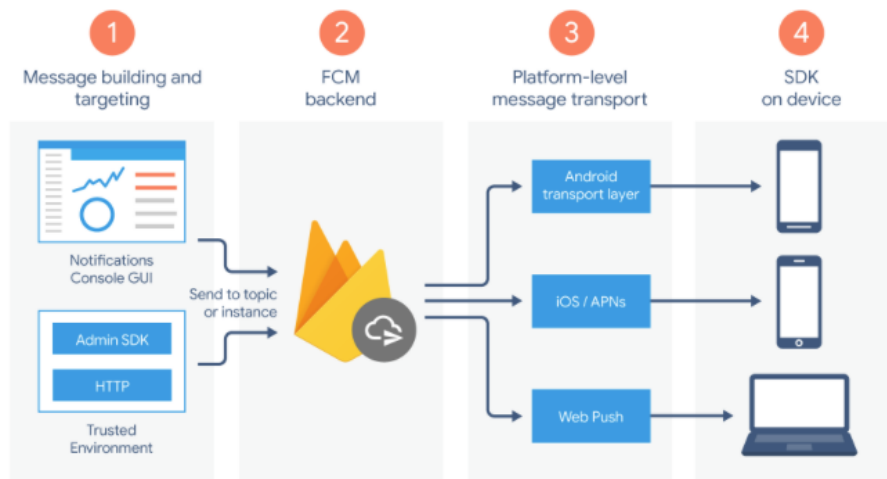


Figure 3.3: Firebase Architectural Overview [82]

In conclusion, Firebase Cloud Messaging is a powerful solution that offers a wide range of messaging options and capabilities, supporting enhanced user engagement and app synchronization [80, 81, 82].

## 3.6 Git - Version Control System

The use of Git, a Version Control System (VCS), is essential in modern software development as it provides a systematic and effective mechanism to manage changes in software projects [83]. A typical Git process includes the following steps:

1. **Project Initiation:** New projects are started or existing ones imported into Git. The initial version of the project is checked out into a private working directory.
2. **Committing Changes:** Developers submit their modifications, such as bug fixes or new features, to the repository along with an explanation. This aids in tracking and comprehending the project's evolution.
3. **Synchronization:** Git allows developers to update their own versions of software with changes submitted by colleagues, ensuring everyone works on a consistent code base and encouraging cooperation.
4. **Versioning:** When a release is complete, Git allows you to tag or label all files with the release's name. This snapshot makes it easier to navigate between project versions.

In addition, Git enables a variety of deployment configurations to meet the demands of individual projects. In some cases, the VCS may be configured as a local instance of GitLab. This local deployment, which includes authentication, guarantees that all apps and code within the project are secure and private [84]. The flexibility of customizing Git's deployment to satisfy security and privacy requirements highlights the technology's adaptability as a software development management tool [83].

## 3.7 Dialogflow

Dialogflow, developed by Google, is a tool designed to understand and interpret natural language, allowing developers to integrate a conversational interface into mobile and web applications. One of the primary features of Dialogflow is its ability to interpret user inputs. It can analyze text inputs, such as those entered into a chatbot interface and generate text responses, which can then be displayed to users [85]. By providing feedback to the users based on their inputs, it essentially closes the loop of the HitL system.

## 3.8 Xamarin

There are numerous approaches that can be adopted in the rapidly growing and competitive world of mobile application development, including native app development, HTML5 mobile sites and cross-platform development frameworks. Xamarin is among the leaders in this field, providing a framework for creating hybrid applications that can operate on several platforms such as iOS, Android and Windows using C# [86].

A paper titled "Mobile Application Development Research Based on Xamarin Platform" dives into the architecture of Xamarin and its utility in mobile app development. The article provides a comparison of Xamarin Forms, Android and React-Native, providing insights into the platforms' strengths and limitations [87].

This idea is further explored in Jared Dickson's "Xamarin Mobile Development," which examines Xamarin as a cross-platform mobile development tool that allows for a single programming language, the usage of native UI elements, code sharing for reuse and near-native performance. The study emphasizes Xamarin's capabilities for geolocation, Bluetooth, shared libraries and native bindings. It also highlights that, while Xamarin is a solid solution, it may not be the solution to all problems and should be implemented properly based on specific use cases [88].

The research article "A performance evaluation of cross-platform mobile application development approaches" provides another perspective to our understanding of Xamarin. It reveals design trade-offs and provides guidance for selecting the most appropriate approach based on performance characteristics by examining the performance characteristics of mobile applications developed using various common tools, including native SDKs of Google Android and Apple iOS and cross-platform tools like Apache Cordova [89], Microsoft Xamarin [86] and the now discontinued Appcelerator Titanium[90] [91].

As evidenced by the information contained in the literature, the development of mobile applications using the Xamarin platform offers promising possibilities for the development of cross-platform applications [87, 88, 91].

In conclusion, it is important to note the appearance of .NET Multi-platform App UI (.NET MAUI), the open-source evolution of Xamarin. .NET MAUI expands from mobile to desktop scenarios, with its UI controls entirely redesigned for performance and flexibility [92]. Our team intends to switch from Xamarin to .NET MAUI in the future.

## 3.9 ASP.NET Blazor

ASP.NET is a free, open-source, cross-platform developer platform for building many different types of applications. This platform allows you to use multiple languages, editors and libraries to create applications for web, mobile devices, desktop, games, IoT and more [93].

Blazor is a framework for creating client-side web user interfaces using .NET. It allows developers to design interactive user interfaces utilizing C# instead of JavaScript. In addition, It supports a broad range of browsers that support WebAssembly [94], rendering user interfaces such as HTML and CSS, even accommodating mobile browsers. Furthermore, Blazor's architecture enables seamless integration with hosting platforms like Docker and supports the creation of hybrid mobile and desktop applications in combination with .NET [95].

## 3.10 Docker

Docker is an open-source platform, providing an environment for developers to build, deploy, run, update and manage containers. At the core of Docker lies the concept of a container. A container is essentially a standard software unit that packages up code and all its dependencies. This include runtime, system tools, libraries and settings. Ensuring a fast and reliable execution of the application across varying computing environments. This platform has brought a revolution in simplifying the development and delivery of distributed applications [96, 97].

## 3.11 SQL

Structured Query Language (SQL) is a programming language primarily utilized for managing and processing information in a relational database. These databases organize data into tables, which SQL commands can manipulate to store, edit, delete, search and retrieve information. SQL is popular among data analysts and developers due to its compatibility with many programming languages and user-friendly syntax. Relationships are formed between database tables in order to optimize data storage. MySQL, a free, open-source system provided by Oracle that is compatible with many operating systems and frequently used in web applications, is an example of a relational database management system that use SQL [98].



## 3.12 Third Party Libraries

Libraries play a crucial role in programming, as they contain collections of pre-written code that developers can leverage to optimize and simplify various tasks [99]. NuGet is the officially recognized code sharing method for .NET. This platform outlines how .NET libraries are built, hosted and consumed, as well as the tools required for each of these activities [100].

The table below contains a list of all NuGet packages used in the project:

Table 3.1: NuGet Packages used in the project

NuGet	Version	Description
RestSharp	106.15.0	Simple REST and HTTP API Client [101]
Xamarin.Essentials	1.7.5	Kit of essential API's for app development [102]
Xamarin.Firebase.iOS.CloudMessaging	4.7.1	C# bindings for Firebase APIs Cloud Messaging iOS Library [103]
Xamarin.Firebase.Messaging	122.0.0.1	C# binding for Firebase APIs Cloud Messaging Android Library [104]
Com.Airbnb.Xamarin.Forms.Lottie	4.1.0	Render After Effects animations natively on Android, iOS, MacOS, TVOs and UWP [105]
IdentityModel	6.1.0	OpenID Connect & OAuth 2.0 client library [106]
Microcharts.Forms	0.9.5.9	Simple, cross-platform chart library [107]
Newtonsoft.Json	13.0.3	Json.NET is a popular high-performance JSON framework for .NET [108]
NGSIBaseModel	1.2.2	This is a multi-purpose parser for the NGS10 standard. The NGS10 standard is a the default standard used by ORION [109]
Plugin.BluetoothLE	6.3.0.19	ACR Bluetooth LE Reactive Plugin for Xamarin and Windows [110]
Rg.Plugins.Popup	2.1.0	Plugin for Xamarin forms. Allows you to open any page as a popup [111]
Socialite Sensors	1.0.5	Manages smartphone sensor data
Xamarin.Forms	5.0.0.2578	Build native UIs for iOS, Android, UWP, macOS, Tizen and many more from a single, shared C# codebase [112]
Xamarin.Forms.Maps	5.0.0.2578	Maps models and renderers for Xamarin.Forms [113]
ZXing.Net.Mobile.Forms	3.1.0	ZXing.Net.Mobile is a C# .NET library based on the open source Barcode Library [114]

# 4 UrsoVerde

## 4.1 Objectives

UrsoVerde is an initiative that uses mobile technology to promote healthy behaviors and sustainability in urban environments. This project is based on the idea that rewarding and encouraging residents for their sustainable practices can result in long-term behavioral changes that benefit the city’s environment [11, 5, 6]. In collaboration with the city council of Coimbra, we were challenged with the objective of creating a mobile application that would achieve this vision.

The genesis of the UrsoVerde project was an Android application prototype developed in collaboration with CoimbraCityLab [115], earning a place among the top ten finalists in a contest that led to an invitation to expand the project further. This early version of the app featured a set of capabilities such as user activity monitoring, access to a ChatBot, a leaderboard and a BLE advertiser that allows IoT devices to detect the user’s smartphone. However, the way it was designed and implemented, as shown in figures 4.1 and 4.2, limited its potential [71] .

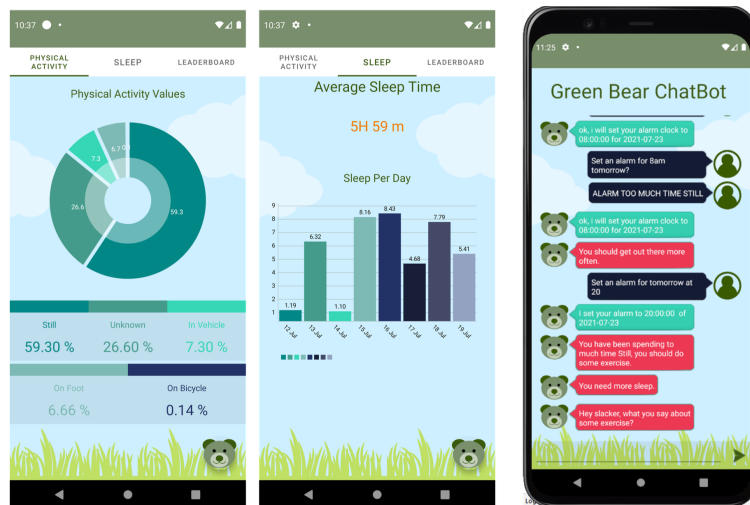


Figure 4.1: User’s activities and ChatBot from the first version of the UrsoVerde project [71]

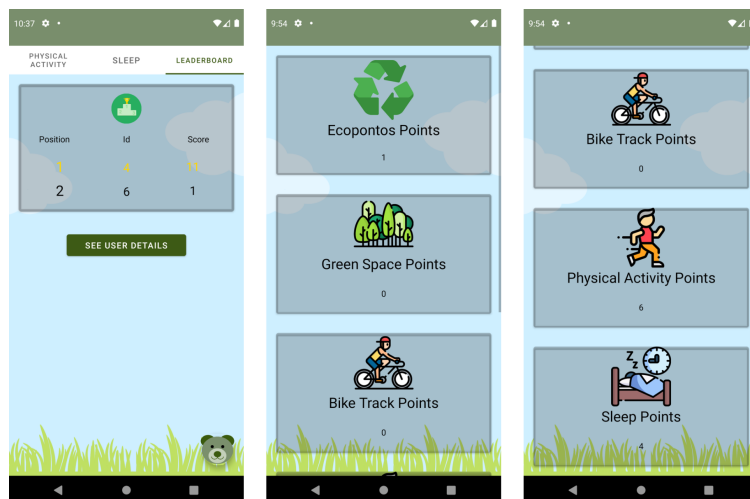


Figure 4.2: Leaderboard and user’s points display from the first version of the UrsoVerde Project [71]

One significant issue was that it was platform-specific. Because the software was designed primarily for Android, iOS users have no access to it, limiting its exposure to a significant portion of prospective customers.

Although the app was simple and effective in terms of design, it did not satisfy contemporary requirements for mobile application UI. Users have come to expect a high level of aesthetic appeal and intuitiveness in mobile apps, which was not delivered by this early version.

Another disadvantage of the app was the lack of user login and registration functionality. To identify the logged-in user, the system relied on an automatically generated token. This configuration created a serious issue since every time the software was reinstalled, all user information was reset.

The application’s point system and user activity validation had various drawbacks in its initial design. The point system was managed in the back-end and was entirely reliant on a connection to the LoRaWAN network each time a user performed a sustainable activity. This technique has an impact on the app’s operation, especially in locations with unreliable or non-existent network connectivity. Simultaneously, user activity validation was completely dependent on IoT devices. The BLE advertiser was the user’s smartphone, which the IoT device would scan for. The IoT gadget would update the user’s activity information through the LoRaWAN network after identifying the smartphone. This strategy severely limited the number of actions that could be recorded and monitored, restricting the application’s ability to add new features and functionality.

The new version of the UrsoVerde project began with many particular goals in mind, all

of which intended to make the application more flexible, comprehensive and accessible to a broader user base. It attempted to combine multiple functionalities in order to increase its effect and establish long-term habits among users. Here are the primary goals set:

- Migrate the application platform from Android to Xamarin to make the app available on both Android and iOS devices, expanding the app's reach.
- Improve the overall user experience by redesigning the UI.
- Add login and account registration tools to deliver safe and specific (exclusive) user experience.
- Develop a service for tracking weekly achievements and user activities at strategic points like recycling bins, public transportation and green zones, to encourage application use.
- Implement a rewards system that provides users with 'citizen points' for successfully completing weekly objectives.
- Reconfigure the application's interaction with IoT devices to allow a user's smartphone to readily detect these devices, allowing the system to update user activities data directly through the smartphone.
- Implement a QR code reader and create custom QR codes for various places of interest to serve as a cost-effective alternative for IoT devices.
- Develop a system to allow exchanging of 'Citizen Points' through the use of the QR Code reader.
- Implement data privacy and security measures in compliance with the General Data Protection Regulation (GDPR).

The proposed objectives serve as the base for the development of the UrsoVerde project. Achieving these goals will enhance the overall functionality of the application serving the needs of its users.

## 4.2 Requirements

This section discusses the functional and non-functional requirements for the mobile application.

### 4.2.1 Functional Requirements

Functional Requirements (FR) are essential components of software development because they outline the exact behaviors and activities that a system or application must execute in order to accomplish its intended function. They are the specifics of "what the product must do" or "what the system should do". The behavioral features of a system are described by functional requirements, which indicate the inputs, the related outputs and the interactions between them [116].

Table 4.1 lists the FR defined for the implementation of this project.

Table 4.1: Functional Requirements defined for the UrsoVerde platform.

FR	Name	FR	Name
FR0	Login	FR8	QR Code Generator
FR1	Register	FR9	Citizen Points Transfer
FR2	Logout	FR10	Notifications
FR3	Settings Page	FR11	Step Counter
FR4	Activity Tracking	FR12	Store Information
FR5	Map Page	FR13	ChatBot
FR6	Objective Page	FR14	Citizen Points Wallet
FR7	QR Code Reader	FR15	Leaderboard

Table 4.2 shows the parameters used to represent the requirements, including priority, description, actors, pre-conditions, event flow and expected outcome.

Table 4.2: Requirements parameters

Parameters	Parameters Description
Priority	<p><b>Must:</b> requirement is essential to the project.</p> <p><b>Should:</b> requirement is important to the project but the system should work without it.</p> <p><b>Could:</b> implemented requirements that wont affect implementation of the important requirements.</p> <p><b>Will not:</b> not implemented requirements for the project, considered as future work.</p>
Description	Overall exposure of the requirement.
Actors	Systems that intervene with the application.
Pre-conditions	Conditions necessary to the requirement functionality.
Events flow	Description of the actions that an actor needs to fulfill in order to get the expected outcome.
Expected outcome	The expect result from the action.

### Login page

- **Priority:** Must.
- **Description:** The application's initial screen should present users with fields to enter their username and password, along with buttons to log in and register.
- **Actors:** User.
- **Pre-conditions:** User has not previously selected the "Remember me" option during a successful login.
- **Event flow:** The user inputs their username and password, then presses the "Login" button. If the user doesn't have an account, they can navigate to a registration page by clicking the "Register" button.
- **Expected outcome:** If the credentials are correct, the user is redirected to the main page of the application. If they are not, an error message is displayed.

## Register

- **Priority:** Must.
- **Description:** The application should provide users with the ability to create a new account.
- **Actors:** User.
- **Pre-conditions:** User has pressed the "Register" button on the Login page.
- **Event flow:** After pressing the "Register" button on the Login page, the user is taken to the Registration page. The user must then enter a valid username and password and click the "Register" button on this page.
- **Expected outcome:** If the provided username and password are valid, the registration process is successful and the user is returned to the Login page. If registration fails, an error message is displayed.

## Logout

- **Priority:** Must.
- **Description:** The application should include a feature to end the user session.
- **Actors:** User.
- **Pre-conditions:** User is logged in.
- **Event flow:** User selects the Logout button from the settings page.
- **Expected outcome:** The application redirects to the Login page, user credentials are cleared and the "Remember me" option is deselected.

## Settings Page

- **Priority:** Should.
- **Description:** The application should include a page where users can access and alter various configuration settings such as 'Logout', 'Change Password', 'Change Language' and 'Change Theme'.



- **Actors:** User.
- **Pre-conditions:** User is logged in.
- **Event flow:** Upon pressing the Settings button, the user is directed to the Settings page.
- **Expected outcome:** The user's settings are successfully modified according to their preferences.

### Activity Tracking Page

- **Priority:** Must.
- **Description:** The application should provide a page to display user activity in the form of charts.
- **Actors:** User.
- **Pre-conditions:** User is logged in.
- **Event flow:** Users are presented with charts reflecting the frequency of their activities.
- **Expected outcome:** The chart values should update every time the user performs an activity. If users click on these charts, a popup should appear with more information about that specific activity.

### Map Page

- **Priority:** Must.
- **Description:** The application needs to feature a map page that displays the user's location and nearby points of interest where sustainable actions can be performed.
- **Actors:** User.
- **Pre-conditions:** User is logged in and has granted permission for location services.
- **Event flow:** Users can access the map, zoom in or out and check for nearby points of interest, represented as clickable map pins.

- **Expected outcome:** When users click on a chosen pin, a popup appears providing information about that specific location.

## Objectives page

- **Priority:** Must.
- **Description:** The application needs a dedicated page for Weekly Objectives. Completion of these objectives rewards the user with 'citizen points'. Objectives should be laid out in a list format, with each entry featuring an icon representing its category (public transport, recycling, green zone, or steps taken), a description and a goal to be achieved for earning one 'citizen point'.
- **Actors:** User.
- **Pre-conditions:** User is logged in.
- **Event flow:** Users can scroll down the list to explore different challenges.
- **Expected outcome:** If the user's score for a challenge is zero, the goal appears in red. If the challenge is in progress, it appears in yellow and upon completion, it turns green and a 'citizen point' is awarded.

## QR Code Reader

- **Priority:** Must.
- **Description:** The application should use the smartphone's camera to scan QR codes, a crucial feature for validating users' sustainable actions and allow the exchange of 'citizen points'.
- **Actors:** User.
- **Pre-conditions:** The user is logged in and has granted permission for camera usage.
- **Event flow:** The smartphone's camera view is displayed on the screen with a central square, guiding users to position QR codes for scanning.
- **Expected outcome:** If a valid QR code is scanned, a success message is displayed and the activity meter corresponding to the QR code type increases. If the QR code is invalid, an error message is presented.

## QR Code Generator

- **Priority:** Must.
- **Description:** The application should incorporate a mechanism for generating a QR code linked to the user's profile, facilitating the transfer of 'citizen points' between users.
- **Actors:** User.
- **Pre-conditions:** The user is logged in.
- **Event flow:** On the QR code reader page, a numeric selector along with a 'Generate QR Code' button is displayed. The user selects the number of points they wish to receive and generates a QR code accordingly.
- **Expected outcome:** A unique QR code is produced, encoding the user's ID and the selected number of points they wish to receive.

## Citizen Points Transfer

- **Priority:** Must.
- **Description:** The application necessitates a mechanism to securely transfer 'citizen points' earned from the completion of weekly objectives
- **Actors:** User.
- **Pre-conditions:** The user is logged in and has sufficient points to make the transaction.
- **Event flow:** The user intending to transfer points must direct the QR code reader towards the QR code generated by the receiving user.
- **Expected outcome:** Following the QR code scan, a success message will be displayed if the transaction is successful. An error message will be displayed if the transaction fails.

## Notifications

- **Priority:** Could.
- **Description:** The application may benefit from a notification system to update users regarding new weekly objectives and successful transactions.
- **Actors:** User.
- **Pre-conditions:** The application must be installed.
- **Event flow:** Notifications may appear on the smartphone at any given moment.
- **Expected outcome:** Upon clicking on the notification, the application should open.

### Step Counter

- **Priority:** Must.
- **Description:** The application is required to gather user step count data from sensor inputs and process this information to step activity metrics.
- **Actors:** User.
- **Pre-conditions:** The user has granted all necessary permissions.
- **Event flow:** Following login, the application initiates a scan for step count data.
- **Expected outcome:** Provided all permissions are in place, the application can collect the needed data and update the step count value.

### Store Information

- **Priority:** Should.
- **Description:** The application is designed to store information locally until an internet connection becomes available.
- **Actors:** None.
- **Pre-conditions:** The application has been installed.

- **Event flow:** Upon gathering sensor data, it is stored in a local database on the smartphone and remains there until successfully sent to FIWARE.
- **Expected outcome:** In the absence of an Internet connection, the application stores sensor information in a local smartphone database. Once the data is successfully transferred to FIWARE, it is erased from the smartphone.

### ChatBot

- **Priority:** Could.
- **Description:** The application could incorporate a ChatBot interface where users can inquire about the UrsoVerde platform, potential sustainable activities they can engage in and tips for environmentally conscious behavior. The ChatBot may also provide users with feedback on their sustainable action scores and advice on improvement, in the logic of HitLCPS.
- **Actors:** User
- **Pre-conditions:** the user must be logged in.
- **Event flow:** Users can interact with the ChatBot by sending text messages via an input field provided on the page, receiving responses in a conversational text bubble format.
- **Expected outcome:** Assuming an active internet connection, every user message should prompt a corresponding response from the ChatBot.

### Citizen Point Wallet

- **Priority:** Must.
- **Description:** The application must include a feature where users can check their current 'citizen points' balance along with a history of spent points.
- **Actors:** User.
- **Pre-conditions:** The user must be logged in.
- **Event flow:** Users can view their current points balance on the activity page.

- **Expected outcome:** When the balance label is pressed, the application should redirect users to the QR Code reader page, allowing the exchange of points.

## Leaderboard

- **Priority:** Must.
- **Description:** The application must feature a leaderboard where users can compare their 'citizen points' standing against other users.
- **Actors:** User.
- **Pre-conditions:** The user must be logged in.
- **Event flow:** From the Objectives page, users can access the Leaderboard page by selecting the Leaderboard button.
- **Expected outcome:** On accessing the Leaderboard page, users should see a ranked list of all users based on 'citizen points'. The user's own rank should be highlighted in a distinctive color to facilitate easy identification.

### 4.2.2 Non-functional Requirements

Non-Functional Requirements (NFRs) are critical in defining the system's operational capabilities and limits. Non-functional requirements, as opposed to functional requirements, address the system's performance characteristics, including but not limited to accessibility, security, support, usability and others [117].

- **Accessibility:** Ensure that the system supports the adequate number of service requests.
- **Performance:** The application should not have a significant impact on the battery life, memory, or performance of the smartphone.
- **Privacy:** All data gathered must be kept private and secure. Respecting the GDPR.
- **Support:** The application should be supported by the majority of the devices currently on the market.

- **Usability:** The application must have an appealing user interface as well as features that are essential to the users.

The system’s accessibility, which is more concerned with the FIWARE infrastructure than with the application itself, guarantees that the system remains accessible regardless of the number of users connecting at any given time. This feature ensures that all users have continuous access to services, providing a high degree of service availability. Although native programming has been considered faster, Xamarin provides a balanced approach that improves both code complexity and overall performance [118]. The decision to use Xamarin and FIWARE came from the fact that the team I joined already made use of both technologies, which made the integration process simpler.

Given the sensitive nature of the data collected, privacy is a significant non-functional requirement for our application. Data security is granted by robust authentication mechanisms and storing data locally on the smartphone for a brief period of time. This strategy protects the privacy of the data while also providing for the essential capabilities.

Broad device support is one of the main non-functional requirements we established for our application. We assure compatibility with a wide range of devices by using Xamarin as our development platform. Our application is specifically built to operate on any device running iOS 9 or above and Android 4.0.3 or higher [119].

The usability of our application is a main concern. We recognize that an application’s aesthetics and user interface may have a major influence on user experience. As a result, we need to create a visually appealing and simple interface for our application in order to deliver a user-friendly experience.

## 4.3 Architecture

An application’s architecture not only specifies the essential structure of the program, but it also explains how distinct software components interact with one another and with other entities. This section presents an overview of the UrsoVerde platform’s general architecture, outlining its connections to external platforms as well as its internal operations.

### 4.3.1 Overall architecture

The overall architecture of the UrsoVerde platform is a synchronized system designed to permit communication between various components, with the mobile application serving as

core. This chapter explains the platform’s functional dynamics, concentrating on its primary parts and how they interact to provide a full service.

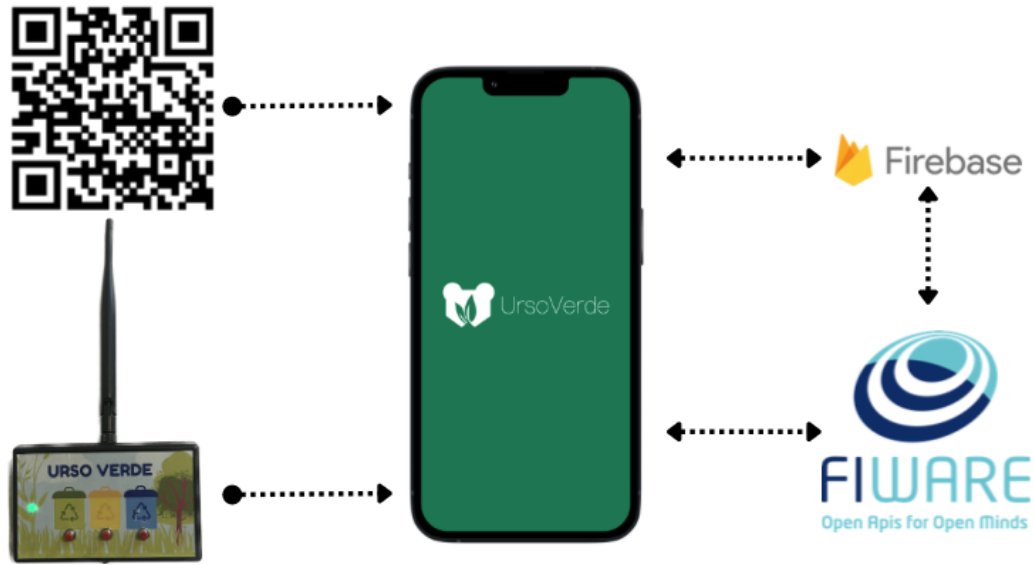


Figure 4.3: UrsoVerde platform architecture

As shown in figure 4.3, the UrsoVerde mobile application is central to the system, establishing direct communication channels with the IoT devices, the FIWARE and Firebase.

The app communicates with IoT devices through BLE technology. Rather than establishing a permanent connection, the app searches for BLE advertisements broadcasted by the IoT devices. As these BLE beacons are identified, the application can assume that the user is engaged in a sustainable activity.

FIWARE serves as the cloud-based platform that the mobile application interacts with for various operations, such as data storage, retrieval and user authentication. Firebase Cloud Messaging is linked to both the mobile application and FIWARE. The app subscribes certain FIWARE topics through Firebase, allowing notification messages to be received every time the subscribed topic is added, deleted or updated.

### 4.3.2 Mobile application architecture

As shown in Figure 4.4, our application’s architecture follows a common application structure, characterized by many threads operating concurrently to perform distinct tasks. The primary UI thread acts as main, as are other background threads specialized for specific activities.



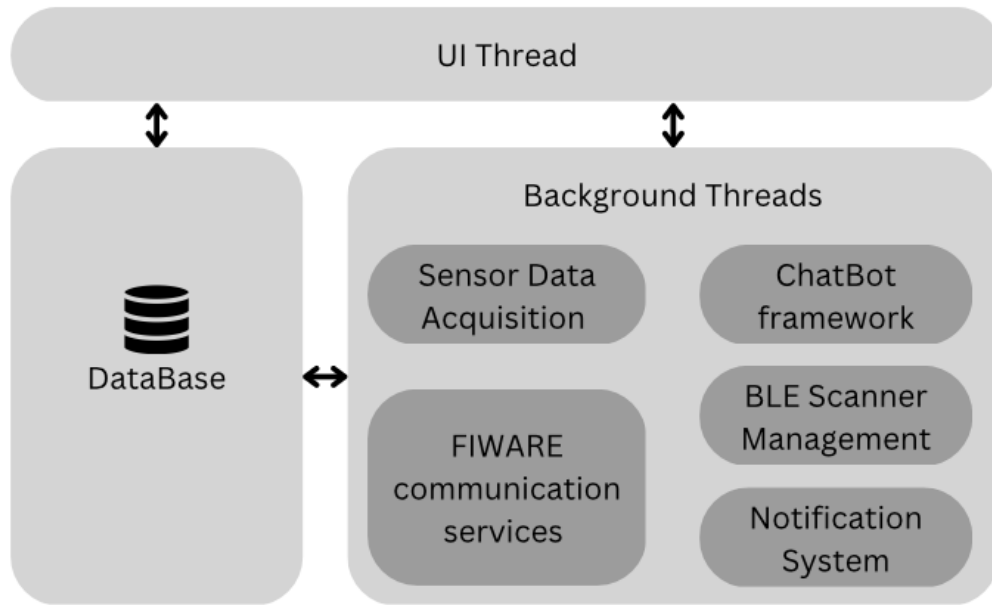


Figure 4.4: UrsoVerde Application Architecture Diagram

### UI Thread

The UI Thread, or User Interface Thread, is the application’s main execution thread. It is in charge of managing all visual interactions with the user, such as Shell navigation and function interaction. This thread also manages the start of all other threads in the program.

Its capabilities include producing the graphical interface, capturing user inputs, updating screen displays and navigating across various pages.

### Background Threads

The application’s background threads manage a variety of functions such as data gathering and storage, notification systems, IoT device scanning and the ChatBot framework. These threads run in parallel with the UI Thread and are responsible for activities that do not require direct user input.

- **Sensor Data Acquisition:** This thread collects data from various smartphone sensors, providing inputs for application functionalities.
- **FIWARE Communication:** This thread handles communication with the FIWARE back-end, sending and receiving data as needed.
- **Notification System:** Using Firebase Cloud Messaging API, this thread allows the app to notify users of updates, notifications and other pertinent information.
- **BLE Scanner Management:** This thread hosts the BLE scanner in the background, allowing for continuous scanning of IoT devices.

- **ChatBot Framework:** This thread allows the ChatBot Framework to wait for new messages or user queries.

### **DataBase**

To handle user data, our program makes use of the device storage system. All user data is stored in the smartphone's internal database as "preferences." If the program cannot connect to the Internet, it temporarily saves data locally. When the connection is restored, the FIWARE communication service sends the previously saved data to the FIWARE backend.

# 5 Urso Verde Development

## 5.1 Partner Meetings

Communication and collaboration was critical during the UrsoVerde project's development. We set a strict meeting schedule to maintain transparency, explore ideas and keep everyone involved updated on the project's status.

### 5.1.1 Internal Weekly Meetings

To streamline the project workflow and make decisions about future steps, we conducted weekly internal meetings. These meetings were attended by the supervisors and occasionally by other students. During these meetings, we would review the tasks and objectives accomplished in the previous week and plan new tasks for the upcoming week. A significant outcome of these meetings was the development of a section of a paper titled "Confluence: An Integration Model for Human-in-the-Loop IoT Privacy-Preserving Solutions Towards Sustainability in a Smart City". As a co-author, I contributed to the section about the UrsoVerde Application (Appendix B). The paper was submitted for review to the IEEE Internet of Things Journal and received favorable feedback.

### 5.1.2 Partnership with Coimbra City Hall

As our project involves a cooperation with the Coimbra city hall, it was critical to engage with them on multiple occasions to verify that we were meeting their expectations. We discussed potential features and offered current updates on the project. These sessions assisted us in adapting the app to the needs of the Coimbra community. Figure 5.1 shows the timeline of all the meetings and events that took place with the Coimbra City Hall, beyond the constant contact by email.



Figure 5.1: Timeline of meetings and events with the Coimbra City Hall.

**Initial Meeting:** On May 5th, 2022, in the introductory meeting, we discussed the objectives of the UrsoVerde application and how it could be deployed in Coimbra. This meeting helped lay the foundation for the project and outline the primary goals that UrsoVerde aims to achieve.

**Update Meeting:** On September 1st, 2022, as the project progressed, we held this meeting to present the mockups and discuss potential features. This session enabled us to receive early feedback on the design and functionality of UrsoVerde, allowing for necessary adjustments to be made in the early stages of development.

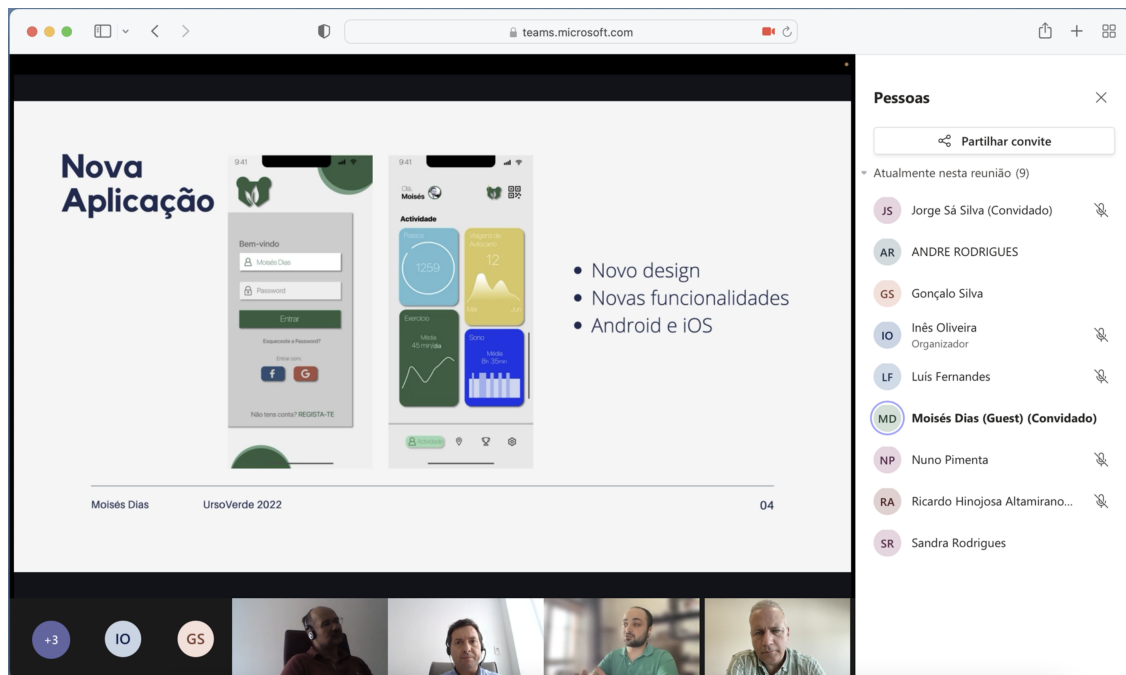


Figure 5.2: Update meeting with the Coimbra City Hall.

**Preparation Meeting:** On October 6th, 2022, a meeting was held to prepare for the upcoming Portugal SmartCities Summit. We discussed logistics and the presentation of prototypes at the summit. The meeting ensured we were prepared to showcase UrsoVerde at the event.

**Portugal Smart Cities Summit 2022:** We attended the Portugal Smart Cities Summit in Lisbon on October 12th, 2022. This summit provided us with an opportunity to present a prototype of our app to a wider audience. This exposure provided us with significant insights and criticism, allowing us to improve our prototype.



Figure 5.3: UrsoVerde Team at Portugal Smart Cities Summit 2022

**Final demonstration to Coimbra City Hall:** On June 12th, 2023, we conducted an extended meeting with Coimbra City Hall to show the final stage of our app's prototype. We began the discussion with a detailed demonstration of the software, emphasizing its functionalities and characteristics. Later, we followed a guide for the demonstration as shown in appendix A. In this demonstration we exemplified real-time app usage by creating UrsoVerde Accounts on two distinct devices, one iOS and one Android. We displayed how to add new objectives via the back-end, use IoT devices to increase activity values and use the QR code reader. We also demonstrated how to complete a challenge and transfer 'citizen points' from one account to another. The demonstration concluded on a positive note, with a proposal from Coimbra City Hall to test the software in a real-life setting in a Coimbra public school, "Escola Jaime Cortesão", and the Municipal Market merchants. This opportunity will undoubtedly provide us with more insights to refine and improve our application further.

Shortly after the meeting, the Coimbra City Hall posted a message on their official LinkedIn profile, which is evidence of its success. The post highlighted the potential of the

UrsoVerde application in encouraging sustainable city practices, including images from the meeting, as seen on appendix D.



Figure 5.4: Meeting with Coimbra City Hall Team.

## 5.2 Cascais CityPoints Meeting

In collaboration with the Coimbra City Hall, we held a meeting with the team behind "Cascais CityPoints", a mobile application similar to ours, which incentivizes sustainable activities, validated through the use of QR Codes, with a point-based reward system [47]. The meeting enabled us to understand their app's functionality and the way they manage activities and user points, providing valuable information for our project. Despite the similarities we highlighted the following aspects:

- Users can exchange accumulated points for vouchers redeemable at municipal activities such as theater shows, cinemas, and concerts.
- At the time of our meeting, the platform, having a user base of 12,026 users, also includes an application where city infrastructure issues can be reported for points.
- The system is accompanied by a PowerBI statistics dashboard that provides significant data analysis capabilities.
- An independent Backoffice dashboard enables management of activities, vouchers, and users.

- Public transportation usage is capped at a daily point maximum of one to prevent potential system abuses.
- The application also includes a map feature that displays nearby locations where users can earn and redeem points.
- To foster competition, the application includes a leaderboard, awarding additional points to top-ranking users.

Our meeting with the Cascais CityPoints team was an excellent learning experience. Exploring the common and distinctive features of our apps while discovering about their strategy and solutions have greatly broadened our vision and guided our next steps.

### **5.3 Partnership with Escuela Politécnica Nacional**

Escuela Politécnica Nacional located in Quito, Ecuador, is collaborating with us with a parallel project to UrsoVerde, named "Colibri". The name Colibri, which represents the national bird of Ecuador, mirrors our concept of represent local symbolism in the platform. Led by Professor Soraya Sinche, the Colibri's characteristics are closely aligned with the UrsoVerde project. It incorporates the same features, providing a similar user interface and experience.

We arranged multiple meetings with the Escuela Politécnica Nacional team at the beginning of the Colibri project. Our aim was to provide a comprehensive understanding of the UrsoVerde platform, its goals, features, and design. These discussions assisted the Colibri project getting a successful start by ensuring that both platforms properly follow our common objectives.

## 5.4 Implementation

### 5.4.1 Xamarin application development

The UrsoVerde application was developed with Xamarin, a sophisticated framework for producing cross-platform apps. Xamarin is especially useful for developers who want to share code, tests and business logic across platforms, as well as those who want to design apps in C# using Visual Studio or any equivalent IDE, in our case, JetBrains Rider IDE[120, 121].

As demonstrated in figure 5.5, one of the primary benefits of Xamarin is its ability to create native UIs on each platform, while the business logic is written in C# and shared across platforms. It is estimated that Xamarin can share roughly 80% of the application code in most circumstances, making it an effective solution for cross-platform development [120].

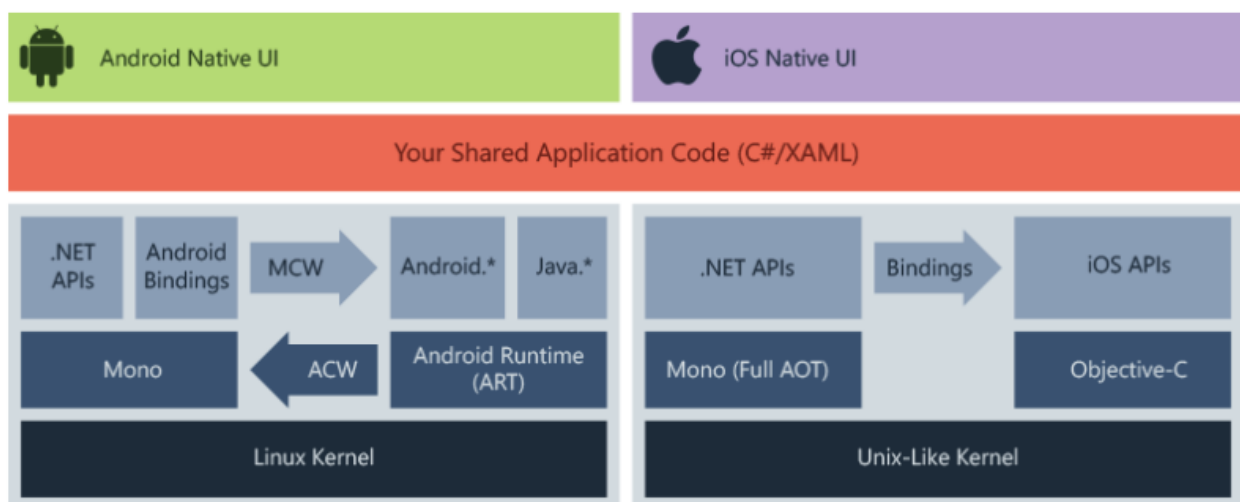


Figure 5.5: Overall architecture of a cross-platform Xamarin application. From [120]

Xamarin, which is built on top of .NET, performs activities like memory allocation, garbage collection and compatibility with underlying platforms automatically, providing developers with a robust and efficient development environment [120].

The creation of a page in Xamarin is structured by the combination of a View and a ViewModel. The View, which is specified in XAML, is in charge of detailing the UI component of the page, displaying the visual layout with which the user interacts. The ViewModel, on the other hand, which is bound to the View, supports the logic behind the UI components in C#.

The View can include a variety of Xamarin components including as frames, buttons, labels and entries. Each of these components is equipped with an array of properties that



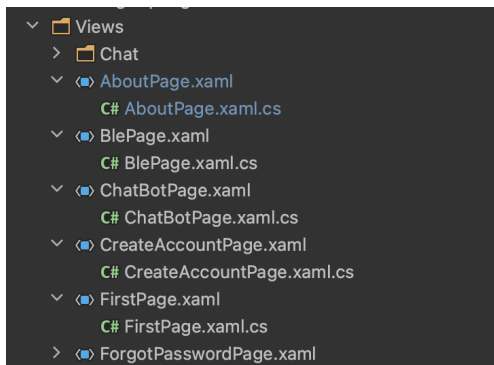


Figure 5.6: View (.XAML) and ViewModel (.CS) structure.

allow customization of appearance, performing a distinct purpose on the application's overall functionality and looks. For instance, a button's "Clicked" action can be bound to a certain piece of code in the ViewModel, thereby enabling the programming of specific outcomes from that interaction. In figure 5.7, we exemplify how we can define a button with the text "Trocar" in XAML and how it appears on the actual page of the application.

```
<Frame
  Margin=" [l:] 0, [t:] 10, [r:] 0, [b:] 0"
  Padding=" [all:] 0"
  BackgroundColor=" #016837"
  CornerRadius="10"
  HasShadow="False">
  <Button TextColor=" ■ White"
    BackgroundColor=" #016837"
    Text="Trocar"
    HorizontalOptions="CenterAndExpand"
    Clicked="QRCode" />
</Frame>
```



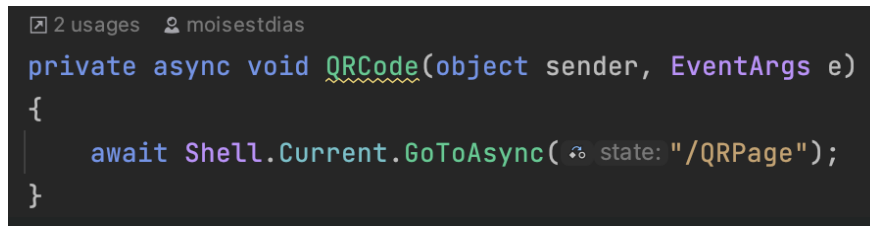
(a) Definition of button inside of a frame in XAML (b) Visual output from the XAML definition

Figure 5.7: Definition of component in XAML and respective output

The Shell structure serves as the foundation for each page in the application, ensuring a seamless transition between pages. This is accomplished by routing each page, which is navigated similarly to URL navigation [122]. Figure 5.8 illustrates the programming of the previously described button action. Here, we configure the application to transition to the QRPage upon the user's click on the button.

The inclusion of the 'async' and 'await' keywords in C# 5 substantially simplified asynchronous programming. They enable developers to create simple code that uses the Task Parallel Library to conduct long-running processes in a different thread and retrieve the results upon completion. The 'await' operator can be used within a 'async' method and can be applied to a Task. This causes the method's execution to stop and await the completion

of this task [123].

A screenshot of a code editor with a dark background. At the top left, it shows '2 usages' and a user icon labeled 'moisestdias'. The code is in C# and defines a private async void method named 'QRCode' that takes an 'object sender' and 'EventArgs e' as parameters. The method body contains a single line: 'await Shell.Current.GoToAsync(state: "/QRPage");'.

```
private async void QRCode(object sender, EventArgs e)
{
    await Shell.Current.GoToAsync(state: "/QRPage");
}
```

Figure 5.8: Button action defined on the ViewModel.

## 5.4.2 FIWARE Communication

Communication with ORION, a component of FIWARE, is a key in our application's development process. REpresentational State Transfer (REST) requests are used to do this. REST is an architectural approach for providing standards across computer systems on the web, allowing systems to communicate more easily with one another [124].

The REST methodology requires a client to make a request to the server in order to retrieve or modify data. Each request generally consists of:

- An HTTP verb, defining the kind of operation to perform.
- An header, which allows the client to pass along information about the request.
- A path to a resource.
- An optional message body containing data.

There are three basic HTTP verbs we use in requests to interact with resources in a REST system:

- GET - to retrieve a specific resource (by id) or a collection of resources.
- POST - to create a new resource or update an existing one.
- DELETE - to remove a specific resource by id.

Figure 5.9 provides an illustration of how we utilized the Postman software to test a REST request. This example showcases a GET request made to the URL "https://socialiteorion2.dei.uc.pt:9012/v2/entities?type=challenge&options=keyValues".

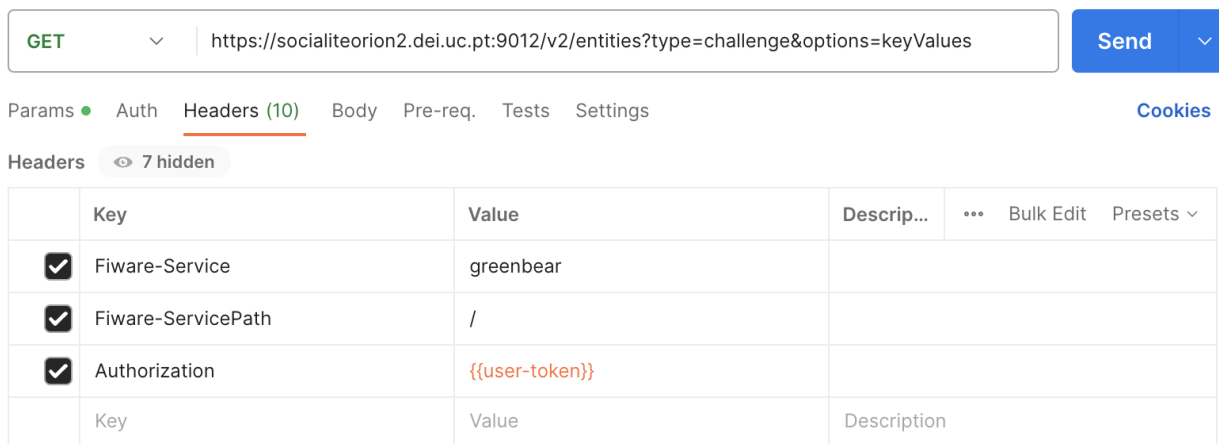


Figure 5.9: Example of a GET request using Postman. URL on top and the list of header parameters.

To deconstruct this URL, "socialiteorion2.dei.uc.pt:9012/v2/entities" is our host. The query parameters following the '?' specify the type of entity we're requesting. In this case, we're seeking an entity of the type "challenge". Furthermore, we use the "options=keyValues" parameter to retrieve the data in a simplified JSON format.

Within the Headers, we are specifying the path as "/greenbear" and using the "user-token" as the Authorization token. Upon providing the correct token, we anticipate receiving a response in JavaScript Object Notation (JSON) format containing information about the "challenges" entities, as depicted in figure 5.10.

```
{
  "id": "3",
  "type": "challenge",
  "challenge_disc": "Idas a zonas verdes",
  "challenge_objective": "5",
  "challenge_type": "green_space",
  "expiration_time": "2023-07-31T00:00:00Z"
},
{
  "id": "4",
  "type": "challenge",
  "challenge_disc": "Transportes públicos",
  "challenge_objective": "5",
  "challenge_type": "public_transport",
  "expiration_time": "2023-07-31T00:00:00Z"
},
```

Figure 5.10: GET request response in JSON format.

The above example shows that we have two 'challenge' entities in the ORION. Each of these entities has the same parameters, but they have distinct string values for each of them.

In order to incorporate this information into the app, it is essential to create a class object that has the same fields as the entities. In this scenario we have created a class called 'challenge' with the following attributes:

```
string id;  
string type;  
string challenge_disc;  
string challenge_type;  
string expiration_time;
```

Following that, we used the JsonConvert library to transform the received JSON data into an object of the 'challenge' class. This allows us to use the ORION data within our Xamarin application [125].

If the goal is to save data within the ORION, we must first declare the parameters of the class object within our application. The JsonConvert library is then used to convert this class instance back into a JSON object. This JSON object is then included in the body of a POST request to the FIWARE server. The server identifies the corresponding entity after receiving the request. If an existing entity matches the information submitted, the server updates that entity with the new values from the POST request. If no matching entity is found, the server creates a new entity based on the data provided.

This approach of using REST requests in conjunction with JSON conversion and data binding to class objects serves as the foundation for all data store and retrieval activities throughout the UrsoVerde application's development.

### 5.4.3 Login and account registration

To manage user login we use KEYROCK, a FIWARE component. When a user initially uses the application, it is requested to enter the username and password. These credentials are transmitted to KEYROCK for verification when you click the login button. KEYROCK then provides a validation answer. If the answer is 'true', meaning the credentials match a registered user, the program allows the user access and navigates to the main shell of the application. If the response is negative, however, the program denies access, preventing unauthorized users from entering.

When a user creates a new account, the program demands information such as a username, email address and password. These information are then forwarded to KEYROCK, which checks to see if the login credentials are unique.

KEYROCK establishes an account for the user if the username is unique and not already in use. In ORION, a matching user entity is equally generated, where all user data for that specific account is saved. Simultaneously, a unique 32-digit identifier is generated and assigned to the user. This identifier serves as a primary key for that particular user's data storage in ORION.

#### 5.4.4 User data

In UrsoVerde, user data is stored through an individual entity in ORION with the following attributes:

- **id:** This is a unique 32-digit identifier assigned to each user, which serves as the primary key for the user's data storage.
- **type:** This attribute denotes the type of the entity which, in this case, is 'User'.
- **username:** This attribute stores the username associated with the account, serving as a unique identifier for the user in the context of the application's interface.
- **challenges\_completed:** This attribute tracks which challenges a user has successfully completed within the app.
- **citizen\_points:** This attribute stores the total number of available citizen\_points a user has.
- **citizen\_points\_accumulated:** This attribute denotes the total number of citizen points a user has accumulated over time.
- **citizen\_points\_spent:** This attribute records the total number of citizen points a user has spent within the app.
- **create\_time:** This attribute records the timestamp of when the user entity was created in the system.
- **ecoponto:** This attribute is used for storing user-specific data related to recycling interactions within the app.
- **public\_transport:** This attribute records user-specific information related to the use of public transportation.

- **time\_greenpace:** This attribute stores the amount of times the user has been in green spaces.

A corresponding User class is constructed locally within the UrsoVerde application alongside the user entity in ORION. This class has the same properties as its ORION equivalent and serves as a mirror, reflecting user data inside the context of the program. Whenever a user attribute, such as completing a new challenge or earning more citizen points, has to be updated, the change is first done in the locally stored User class object. Following the local update, the modified data is sent to the associated ORION user entity. This procedure guarantees that the user data kept within the application is always synchronized with the data hosted in the ORION back-end.

## 5.4.5 Activity tracking

Sustainable user activities are tracked within the UrsoVerde application using two main methods: QR codes and IoT devices.

### 5.4.5.1 QR codes

To take advantage of QR codes' simplicity, we implemented a view that uses the ZXing library to read these codes [126]. These QR codes are generated with a structured message following this pattern:

```
"ENTITY_ID:user_Id,POINTS:transferPoints,ENTITY_TYPE:activity"
```

The "ENTITY\_ID" and "POINTS" parameters in this structure are unrelated to activity monitoring (they will be described in a later chapter), but the "ENTITY\_TYPE" is critical. The "activity" label is replaced with predefined values such as "eco" for recycling bins, "green\_space" for green zones and "public\_transport" for usage of public transportation.

When the QR code reader scans the QR code, the contained message is divided into the three components. Depending on the last parameter, we increment the corresponding user activity type and update the user data on the ORION back-end.



Figure 5.11: Generated QR code associated to a recycling bin.

#### 5.4.5.2 IoT Devices

IoT devices, alongside QR codes, provide an option for tracking user actions. Currently, we have developed a way to track recycling activities using IoT devices.

These devices are programmed to emit a BLE beacon with the identifier "ecopoint\_greenbear" for 20 seconds whenever a user presses a button on the device.

The UrsoVerde application runs a background service that periodically scans for BLE devices every 10 seconds. When it detects the BLE beacon emitted by the IoT device, the user's "ecoponto" parameter is incremented to reflect the recycling activity and the scanning process is paused for 60 seconds.

#### 5.4.6 Step Tracker

The UrsoVerde application uses the SocialiteSensors library to track the amount of steps a user takes. This library can collect a variety of data from the smartphone's sensors and store it in a local database named "SmartphoneData."

Among the data types that the SocialiteSensors library can track, are variables like sleep time, user location, battery life and most importantly for our purposes, the step count.

To incorporate this functionality into our app, we access the SmartphoneData database and specifically retrieve the 'step\_count' value. This value represents the number of steps a user has taken.

### 5.4.7 User Objectives

The UrsoVerde application relies heavily on objectives to drive user engagement and motivation. These objectives, stored in ORION as a "challenge" entity, are intended to reward users for completing environmental friendly activities.

A 'challenge' entity in ORION has several parameters:

- **id:** Unique identifier for the objective.
- **type:** This always remains 'challenge'.
- **challenge\_disc:** Provides a description of the objective.
- **challenge\_objective:** Specifies the value that the user needs to reach to earn a point.
- **challenge\_type:** Specifies the type of the objective, which can be one of the following - 'public\_transport', 'walk', 'green\_space', or 'ecoponto'.
- **expiration\_time:** The date when the objective will no longer be available.

Currently, objectives require manual input into the system using Postman. However, we plan to develop a user-friendly dashboard website that allows the UrsoVerde challenges to be managed easily.

Within the user class, we use the "challenges\_completed" parameter to keep track of each user's progress towards their objectives. To track this progress, we first perform a GET request to acquire the complete list of objectives. For each of these objectives, we compare its challenge\_objective value with the respective user's activity data. If the user's activity matches or exceeds the specified objective and the id of that objective isn't already present in the user's "challenges\_completed" field, we increase the user's point count by one. Simultaneously, the id of the accomplished objective is appended to the user's "challenges\_completed" field. This procedure prevents a user from accumulating points from a single objective that they have previously accomplished.

### 5.4.8 Points Transfer

In this sub-section, we will explore the mechanism for transferring points between users. Assume that User X wants to transfer points to User Y. This process can be described in the following steps:



1. **User Y:** User Y has to generate a QR code from the QR code reader page with the quantity of desired points to be received (`transferPoints`). The generated QR code carries the message:

`ENTITY_ID:userY_Id,POINTS:transferPoints,ENTITY_TYPE:user`

2. **User X:** User X scans this QR code with their reader. Since the third parameter of the QR code is 'user', a function is called that handles the transfer of points from one account to the other.
3. **Point Transfer Function:** This function receives the User Y's ID (`userY_id`) and the number of points (`transferPoints`) to be transferred as parameters. It then checks whether the 'transferPoints' value is less than User X's 'citizen\_points' value and if this condition is met, the function updates the 'citizen\_points' of both User X and User Y in ORION. Specifically, it deducts the 'transferPoints' from User X's 'citizen\_points' and adds the same value to User Y's 'citizen\_points'.
4. **User X and User Y:** Both User X and User Y are updated with the new 'citizen\_points' values.

### 5.4.9 Leaderboard

The LeaderBoard page introduces "gamification" to the application, there we present the list of users in descending order of their 'citizen\_points'. This is done by making a GET request to the ORION back-end. The URL for this request is "`https://socialiteorion2.dei.uc.pt:9012/v2/entities?type=user&attrs=username,citizen_points&orderBy=!citizen_points`".

In this request, we are asking for 'user' type entities. The attributes ('attrs') we need are 'username' and 'citizen\_points'. We also specify 'orderBy' as '!citizen\_points', which sorts the entities in descending order based on the 'citizen\_points' value.

We then deserialize the JSON response into a list of objects from the "Rank" class. The list is then displayed on the LeaderBoard page in the same order that they were received.

To highlight the logged-in user from the rest of the users on the LeaderBoard, we compare the 'username' attribute of each 'Rank' object with the 'username' of the User. If they match, it signifies that the 'Rank' object corresponds to the logged-in user. In such a case, we use a different color to represent the user's ranking frame, as seen in figure 5.20b, setting them apart from the other users in the LeaderBoard.

### 5.4.10 Map

The Map page of the application was implemented using Xamarin.Forms.Maps library, which dynamically displays the map based on the device being used. For iOS devices, it will display Apple Maps, while Android devices will show Google Maps [127].

The Xamarin.Forms.Maps library has features that make it an ideal choice for our implementation, as it provides functionalities to display the user's location on the map, include a search bar, apply filters, zoom in and out and allows to display customized pins.

In the ORION back-end we store a list of pins, each with specific parameters:

- **id:** The identifier of the location.
- **type:** The type of location, which could be 'green\_space', 'eco', or 'public\_transport'.
- **city:** The city where the location is situated.
- **lat:** The latitude coordinate of the location.
- **lon:** The longitude coordinate of the location.
- **message:** A description of the location.

To display these pins on the map, we make a GET request to retrieve the list of pins. With the latitude and longitude data, we plot a pin on the map at the corresponding location.

We've also incorporated a feature such that, when a pin is clicked, an image popup appears, displaying information about the city, a description of the location (message) and the type of point, as shown in figure 5.19.

### 5.4.11 ChatBot

The ChatBot page is designed to allow a two-way communication between the user and a server that gives user feedback and answers to questions.

The process of interaction begins when the user sends a message. This message is encapsulated into a payload message consisting of three parameters:

1. **text:** This carries the content of the user's message.
2. **session\_id:** This is the user's id, used for maintaining the context of the conversation.
3. **language\_code:** This denotes the language in which the message is written.

Once the payload message is structured, it is sent over, through a REST request, to a server specifically set up to handle Dialogflow interactions [85]. This server processes the user's intent, manages an appropriate response and sends it back to the UrsoVerde application. The response from the server is then displayed to the user.

#### 5.4.12 Notifications

The notification system is handled by the Firebase Cloud Messaging (FCM) API. Our notification system operates on a background thread that subscribes to specific entities from the ORION back-end. Any changes or updates to these subscribed entities trigger the FCM to push a notification to the user's smartphone.

Upon launching the app, we automatically subscribe the user to two topics: "challenge" and the "user.id" specific to the logged in user, this way:

```
firebaseService.SubscribeToSingleTopic("challenge");  
firebaseService.SubscribeToSingleTopic(user.id);
```

The "challenge" subscription ensures that the user is notified whenever a new challenge is added or an existing one is updated in ORION, as shown in figure 5.12. Subscribing to "user.id", on the other hand, allows users to receive updates on changes to their specific "User" entity. This could include changes in various parameters, such as their accumulated points. Therefore, for example, if points are added to a user's account, an update will immediately be pushed to their device via a notification.



Figure 5.12: Push Notification in iOS devices.

## 5.5 User Interfaces

### 5.5.1 App Icon and Initial Mockups

The icon is the initial point of interaction a user has with any app. The UrsoVerde icon, shown in figure 5.13, was created using Adobe Illustrator to represent the essence of the UrsoVerde program. It includes a bear, as a reference to the app's name and two leaves, which symbolize sustainable environments.



Figure 5.13: UrsoVerde logo.

Adobe Illustrator was also used to develop preliminary design mockups show in figure 5.14. These prototypes were critical in shaping the application's initial concept, which included all of the necessary features. They also served as the base for Xamarin's UI design.

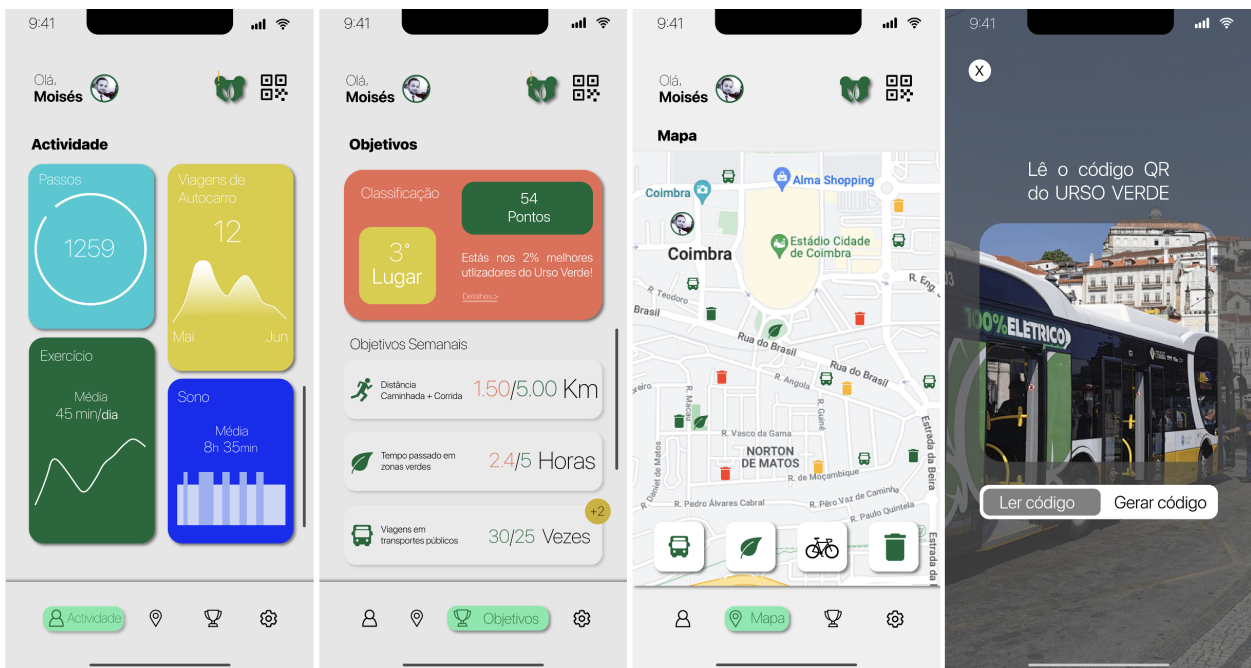


Figure 5.14: First application Mockups. From the left to the right: Activity Page, Objectives Page, Map Page, QR code reader Page.

It is crucial to note, however, that these designs were created prior to knowing the challenges of designing UI with Xamarin’s XAML markup language. As a result, during the development process, some modifications were required.

## 5.5.2 Application Flux

The application flux outlines how users can transition from one page to another. Figure 5.15 provides a diagram that illustrates how the navigation within the UrsoVerde application is structured:

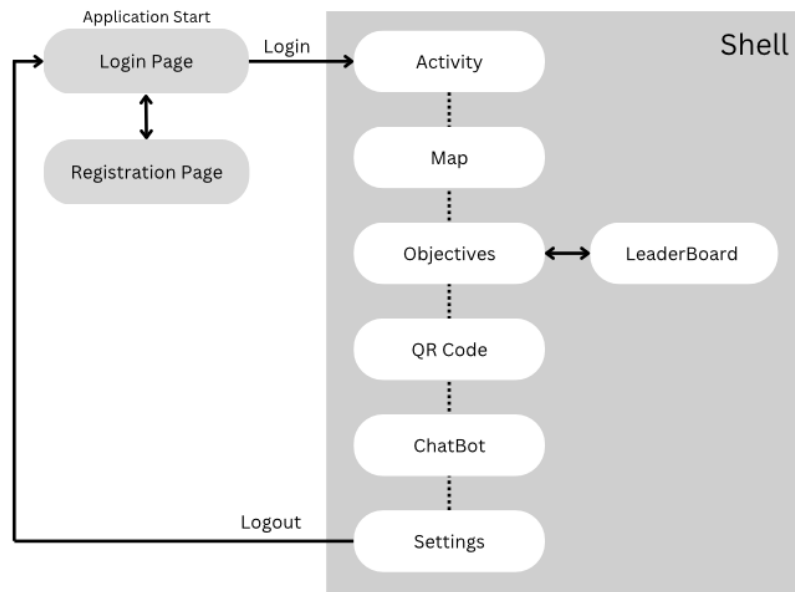


Figure 5.15: Application UI Flux Diagram.

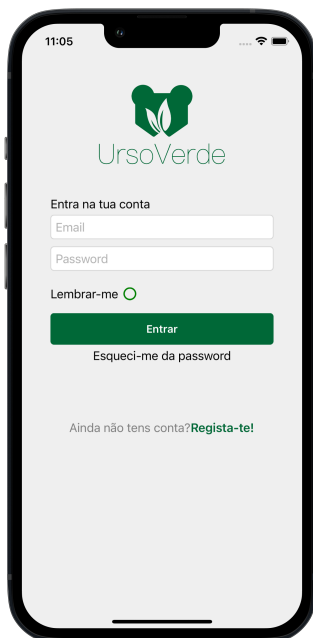
As illustrated in the diagram:

1. The application starts on the Login Page. From there, the user has the option to navigate to the Registration Page and vice versa.
2. Upon successful login, the user enters the application’s Shell, in which all the main features pages are accessible.
3. The default page within the Shell is the Activity Page. From there, users can freely navigate to the Map Page, Objectives Page, QR code Page, ChatBot Page and Setting Page.
4. The Leaderboard Page can be accessed through the Objectives Page.
5. The Settings Page offers an option to Logout. Selecting this will log out the user and redirect them back to the Login Page. Leaving the Shell of the application.

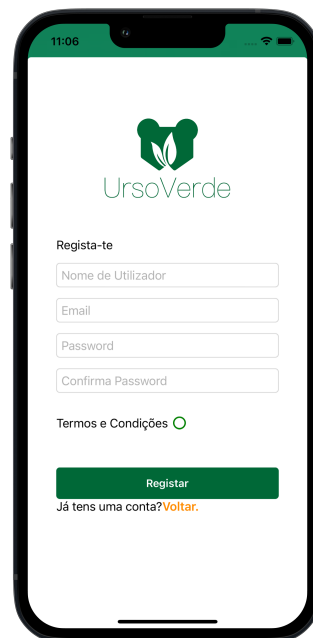
### 5.5.3 Login and Registration Page

The Login Page, shown in figure 5.16a, is the initial page that users encounter when opening the app. It includes two fields where users can input their email and password. The password field is of the 'password' type, meaning that the text input will appear as asterisks (\*) for privacy and security purposes. There is a 'Remember Me' checkbox that, when checked, saves the users' login details for future sessions. Also, there is a 'Forgot my password' clickable label that users can use if they forgot their credentials. Additionally, there is a clickable 'Registration' label that navigates the user to the Registration Page.

The Registration Page, shown in figure 5.16b, allows for user account creation. This page includes four entry fields: 'Username', 'Email', 'Password' and a confirmation 'Password' field. Both 'Password' fields also hide characters for privacy and security reasons. A clickable 'Terms and Conditions' label directs users to the terms and conditions texts. To finish registration, the user needs to check a box confirming that they agree with these terms and conditions. Once all the necessary fields are filled out and the terms and conditions are agreed, the user can press the 'Register' button to create their account. A 'Return' clickable label is also provided to allow the return to the Login page.



(a) Login Page

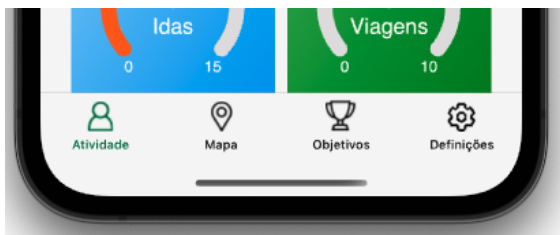


(b) Registration Page

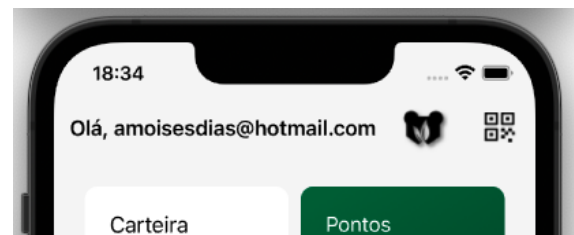
Figure 5.16: Login and Registration Pages

## 5.5.4 Shell

The Shell serves as the primary navigation container within the app. It allows us to define both a tab bar and a tool bar, which act as the main controls for navigating between the various pages of the app. As illustrated in Figure 5.17a, the Activity, Map, Objective and Settings pages are all accessible via the tab bar. The page currently being viewed is highlighted in green. The toolbar, as shown in figure 5.17b houses the more interactive pages of the app. These include the ChatBot page and the QR code reader.



(a) Tab bar



(b) Tool Bar

Figure 5.17: Application Shell Tab Bar and Tool Bar

## 5.5.5 Activity Page

The Activity page, displayed in figure 5.18 has been designed with a two-section layout, delivering a comprehensive overview of the user's activity and progress at a glance.

The first section, referred to as the 'Wallet', provides an up-to-date display of the user's Citizen Points. Here, the user can view their currently available points, the total number of points accumulated over time and the points they've already spent. There is also a button labeled 'Trocar' that, upon being pressed, redirects the user to the QR code Reader page.

The second section is dedicated to tracking and visualizing the user's activities. This area features four radial charts, each representing a different activity tracked by the app. The four activities are: the number of steps taken, usage of public transportation, visits to green zones and recycling bins.



Figure 5.18: Activity Page

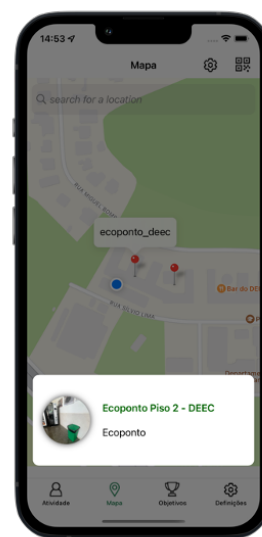
### 5.5.6 Map Page

The Map page, as shown in figure 5.19a, introduces several interactive features to help users locate areas where they can participate in sustainable activities promoted by UrsoVerde.

The map is marked with red pins, which represent sites where sustainable activities may be engaged. When you click one of these pins, a popup page will appear, as shown in figure 5.19b, showing a photograph of the area, a brief description of it and the type of sustainable activity it is linked with.



(a) Map Page



(b) Map Pin

Figure 5.19: Map page and pin interaction



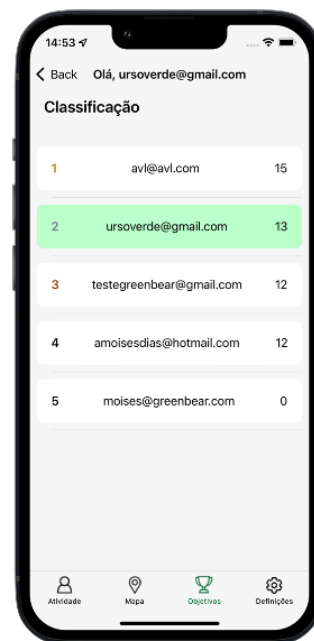
## 5.5.7 Objectives Page

The Objectives page, shown in figure 5.20a, is designed in two parts. In the 'Classification' section located at the top of the page, users can view their current Citizen Points balance, their standing on the LeaderBoard and access the LeaderBoard page via a button. The bottom half of the page is dedicated to listing the available objectives. Each objective is represented with a distinct icon that reflects the type of activity involved, followed by a short description and the goal users need to surpass to earn points. This goal is color-coded to reflect the user's progress: red indicates that the user hasn't started working towards the objective, yellow denotes that the user has initiated the task but hasn't achieved the goal yet and green signifies that the user has successfully completed the objective.

The LeaderBoard page, shown in figure 5.20b, is designed to display a list of users, ordered in descending order by the number of Citizen Points. Each entry includes the user's numerical position, username and the count of their available Citizen Points. To make it easy for the user to find their standing, their position is highlighted with a green tint. This page allows users to compare their standings with others.



(a) Objectives Page



(b) Leaderboard Page

Figure 5.20: Objectives and LeaderBoard Page

### 5.5.8 QR code Page

The QR code page shows the live feed from the smartphone camera. Right in the center of the screen, a square frame with a leaf symbol indicates where users should position the QR code they wish to scan.

At the bottom of the screen, there is an input field where users can specify the number of points they would like to receive. Adjacent to this field, there's a button that, when clicked, generates an unique QR code associated with the user's specified point value.



Figure 5.21: QR code Reader Page

### 5.5.9 ChatBot Page

The ChatBot page design follows typical chat applications. This page features a text input field located at the bottom of the screen for users to type their messages. Adjacent to this field is a 'send' button to submit the user's queries.

Above this, the page hosts a scrollable list that chronologically displays the conversation between the user and the ChatBot. Messages sent by the user appear on the right side, while responses from the ChatBot appear on the left. Each message is encapsulated within a distinct text bubble, helping differentiate between user inputs and ChatBot responses.

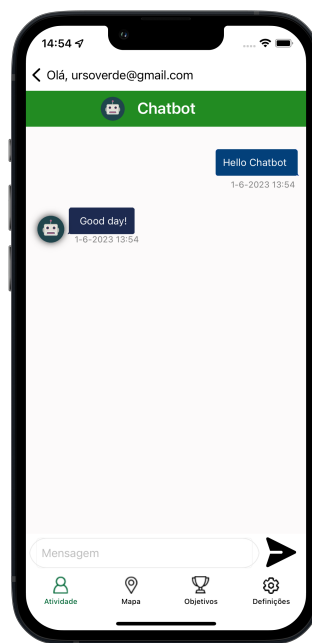


Figure 5.22: ChatBot Page

## 5.6 Tests

This section is dedicated to presenting the tests performed on the UrsoVerde application. Through these tests, we assured the functionality, performance and usability of the system.

### 5.6.1 Devices Used for Testing

During the testing process, we used a variety of devices to ensure that the application could function effectively across different platforms. This included both physical devices and emulators, described in table 5.1.

Table 5.1: Devices used during testing and their main specifications.

Device	Operating system	RAM	CPU	Battery
iPhone 11 (emulator)	iOS 16	6GB	-	-
Pixel 3a (emulator)	Android 12L	1.5GB	-	-
Samsung S9	Android 10	6GB	Exynos 9810	3500 mAh
iPhone 13	iOS 16.5.1	6GB	A15 Bionic	3095mAh

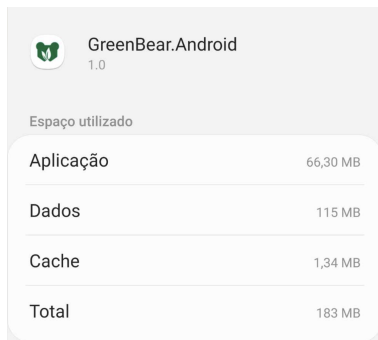
For the emulators, the iPhone 11 and Pixel 3a emulators, CPU and battery specifications are not provided. This is due to the fact that these emulators run on a computer and therefore use the computer’s resources, including the CPU and battery.

### 5.6.2 Performance Metrics

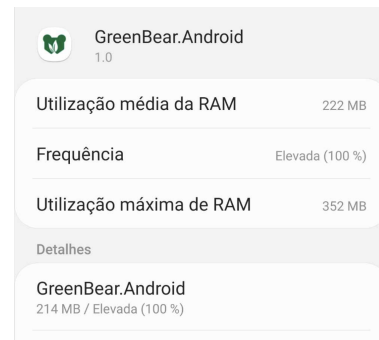
Some metrics were analysed in order to understand the impact of the application on the devices. However, the iOS operating system restricts the access to certain system-level metrics for third-party applications. Emulators also don’t provide real life metrics because they use the host computer resources. Therefore, we could only obtain specific metrics on the Samsung device.

We run the application during one hour of intensive use. The test showed satisfactory performance with the following results:

- **Storage:** Figure 5.23a shows the storage metrics. The application consumed a total of 183MB, distributed as follows:
  - **Application size:** 66.30MB. The main application size.
  - **Data:** 115MB. This includes every saved sensor data.
  - **Cache:** 1.34MB. Temporary data that enhances load times.
- **RAM:** Figure 5.23b shows the RAM usage metrics. The test demonstrated reasonable memory usage:
  - **Average memory used:** 222MB.
  - **Maximum memory usage:** 352MB.



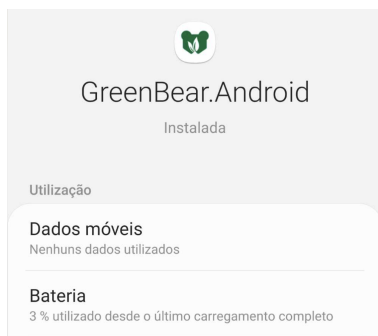
(a) Storage usage metrics



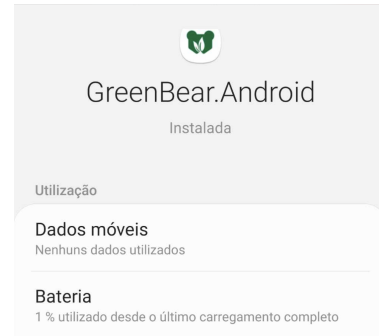
(b) RAM usage metrics

Figure 5.23: Test metrics on the Android device.

- **Battery:** Figure 5.24 shows the battery usage metrics. During an hour of intense testing, the application consumed 3% of the device’s battery. During twenty four hours of background usage, the application consumed 1% of the device’s battery. Indicating a low impact on battery life.



(a) Battery consumption in intensive usage for one hour.



(b) Battery consumption in background usage for twenty four hours.

Figure 5.24: Battery usage metrics on the Android device.

### 5.6.3 Real Scenarios

During a meeting with Coimbra City Hall, we demonstrated the UrsoVerde application in a real-life scenario simulation. The guests tested the program and offered excellent comments, with the application proving to be intuitive and stable, with no evidence of crashes or slowdowns. While UrsoVerde’s functionality was carefully tested throughout development, we recognize that certain defects may only become evident when the application is tested by a wider user base in real-life settings.

# 6 Engenheiros Sem Fronteiras

During the development of the UrsoVerde platform, we concurrently worked on two other significant projects: Engenheiros Pelo Mundo (EPM) and Engenheiros Sem Fronteiras (ESF). These projects were both developed in collaboration with the Ordem dos Engenheiros da Região Centro (OERC) [128].

The EPM, which already existed before our current project, assists engineers intending to work in foreign countries by connecting them with engineers already employed in those locations. The ESF platform is being developed to promote voluntary work among engineers and engineering students. We aim to combine the two platforms into a single one as part of our ongoing strategy. This integrated platform will incorporate the UrsoVerde Citizen Points as a reward mechanism for voluntary work.

In the following sections we will explore each one of them in more detail.

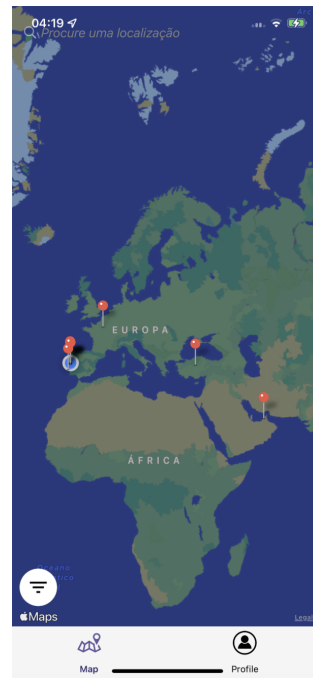
## 6.1 Engenheiros Pelo Mundo

EPM is a mobile application, displayed in figure 6.1, developed using Xamarin, designed to enable engineers affiliated with the OERC to share their location in foreign countries. The purpose of this is to assist newer engineers intending to work overseas by enabling them to connect with Portuguese engineers already living there.

The application's structure is straightforward, comprising a login and registration page that prompts the user to input their Ordem dos Engenheiros (OE) credentials. Then, the application provides a map page that displays the location and contact of registered users, as shown in figure 6.1b.



(a) EPM Login Page



(b) EPM Map Page

Figure 6.1: Login and Map pages of the EPM application.

My contribution to this project began with the investigation of particular problems related to iOS devices with smaller displays. I actively contributed to the creation of the first stable version of the application. My tasks included setting up and releasing the application on the Apple AppStore. I was able to have an in-depth understanding of the application's development and deployment procedures

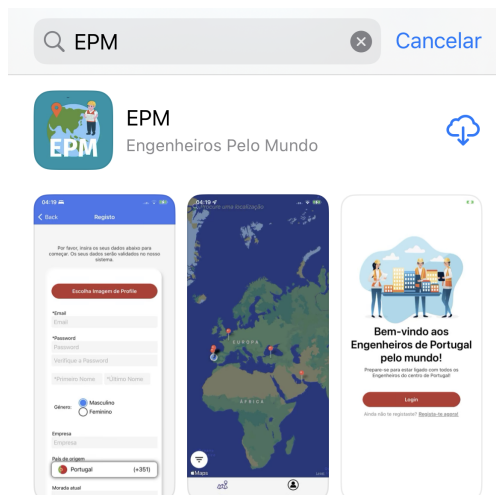


Figure 6.2: EPM application on the AppStore

## 6.2 Engenheiros Sem Fronteiras

ESF is a web and application platform created to bring together engineers and associations. It acts as a connecting link, in which engineers volunteer their talents and knowledge to address issues presented by associations.

The website is currently being developed with Blazor, a programming syntax for building dynamic web pages with C#. We are using Docker Containers. These containers encapsulate the website's environment, making it easier to manage, scale and deploy. To manage the data we are using SQL databases.

It is important to note that the ESF project is still under development. However, we have made significant planning and conceptualizing of the project. This includes creating mockups and a project presentation document to aid in envisioning the platform's final look and functionality [Appendix C].

### 6.2.1 Objectives

The platform's primary aim is to enable engineers, or engineering students, to address real-world engineering problems possessed by associations. Engineers can register their accounts as Problem Solvers (PS), while associations register as Problem Makers (PM). PMs can post problems on the platform, which are then made available to the PSs. PSs can link their UrsoVerde accounts to the Engenheiros Sem Fronteiras platform and upon resolving the problems, they receive Citizen Points as a reward.

### 6.2.2 Requirements and Functionalities

This subsection will look into the functional and non-functional requirements of the web application.

#### 6.2.2.1 Functional Requirements

The Functional Requirements will be delineated based on Table 4.2, as was done for the UrsoVerde application.

They are listed as depicted in table 6.1 below:



Table 6.1: ESF Functional requirements

FR	Name	FR	Name
FR0	Login	FR4	Apply to Problems
FR1	Register	FR5	Link UrsoVerde Account
FR2	Differentiate PS from PM	FR6	Match System
FR3	Post Problems	FR7	Connect to a Database

## Login

- **Priority:** Must.
- **Description:** The website initial screen should present users with fields to enter their username and password, along with buttons to log in and register.
- **Actors:** User.
- **Pre-conditions:** User has not previously selected the "Remember me" option during a successful login.
- **Event flow:** The user inputs their username and password, then presses the "Login" button. If the user doesn't have an account, they can navigate to a registration page by clicking the "Register" button.
- **Expected outcome:** If the credentials are correct, the user is redirected to the main page of the website. If they are not, an error message is displayed.

## Register

- **Priority:** Must.
- **Description:** The website should provide users with the ability to create a new account.
- **Actors:** User.
- **Pre-conditions:** User has pressed the "Register" button on the Login page.
- **Event flow:** After pressing the "Register" button on the Login page, the user is taken to the Registration page. The user must then enter a valid username and password and click the "Register" button on this page.

- **Expected outcome:** If the provided username and password are valid, the registration process is successful and the user is returned to the Login page. If registration fails, an error message is displayed. The new user must be accepted from an administrator on a dashboard.

## Logout

- **Priority:** Must.
- **Description:** The website should include a feature to end the user session.
- **Actors:** User.
- **Pre-conditions:** User is logged in.
- **Event flow:** User selects the Logout button from the main page.
- **Expected outcome:** The application redirects to the Login page, user credentials are cleared and the "Remember me" option is deselected.

## Differentiate PS from PM

- **Priority:** Must.
- **Description:** The application should distinguish PS users from PM Users.
- **Actors:** None.
- **Pre-conditions:** None.
- **Event flow:** None.
- **Expected outcome:** The website shows different pages with different functionalities depending on the type of user.

## Post Problems

- **Priority:** Must.
- **Description:** The website should allow PM to post a problem they want to get solved by PS.
- **Actors:** PM.

- **Pre-conditions:** PM is logged in.
- **Event flow:** PM write a form with the problem information.
- **Expected outcome:** The website posts the problem in a list of problems available to PS.

### Apply to Problems

- **Priority:** Must.
- **Description:** The website should allow PS to apply problem posted by PS.
- **Actors:** PS.
- **Pre-conditions:** PS is logged in.
- **Event flow:** PS applies to the desired problem.
- **Expected outcome:** The website associates the PS to the problem.

### Link UrsoVerde Account

- **Priority:** Should.
- **Description:** The website should allow PS to link their UrsoVerde account.
- **Actors:** PS.
- **Pre-conditions:** PS has an UrsoVerde Account.
- **Event flow:** PS enters his UrsoVerde Account ID in an appropriate entry for that.
- **Expected outcome:** The website associates the PS to UrsoVerde Account and adds Citizen Points to that account.

### Match System

- **Priority:** Must.
- **Description:** The website should have a mechanism that automatically attributes on PS to a problem based on the Engineering College.

- **Actors:** None.
- **Pre-conditions:** PS has the same Engineering College as the problem.
- **Event flow:** PS enters his Engineering College in an appropriate entry for that.
- **Expected outcome:** The website associates the PS to Problem and notifies both PS and PM.

### Connect to a DataBase

- **Priority:** Must.
- **Description:** The website must be able to communicate with a Database to store and retrieve data.
- **Actors:** None.
- **Pre-conditions:** None.
- **Event flow:** None.
- **Expected outcome:** The website stores and retrieves data every time it is needed.

### 6.2.2.2 Non-functional Requirements

Similarly to the UrsoVerde non-functional requirements, we will describe them based on 'Accessibility', 'Privacy', 'Support' and 'Usability'.

- **System Accessibility:** This is predominantly determined by the Razor Pages infrastructure used in the project rather than the website itself. It's critical to ensure that the system remains accessible regardless of the number of users accessing it simultaneously. This requirement assures uninterrupted service availability to all users.
- **Privacy:** Data protection is ensured through authentication mechanisms.
- **Broad Browser Support:** By using Razor Pages for our web development, we can ensure that our platform operates efficiently across all modern browsers.
- **Usability:** Creating a user-friendly interface for our website that delivers an aesthetically pleasing experience is one of our main goals.

### 6.2.3 Similar Applications

It was essential to understand the environment of similar apps when establishing ESF. These platforms offer information on best practices, design considerations and potential problems. Here are a few important examples of related applications:

- **Taproot Foundation:** The Taproot Foundation connects nonprofits and social change organizations with skilled volunteers who share their expertise pro Bono. Through their online marketplace, nonprofits seeking assistance are matched with volunteers with the appropriate skill set [129].
- **VolunteerMatch:** This web-based platform connects volunteers with nonprofit organizations. Like ESF, organizations can post their needs and volunteers can search for opportunities that match their skills and interests [130].
- **Catchafire:** Catchafire is a platform that connects skilled professionals looking to volunteer their services with nonprofit organizations in need of their skills. Just like ESF, they aim to drive social change by enabling talented professionals to share their skills with organizations that need them [131].

While all of these platforms provide excellent points of comparison and inspiration, it is vital to note that ESF fills a specific need in this environment. Unlike other volunteer platforms, ESF is intended exclusively for engineers and engineering students.

# 7 Conclusion

## 7.1 Final Results and Future Work

The UrsoVerde project aimed to create an application that could effectively monitor people's sustainable actions and encourage such behaviors through a reward-based system. This goal involved the investigation and use of a number of concepts, including HitLCPS, IoT and the use of Xamarin for mobile application development.

Our efforts produced a prototype application with an array of capabilities that suit the demands of any city. The application, which is designed as a cross-platform mobile app, can detect user activity using smartphone sensors, connect to the FIWARE back-end, act as a QR code reader and connect with IoT devices through BLE.

While the application is at a stage where it can be customized to meet the requirements of any city, project, or even particular organizations such as a school or a hospital, there is still work to be done and improvements to be made. These future works may include monitoring recycling bins for material collection planning, supporting healthy living through sleep tracking, facilitating volunteer work, encouraging visits to cultural spaces and participation in cultural events, inviting friends to install the app, promoting active citizenship through anomaly reporting, tracking healthy meal consumption and supporting animal adoption from shelters.

However, one issue that must be addressed soon in the future is the development of a dashboard website where the platform can be managed, security and anti-cheat methods in the app's reward systems. Several changes were made to the app throughout the development process, the most notable of which was the decision to abandon the usage of LoRaWAN in IoT devices.

The Coimbra City Hall has included this idea as a part of their application for the "Bairros Digitais Comerciais" program. Also, "Escola Jaime Cortesão", a public school

in Coimbra, will host a pilot test of the UrsoVerde application which is an important step towards real implementation.

Though we have made significant progress, further testing is necessary to improve the app's reliability and reduce possible concerns. As a result, we see our current position as a catapult for future generations of the UrsoVerde application that are more sophisticated and flexible.

Throughout the course of my dissertation, I also contributed to a couple of projects initiated by the Ordem dos Engenheiros:

**Engenheiros Pelo Mundo:** My contribution in this project was moderate, however it was helpful in improving my understanding of AppStore app publication processes. Furthermore, it provided me the opportunity to look deeper into Xamarin Projects, expanding my experience in app development.

**Engenheiros Sem Fronteiras:** This ongoing project shares similarities with mobile app development, but it has presented a different perspective on building digital platforms. The project's objective is to promote voluntary work among engineers and associations through web platforms. To create a robust platform, we have integrated web development with SQL databases, demonstrating the potential of such a combination.

## 7.2 Personal Considerations

On a personal note, I consider the work done in the last several months to be extremely important to my own growth and development. It has not only enabled me to further develop the abilities I gained during my academic journey, but it has also reiterated the notion that the knowledge we gain during our academic years acts as a strong foundation upon which we may acquire and apply new tools. This experience has been extremely gratifying, shaping my abilities in dealing with the world of app and web development.

## 8 Bibliography

- [1] Lionel Sujay Vailshery. Number of internet of things (iot) connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030. <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>. Accessed : 2023-06-19.
- [2] Evangelos Theodoridis, Georgios Mylonas, and Ioannis Chatzigiannakis. Developing an iot smart city framework. In *IISA 2013*, pages 1–6, 2013.
- [3] Andres Monzon. Smart cities concept and challenges: Bases for the assessment of smart city projects. In *2015 International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*, pages 1–11, 2015.
- [4] Mariluz Guenaga, Iratxe Mentxaka, Andoni Eguíluz, Susana Romero, and Javier García Zubía. Smartphones and teenagers, threat or opportunity. In *2012 15th International Conference on Interactive Collaborative Learning (ICL)*, pages 1–5, 2012.
- [5] Carmen Isensee, Frank Teuteberg, and Kai Michael Griese. Exploring the use of mobile apps for fostering sustainability-oriented corporate culture: A qualitative analysis. *Sustainability*, 14:7380, 6 2022.
- [6] Maren M. Michaelsen and Tobias Esch. Motivation and reward mechanisms in health behavior change processes. *Brain Research*, 1757:147309, 4 2021.
- [7] David Sousa Nunes, Pei Zhang, and Jorge Sá Silva. A survey on human-in-the-loop applications towards an internet of all. *IEEE Communications Surveys Tutorials*, 17(2):944–965, 2015.
- [8] Gerard Wagenaar, Sietse Overbeek, Garm Lucassen, Sjaak Brinkkemper, and Kurt Schneider. Working software over comprehensive documentation – rationales of agile teams for artefacts usage. *Journal of Software Engineering Research and Development*, 6(1), July 2018.



- [9] W.B. Frakes and Kyo Kang. Software reuse research: status and future. *IEEE Transactions on Software Engineering*, 31(7):529–536, July 2005.
- [10] IAPMEI. Bairros comerciais digitais. <https://www.iapmei.pt/Paginas/Bairros-Comerciais-Digitais.aspx>. Accessed : 2023-07-10.
- [11] Linda Steg and Charles Vlek. Encouraging pro-environmental behaviour: An integrative review and research agenda. *Journal of Environmental Psychology*, 29:309–317, 9 2009.
- [12] Google. Google play. <https://play.google.com/store/apps>. Accessed : 2023-06-16.
- [13] Benjamin Brauer, Björn Hildebrandt, and Gerrit Remane. Green by app: The contribution of mobile applications to environmental sustainability large-scale agile software development view project, 2016.
- [14] Chiara Bresciani, Alberto Colorni, Federico Lia, Alessandro Luè, and Roberto Nocerino. Behavioral change and social innovation through reward: An integrated engagement system for personal mobility, urban logistics and housing efficiency. *Transportation Research Procedia*, 14:353–361, 2016.
- [15] Robert A. Rescorla. A pavlovian analysis of goal-directed behavior. *American Psychologist*, 42(2):119–129, February 1987.
- [16] Taewoo Nam and Theresa A. Pardo. Conceptualizing smart city with dimensions of technology, people, and institutions. In *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times*, dg.o '11, page 282–291, New York, NY, USA, 2011. Association for Computing Machinery.
- [17] Abbas Shah Syed, Daniel Sierra-Sosa, Anup Kumar, and Adel Elmaghraby. Iot in smart cities: A survey of technologies, practices and challenges. *Smart Cities*, 4(2):429–475, 2021.
- [18] Zaheer Khan, Ashiq Anjum, Kamran Soomro, and Muhammad Atif Tahir. Towards cloud based big data analytics for smart future cities. *Journal of Cloud Computing*, 4:2, 12 2015.
- [19] Yun Wang, Sudha Ram, Faiz Currim, Ezequiel Dantas, and Luiz Alberto Sabóia. A big data approach for smart transportation management on bus network. In *2016 IEEE International Smart Cities Conference (ISC2)*, pages 1–6, 2016.

- [20] Anis Koubaa, Abdulrahman Aldawood, Bassel Saeed, Abdullatif Hadid, Mohamed Ahmed, Abdulrahman Saad, Hesham Alkhouja, Adel Ammar, and Mohamed Alkanhal. Smart palm: An IoT framework for red palm weevil early detection. *Agronomy*, 10(7):987, July 2020.
- [21] M.J. O'Grady, D. Langton, and G.M.P. O'Hare. Edge computing: A tractable model for smart agriculture? *Artificial Intelligence in Agriculture*, 3:42–51, September 2019.
- [22] Elham Shirazi and Shahram Jadid. Autonomous self-healing in smart distribution grids using agent systems. *IEEE Transactions on Industrial Informatics*, 15(12):6291–6301, 2019.
- [23] Sergey Konovalov, Denis Mayorov, Yuri Ponomarev, and Tatiana Soloveva. High-precision smart system on accelerometers and inclinometers for structural health monitoring: development and applications. In *2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics*, pages 52–57, 2018.
- [24] Siham G. Farag. Application of smart structural system for smart sustainable cities. In *2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)*, pages 1–5, 2019.
- [25] Kellow Pardini, Joel J. P. C. Rodrigues, Sergei A. Kozlov, Neeraj Kumar, and Vasco Furtado. IoT-based solid waste management solutions: A survey. *Journal of Sensor and Actuator Networks*, 8(1):5, January 2019.
- [26] Izabela Rojek and Jan Studzinski. Detection and localization of water leaks in water nets supported by an ICT system with artificial intelligence methods as a way forward for smart cities. *Sustainability*, 11(2):518, January 2019.
- [27] Joy Dutta, Chandreyee Chowdhury, Sarbani Roy, Asif Iqbal Middy, and Firoj Gazi. Towards smart city. In *Proceedings of the 18th International Conference on Distributed Computing and Networking*. ACM, January 2017.
- [28] Rodrigo Varejão Andreão, Matheus Athayde, Jérôme Boudy, Paulo Aguilar, Italo de Araujo, and Rossana Andrade. Raspcare: a telemedicine platform for the treatment and monitoring of patients with chronic diseases. In *Assistive Technologies in Smart Cities*. IntechOpen London, UK, 2018.
- [29] Pearse A. Keane and Eric J. Topol. With an eye to AI and autonomous diagnosis. *npj Digital Medicine*, 1(1), August 2018.

- [30] Boudewijn R. Haverkort and Armin Zimmermann. Smart industry: How ICT will change the game! *IEEE Internet Computing*, 21(1):8–10, January 2017.
- [31] Fei Tao, Jiangfeng Cheng, and Qinglin Qi. Iihub: An industrial internet-of-things hub toward smart manufacturing based on cyber-physical system. *IEEE Transactions on Industrial Informatics*, 14(5):2271–2280, 2018.
- [32] Panagiotis Trakadas, Pieter Simoens, Panagiotis Gkonis, Lambros Sarakis, Angelos Angelopoulos, Alfonso P. Ramallo-González, Antonio Skarmeta, Christos Trochoutsos, Daniel Calv, Tomas Pariente, Keshav Chintamani, Izaskun Fernandez, Aitor Arnaiz Irigaray, Josiane Xavier Parreira, Pierluigi Petrali, Nelly Leligou, and Panagiotis Karkazis. An artificial intelligence-based collaboration approach in industrial IoT manufacturing: Key concepts, architectural extensions and potential applications. *Sensors*, 20(19):5480, September 2020.
- [33] Toh Mun Heng and Linda Low. The intelligent city: singapore achieving the next lap. *Technology Analysis & Strategic Management*, 5(2):187–202, January 1993.
- [34] Martin Bauer, Luis Sanchez, and JaeSeung Song. IoT-enabled smart cities: Evolution and outlook. *Sensors*, 21(13):4511, June 2021.
- [35] SmartSantander. Smartsantander. <https://www.smartsantander.eu>. Accessed : 2023-06-21.
- [36] FIWARE. Fiware. <https://www.fiware.org/about-us/>. Accessed : 2023-06-21.
- [37] Jose A. Galache, Juan R. Santana, Veronica Gutierrez, Luis Sanchez, Pablo Sotres, and Luis Munoz. Towards experimentation-service duality within a smart city scenario. In *2012 9th Annual Conference on Wireless On-Demand Network Systems and Services (WONS)*. IEEE, January 2012.
- [38] Flavio Cirillo, Gurkan Solmaz, Everton Luis Berz, Martin Bauer, Bin Cheng, and Erno Kovacs. A standard-based open source IoT platform: FIWARE. *IEEE Internet of Things Magazine*, 2(3):12–18, September 2019.
- [39] Fiona Chang and Diganta Das. Smart nation singapore: Developing policies for a citizen-oriented smart city initiative. In *Developing National Urban Policies*, pages 425–440. Springer Nature Singapore, 2020.

- [40] SNClavalin. North avenue smart corridor. <https://www.snclavalin.com/en/beyond-engineering/north-avenue-smart-corridor>. Accessed : 2023-06-21.
- [41] SoundThinking. Make sound decisions that keep your community safe. <https://www.soundthinking.com>. Accessed : 2023-06-21.
- [42] ATL. Atl311 non-emergency city services. <https://www.atl311.com/311>. Accessed : 2023-06-21.
- [43] ATL. Notifyatl. <https://member.everbridge.net/3091830992273427/login>. Accessed : 2023-06-21.
- [44] CityIQ. About cityiq. <https://docs.cityiq.io>. Accessed : 2023-06-21.
- [45] Bee Smart City. Amsterdam smart city: A world leader in smart city development. <https://www.beesmart.city/city-portraits/smart-city-portrait-amsterdam>. Accessed : 2023-06-21.
- [46] Lauren Macpherson. 8 years on, amsterdam is still leading the way as a smart city. <https://towardsdatascience.com/8-years-on-amsterdam-is-still-leading-the-way-as-a-smart-city-79bd91c7ac13>. Accessed : 2023-06-21.
- [47] Innwave Technologies. Citypoints cascais. <https://cidadania20.com/projectos/citypoints-cascais/>. Accessed : 2023-06-21.
- [48] Câmara Municipal de Cascais. City points cascais. <https://www.cascais.pt/citypoints>. Accessed : 2023-06-21.
- [49] Ben Agger. itime: Labor and life in a smartphone era. *Time & Society*, 20(1):119–136, 2011.
- [50] Andrew Lim. Nikola tesla predicted mobile phones in 1909. [https://editorial.recombu.com/mobile/article/nikola-tesla-predicted-mobile-phones-in-1909\\_m11683-html](https://editorial.recombu.com/mobile/article/nikola-tesla-predicted-mobile-phones-in-1909_m11683-html). Accessed: 2023-06-19.
- [51] Keith Mallinson. Smartphone revolution: Technology patenting and licensing fosters innovation, market entry, and exceptional growth. *IEEE Consumer Electronics Magazine*, 4(2):60–66, 2015.
- [52] Yu-Kang Lee, Chun-Tuan Chang, You Lin, and Zhao-Hong Cheng. The dark side of smartphone usage: Psychological traits, compulsive behavior and technostress. *Computers in Human Behavior*, 31:373–383, 2014.

- [53] Barry Wellman and Lee Rainie. If romeo and juliet had mobile phones. *Mobile Media Communication*, 1:166–171, 01 2013.
- [54] Muhammad Sarwar and Tariq Soomro. Impact of smartphone’s on society. *European Journal of Scientific Research*, 98, 02 2013.
- [55] Mobile Phone Museum. Ibm - simon. <https://www.mobilephonemuseum.com/phone-detail/ibm-simon>. Accessed : 2023-06-19.
- [56] BlackBerry. Blackberry. <https://www.blackberry.com/us/en>. Accessed : 2023-06-19.
- [57] Apple. Apple. <https://www.apple.com>. Accessed : 2023-06-19.
- [58] Mashable. Revisiting the iphone launch keynote, 15 years on. <https://mashable.com/article/first-iphone-keynote-15-year-anniversary>. Accessed : 2023-06-19.
- [59] The Verge. See one of google’s first android build on the 2007 sooner phone (update). <https://www.theverge.com/2012/5/5/3001163/google-early-android-design-prototype-phone>. Accessed : 2023-06-19.
- [60] Jack Flynn. 20 vital smartphone usage statistics [2023]: Facts, data, and trends on mobile use in the u.s. <https://www.zippia.com/advice/smartphone-usage-statistics/>. Accessed : 2023-06-19.
- [61] Marktest. Target group index. <https://www.marktest.com/wap/a/grp/p~16.aspx#>. Accessed : 2023-06-19.
- [62] Marktest. Marktest. <https://www.marktest.com/wap/>. Accessed : 2023-06-19.
- [63] Marktest. 7,2 milhões de portugueses usam smartphone. <https://www.marktest.com/wap/a/n/id~2700.aspx>. Accessed : 2023-06-19.
- [64] Michael DeGusta. Are smart phones spreading faster than any technology in human history? <https://www.technologyreview.com/2012/05/09/186160/are-smart-phones-spreading-faster-than-any-technology-in-human-history/>. Accessed : 2023-06-19.
- [65] Katie Anderson. Getting acquainted with social networks and apps: Instagram’s instant appeal. *Library Hi Tech News*, 33:11–15, 05 2016.
- [66] Karen Rose, Scott Eldridge, and Lyman Chapin. *OCTOBER 2015 THE INTERNET OF THINGS: AN OVERVIEW Understanding the Issues and Challenges of a More Connected World*. 2015.

- [67] Mohammad Aazam, Marc St-Hilaire, Chung-Horng Lung, and Ioannis Lambadaris. Pre-fog: Iot trace based probabilistic resource estimation at fog. In *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 12–17, 2016.
- [68] Debasis Bandyopadhyay and Jaydip Sen. Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58:49–69, 5 2011.
- [69] P. Middleton, P. Kjeldsen, and J. Tully. Forecast: The internet of things worldwide 2013. Nov 2013.
- [70] Pycom. Micropython modules. <https://docs.pycom.io/firmwareapi/micropython/>. Accessed : 2023-06-26.
- [71] Oscar Torres Sanchez, José Marcelo Fernandes, André Rodrigues, Jorge Sá Silva, Fernando Boavida, Jorge Eduardo Rivadeneira, Afonso Viana De Lemos, and Duarte Raposo. Green bear-a lorawan-based human-in-the-loop case-study for sustainable cities.
- [72] Pycom. Fipy - pycom - five network development board for iot. <https://pycom.io/fipy-kickstarter-campaign/>. Accessed : 2023-06-26.
- [73] Yue Liu, Ju Yang, and Mingjun Liu. Recognition of qr code with mobile phones. In *2008 Chinese Control and Decision Conference*, pages 203–206, 2008.
- [74] Sumit Tiwari. An introduction to qr code technology. In *2016 International Conference on Information Technology (ICIT)*, pages 39–44, 2016.
- [75] Jacopo Tosi, Fabrizio Taffoni, Marco Santacatterina, Roberto Sannino, and Domenico Formica. Performance evaluation of bluetooth low energy: A systematic review. *Sensors*, 17(12):2898, December 2017.
- [76] Bluetooth. Bluetooth technology overview. <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>. Accessed : 2023-06-27.
- [77] Sebastian Bräuer, Anatolij Zubow, Sven Zehl, Mehran Roshandel, and Soroush Mashhadi-Sohi. On practical selective jamming of bluetooth low energy advertising. In *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–6, 2016.
- [78] Ngombo Armando, Duarte Raposo, Marcelo Fernandes, André Rodrigues, Jorge Sá Silva, and Fernando Boavida. Wsns in fiware – towards the devel-

- opment of people-centric applications. In Javier Bajo, Zita Vale, Kasper Hal- lenborg, Ana Paula Rocha, Philippe Mathieu, Pawel Pawlewski, Elena Del Val, Paulo Novais, Fernando Lopes, Nestor D. Duque Méndez, Vicente Julián, and Johan Holmgren, editors, *Highlights of Practical Applications of Cyber-Physical Multi-Agent Systems*, pages 445–456, Cham, 2017. Springer International Pub- lishing.
- [79] FIWARE. Fiware catalogue. <https://github.com/FIWARE/catalogue/>. Ac- cessed : 2023-06-27.
- [80] Firebase. Firebase cloud messaging. [https://firebase.google.com/docs/ cloud-messaging](https://firebase.google.com/docs/cloud-messaging). Accessed : 2023-06-27.
- [81] Firebase. About fcm messages. [https://firebase.google.com/docs/ cloud-messaging/concept-options#notifications\\_and\\_data\\_messages](https://firebase.google.com/docs/cloud-messaging/concept-options#notifications_and_data_messages). Accessed : 2023-06-27.
- [82] Firebase. Fcm architectural overview. [https://firebase.google.com/docs/ cloud-messaging/fcm-architecture](https://firebase.google.com/docs/cloud-messaging/fcm-architecture). Accessed : 2023-06-27.
- [83] D. Spinellis. Version control systems. *IEEE Software*, 22(5):108–109, 2005.
- [84] GitLab. About gitlab. <https://about.gitlab.com>. Accessed : 2023-06-28.
- [85] Google Cloud. Dialogflow. <https://cloud.google.com/dialogflow/docs>. Ac- cessed : 2023-07-05.
- [86] Microsoft. Microsoft xamarin. [https://dotnet.microsoft.com/en-us/apps/ xamarin](https://dotnet.microsoft.com/en-us/apps/xamarin). Accessed : 2023-06-16.
- [87] Kumar Vishal and Ajay Shriram Kushwaha. Mobile application development research based on xamarin platform. In *2018 4th International Conference on Computing Sciences (ICCS)*, pages 115–118, 2018.
- [88] Jared Dickson. Xamarin mobile development, 12 2013.
- [89] Apache. Apache cordova. <https://cordova.apache.org>. Accessed : 2023-06- 16.
- [90] Axway. Appcelerator titanium. [https://www.axway.com/en/ appcelerator-end-of-life](https://www.axway.com/en/appcelerator-end-of-life). Accessed : 2023-06-16.
- [91] Xiaoping Jia, Aline Ebone, and Yongshan Tan. A performance evaluation of cross-platform mobile application development approaches. pages 92–93. ACM, 5 2018.

- [92] Microsoft. What is .net maui? <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui>. Accessed : 2023-07-10.
- [93] Microsoft. What is .net? <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>. Accessed : 2023-07-06.
- [94] Microsoft. Asp.net. <https://dotnet.microsoft.com/en-us/apps/aspnet>. Accessed : 2023-07-06.
- [95] Microsoft. Asp.net core blazor. [https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-7.0&WT.mc\\_id=dotnet-35129-website](https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-7.0&WT.mc_id=dotnet-35129-website). Accessed : 2023-07-06.
- [96] IBM. What is docker? <https://www.ibm.com/topics/docker>. Accessed : 2023-07-06.
- [97] Docker. Use containers to build, share and run your applications. <https://www.docker.com/resources/what-container/>. Accessed : 2023-07-06.
- [98] AWS. What is sql (structured query language)? [https://aws.amazon.com/what-is/sql/?nc1=h\\_ls](https://aws.amazon.com/what-is/sql/?nc1=h_ls). Accessed : 2023-07-06.
- [99] Ryan Barone. What are libraries in programming? <https://www.idtech.com/blog/what-are-libraries-in-coding>. Accessed : 2023-07-05.
- [100] Microsoft. An intruduction to nuget. <https://learn.microsoft.com/en-us/nuget/what-is-nuget>. Accessed : 2023-07-05.
- [101] nuget. Restsharp. <https://www.nuget.org/packages/RestSharp>. Accessed : 2023-07-05.
- [102] nuget. Xamarin.essentials. <https://www.nuget.org/packages/Xamarin.Essentials>. Accessed : 2023-07-05.
- [103] nuget. Xamarin.firebase.ios.cloudmessaging. <https://www.nuget.org/packages/Xamarin.Firebase.iOS.CloudMessaging>. Accessed : 2023-07-05.
- [104] nuget. Xamarin.firebase.messaging. <https://www.nuget.org/packages/Xamarin.Firebase.Messaging>. Accessed : 2023-07-05.
- [105] nuget. Com.airbnb.xamarin.forms.lottie. <https://www.nuget.org/packages/Com.Airbnb.Xamarin.Forms.Lottie>. Accessed : 2023-07-05.
- [106] nuget. Identitymodel. <https://www.nuget.org/packages/IdentityModel>. Accessed : 2023-07-05.



- [107] nuget. Microcharts.forms. <https://www.nuget.org/packages/Microcharts.Forms>. Accessed : 2023-07-05.
- [108] nuget. Newtonsoft.json. <https://www.nuget.org/packages/Newtonsoft.Json>. Accessed : 2023-07-05.
- [109] nuget. Ngsibasemodel. <https://www.nuget.org/packages/NGSIBaseModel>. Accessed : 2023-07-05.
- [110] nuget. Plugin.bluetoothle. <https://www.nuget.org/packages/Plugin.BluetoothLE>. Accessed : 2023-07-05.
- [111] nuget. Rg.plugins.popup. <https://www.nuget.org/packages/Rg.Plugins.Popup>. Accessed : 2023-07-05.
- [112] nuget. Xamarin.forms. <https://www.nuget.org/packages/Xamarin.Forms>. Accessed : 2023-07-05.
- [113] nuget. Xamarin.forms.maps. <https://www.nuget.org/packages/Xamarin.Forms.Maps>. Accessed : 2023-07-05.
- [114] nuget. Zxing.net.mobile.forms. <https://www.nuget.org/packages/ZXing.Net.Mobile.Forms>. Accessed : 2023-07-05.
- [115] Câmara Municipal de Coimbra. Coimbracitylab: Acolhemos ideias, criamos futuros! <https://www.cm-coimbra.pt/areas/viver/smartcity/coimbracitylab>. Accessed : 2023-06-28.
- [116] Martin Glinz. On non-functional requirements. In *15th IEEE International Requirements Engineering Conference (RE 2007)*, pages 21–26, 2007.
- [117] altexsoft software rd engineering. Non-functional requirements: Examples, types, how to approach. <https://www.altexsoft.com/blog/non-functional-requirements/>. Accessed : 2023-06-30.
- [118] altexsoft software rd engineering. Performance comparison: Xamarin.forms, xamarin.ios, xamarin.android vs android and ios native applications. <https://www.altexsoft.com/blog/engineering/performance-comparison-xamarin-forms-xamarin-ios-xamarin-android-vs-android>. Accessed : 2023-06-30.
- [119] Microsoft. Xamarin.forms supported platforms. <https://learn.microsoft.com/en-us/xamarin/get-started/supported-platforms?tabs=macos>. Accessed : 2023-06-30.

- [120] Microsoft. What is xamarin? <https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>. Accessed : 2023-07-02.
- [121] JetBrains. Rider - fast & powerful cross-platform .net ide. <https://www.jetbrains.com/rider/>. Accessed : 2023-07-02.
- [122] Microsoft. Xamarin.forms shell. <https://learn.microsoft.com/en-us/xamarin/xamarin-forms/app-fundamentals/shell/>. Accessed : 2023-07-02.
- [123] Microsoft. Async support overview. <https://learn.microsoft.com/en-us/xamarin/cross-platform/platform/async>. Accessed : 2023-07-02.
- [124] Codecademy. What is rest? <https://www.codecademy.com/article/what-is-rest>. Accessed : 2023-07-02.
- [125] Newtonsoft. JsonConvert class. [https://www.newtonsoft.com/json/help/html/t\\_newtonsoft\\_json\\_jsonconvert.htm](https://www.newtonsoft.com/json/help/html/t_newtonsoft_json_jsonconvert.htm). Accessed : 2023-07-02.
- [126] ZXing. Zxing - "zebra crossing" barcode scanning library for java, android. <https://github.com/zxing/zxing>. Accessed : 2023-07-03.
- [127] Microsoft. Xamarin.forms map. <https://learn.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/map/>. Accessed : 2023-07-04.
- [128] Ordem dos Engenheiros. Ordem dos engenheiros - região centro. <https://www.ordemengenheiros.pt/pt/contactos/regiao-centro/>. Accessed : 2023-07-05.
- [129] Taproot. We are connectors. <https://taprootfoundation.org>. Accessed : 2023-07-05.
- [130] VolunteerMatch. Volunteermatch. <https://www.volunteermatch.org>. Accessed : 2023-07-05.
- [131] Catchafire. Where talent meets purpose. <https://www.catchafire.org>. Accessed : 2023-07-05.
- [132] Gerard Wagenaar, Sietse Overbeek, Garm Lucassen, Sjaak Brinkkemper, and Kurt Schneider. Working software over comprehensive documentation – rationales of agile teams for artefacts usage. *Journal of Software Engineering Research and Development 2018 6:1*, 6:1–23, 7 2018.
- [133] William B. Frakes and Kyo Kang. Software reuse research: Status and future. *IEEE Transactions on Software Engineering*, 31:529–536, 7 2005.

- [134] Ruowei Xiao, Zhanwei Wu, and Juho Hamari. Internet-of-gamification: A review of literature on iot-enabled gamification for user engagement. *International Journal of Human-Computer Interaction*, 38:1113–1137, 2022.

## Appendix A

UrsoVerde - Guião para demonstração



# Urso Verde

URSOVERDE  
2023

---

## Guião para demonstração

---

António Moisés Tomás Dias  
Junho de 2023

Neste guião vamos explorar a aplicação UrsoVerde e as suas funcionalidades principais, oferecendo uma visão prática de como a aplicação pode beneficiar os utilizadores ao promover práticas sustentáveis e recompensar essas mesmas ações. Durante a demonstração, iremos testar, passos a passo, as principais funcionalidades do UrsoVerde, desde a criação de conta, cumprir objetivos e a transferência de pontos entre contas.

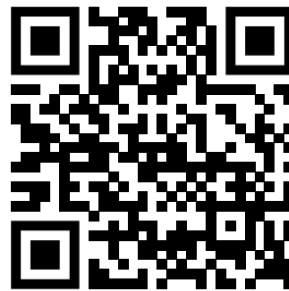
## 1 Preparação

Antes de iniciarmos a demonstração deveremos garantir que as aplicações estão devidamente instaladas em pelo menos dois dispositivos, um Android e um iOS. Poderemos instalar a aplicação em configuração *Debug* recorrendo a um computador com o código fonte.

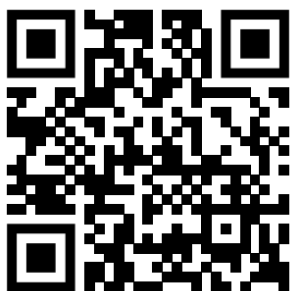
Deveremos dispor o dispositivo externo, devidamente carregado, em conjunto com três códigos QR, para simular uma zona verde, um autocarro e um ecoponto.



(a) Dispositivo externo



(b) Código QR para ecoponto



(c) Código QR para zona verde



(d) Código QR para autocarro

Figure 1: Material necessário para a demonstração

Depois de assegurarmos que tudo está devidamente colocado ao dispor dos utilizadores devemos prosseguir com a apresentação.

## 2 Criação de Conta

Agora, podemos optar por criar uma nova conta dentro da aplicação ou utilizar uma conta já existente. No caso de preferir usar uma conta existente poderá utilizar as seguintes credenciais:

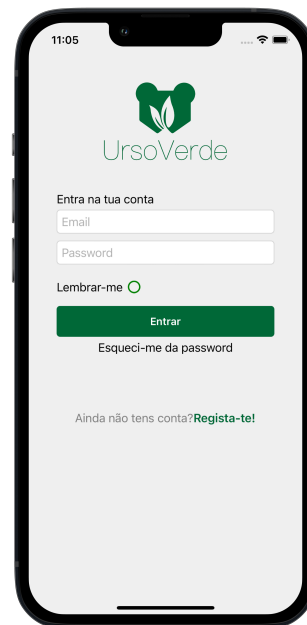
Username	Password
ursoverde@gmail.com	ursoverde
greenbear@gmail.com	greenbear

Para criar uma conta basta seguir as etapas de registo fornecidas pela aplicação. Insira as informações necessárias, como nome de utilizador, endereço de email e senha. Aceite os "Termos e Condições" e carregue em "Criar Conta". Em seguida, volte à pagina de Login para inserir as credenciais da conta que acabou de criar.

## 3 Explorar a aplicação

Agora que entramos na aplicação com as novas credenciais, podemos observar que os dados da atividade estão todos a zero.

Ao navegarmos pela página de atividade, podemos clicar em cada uma das atividades listadas para obter mais informações sobre como funciona cada uma delas.



Durante a demonstração desta página, é importante destacar a arquitetura da aplicação. Na parte superior da interface, temos acesso ao leitor de QRCode, que permite validar ações usando códigos QR associados a pontos específicos. Também temos acesso a um Chatbot, que, no futuro, poderá fornecer suporte e informações adicionais aos utilizadores.

Na parte inferior da aplicação, encontramos os ícones que nos direcionam para as outras páginas principais, incluindo a página de atividade, o mapa, os objetivos e as opções.

Encorajamos o utilizador a explorar todas as páginas de forma livre, experimentando as diferentes funcionalidades e interagindo com a aplicação para obter uma experiência completa do seu funcionamento.

## 4 Adicionar um objetivo

Deveremos pedir ao utilizador que coloque a aplicação em segundo plano para verificar o sistema de notificações. Em seguida, utilizando o *Postman*, deveremos adicionar um novo objetivo.

Cada objetivo tem seis parâmetros obrigatórios:

1. `id` - o `id` é único para cada objetivo e deverá ser sempre diferente de qualquer objetivo existente, caso contrário estaremos a atualizar um objetivo que já existe no Orion.
2. `type` - o parâmetro "type" define o tipo da entidade no Orion. Para o caso dos objetivos deverá ter sempre o valor "challenge".
3. `challenge_disc` - representa a descrição do objetivo. Deverá ser uma descrição sucinta, por exemplo "Idas a ecopontos".
4. `challenge_objective` - define a meta do objetivo que quando cumprida adiciona um ponto ao utilizador.
5. `challenge_type` - define o tipo de objetivo que estamos a inserir, pode ser um objetivo relacionado com ir a ecopontos ("ecoponto"), andar de autocarro ("public\_transport"), passar tempo em zonas verdes ("green\_space") ou um número de passos que utilizador deverá cumprir ("walk").
6. `expiration_time` - Neste campo definimos a data limite do objetivo. Após esta data o objetivo não será mais listado na aplicação.



Para efeitos de demonstração a meta deverá ser pequena para ser mais fácil de completar. Para tal deveremos fazer um Post Request para o *url* <https://socialiteorion2.dei.uc.pt:9012/v2/op/update>, com o seguinte *body* em formato *JSON*:

```
{
  "actionType": "APPEND",
  "entities": [
    {
      "id": "5",
      "type": "challenge",
      "challenge_disc": {
        "type": "Text",
        "value": "Idas a ecopontos",
        "metadata": {}
      },
      "challenge_objective": {
        "type": "Text",
        "value": "2",
        "metadata": {}
      }
    }
  ]
}
```



```
    },  
    "challenge_type": {  
      "type": "Text",  
      "value": "ecoponto",  
      "metadata": {}  
    },  
    "expiration_time": {  
      "type": "Text",  
      "value": "2023-07-31T00:00:00Z",  
      "metadata": {}  
    }  
  }  
]  
}
```

Depois de adicionarmos o novo objetivo deveremos verificar que uma notificação foi recebida no *smartphone* e podemos novamente abrir a aplicação. Deveremos dirigir o utilizador para a página dos objetivos onde irá encontrar o novo objetivo que acabamos de criar.

## 5 Obter pontos

Para obter pontos, o utilizador pode utilizar o botão da caixa na aplicação para sinalizar o uso do ecoponto e incrementar o valor no objetivo correspondente. No entanto, é importante ter em mente que a aplicação possui um intervalo de espera de 1 minuto entre as ações com os dispositivos externos. Durante esse período de espera de 1 minuto, a próxima ação relacionada ao ecoponto deverá ser validada utilizando o leitor de QRCode. Para isso, solicitamos ao utilizador que utilize o scanner na aplicação para ler o código QR associado ao ecoponto.

Agora, por ter cumprido o objetivo definido anteriormente o utilizador deverá ter 1 ponto disponível na aplicação.

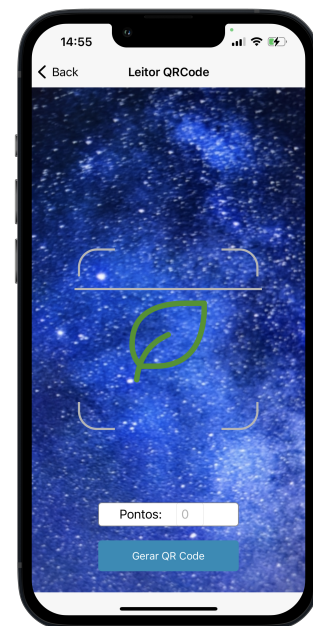
## 6 Trocar Pontos

Para testarmos a funcionalidade de troca de pontos entre contas e demonstrar como essa interação acontece na aplicação, vamos simular uma situação em que um utilizador deseja transferir 1 ponto para outra conta.

Primeiro, geramos um QRCode com o valor de 1 ponto associado a uma conta específica, diferente daquela que estamos a utilizar para a demonstração. Em seguida, pedimos ao utilizador que utilize o scanner de QRCode da aplicação para ler o código gerado.

Ao ler o QRCode, a aplicação reconhecerá a informação contida nele e realizará a transferência dos pontos especificados para a conta designada.

Ao completar a leitura do QRCode, veremos o saldo de pontos da conta de destino ser incrementado em 1 ponto, enquanto o saldo de pontos da conta de origem será reduzido pelo mesmo valor.



## 7 Conclusão

Após as demonstrações, é fundamental abrir espaço para discussões e sugestões, onde todos os participantes têm a oportunidade de compartilhar suas ideias, contribuições, opiniões e dúvidas.

# Appendix B

## Confluence: An Integration Model for Human-in-the-Loop IoT Privacy-Preserving Solutions Towards Sustainability in a Smart City

Green Bear is a service included in the pool of HiTL-IoT services that the IoT Gateway's implementation offers and its aim is to promote sustainable living habits among citizens namely encouraging recycling, the use of bicycles, using public transportation, and participating in voluntary actions. To develop this service, we identified functional requirements by considering the target audience (in our case, citizens) and the desired features. This service encompasses several features and components, such as activity tracking, objectives displaying, a chatbot, a QR code validator, and a Bluetooth low-energy (BLE) advertiser. To encourage participation, the service implements a point-based reward system called Citizen Points based on our proposed incentive mechanism.

For implementing this service, we have followed an agile methodology, particularly the principle of "working software over comprehensive documentation" [132] and the practice of "code reuse" [133], this involved employing a code reuse approach, which entails adapting previous code to enhance the efficiency and efficacy of the development process.

One of the services' capabilities is to communicate with public IoT resources through

a BLE advertiser to validate user actions. For instance, when a citizen performs a recycling task and interacts with a public IoT resource (e.g., pressing a button), this resource is capable of fetching the advertised user pseudo ID from the IoT Gateway to validate the interaction. Later, the collected data are sent to the IoT orchestrator for the Citizen Point assignment.

However, there is another validation method supported by this service. Through the QR validator, the actions of the citizens can be verified instead of interacting directly with the hardware of IoT resources. For example, after completing a task, the IoT resource can generate a QR code, and a citizen leveraging this smartphone camera can read the QR code to validate the action.

To provide the citizens with a friendly interaction interface, this service implements three main views: Activity, Chatbot, and Objectives (Figure B.1). The first view displays charts with user activity metrics, such as steps taken, time spent in green zones, and the number of Citizen Points (accumulated and spent).

The second view retrieves the objectives and shows the user's Citizen Points, as well as their classification. Each objective is color-coded to show the progress, with red indicating that the objective is far from completion, yellow indicating that it is almost complete, and green indicating that the objective is complete. In addition to the weekly objectives, there are monthly objectives. These tasks are typically difficult to complete and may require more effort from citizens, such as through voluntary work. However, they offer more points and can be a great way to earn rewards through the Citizen Points system.

The Chatbot view provides users with a convenient and user-friendly way to ask questions and receive feedback regarding their activities. Using Dialogflow<sup>1</sup>, the Chatbot can analyze citizen intents and provide accurate and relevant answers to their queries. This is achieved through machine learning algorithms that enable the Chatbot to recognize and interpret natural language inputs from users, allowing for more natural and conversational interaction. In the future, the integration of ChatGPT with Dialogflow could further enhance the user experience by allowing for more complex and engaging conversations.

Green bear services play a crucial role as catalysts for human-system interaction, necessitating the design of a citizen participation platform. This platform supports sus-

---

<sup>1</sup>Dialogflow: <https://cloud.google.com/dialogflow/docs?hl=en-en>

tainable initiatives, enabling individuals to provide real-time feedback on system usage and its proposed activities. Furthermore, Human-in-the-Loop (HITL) systems, such as the one proposed, enable interventions to promote sustainable behavior among citizens. These interventions specifically aim to encourage sustainable practices related to outdoor activities and recycling. Also, implementing and integrating the Green Bear service represents a significant milestone in developing the IoT Gateway to promote sustainable living habits. Finally, the incentive systems within the proposed framework aim to facilitate the generation of green businesses and investments by local companies, leveraging citizen points as a mechanism. By incorporating a transparent scheme, the platform minimizes investment risks for companies, thereby supporting the transition towards sustainable habits within the city. Furthermore, as observed in the reviewed literature [134], gamification systems have effectively motivated user participation, leading to higher acceptance rates and improved performance of systems such as CONFLUENCE.

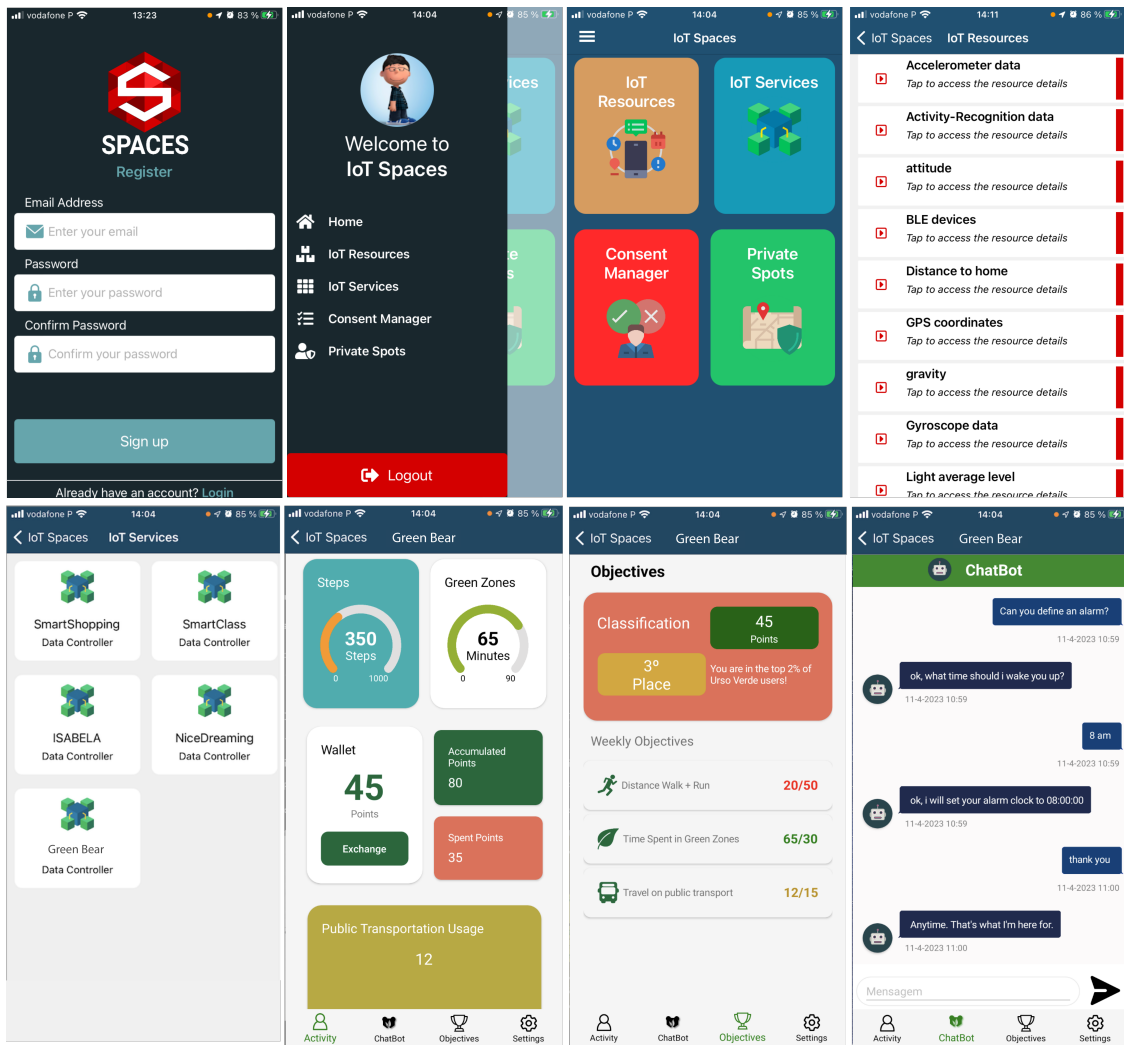
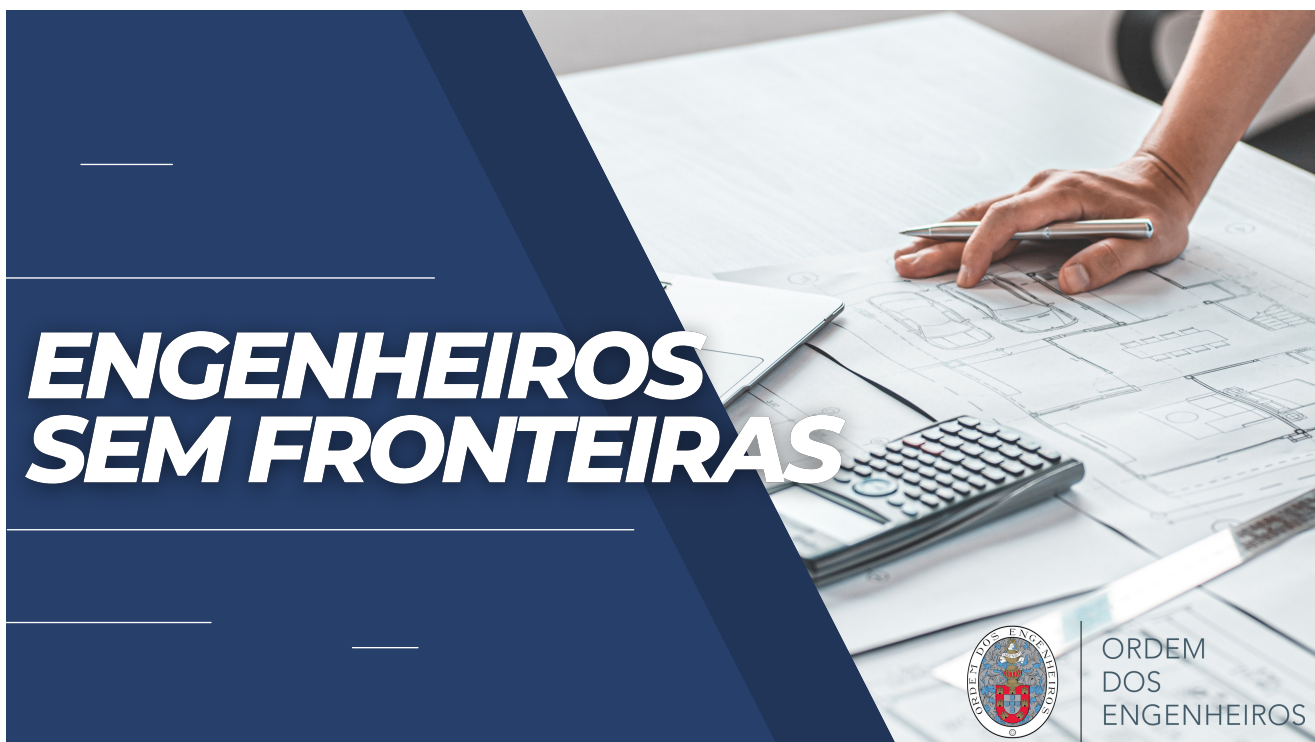


Figure B.1: IoT Spaces Mobile Application as part of the IoT Gateway component including the Green Bear service views.

## Appendix C

# Engenheiros Sem Fronteiras Presenta- tion



# ENGENHEIROS SEM FRONTEIRAS



ORDEM  
DOS  
ENGENHEIROS

## ESF

Engenheiros Sem Fronteiras é uma plataforma que liga engenheiros a problemas de engenharia apresentados por associações.

### Problem-Makers

Associações  
Ações de voluntariado  
Comunidades

### Problem-Solvers

Engenheiros  
Estudantes de Engenharia



# MATCH



## Caritas

"Desenvolver um sistema de gestão de utentes"



## João

Engenheiro Informático  
SQL  
C#



## AGIR

"Projetar um sistema de escoamento de água da chuva"



## Marta

Engenheira Civil  
Urbanismo

# PÁGINA INICIAL

ENGENHEIROS SEM FRONTEIRAS

Sobre Contactos Colégios

Sign up Login

## Ligamos Associações a Engenheiros

Engenheiro Sem Fronteiras conecta engenheiros e estudantes com vontade de resolver problemas com associações que precisam das suas capacidades. Como **Problem Solver** podes navegar por uma lista de projetos que precisam da tua ajuda e fazer a tua candidatura. Como **Problem Maker**, podes publicar projetos e ser ligado a profissionais adequados ao teu problema. Liga-te a nós para fazeres parte de uma comunidade que tem um impacto positivo na sociedade.

Começa já!

ORDEM DOS ENGENHEIROS

### Sobre

Bem-vindo ao Engenheiros Sem Fronteiras, uma plataforma da Ordem dos Engenheiros que conecta engenheiros e estudantes com vontade de resolver problemas com associações que precisam das suas capacidades. Como **Problem Solver** podes navegar por uma lista de projetos que precisam da tua ajuda e fazer a tua candidatura. Como **Problem Maker**, podes publicar projetos e ser ligado a profissionais adequados ao teu problema. Liga-te a nós para fazeres parte de uma comunidade que tem um impacto positivo na sociedade.

200 Engenheiros

50 Associações

## Colégios da Engenharia

Vê todos as áreas

## Projetos

Vê todos os projetos

**Parque infantil para crianças necessitadas**  
Procuramos Engenheiro Civil que possa ajudar com a concepção e planificação de um parque infantil para crianças...

**Website para controlar inscrições em eventos de voluntariado**  
Precisamos de Engenheiro Informático para criar um website de gestão de inscrições em eventos de voluntariado.

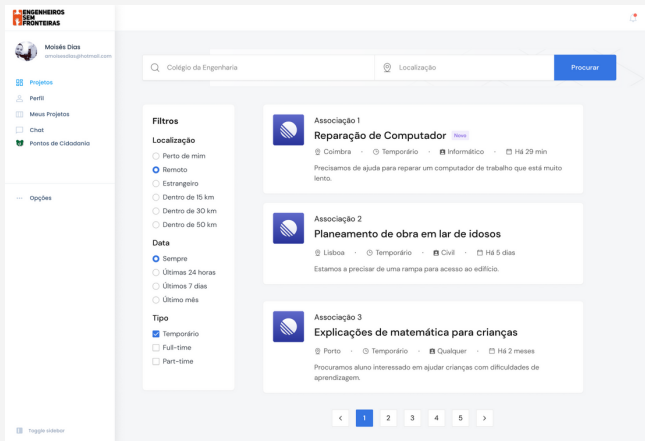
**Pequeno sistema de tratamento de águas para comunidade**  
Um engenheiro ambiental poderia desenvolver um pequeno sistema de tratamento de águas residuais para uma comunidade...

ENGENHEIROS SEM FRONTEIRAS

Menu  
Contactos  
Ordem dos Engenheiros  
Universidade de Coimbra

Follow us on:

# ENGENHEIROS



- Procurar projetos ativos
- Filtrar
- Candidatar-se a projetos

# ASSOCIAÇÕES

- Listar projetos ativos
- Publicar novos projetos
- Ver e atualizar detalhes

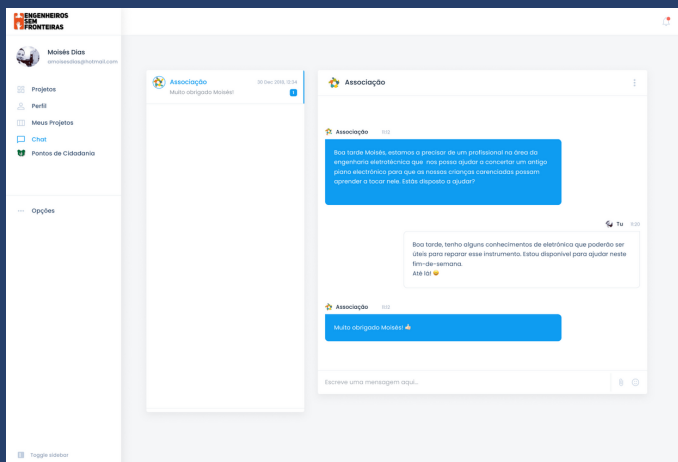
The screenshot shows a list of projects for an association. The table has columns for 'Nome', 'Votos', 'Data', 'Status', and an 'Info' icon. The data is as follows:

Nome	Votos	Data	Status	Info
Projeto 1	1/2	07.05.2022	Pendente	Info
Projeto 2	3/3	23.04.2022	Ativo	Info
Projeto 3	1/1	08.01.2022	Ativo	Info
Projeto 4	1/1	18.03.2022	Ativo	Info
Projeto 5	2/2	22.05.2022	Ativo	Info
Projeto 6	1/4	04.12.2021	Pendente	Info
Projeto 7	0/2	27.03.2022	Cancelado	Info
Projeto 8	0/1	07.07.2022	Cancelado	Info

○ ○ ○ ○

# MATCH

- Após candidatura é iniciado o diálogo entre Engenheiros e Associações



○ ○ ○ ○

○ ○ ○ ○

# ENGENHEIROS SEM FRONTEIRAS



ORDEM  
DOS  
ENGENHEIROS

# Appendix D

## LinkedIn Post



Câmara Municipal de Coimbra

3,700 followers

3w · 🌐

+ Follow ...

### CM de Coimbra visita o projeto Urso Verde no Departamento de Engenharia Eletrotécnica e de Computadores da FCTUC 🌱♻️

A Câmara Municipal (CM) de Coimbra, no âmbito do [#CoimbraCityLab](#), esteve ontem representada pelos Departamentos de Tecnologias de Informação e Inovação Digital (DTIID) e de Educação e Saúde numa visita ao laboratório onde está a ser desenvolvido o projeto **Urso Verde**, no Departamento de Engenharia Eletrotécnica e de Computadores.

O projeto **Urso Verde** visa, pois, a promoção da prática de hábitos saudáveis e sustentáveis. O laboratório vivo da CM de Coimbra, [#CoimbraCityLab](#), inserido no âmbito das competências do Centro de Inteligência de Coimbra do DTIID, tem vindo a acompanhar o desenvolvimento e prototipagem do projeto desde 2021.

[#coimbra](#) [#municipiodecoimbra](#)

[See translation](#)

