Blockchain based Android application for collective defence against sexist actions

By

Elena Pérez Tirador

Departmento de Ingeniería del Software e Inteligencia Artificial
Universidad Complutense de Madrid

A dissertation submitted to Universidad Complutense de Madrid in accordance with the requirements of the Doble Grado en Ingeniería Informática y Matemáticas in the Facultad de Informática.

April 2019

Directors and collaborators:
Samer Hassan Collado (Director)
Silvia Diaz Molina (Codirector)
David Llop Vila (Collaborator)

Declaration of Authorship

I, Elena Pérez Tirador, declare that this thesis titled, "Blockchain based Android application for collective defence against sexist actions" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Abstract

Throughout the last years, **sexual harassment** and abuse in public spaces have become major concerns for institutions and feminist organizations, since they are serious problems. Legislation has been improved in order to prosecute the perpetrators. However, these measures work *a posteriori*, when the problem has already happened.

This project aims to provide a system for preventing and helping in cases of **sexual harassment** in the streets. It centers on the implementation of an **Android** application that allows users to **cooperate** and help each other in case of need, thus creating a community of aware people that are able to react to these problems right where they happen.

A system has been designed using a User-Centered Design approach, gathering data from potential users and adapting it to their needs. The core of the application runs on an **Ethereum blockchain**. By using this technology, it is possible to create a **decentralized** system for sharing trustworthy information. Also, with its **cryptographic** basis, it provides the **privacy** and anonimity that are needed in such sensitive situations.

The proposed system has been reviewed by potential users and professionals and has shown success in providing an intuitive mobile interface to report **sexual harassment** incidents and in showing that blockchain technologies are ready to be integrated in consumer level secure applications.

Keywords: Android, blockchain, cooperation, cryptography, decentralization, Ethereum, privacy, sexual harassment

Resumen

Durante los últimos años, el **acoso sexual** en espacios públicos se ha convertido en una preocupación fundamental para diversas instituciones y organizaciones feministas, en tanto es un problema serio. Se han tomado medidas a nivel legislativo para poder condenar a los delincuentes, pero estas medidas funcionan *a posteriori*, cuando los problemas ya han sucedido.

Este proyecto busca ofrecer un sistema para prevenir y ayudar en casos de **acoso sexual** en las calles. Se centra en la implementación de una aplicación **Android** que permite a quienes la usan **cooperar** y ayudarse entre sí en casos de necesidad, creándose de esta forma una comunidad de personas implicadas, capaces de reaccionar a este tipo de problemas en el momento en que suceden.

Se ha diseñado un sistema usando las técnicas del Diseño Centrado en el Usuario, reuniendo información de usuarios potenciales y adaptándolo a sus necesidades. El núcleo de la aplicación se ejecuta sobre una **blockchain** de **Ethereum**. Usar esta tecnología permite crear un sistema **descentralizado** para compartir información fiable y fidedigna. Además, con su base criptográfica, ofrece la **privacidad** y el anonimato que se requieren en situaciones tan sensibles como estas.

El sistema propuesto ha sido evaluado por usuarios potenciales y por profesionales y ha dado buenos resultados ofreciendo una interfaz móvil intuitiva para denunciar incidentes de **acoso sexual** y mostrando que las tecnologías blockchain están listas para ser integradas en aplicaciones seguras a nivel del consumidor.

Keywords: Android, blockchain, cooperación, criptografía, descentralización, Ethereum, privacidad, acoso sexual

Acknowledgements

I want to use this space to be grateful to everyone that has been with me during the whole experience this Double Degree has been.

I would like to thank:

- Samer and Silvia, for showing me the world of decentralization, and giving me the oportunity to carry out this project.
- The rest of the P2P Models team for providing us help: Antonio, Elena, Jordi, and specially Sem.
- My parents, for letting me begin (and continue) this journey.
- Pablo, for always cheering me up with his unique point of view and for his priceless help.
- Patri and Bea, because we have gone through this together. And hopefully we still will.
- Inés and Ico, for always being there. I'm glad you are alive.
- All my residence hall friends, for sharing their life with me. It has trully ment a lot.
- All the volunteers that participated in the interviews: Almudena, Álvaro, Ana, Anto, Carlota, Dani, Enrique, Helena, Kattalin, Luis, Oier, Paula, Rafa, Raquel García, Raquel Sánchez and Zoraida.

And, in general, anyone that was there at some point this last 6 years. Thank you very much.

Contents

List of Figures

List of Tables

List of Abbreviations

API	**A**pplication **P**rogramming **I**nterface
CAS	**C**ontent **A**dressed **S**torage
ECDSA	**E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm
FCM	**F**irebase **C**loud **M**essaging
GDD	**G**oal **D**irected **D**esign
GPS	**G**round **P**ositioning **S**ystem
IDE	**I**ntegrated **D**evelopment **E**nvironment
IPFS	**I**nter**P**lanetary **F**ile **S**ystem
JSON	**J**ava**S**cript **O**bject **N**otation
MVC	**M**odel **V**iew **C**ontroller
MVP	**M**odel **V**iew **P**resenter
SDK	**S**oftware **D**evelopment **K**it
UCD	**U**ser-**C**entred **D**esign

To my family

Chapter 1

Introduction

1.1 Motivation

Sexual harassment and abuse in public spaces are severe problems that a significant number of women experience in our society. Throughout the last years, legislative progress has been made concerning these issues, but there is still work to do to help making this kind of problems disappear.

According to the *INE* (*Instituto Nacional de Estadística*) [16], in 2017 a total of 337 sexual offences were commited. 336 of them were perpetrated by men. 103 of these offences were cases of sexual agression and 107 were cases of sexual abuse [8]. These are only the reported cases, and they keep increasing every year. It is indeed a major problem, and solutions are needed to reduce these numbers.

Several ways exist for women to report these issues. However, in most cases, either help will be slow or the report will have to go through an intermediate party, which may make the flow of information both slow and cumbersome.

Situations like the one described, where the existence of intermediate elements is not adequate can be solved by using *decentralized technologies* such as *Blockchain* or *IPFS*. *Decentralized technologies* provide the perfect environment for aplications to be fair and solid, since it is almost impossible to manipulate the information stored.

In the particular context of sexual abuse, sharing truthful and trustworthy information about people's experiences in an appropriate way can help prevent the persistence of this kind of situations in the future.

These technologies also provide *anonimity* and *privacy*, which are essential for people to feel safe, specially in situations like the ones mentioned, where victims need to be sure they will be safe and no reprisal can ever be taken against them.

In this context, initiatives to create decentralized reporting systems have begun to appear. This TFG is the result of one such initiatives led by the *ISIA* department at UCM.

1.2 Objectives

The main goal of this project is to design and create a mobile app that provides tools for preventing and, if not possible, helping in cases of sexual harassment or abuse. This application will be developed for *Android* devices and will use *Blockchain* as base technology.

The basic functionality of the app will be divided in two main aspects.

First, an *SOS alert system*. This will allow women using the app to send their GPS coordinates to both contacts selected by them and app users that are geographically near.

Second, a *map* displaying dangerous places. This will allow users to upload coordinates of places that are not safe and describe the reasons, thereby giving other users the possibility to check whether some place they are interested in is safe or not.

The first functionality is centerd around helping in cases of abuse and the second, around preventing them.

1.3 Planning

This project is developed by two people in a total of seven months (since October 2018 to April 2019). The process is divided in two parts:

- From Octover to December 2018, the main concerns are learning the technologies and completing the first phases of *GDD*.
- From January to April 2019, the last phases of *GDD* are carried out, along with the programming of the application (including final testing and evaluations).

The workload is equally divided between the two people in the project. Research and design is carried out by the two members concurrently and results are then pooled in order to determine the starting point for the application. This also applies to the final testing and results.

The programming of the application is divided. Since the application has two discernible funcitonalities, one of each is responsibility of one of the members in the group. Namely:

- The *alert system* corresponds to this TFG. This includes sending and receiving alerts and storing the necessary information, both in the phone and the blockchain.
- The *map* corresponds to Patricia Barrios Palacios' TFG *"Aplicación Android con integración blockchain en respuesta al acoso en espacios públicos"*. This includes managing the map and the alert history.

Therefore, only the corresponding part is explained in this document.

1.4 Structure of this document

This document explains the development process of the app, along with initial considerations and final results.

The first two chapters (chapters 2 and 3) introduce the practical and theoretical context in which the project is developed. Chapter 2 explains the main development methodology followed. Chapter 3 introduces the technologies used in the project and explains the theoretical background of its core technology: *blockchain*.

The next two chapters (chapters 4 and 5) are the central chapters, in which the whole development process is explained. Chapter 4 details the design process and 5 explains the implementation, giving a precise description of the alert system in the app.

The final two chapters (chapters 6 and 7) provide results and conclussions of the project. Chapter 6 analyzes the results obtained in the final testing of the application. Chapter 7 gives a conclusion and lays out a series of suggestions to be implemented in the future.

Finally, there are 7 appendices that complement the explanations given in the present document.

Chapter 2

Methodology

In this chapter, the methodology followed during the developement of this project is explained. An introduction to the design methodology of choice, Goal-Directed Design (from this point forward, GDD) is given, along with further description of its phases.

2.1 Introduction

Goal-Directed Design is one of the methodologies that belong to *User-Centred Design*. This is, the whole process is developed having the final user as a central reference. This methodology focuses on the interactions between the user and the system as well as the goals the user is trying to achieve by using the system. A detailed explanation of this methodology is presented by Alan Cooper et al in "About Face: the essentials of interaction desing" [7].

This methodology is divided in several steps. The first three steps are part of the *Investigation phase*, the fourth step is part of the *Framework phase* and the last two steps are part of the *Design and Development Phase*. Namely:

1. Research
2. Modeling
    - Personas
3. Requirements definition
4. Framework definition
5. Design
6. Development

These phases and steps will now be further explained.

2.2 Investigation Phase

The *investigation phase* is the first phase in *GDD*. It focuses on gathering relevant information concerning the problem to be solved, and organizing it so it is useful and meaningful for the next phases of the process.

The goal of the *investigation phase* is knowing the potential users of the final system. This is essential, because a system should be designed to fulfill the needs of its final users, with as less bias as possible. The process consists in a series of studies in order to obtain quantitative results. Once these results are obtained, they are processed and interpreted, and eventually used to create *personas*. And, finally, these *personas* are involved in some scenarios, in which they face a problem that can be solved using the system.

The *investigation phase* has three parts, namely:
1. Research
2. Modeling
3. Requirements definition

2.2.1 Research

Research is the first step in the whole *GDD* process. It centers on obtaining the quantitative results that will be used in subsequent steps of the process. This results will provide information such as who are the potential users of the system, what are their goals when they use the system and how do they behave. They will also explain the context in which the users would interact with the system, as well as the state of the art — other systems that already exist and are similar to the one being developed.

There are plenty of techniques and processes to carry out this research. Among them, two have been used during the development of this project: *interviews* and *competitor analysis*.

Interviews

First, it is necessary to do the *recruiting*. This is, find a representative sample of users among the target users of the system. In order to do this, the types of users must be briefly defined.

Then, a list of questions is written. The first questions are used to check if the person interviewed is actually a target user. The rest of the questions should cover the following aspects:
- Goals and motivations for using the system (*Questions oriented to objectives*).
- Tasks and activities carried out with the product (*Questions oriented to the system*).
- Mental model - how users think and what do they do (*Questions oriented to work flows*).
- General motivations and preferences (*Questions oriented to attitudes*).

The duration of the interviews should not exceed one hour. At the end, the interviewee should have the opportunity to summarize the main ideas or put them into context.

Competitor analysis

Competitor analysis focuses on already existing products that share features with the system being developed. It consisits in analyzing how users interact with these products, how do these products help the users fulfill their needs and what probems do they find while using them. All this information is then used in the development of the system, in order to avoid usability problems and overall imporve user experience.

This analysis can be divided in four stages:

First, the competitors must be identified. In order to do this, the main tasks that the system is going to cover must be defined and listed. The competitors are found based on these tasks. If a system only covers some of the tasks listed, it is considered a *partial competitor*.

Second, the *product dimensions* must be established. Analyzing every possible attribute would be infeasible, so the most essential ones are listed and priorized. The subsequent study will be done relating to them.

Third, analyzing the competitors. The competitors found in the first step are analyzed based on the product dimensions defined in the second one. The results are organized in a comparative matrix indicating how each competitor covers each product dimension.

Finally, the results obtained in the third step are analyzed with the purpose of gaining knowledge about the competitors' flaws and strengths, the problems the users have found with them and how have they solved them. This aims to know what can be improved and to identify areas of differentiation.

After finishing the *research* (this is, both *interviews* and *competitor analysis*), all the information obtained is put together and the summarized data is organized in a list of simple points, which is called a *list of factoids*.

2.2.2 Modeling

The next stage after *research* is modeling. The goal of the *modeling* phase is producing *personas*. A *persona* is a detailed description of a fictional individual that ideally represents and summarizes a group of target users. It is created based on the information obtained after the *research*. The description of a *persona* is very detailed: it is represented using a picture and given a name, surname, age and some more personal information. This is usefull since it is easier to understand a particular person than an abstract group of people. The *persona* also encapsulates the goals, motivations and wills of the group of people it represents.

For the purpose of this project, and given the narrow variety of users, only one *persona* has been designed. It is a *primary persona*, which means it represents the target users of the system.

As it is described by Tamara Adlin and John Pruitt in "The Essential Persona Lifecycle" [1], the *persona* is designed following a *top-down* process.

Top-down design process

Several steps are carried out in this process.

First, the *user categories* are defined. This is, the different types of potential users the system will have. In order to define these categories, several approaches can be taken: they could be defined based on the tasks the users will acomplish with the system, or based on the goals of the users, or based on segments, as in a market study.

Second, the data is processed by organizing it in *affinity diagrams*. The information from the *factoids* is filtered and organized in groups based on the user categories previously defined.

Third, *presona frameworks* are created. The data organized in the previous step is compared with the user categories defined in the first step, thus creating subgroups of information. Each of these subgroups is used to create a *persona* framework, which is basically a list of factoids, and a summary of the qualities of the user category it represents. After that, it is decided which of this frameworks are going to actually turn into *personas*.

Finally, personas are developed based on these frameworks. Basically the frameworks are completed with more detail and context, in a narrative way. The finished *persona* should contain, at least:

- Name and surname
- Age and sex
- Quote
- Description
- Picture
- Job
- Short-term and long-term goals
- Academic background and knowledge
- Environment and context

To finish the process, these *personas* are validated by checking if they actually fit with the original data.

2.2.3 Requirements

The *requirements* phase is the last step of the *investigation phase*. It, in turn, is divided in two phases.

Scenarios

A *scenario* is a narration inolving a *persona* pursuing a goal, and describing the tasks the *persona* carries out in order to satisfy the goal.

Requirements

The requirements are the functionalities that the system needs to provide, in order to make the *scenarios* possible. This is, what the system has to be able to do, so the *persona* can fulfill their goals. The key of this phase is defining *what* has to be done, rather than *how* it has to be done. The process for defining these *requirements* can be broken down into several steps.

First, the *problems* are formulated based on the previous stages (research and *personas*). This is, a problematic situation that has to be changed, from the perspective of the *persona*, is explained. The way the system is going to solve this problem is also formulated, becoming a design objective.

The next step is a *brainstorming* in which all members of the team share their opininons about the way the solutions could work. The goal of this activity is forgetting these preconceived ideas that are shared in the *brainstorming*, thus being able to center in the *scenarios* and goals with less subjectivity.

After that, the next step is identifying the expectations of the *personas*. This is, social factors influencing their expectations, how they hope their experience using the system is, how they think the system will work and how they would interact with it.

Then, the actual *context scenarios* are built. They describe situations in which a *persona* uses the system to cover a need. It is described from its perspective, so it centers on its expectations, motivations and perceptions. These *scenarios* do not describe the system in detail nor the specific interactions the user executes, but give a high level description of the actions that occur.

Once the *scenarios* are written, they are analyzed to obtain the needs and requirements of the *personas*, that is, context and actions to acomplish a specific goal. These requirements can be divided in several classes:

- Data requitements (information to be represented in the system)
- Functional requirements (actions over the objects in the system)
- Context requirements (system restrictions)
- Other requirements (planification and cost of the system)

2.3 Framework Phase

The *framework phase* is the second phase in *GDD*. It focuses on defining the main structure of the interface, in terms of interaction with the user. This is, how the *functional elements* are laid out in the interface, their behavior and how they are used.

The *scenarios* and *requirements* from the previous phase are used to create the first prototypes for the system. To validate these prototypes, they are checked against new *scenarios* created from the previous ones, and with usage tests from real users. The feedback obtained from these activities is used to improve the prototypes and create new ones, which are again validated using the same techniques. This process is iterated as many times as needed.

This phase corresponds with the *framework deffinition* mentioned in 2.1. Different kinds of related frameworks can be defined at this point, such as

- Interaction framework
- Visual design framework definition
- Industrial design framework definition

Only the first one will be explained in detail.

2.3.1 Interaction framework definition

The *interaction framework* defines the interface screens that constitute the system and how they are organized, the flow and the general behaviour of the system, from a high-level point of view.

This phase is divided in six stages. And, since it is an iterative phase, they are repeated several times, and the order of some of them is interchangeable.

Defining form factor, posture and input methods

The *form factor* is the context in which the information is presented and the constraints that apply. This is, for example, the device where the system is going to be used, the size of the screen, the quality of the screen or the climate conditions in which the system is going to be used. These information can usually be obtained from the context scenarios.

The *posture* represents how much attention the user pays to the system while using it and how much is actually needed to use the system. There are different types of postures:

- *Sovereign posture.* Systems that have lots of functions and features. The user interacts with them for a long period of time, directing the work flow. Therefore, they monopolize the users' attention and tend to use the full screen.
- *Transient posture.* Systems with a specific functionality. The user interacts with them for short periods of time, so this posture is strictly temporary. Therefore, these systems should have a clear and simple interface.
- *Daemonic posture.* Systems that run in the backround. The user has little to no interaction with them, so, whenever the user interacts with them, they behave as transient, so the princliples described for *transient posture* also apply. Sometimes they show their state via an icon that is constantly visible for the user.

The *input methods* are usually related to the form factor and the posture of the system, as well as the personas' preferences, attitudes and aptitudes. They represent the way the user interacts with the system. For example, using a keyboard, a mouse, a touch screen, etc.

Defining functional and data elements

During this stage, the elements in the interface are described, those based on data and functional requirements described in 2.2.3.

First, the *data elements* — the data to be represented in the system — are listed and then the *functional elements* (interface elements that allow the user to manipulate the *data elements*) are defined. The context scenarios, *personas*' goals and *mental models* are essential to define all these elements.

Determine functional groups and hierarchy

During this stage, the elements defined in the previous stage are organized in a hierarchical structure. They are grouped into *functional units*, as follows: *functional* and *data elements* are organized in *containers* and *views*, which are elements such as screens, frames or panes.

Some issues must be considered. Such as which elements should be larger in the interface, or how they should be organized in order to make the interaction fluent.

Sketch the interaction framework

In these phase, the first sketches and mockups are created. The first ones are usually very simple sketches in paper. Later, they become more complex and, eventually, digital mockups are made. This complexity increase happens as more iterations are performed.

Construct keypath scenarios

During this phase, *key path* scenarios are defined. They describe how a *persona* interacts with the interface designed during the previous stages of this process. They are created based on the *context scenarios*, but in this case actual interactions are described. At this stage, both the objectives and the process to achieve them are described.

Check designs with validation scenarios

Once one iteration of the interaction framework is finished, it is validated using *validation scenarios*. These scenarios are more simple than the previous ones. They lay out questions about situations in which the system may fail and try to test the boundaries of the system. They also test different ways of completing the same task in the system. The goal of this testing is finding possible holes in the system design and fixing them in order to cover all the possible needs of the *persona*.

Chapter 3

Technologies and mathematical background

In this chapter, the different technologies used in this project are described, as well as some of the methodologies employed.

The main technology used as a base for this project is *blockchain*. In the current chapter, the mathematical background in which it is based shall be studied. First, basic definitions concerning *cryptographic hashes* and *digital signatures* will be introduced. Then, basic theory about *elliptic curves* will be explained.

3.1 Technologies

Several technologies have been used to develop the present project. They shall now be listed and the use of some of them will be briefly explained.

3.1.1 Ethereum and decentralized technologies

The base of this project is providing a decentralized system. Therefore, blockchain and *IPFS* are suitable solutions. *Ethereum* is the chosen blockchain, because it it open source. It provides functionality for implementing *smart contracts*. The *smart contracts* are implemented using *Solidity* as a language. Related to this, the following tools have been used.

IPFS (InterPlanetary File System)

IPFS [17] is a decentralized file system based on a P2P (peer-to-peer) Content Adressed Storage (CAS) net. It provides hyperlinks to the stored content. It gives the posibility to store heavier data than a blockchain, such as media.

Remix

Remix [38] [28] is an online *Solidity* compiler. It provides functionality for testing, debugging and deploying *smart contracts*.

Rinkeby

Rinkeby [30] is an *Ethereum* testnet. It provides functionality for testing without spending actual money. Funding is obtained via the *Rinkeby Faucet* [29].

Metamask

Metamask [21] is a browser extension that provides functionality to run *distributed applications* without being a node in an *Ethereum* net. It is also a *wallet manager*.

Infura

Infura [15] is a set of tools to facilitate interaction with *Ethereum* or *IPFS*. It provides the infrastructure, meaning the node in the main net, as used, for example, in *Metamask* (3.1.1).

Web3j

Web3j [18] is a Java and Android library that provides functionality for working with *Smart Contracts* and conneacting with nodes, such as *Infura* (3.1.1).

3.1.2 Design and implementation

Marvel App

Marvel [20] is a mobile app for prototyping. It allows the user to create an interactive prototype using pictures of drawings made in paper. Pictures are taken and then interconnected by hotspots.

Pencil

Pencil [14] is a desktop application for prototyping. It provides functionality to create digital interactive prototypes.

Gravit Designer

Gravit [25] is an online vector graphic design application. It was used to create some of the icons in *SpreadApp*.

Android Studio

Android Studio [9] is the official IDE for Android development, developed by Google. Combined wih the SDK, it provides functionality to implement, compile and debug Android applications.

Mapbox

Mapbox [19] is a platform for creating custom on-line maps for applications.

Firebase (FCM API)

Firebase [10] is a platform for mobile and web app developmenet. *FCM* [11] is a cross-platform solution for message delivery.

Telegram Bot API

Telegram [33] provides an API for developers to connect and manage Telegram bots [34].

3.1.3 Other technologies

GitHub

GitHub [4] is a platform for collaborative software development and version control (using Git). The code for the applicaation is stored in a repository.

Zotero

Zotero [39] is a bibliographic reference manager. It provides functionality for organizing bibliography and easy citing.

TexWorks

TexWorks [35] [36] is a desktop application for writing and compiling LaTeX documents.

Overleaf

Overleaf [26] is an online collaborative LaTeX editor.

3.2 Architectural patterns

3.2.1 Model-View-Presenter

Model-View-Presenter [27] is an architectural pattern based on a generalization of *MVC* (Model-View-Controller). It consists of:

- The *model* is an interface that is implemented in the *view*. It encapsulates the data displayed in the interface, along with the methods for modifying it.
- The *view* displays the data in the interface and manages the interactions with the user, by accepting the user input and calling the *presenter*.
- The *presenter* manages and modifies the data encapsulated in the *model* in order to correctly display it in the *view*.

3.3 Mathematical background. Introduction (Cryptographic hashes and digital signatures)

Traditional economic systems have mechanisms to prevent counterfeiting, double payments and similar problems. In these systems, banks act like chore entities that ensure the soundness of the payments, and laws exist to regulate the transactions. However, in cryptocurrency systems there is no intermediate entity nor legislation to ensure the security of the system. Under these circumstances, the soundness of the transactions is fundamented in *cryptography*[22].

3.3.1 Cryptographic hash functions

First, it is necessary to provide a definition of what *hash functions* are, as well as the properties that a hash function has to verify to be considered cryptographic.

Definition 1 *A **hash function** is a function that verifies:*

1. *Its input is a string of arbitrary length.*
2. *Its output's length is fixed (e.g. 256 bits).*
3. *Its complexity is $\mathcal{O}(n)$, n being the length of the input string.*

Definition 2 *A hash function is **cryptographic** if it verifies:*

1. *Collision resistence*
2. *Hiding*
3. *Puzzle friendliness*

This properties will now be further analyzed.

Properties

Definition 3 *A function H is **collision resistant** if it is infeasable to find two values x and y such that $x \neq y$, $H(x) = H(y)$.*

It is important to realize that collisions may exist in practice (in fact, they will most likely exist, since the input is, in general, larger than the output). What matters is that finding a collision should be impracticable.

Definition 4 ***Hiding** property states that, given the output of a hash function $y = H(x)$ it is infeasible to find the value of x.*

To facilitate the existence of this property, the input x is concatenated to a secret value r chosen with a probability distribution that has high min-entropy. Min-entropy is a measure of how predictable a result is. Therefore, "high min-entropy" encapsulates the idea that the distribution is very sparse. Given this concepts, the property of *hiding* can be redefined.

Definition 5 *A hash function H is **hiding** if, given the function H and a value $H(r||x)$[1] it is infeasible to find x. The value of r is unknown.*

Now, as in definition 5, a value k is chosen from a distribution with high min-entropy, and it is concatenated to an input x before applying the hash function.

Definition 6 *A hash function H is **puzzle friendly** if, given the function H, an n-bits output value y and the value k it is infeasible to find x such that $H(k||x) = y$ in a time significantly lower than 2^n.*

Merkle-Damgård construction

As previously stated in definition 1, a hash function must accept arbitrary-length inputs. This can be achieved by defining a function with fixed input size, which shall be called *compression function*. The *Merkle-Damgård Transform* is applied to this function to turn it into an equivalent one, which admits arbitrary-length inputs. If the compression function is collision resistent, so is the transformed one.

The *Merkle-Damgård Transform* works as follows: Let us suppose defined a compression function that accepts inputs of length m and generates outputs of length n, being $n < m$. Given an input of arbitrary length k, it is divided in blocks sized $m - n$ and, in every iteration of the algorithm, an input formed by the block and the previous iteration's output is given to the compression function (in effect, $m - n + n = m$). This is repeated as many times as necessary to process the whole input. In the first iteration, since no previous output exists, a certain *initialization vector* is used.

[1] The symbol "||" denotes concatenation.

This system is used, for example, in *SHA-256*, which is the hash function used by Bitcoin. In this case, the input of the compression function is 768 bits, the output is 256 bits and the block is, therefore, 512 bits.

Data structures

Definition 7 *A **hash pointer** is a pointer that not only points to the location of the data, but also stores a hash of that data. This makes it possible to verify if they have been modified.*

Using this type of pointers, data structures based in usual structures formed by linked nodes can be defined, as long as they do not have cycles. Two examples will now be briefly explained: Blockchain and Merkle Trees.

Blockchain

Definition 8 *A **blockchain** es a linked list using hash pointers. Each block stores a pointer to the previous block.*

Blockchain constitutes an efficient system to build a register that is resistent to forgery: if the data within a block is modified, the next block's hash pointer is consequently incorrect. To hide the modification of the data, the next block would have to be modified, thus making the next one store an incorrect hash, and so on until the header is reached. If a hash of the header is stored in a way that it can not be modified, it will always be possible to detect changes. In this way, storing a single hash — the one of the header — the integrity of the whole chain is ensured. This special block that stores the hash is called *genesis block*.

Merkle Trees

Definition 9 *A **Merkle Tree** is a binary linked tree with hash pointers.*

The blocks containing the data are the leaves. A group is made with every two blocks and, for each pair, a structure with two hash pointers is created, one pointer for each block. Then, this new level is also grouped in pairs, and for each one a structure with their hashes is created. This process is repeated until the root of the tree is reached.

As it was the case with blockchain, any modification in the data will be propagated to the top of the tree and will finally reach the root, so it is a safe structure and it is only necessary to store a hash of the root to guarantee this. In addition, with Merkle Trees it is possible to check efficiently if a given piece of information is stored in the structure.

Proof of membership In Merkle Trees, given a block and the path to reach it from the root, it is possible to check if the block is in the tree. For a tree with n nodes, this can be achieved with complexity $\mathcal{O}(\log\ n)$.

Definition 10 *A Merkle Tree is said to be **ordered** if the leaves of the tree are ordered following a defined system of criteria.*

Proof of non-membership In an ordered Merkle Tree it can be proved that an element is not in the tree with complexity $\mathcal{O}(\log\ n)$. It is sufficient to show the path to the previous and the next elements given the ordering criteria. Since these two elements are consecutive in the tree, it can therefore be proven that the searched element is not in the tree.

3.3.2 Digital signatures

By analogy to traditional signatures, two properties are required for digital signatures:

1. Only the owner of the signature can produce it, but anyone who sees it can verify it is valid.
2. A signature must be bound to a document, so it can not be used to show consent with a different one that has not been signed.

The scheme for building a digital signature uses three functions:

`(sk,pk):= generateKeys(keysize)` This function geeets a key size and generates a pair of keys: `sk` (secreet key) is uses for signing messages; `pk` (public key) is used to verify the signature.

`sm:=sign(sk,message)` This function gets a message and a private key and generates the signed message (`sm`) under the key.

`isValid:=verify(pk,message,sm)` This funciton gets a message, a signed message and a public key. It produces a boolean value that is **`true`** if `sm` is a valid signature for `message` under `pk` and **`false`** otherwise.

Two properties are required for this scheme:

1. First, every valid signature must verify:

```
verify(pk, message, sign(sk, message)) == true
```

2. Second, signatures are *existentially inforgeable*, meaning it must be computationally infeasible to forge the signatures. That is, given a public key and the signatures of various messages, it must not be possible to forge the signature in a new message.

To illustrate this last property, a sort of game can be considered. The game involves two parts: a *challenger* and an *attacker*. The *challenger* knows his public and private kay and the *attacker* only knows the public key. The *attacker* can send messages to the *challenger* at his own will, and the latter has to sign them with his private key and resend them (since the *attacker* has the public key, he can check the signatures are correct). After a certain number of iterations, the *attacker* tries to guess the *callenger*'s private key.

If the *attacker* is able to guess the key correctly after k messages, k being a polynomial function of the key's length, then the signaature system is not infeasible. The system is considered infeasible if, whatever algorithm the *attacker* uses, it is not possible for them to guess the private key in polynomial time.

Other considerations

Many signature systems have parts that depend on random generation of numbers. For the algorithm to be sound, it is necessary to have a good system for generating these random numbers as, otherwise, it could allow an attacker to easily find the secret key.

Given the nature of signature functions, there must be a limitation in the length of the input message. One way to allow arbitrary length signatures is signing the hash of the message, that has a fixed length - and since the hash function is collision resistent, this hash represents a good summary of the message. A hash pointer could also be signed, so the signature would cover the whole structure the pointer points to (for example, the whole blockchain).

Identities

A public key can be understood as an identity or an actor in a system, as it allows for vierifying if several messages come from the same issuer. Given this point of view, for someone to talk under the identity of `pk` they need `sk`.

In this way, an identity can be created by generating a new pair `(pk, sk)`. Whoever uses this identity will communicate using `sk` and it will be possible to verify if a message comes from them using `pk`.

Since the public key is seemingly random, it does not reveal information about the issuer of the message, unless it is contained in the message itself. In addition, multiple identities can be created by creating new pairs `(pk, sk)`.

Decentralized Management of Identities

In traditional systems, a central authoriry is responsible for registering new users in the system. With a decentralized management, it is considered thar each user registeres themselves, by creating an identity. As explained before, because of the nature of the system for creating identities, each user can have as many identities as they will.

Despite the seeming anonimity this system seems to provide, a user is never absolutely anonymous. Since the key identifies them unambiguously, their transactions and the actions they carry out give enough information for an external observer to discover who is the person behind a certain identity. This means the systme is actually semi-anonymous.

3.4 Mathematical Background. Elliptic Curves

One of the most important signature algorithms nowadays is *ECDSA*: Elliptic Curve Digital Signature Algorithm. For instance, it is used as the signature algorithm in Bitcoin. This algorithm is based on a complex mathematical sctructure known as *Elliptic Curves*[12].

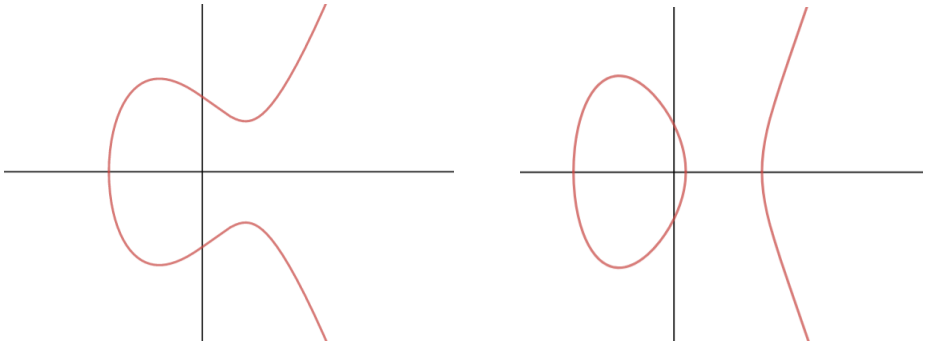
Definition 11 *An **elliptic curve** E is the set of sollutions to an equation of the form:*

$$Y^2 = X^2 + AX + B$$

(These equations are called ***Weirstrass equations****). The constants A and B must satisfy the condition:*

$$4A^3 + 27B^2 \neq 0$$ [2]

Examples of elliptic curves are illustrated in figure 3.1.



(A) $E\colon y^2 = x^3 - 2x + 2$ (B) $E\colon y^2 = x^3 - 4x + 1$

FIGURE 3.1: Examples of elliptic curves

An internal addition operation can be easilly defined. This is, an operation that takes two points in the curve and generates a third one that is also in the curve; and it is a conmutative and associative operation, and an identity element can be defined. It is represented in figure 3.2.

Let E be an elliptic curve, and let P and Q be two points in E. The line L connecting P and Q is traced. L intersects E at 3 points: P, Q and R.The new pont R is taken and its reflection across the x-axis is calculated (this is, its y coordinate is multiplied by -1). This operation produces a new point R', which is considered the sum of P and Q. It is denoted:

$$P \oplus Q = R'$$

Adding a point to itself is the limit of the prevoius situation. In this case, Q tends to P, so the limit is $P = Q$. The line L intersects E in the point P with multiplicity 2. This is, L is *tangent* to E in P. Since P is a double intersection point, L still intersects E in three points: P (twice) and R. The reflection is again calculated, and R' is obtained, as before. This process is represented in figure 3.3.

Two points with the same x coordinate, this is, a point P and its reflection across the x-axis, can not be added using the previous procedure, since a vertical line intersects E in only two points. It is, therefore, necessary to define a point $\mathcal{O}$ at infinity (this is, not in the XY-plane), such that:

$$P \oplus P' = \mathcal{O}$$

This case is illustrated in figure 3.4.

[2] This condition is equivalent to saying the polynomial $X^3 + AX + B$ has no repeated roots (it has no singular points).

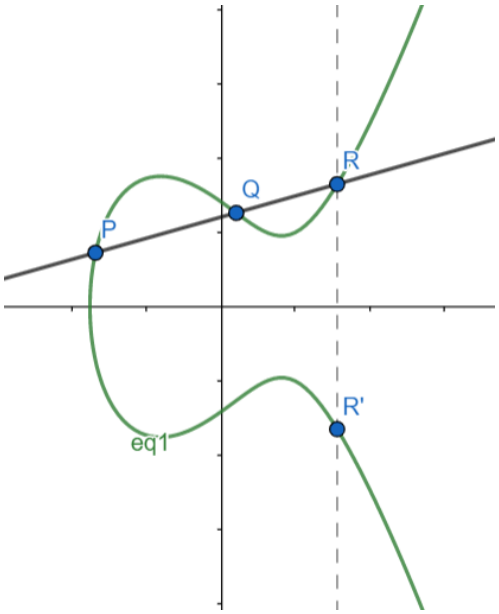


FIGURE 3.2: Elliptic curve addition

This point can be imagined as a third invisible intersection in the vertical line. With that in mind, it is easy to conclude:

$$P \oplus \mathcal{O} = P$$

Therefore, $\mathcal{O}$ acts like zero. This is, as an identity element.

Notation 1 *Given E, an elliptic curve, and $P = (a, b)$, a point in E, the reflection of P across the x-axis is denoted as $\ominus P = (a, -b)$, or simply as $-P$. With this notation, is easy to define the substraction as:*

$$P \ominus Q = P \oplus (-Q)$$

The multiple addition of a point can be represented as multiplication of a point by a scalar:

$$nP = P \oplus \overset{n}{\ldots} \oplus P$$

Theorem 1 *Let E be an elliptic curve. The addition operation defined by $\oplus$ has the following properties:*

1. Identity. $P \oplus \mathcal{O} = \mathcal{O} \oplus P = P \ \forall P$
2. Inverse. $P \oplus (-P) = \mathcal{O}$
3. Associative. $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$
4. Commutative. $P \oplus Q = Q \oplus P$

Therefore, E is an abelian group.

Elliptic Curve Addition Algorithm

Let E be an elliptic curve:

$$E\colon Y^2 = X^3 + AX + B$$

And let P_1 and P_2 be two points in E. The algorithm works as follows:

1. *If* $P_1 = 0$, $\Rightarrow P_1 \oplus P_2 = P_2$

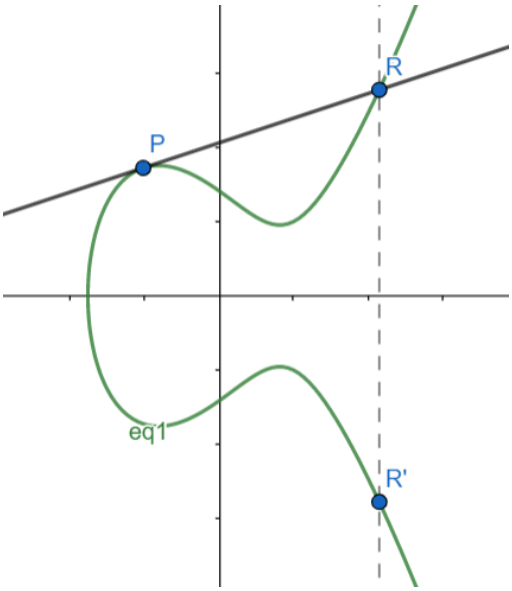


FIGURE 3.3: Elliptic curve addition. Same point

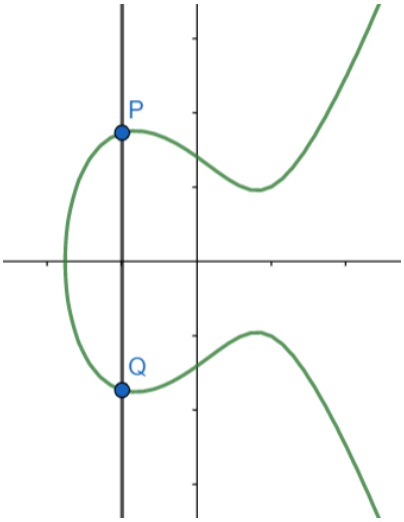


FIGURE 3.4: Elliptic curve addition. Vertical

2. *Otherwise, if* $P_2 = 0$, $\Rightarrow P_1 \oplus P_2 = P_1$
3. *Otherwise*, let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$
4. *If* $x_1 = x_2$ and $y_1 = y_2$, $\Rightarrow P_1 \oplus P_2 = \mathcal{O}$
5. *Otherwise*, define λ as:

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } P_1 \neq P_2 \\ \frac{3x_1^2 + A}{2y_1} & \text{if } P_1 = P_2 \end{cases}$$

and let

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_1 - x_2) - y_1$$

As a result, $P_1 \oplus P_2 = (x_3, y_3)$.

Chapter 4

Applied GDD

In this chapter, the way the *GDD* process (described in chapter 2) has been applied to the present project is explained. The variations over the original process are described, and the documentation and results obtained are shown.

4.1 Introduction

As it was described in chapter 2, the methodology followed during the development of this project has been *Goal-Directed Design* (*GDD*). However, given the short duration of this project, it has been slightly simplified, in order to adjust to the actal schedule of the project. The specific steps shall be explained in this chapter.

4.2 Research. Competitor analysis

The *research* phase started with the *competitor analysis*. The main features of *SpreadApp* were defined and similar applications in the market were analyzed in depth.

4.2.1 Features

The first step in *Competitor Analysis* applied to *User-Centered Design* is defining the main features and functionalities that the application, *SpreadApp*, should possess. Once listed, they will be used as a base to identify wich existing applications are its partial competitors.

The main functionalities of *SpreadApp* are divided in two main groups, within which more specific functionalities can be identified. Namely:

- **SOS Alert**
  - "SOS" button to send a notification.
  - List of trustworthy contacts.
  - Sending precise geographic location via GPS.
  - Sending alerts to both contacts in the contact list and app users that are geographically near.
  - Receiving alerts from nearby users and users whose contact list the user in.
  - Visualizing a user's information, if they have sent an alert.
- **Dangerous spots map**
  - Visualizing a map showing dangerous spots.
  - Specific information about a spot (why it is dangerous, who it is dangerous for, when it is dangerous...)
  - Sending dangerous locations, manually of automatically.
  - Sending information about a dangerous location.

  - Reporting misogynistic or offensive posters or advertisements.

Based in these points, existing applications that share features with *SpreadApp* have been researched. These applications are found to share features with *SpreadApp*:

- Circle of 6
- No more
- Ushahidi
- SafeCity
- Hollaback
- bSafe
- Alpify

The functionalities and features of these applications will next be analyzed.

4.2.2 Analyzed Applications

Circle of 6

Circle of 6 [6] is an application in which the user can add up to 6 people to a list of reliable contacts (such as close friends or family), so that they can be asked for help in case of need. These 6 people will form the user's *circle*. The user has several options available:

- Sending a message to the people in the circle saying "Come and get me. I need help getting home safely.", also providing GPS location.
- Sending a message to the people in the circle saying "Call and pretend you need me. I need an interruption.".
- Sending a message to the people in the circle saying "I need to talk."
- Checking useful links with information about sexuality, relationships and safety.
- Calling phone numbers specialized in abuse and similar subjects.

Circle of 6 is supposed to be available for both Android and iOS, although the Android version is no longer accessible.

Circle of 6 is available in English, Hindi and Spanish, and it has been rated with an average of $4.7/5$ in the AppStore.

No more

No more [24] is an application that allows the user to send an alert with the GPS location to both their contacts and other users of the app that are nearby.

It seems that this application is only a project idea, that has not been put into practice yet, since it is not possible to find it in Google Play nor the AppStore.

Ushahidi

Ushahidi [37] is a tool for gathering and visualizing data using maps, created to be primarly used in scenaries of need, such as crisis situations or support of human rights.

It is open source and plenty of help applications have been developed using this tool.

SafeCity

SafeCity [32] is an application in which users can anonymously upload their experiences with sexual abuse in public places and point out those places. All this information is summarised in a map, that also shows the most commonly reported places, so all these data, once aggregated, can help make cities safer.

This application is mainly used in India, Kenia, Cameroon, Nepal, Nigeria and Trinidad and Tobago.

SafeCity is available for both Android and iOS, in both cases for free.

SafeCity is available in English, Hindi, Malayan, Marathi, Spanish and Swahili, and it has been rated with an average of $4.7/5$ in Google Play, with more than 1000 downloads (i.e., between 1000 and 5000).

Hollaback

Hollaback [13] is a movement aimed to end public problems of harassment and abuse all over the world. In particular, it has an application available in which people can publish stories related to abuse and locate them in a map. They can publish stories they have either experienced or witnessed. A story must have a location, a description text, a category (the type of abuse) and, optionally, the name of the person who published it (it can also be publishd anonymously if desired). Once a story is published in the map, other users can send their support to the poster by pressing the "I Got Your Back" button.

This application also gives the user the possibility to display their own abuse stories map and check, for each story, which users have offered their support.

Hollaback is available for both Android and iOS, in both cases for free.

Hollaback is available in English only. It has been rated with an average of $4.4/5$ in the AppStore and $3.8/5$ in Google Play, where it has been dowloaded more than 10000 times (i.e., between 10000 and 50000).

bSafe

bSafe [2] [3] is an application that allows the user to create a network of reliable contacts to send them an alert in case of emergency. The user can send these contacts his/her GPS location, set a timer to send the contacts an alarm or fake a call to force an interruption.

bSafe allows the user to activate the alert by voice, so getting the phone out of the pocket is not necessary. Another option given by the application is for the contacts in the network to see and hear everything recorded live by the phone, once the alarm has been sent.

People in the contact list must have the application installed, in order to receive the alarms.

bSafe is available for both Android and iOS, in both cases for free, although only the base application is free. The full application costs €8.49 for iOS. For Android, in-app products can be purchased for up to €17.99. The features that can be purchased are the ones related with voice activation, video and audio recording and streaming.

bSafe is available in English and Norwegian. It has been rated with an average of $4.1/5$ in Google Play.

Alpify / Safe 365

This application was originally created as *"Alpify"* [5] and its target market was people who practiced mountain sports, such as hiking or skiing. It allowed the users to alert the emergency services, sending them additional useful data, such as precise geographic location, the path followed by the user or other relevant personal data.

At this moment, the application is called *"Safe 365"* [31] and its target market is the elderly. It allows the users to know if their loved ones are well. Users can add a number of contacts to the application. These contacts will be their "protegés". Once added, the user can check information about them, such as their location, their battery level, whether they have entered or left their house, neighborhood, etc.

Safe 365 is available for both Android and iOS, in both cases for free.

Safe 365 is available in English, Catalan, French, Italian, Portuguese and Spanish. It has been rated with an average of $3.8/5$ in the AppStore and $4.4/5$ in Google Play, where it has been downloaded more then 1000000 times (i.e., between 1000000 and 5000000).

4.2.3 Product dimensions

Next, the *product dimensions* in which the *Competitor Analysis* has been based are described. Essentially, elements described in section 4.2.1 are analyzed by checking to what extent the applications cover them. As before, and for the sake of clarity, *product dimensions* are divided in two groups: SOS alert and map. The main reason for this division is that existing applicatios cover functionalities from either of the groups, but no application exists that covers functionalities from both.

SOS alert

For the SOS alert, the following *product dimensions* are analyzed:

- **SOS button** (Big/Small/None)
- **Contact list** (Yes/No)
- **Send GPS location** (Yes/No)
- **Send alert** (Contacts/People nearby/Both)
- **Receive alerts from other users** (Yes/No)
- **Display information of person in danger** (Yes/No)

Dangerous spots map

For the dangerous spots map, the following *product dimensions* are analyzed:

- **Display map** (Yes/No)

- **Spot information** (Type of information/No)
- **Send locations** (Yes/No)
- **Send information about locations** (Yes/No)
- **Report offensive posters** (Yes/No)

Comparison matrices

Matrices containing results obtained from *product dimensions* analysis are shown below (tables 4.1 and 4.2). For each cell, the value of the corresponding dimension is indicated, and a '-' sign is used in case there is no information available for a particular dimension.

	Circle of 6	No more	Ushahidi	SafeCity	Hollaback	bSafe	Safe 365
SOS button	Medium	-	No	No	No	Small (Voice activation)	Big
Contact list	Yes (6)	Yes	No	No	No	Yes	Yeí
GPS location	Yes	Yes	No	No	No	Yes	Yes
Send alert	6 contacts	Contacts Nearby users	No	No	No	Contacts	Contacts Emergency services
Receive alert	Not in-app	Yes	No	No	No	No	Yes
Display information	Not in-app	-	No	No	No	No	Yes

TABLE 4.1: Comparative matrix for the SOS alarm feature

	Circle of 6	No more	Ushahidi	SafeCity	Hollaback	bSafe	Safe 365
Display map	No	No	Yes	Yes	Yes	No *3	No *3
Spot information	No	No	Anything	*1	*2	No	No
Send locations	No	No	Possible	Yes	Yes	No	No
Send information	No	No	Possible	Yes	Yes	No	No
Report posters	No	No	Possible	No *4	No	No	No

*1 Type of problem, problem description, other users' advise, surveys
*2 Problem description, date, counter of people supporting, difference between experienced or presenced situations
*3 The app has a map, but it has another purpose (different from the one studied)
*4 Creation of different type of events, such as surveys, etc. is supported. Therefore, reporting posters might be possible, although there is not a specific functionality for that purpose.

TABLE 4.2: Comparative matrix for the map feature

4.2.4 Conclusions

As seen in the comparison matrices, some of the applications cover most of the functionalities in the SOS alert group and other applications cover part of the functionalities

in the map group. Nevertheless, none of the aplications covering functionalities from one of the groups cover functionalities from the other one.

It can be concluded that no application is a total competitor for *SpreadApp*. A user wanting to have every feature would need to download several of the applications analyzed.

Regarding each of the groups in particular, the following can be inferred:

SOS alert

At first glance, *Ushahidi*, *SafeCity* and *Hollaback* can be discarded as competitors.

Among the remaining applications, both *Circle of 6* and *bSafe* cover part of the functionalities, but they handle the alerts outside the app, so alerts are never sent to nearby users — only to the user's contact list.

No more seems to cover most functionalities, but it is just a project in developement. That is, the app doesn't exist yet.

Safe 365 covers all studied dimensions. However, the application is meant for a completely different use from the one studied. First, the target market of the application is not women, who could potentially suffer abuse, but elders who want to know if their family is well and would want to alert their family if they have any problem. The application is fully designed towards this goal. Thus, it would not be possible to use it for other purposes, as the one being researched.

Dangerous spots map

As before, at first glance several applications can be discarded. In this case, *Circle of 6*, *No more*, *bSafe* and *Safe 365*.

Amongst the remaining ones, *Ushahidi* should be discarded, as it is not a mobile app, but a tool for collecting and visualizing data. It can be used to create applications covering the studied functionalities, but it is not a competitor by itself.

Lastly, both *SafeCity* and *Hollaback* work similarly, regarding the studied features. Although, in practice, *SafeCity* is a more finished application, with more functionalities. In any case, none of the two applications gives the possibility to report offensive posters or advertisements.

4.3 Research. Interviews

The subsequent step is *Interviews*. For the purpose of this project, users were recruited in a student residence (Colegio Mayor Nuestra Señora de África) and at the Faculty of Computer Science at the Complutense University of Madrid. All the users interviewed were women, aged between 18 and 25 years old. All of them were university students, from different backgrounds and studying different degrees. A total of 11 people were interviewed, as shown in table 4.3.

	Date	Time	Location	Interview record
Paula	30/10/2018	11:00	Colegio Mayor África	Video
Almudena	30/10/2018	12:45	Faculty of Computer Science	Audio
Raquel G.	30/10/2018	19:30	Colegio Mayor África	Video
Ana	31/10/2018	10:30	Colegio Mayor África	Video
Isabel	31/10/2018	10:50	Faculty of Computer Science	Audio
Raquel	31/10/2018	11:00	Colegio Mayor África	Video
Clara	31/10/2018	12:20	Faculty of Computer Science	Audio
Lucía	1/11/2018	15:20	San Martín de la Vega	Audio
Diana	1/11/2018	11:30	Via Skype	Audio
Carlota	2/11/2018	17:00	Colegio Mayor África	Video
Zoraida	2/11/2018	17:30	Colegio Mayor África	Video

TABLE 4.3: Interview planning

All interviews were recorded - video or audio -, with consent of the person interviewed. The template of the document of authorization can be found in appendix A. All of these recordings are kept private, because of their personal nature.

Before the interviews, a list of questions was written, so as to cover all aspects needed. The list of questions can be found in appendix B. It includes questions about feeling isecure in the streets, finding dangerous places in a city, reacting in case of emergency and general habits related to mobile phone usage.

The interviews took around a half an hour each.

The information obtained from these interviews was lateer summarized in a *list of factoids*. First, a list of unclassified factoid was extracted from each interview. Some of the unclassified factoids can be found in appendix C.

4.4 Modeling

Once all the factoids were extracted from the interviews, they were classified by types and frequency. The results of this classification can be found in table 4.4.

Because a *top-down* design process was chosen, the first step was defining *user categories*. The decission was made to focus on only one user type, which was defined as a *basic user*.

After the category was decided and the factoids were organized, a *persona framework* was created. This *persona framework* can be found in appendix D. And based on the *persona framework*, a *primary persona* was created, and it can be found in appendix E.[1]

Essencially, this persona is a young woman that lives in a big city and has a smartphone. She sometimes feels insecure when she is in the street, specially at night. She would like to feel safer, and be able to help other people, if possible.

4.5 Framework

SpreadApp has different functionalities, and each of them is associated with a different posture.

The *map*, and all the associated features require a *sovereign posture*: a user checks the *map* if they are looking for safe or unsafe places, checking the latest or nearest alerts or introducing information about a new dangerous spot or an offensive poster. In all these interactions, the user utilizes the system for medium to long periods of time, and directs the work flow.

Sending an alert requires a *transient posture*: a user activates the alert in case they are in danger. In this situation, they do not have time, nor possibility to pay attention to the system. The interaction is short and has to be as simple as possible.

Receiving an alert requires a *daemonic posture*: when a user receives an alert, they are not actively interactng with the system. The system runs in the background, as a daemon. The interaction is short, and the information must be displayed in a clear way.

These *postures* are taken into account while designing the system.

Given the on-the-go nature of the system, a mobile phone was the most suitable device, so the *input method* was necessarily a touch screen.

The functional and data elements were defined and grouped in an initial brainstorming session, in which the general structure for the system was decided. The functional elements were grouped in several screens, inside of which some of the elements were also grouped in smaller categories. The final hierarchy resulted as follows:

SCREENS

- **Emergency alert** (main)
  - SOS BUTTON
  - Map button
  - Settings
  - Alert history
- **Settings**
  - Contact list
  - Alert range
  - Profile data

[1] The photography used in the *persona* document is a work by Katalin Kleemann: https://www.flickr.com/photos/kataklee/34673444211/

    - Wallet data
        * Address
        * Password
- **Alert history**
    - Your own and other people's
- **Map**
    - Map
    - Add
        * Automatic
        * Manual
    - Dangerous places/posters pins
        * Dangerous place: different colors for different times of the day
        * Posters
    - Pin: display information
- **Point information screen**
    - Location
    - Date and time
    - Description
    - Picture (for posters and establishments)
    - Support button "**YES**"
    - Button for showing the point in the map (for the history)
- **Login**
    - Password
    - Accept button

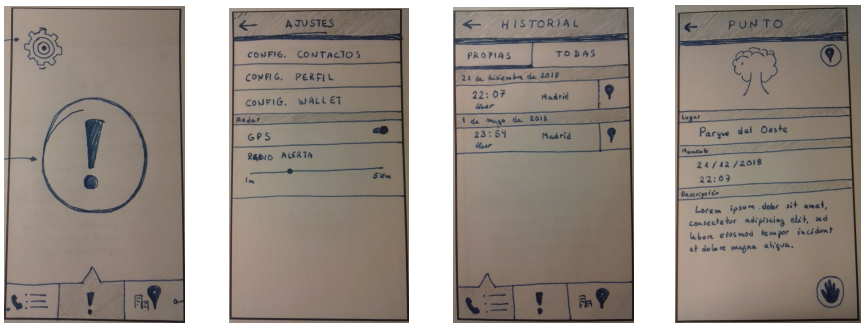
Once the data elements were defined, approximate dimensions of the interface were set, in order to start sketching the prototypes. Dimensions were calculated estimating a screen of $720mm \times 1280mm$, giving an aspect ratio of 0.5605. These are common dimensions in mobile phone nowadays and were only used for the prototypes (the final version would adapt to the actual dimensions of the device).

Several designs were sketched for each screen, shown in figures 4.1 and 4.2. After a brainstorming session based on the initial sketches, a general structure was laid out, as shown in figure 4.3. With these design decissions taken, the first paper mockup was made. The screens were sketched in paper and digitalized using *MarvelApp* [20].

This first paper mockup (figure 4.4) was tested with 5 users. The feedback obtained from these tests was taken into account in order to redefine some of the design elements and a second mockup was made. In particular, some major problems were detected, such as:

- Most of the users had problems finding the alert button, wich was a button in the main screen with the word "SOS". They explained that they thought it was an asset in the background, not and interactive button.
- Part of the users had problems finding the contacts. They explained that, given the icon for the profile tab, they thought it was more likely to find the contacts there.

These problems were corrected for the second prototype. Also, the map tab was interchanged with the alerts tab, since users seem to pay more attention to the central tab.



(A) Home (B) Settings (C) History (D) Info

(E) Log in (F) Profile (G) Map

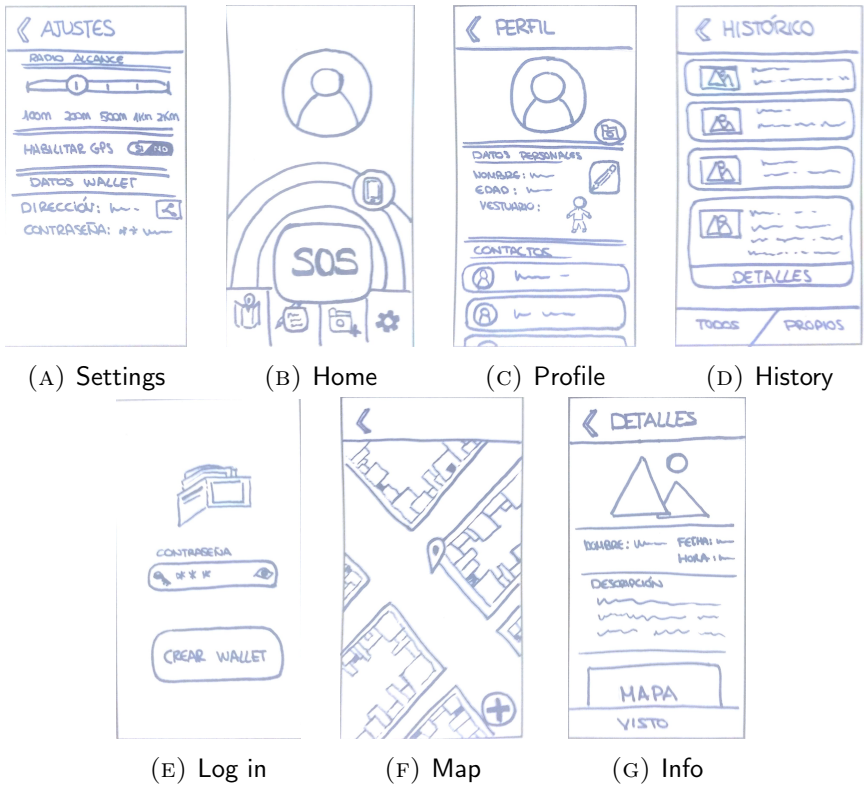
FIGURE 4.1: Preliminary mockup 1

The second mockup was a digital mockup built with *Pencil* [14]. This second mockup (figure 4.5) was tested with a different group of 5 users. The feedback obtained from these tests was taken into account in order to redefine again some of the design elements. The final design was implemented in Android.

Once the application was working, having been designed with the knowledge gained in the previous iterations, it was again tested with the users. It was tested with 9 different users, whose feedback was used to improve the interface and create the final version of the app. The feedback given by the users in this final testing can be consulted in appendix F.

Some problems that were detected in these tests and fixed in the final version were:

- In the *login* screen, most users pressed the *Register* button first, before entering the password. The screen was modified to make it more clear.
- Most users had trouble finding the way to add contacts. The section was labeled as "Grupos de Telegram" insted of "Contactos", so they found it confusing. That title was changed.
- The icon in the button for coloring the areas in the map was confusing for the users, so it was changed to a more intuitive one.
- Almost none of the users knew what the field "Santo y seña" was for, so the option to show information about it was added.
- When modifying the alert range, some users did not understand the existence of the option "Nadie", so information was added to clarify this.
- Some users suggested different types of information to be added in the profile, so an open field was added, for users to write down whatever they will.



(A) Settings (B) Home (C) Profile (D) History

(E) Log in (F) Map (G) Info

FIGURE 4.2: Preliminary mockup 2

The application was also tested by two experts in *usability*, that gave comprehensive feedback. They were asked to fill a template with the errors they found, according to Nielsen's 10 usability heuristics [23]. Namely:

- *N01.* Visibility of system status.
- *N02.* Match between system and the real world.
- *N03.* User control and freedom.
- *N04.* Consistency and standards.
- *N05.* Error prevention.
- *N06.* Recognition rather than recall.
- *N07.* Flexibilityand efficiency of use.
- *N08.* Aesthetic and minimalist design.
- *N09.* Help users recognize, diagnose, and recover from errors.
- *N10.* Help and documentation.

The feedback given by the experts can be found in appendix G. This feedback was later used to improve the interface for the final version of the app and remove the errors.

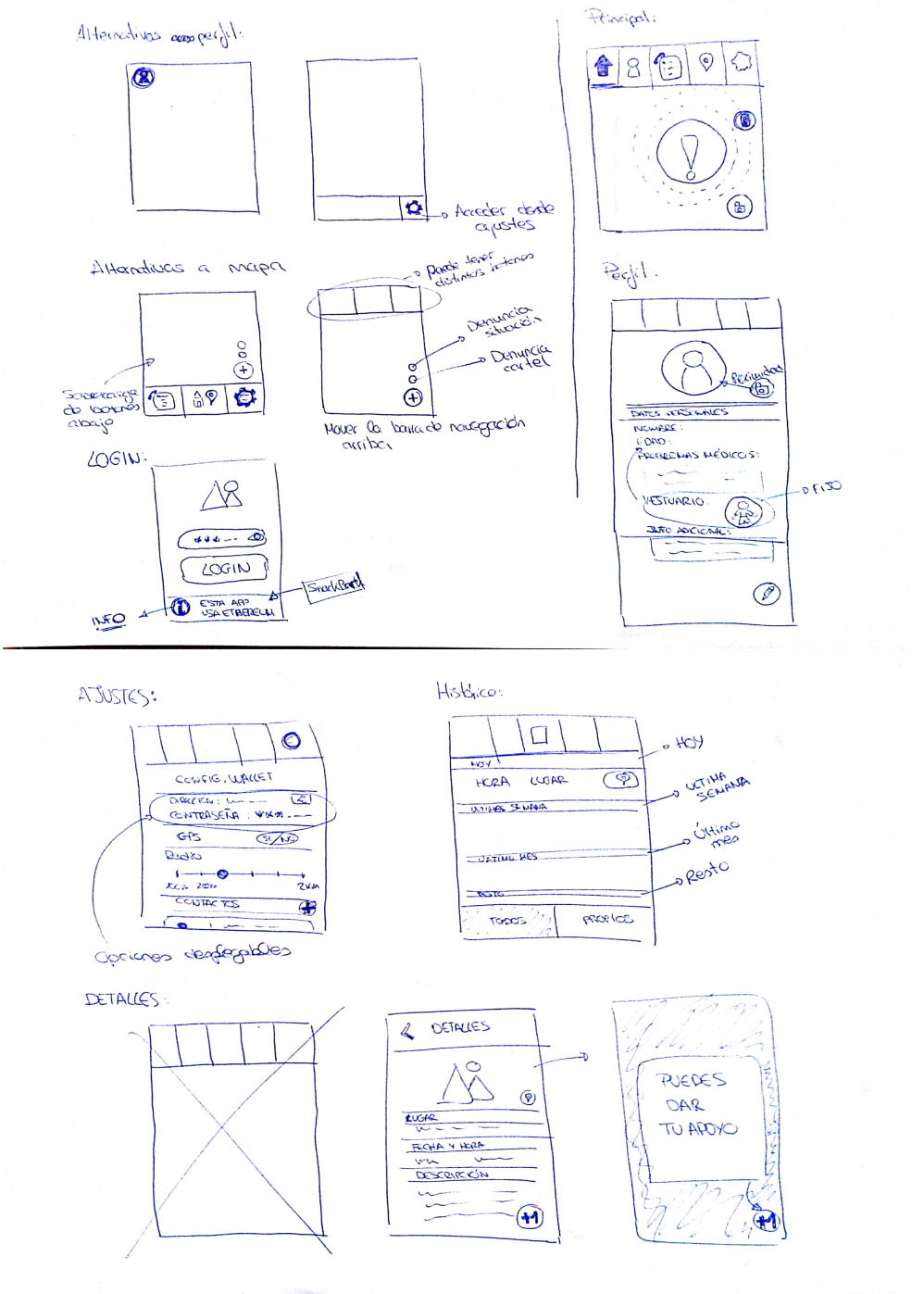
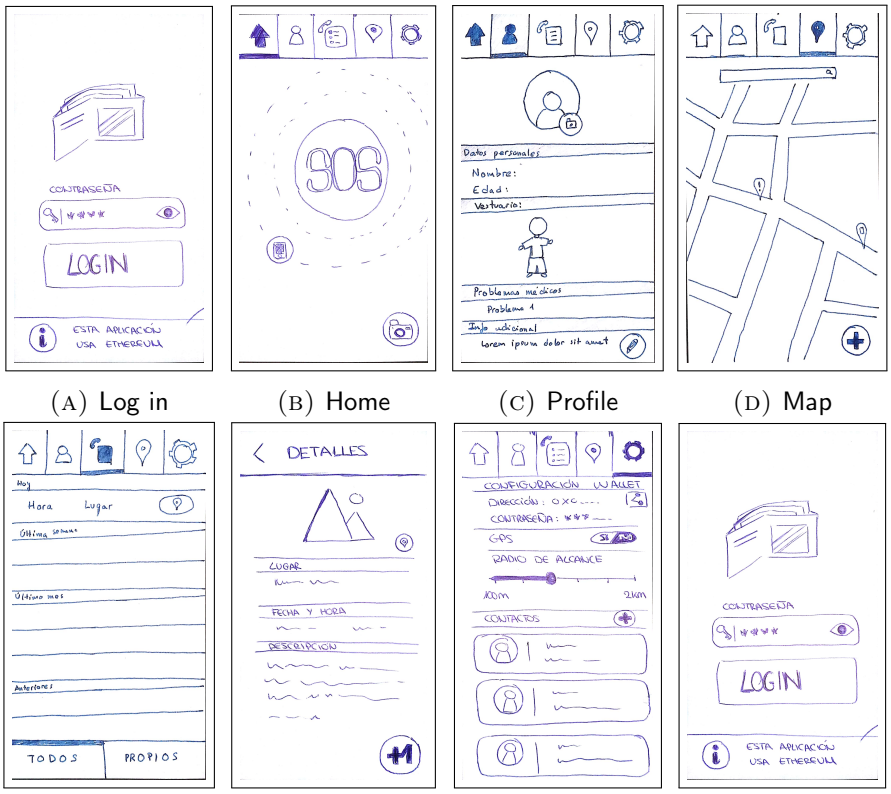


FIGURE 4.3: Initial interface screens



(A) Log in (B) Home (C) Profile (D) Map

(E) History (F) Info (G) Settings (H) Log in

FIGURE 4.4: First Mockup. Marvel.

(A) Log in (B) Home (C) Profile (D) Profile 2

(E) Map (F) History (G) Info (H) Settings

FIGURE 4.5: Second Mockup. Pencil.

Basic User Factoids	More frequent	Less frequent	Unusual
Dangerous areas	· Suburbs · Poor neighborhoods · Big cities · Lonely places	Areas with pubs or alcohol	· Public transport, underground · Drug selling areas
Why those areas	· Personal experience · Heard from close people	· News in the media · How people look	How streets and buildings look
Difference Madrid / smaller towns	Similar	Madrid is worse	In a smaller town, danger can be generalized
Not knowing the city	Ask people	Look up in forums/internet	People warn you before asking
Gender matters	Yes	Males can also be in danger, but more unlikely	-
The time of the day matters	Night	-	· No · Time of the year: summer
Have felt insecure in the street	Yes	-	No
Reaction in case of insecurity	Escape from the situation	Ignore	-
Reaction in case of emergency	· Escape from the situation · Look for help: people nearby, trusted people or emergency services.	Freeze because of fear	Defend themselves
Reaction seeing somebody in emergency (being with a group)	Take part	Call to ask for help	-
Reaction seeing somebody in emergency (being alone)	Call to ask for help	Take part	Nothing, if context unknown
Think of avoiding dangerous areas	Yes	No, usually with a group of people	-
How to avoid dangerous areas	Circulate crowded streets, even if it takes longer	-	-
Reporting posters			
Places where offensive posters are	· Establishments · Public transport stops	Business advertisements	· Street lamps · Bar menu
React against offensive poster?	· Yes, they should be removed · Haven't thought of doing something	No, people is too sensitive	-
Reaction against offensive poster	Tell to friends/familly	-	· Strip off, if possible · Talk to owner of establishment
Benefits of sharing posters	· More effective report · Visibilization	Awareness-raising	It would be needed to speak directly to the anouncer
System usage			
Share anonymous location	Yes	· Emphasize anonymity · Not with strangers	No
Help a nearby stranger	Yes	Yes, if able to take part	Yes, if police not there yet
Times not carrying phone	They always carry it	When it is charging	· When they are with more people · When she has an exam
Place for phone if not pockets	Hand	· Waist of trousers · Backpack · Purse	Case hanging on neck
Times activating GPS	· Occasionally · Certain applications, maps	Always	-
Why not activated	Privacy issues	Battery issues	Efficiency issues
Looking up new places	At home, before going out, computer and phone	· While going, phone · Asking people	Underground maps and similar sources
Puntos violeta	· Know they exist, but not what they are · Know what they are, but are not familiar	Know what they are and where to find them	Don't know what they are
Opinion about puntos violeta	Good idea	They should be everywhere	-
Spread existence and location of puntos violeta	Agree	-	-

TABLE 4.4: Classified factoids

Chapter 5

Implementation

In this chapter, details about the implementation of the app are given. The different parts in which the system is divided are introduced, followed by further explanation of one of the main functionalities, the *alert system*, which shall be explained in detail.

5.1 The application

For the purpose of this project, an Android application was generated. It was programmed in *Java*. The code for the application is open source[1] and it is available in the following *GitHub* repository:

`https://github.com/ElenaPT/SpreadApp`

As explained in previous chapters, *SpreadApp* is divided in two main functionalities: an *alert system* and a *map*.

The *alert system* focuses on sending and receiving emergency alerts. Using *SpreadApp*, a user is able to configure a series of personal data. In case of an emergency, the user can press a button and send those data, along with the GPS location, to other users of the app. Depending on the range the user sets for the alert, it reaches a certain number of nearby app users. It is also sent to the user's contact list. Finally, the information about the time and location is stored in an *Ethereum blockchain*, in order to be represented in the *map*. The application also receives the alerts sent by nearby users in danger, and allows the user to check the information that has been sent.

The *map* focuses on displaying and storing alerts that have already happened, giving a clear geographic representation of dangerous and safe areas. It allows the user to see the places where alerts have been sent, check the information of those alerts and see a graphic representation of the most dangerous areas, using *Voronoi diagrams*. It also allows the user to post alert situations and offensive posters found on the streets. All this information is stored in a *blockchain*.

In order to provide these functionalities, the app has been divided in several screens (as designed in 4.5). Each screen corresponds to one Android *fragment*. There is a total of 6 screens.

5.1.1 Log in

The first fragment corresponds to the *log in* screen. It only runs the first time a user opens the app after installing it. Since the app stores data in the *Ethereum blockchain*,

[1] The code is licensed under a MIT license

the user needs to have a *wallet* associated to the app, in order to be able to upload and read information from the *blockchain*. Therefore, the first time a user opens the app, they must create a new *wallet*. This screen asks the user to provide a new password. Once the user writes it and accepts, the new *wallet* is created in the phone and associated to the app.

There is an option for the user to read some information about the process while in this screen.

The user us requested to give the app permission to use the GPS on the device as it is required for its functioning. In case the permission is not given, the whole *alert system* can not be used, and most of the app remainis useless.

Before the app starts, a tutorial is desplayed, to explain the user the main functionalities of *SpreadApp*. It consists of five screens, each containing a short text and an iimage.

5.1.2 Home

This fragment corresponds to the *home* screen. It contains the main screen in the app, and the trigger for the alert system. Its main element is a big button with the icon of a megaphone that is used to send an alert.

There is also a small button with the drawing of a camera. This button allows the user to take a photo of a poster and store it on the *blockchain*. Once this information is uploaded, the poster will appear in the map, with the information provided by the user.

5.1.3 Profile

This fragment corresponds to the *profile* screen. It is the screen where the user's personal data is displayed and where it can be modified. Unlike most of the information stored by the app, this personal data is not stored in the *blockchain*, but locally. The management of this information is further explained in 5.2, in terms of its usage in the *alert system*.

The data the user can edit in the profile includes:

- Name
- Age
- Description
- Outfit
- Watchword system (Santo y seña)[2]
- Contact list

5.1.4 Map

This fragment corresponds to the *map* screen. It is the core of the map functionality. It uses Mapbox [19] to display a map with the points stored in the blockchain.

[2] *Santo y seña* are two sentences for the user who asks for help and the one who provides help to recognize each other. When the user who helps arrives to the location, they say the message in the field "Mensaje". The person asking for help recognizes the message and replies the message in the field "Respuesta". Once this exchange is done, they both know who the other person is.

When the user opens the app, the information stored in the *blockchain* is read and processed by the app. Each point is represented in the location given by its coordinates. There are two different kind of points, the *alerts* (represented by 5.1a) and the *posters* (represented by 5.1b). When the user presses one of the points, the name and a brief description are displayed.

(A) Alert pin

(B) Poster pin

FIGURE 5.1: Pins

There is a small button with a +, which allows the user to upload new points to the map, choosing between *alerts* and *posters*. This button is located in the bottom right corner.

There is another small button in the bottom left corner (represented by 5.2). This button divides the map in different areas, each one painted in a different color, depending on the level of danger that area represents. Safe areas (this is, areas whithout any points) are represented in green; medium alert areas (this is, areas whith one or two points) are represented in yellow; dangerous areas (this is, areas whith three or more points) are represented in red. This representation is based on a *Voronoi tessellation*, also known as *Thiessen polygons*.

FIGURE 5.2: Map areas

5.1.5 History

This fragment corresponds to the *history* screen. It complements the functionality of the *map* screen, by showing more detailed information of the alerts.

This screen displays a list of recent alerts. This list is implemented using a *RecyclerView*. The alerts and posters shown in this list depend on the current state of the map. Only the points that are visible on the map at the moment are listed. This means zooming in restricts the number of points shown and zooming out makes it bigger.

When the user presses the name of a point, more information is displayed. There is a small button that allows the user to view the point in the map when pressed.

5.1.6 Settings

This fragment corresponds to the *settings* screen. It provides the user the possibility to alter some configuration aspects.

When a user runs the app for the first time, they create an *Ethereum wallet*. The address, password and balance of the account is displayed in the *settings* screen. If the address is pressed, it is automatically copied to the clipboard.

There is also a section for configuring the range of the alerts. There are four possible configurations, from *no one* to *far*. When the range is set to *no one* the alert is only sent to the user's contact list. All the other options depend on the zip code, as it is further explained in 5.2.1.

There is an option to show the same tutorial that is displayed when the app is opened for the first time.

5.2 Alert system

The *alert system* is one of the two main functionalities the app provides. A user can send alerts to other users, and they can also receive alerts that other users send. These two situations (sending and receiving notifications) must be studied separately, since the pocesses are very different, although they are related.

5.2.1 Sending alerts

When an alert is sent, two main pieces of information are used:

- Profile data
- Location

Profile data are, as the name suggests, the data stored in the *profile* screen. The available fields are listed in 5.1.3. All of them, except for the outfit, are text fields that are stored as strings. The outfit, on the other hand, is represented as two arrays of possible colors, where ony one color can be active in each of them. The user interacts with the outfit by pressing a button that changes the colors in rotation. All this data is stored locally in a *SharedPreferences* object, that is saved every time the data are modified and charged every time the user starts the app.

Location is obtained from the device's GPS. It works as a service that runs in the background, even when the app is inactive.

In order to manage the exchange of alerts, the app connects to a *Firebase* API (*FCM*)[10] [11]. This API provides functionality to send notifications to a client via certain channels (*topics*). The way *SpreadApp* uses this is as follows: The location, obtained from the GPS, is constantly updated. Based on the location, the app calculates the corresponding *zip code*. When the user moves to a new location, the *zip code* is recalculated, the app disconnects from the previous channel and connects to the new one.

When a user presses the megaphone button in the *home* screen, a four step process is triggered:

1. The alert range is checked
2. The message is sent via *Firebase* [10] [11]
3. The message is sent via *Telegram* [33] [34]
4. The data is uploaded to the *blockchain*

Alert range and *Firebase*

The *alert range* can be set by the user in the *settings* screen. As the *profile data*, this is stored locally in a *SharedPreferences* object. There are four possible values for the alert range, each meaning the following:

- *Nadie.*[3] No alert is sent via *Firebase*. Only people in the user's contact list are notified via *Telegram*.
- *Cerca.*[4] The alert is sent to those users in the same *zip code* as the one activating the alert. This is, the alert is sent through a single *Firebase topic*.
- *Medio.*[5] The alert is sent to those users in a *zip code* sharing the first 4 digits with the one activating the alert. This is, a user sending an alert with a *medio* range from, assume, *zip code* 00000, will nottify every user in *zip code* 00000 up to 00009. Therefore, the alert is sent through 10 *Firebase topics*.
- *Lejos.*[6] The alert is sent to those users in a *zip code* sharing the first 3 digits with the one activating the alert. This is, a user sending an alert with a *lejos* range from, assume, *zip code* 00000, will nottify every user in *zip code* 00000 up to 00099. Therefore, the alert is sent through 100 *Firebase topics*.

Using *Firebase* it is only possible to send notifications through 5 *topics* at a time. Therefore, when *medio* of *lejos* ranges are activated, the alert is sent through 5 *topics* and the process is iterated with the subsequent *topics* until all of them are done.

Blockchain

After sending the alert to nearby users via *Firebase* and contacts via *Telegram*, the alert is stored in the *blockchain*, so it can be displayed in the map. This step can only be done if the user has enough *Ether* in their *Ethereum wallet*.

Although the *profile data* is sent to other users with the alert, no personal data is stored in the *blockchain*, for the sake of privacy. Consequently, the data stored is: location, date and time, title and description. In this automatic alerts, title and description are automatically created by the system. This data is written in the *blockchain* using a method *setAlerta* defined in the *Smart Contract*. It sends the information to the *blockchain* and after a while it is saved permanently.

This process takes a while, so ir runs in a parallel thread, so that the user can still interact with the application in the meanwhile. Once the data is successfully uploaded to the *blockchain*, the user is notified with a pop-up.

5.2.2 Receiving alerts

An alert can be received whether the app is open and active or not. In each of these cases, a different management of the alert has to be done.

If the app is not in the foreground, the receipt of the alert is managed by the Android notification system, which is low level and, hence, can not be modified. However, the type of notification that is shown to the user can be configured. In particular, a title, a

[3] Nadie ≡ Nobody
[4] Cerca ≡ Nearby
[5] Medio ≡ Medium
[6] Lejos ≡ Far

description and a small image are the available fields. The notification is displayed to the user. When the user presses the notification, the app is opened and all the information can be handled there.

If the app is in the foreground, the notification can be handled when it arrives. A flag is activated indicating a notification has arrived. Then, the same management that Android would do in the background has to be done in the application.

The message received consists of a title, a description and a small image (in this case, the one shown in figure 5.3). In the description field all the *profile information* is sent encapsulated in a *JSON* file. The reason for this is that only one set of data can be sent. It is not possible to send different fields as different parameters. So all the information is summarized in a single string of data. Since the model has been previously built for the needs of the application, it can be easily decomposed following the same criteria.

FIGURE 5.3: Alert icon

When the notification in opened in the app, the *detail* screen is displayed. This screen shows all the information sent with the alert:

- Location
- Name
- Age
- Free field (description)
- Outfit
- *Santo y seña*

Once an alert is closed, it can not be reopened, since the app does not keep a register of the received alerts.

Chapter 6

Testing results

Having finished the implementation of the application, it is necessary to do some testing, in order to ensure the main objectives have been achieved. In this chapter, the relevant results of this testing are presented and explained.

6.1 Testing with final users

At the end of the *GDD* process, the app was tested with 9 different users, which could potentially be final users of the app. The testers were people between 18 and 25 years old, both male and female. A total of 6 females and 3 males tested the app. Even though the app is primarly focused on women and the first interviews were only done with women, male testers would possibly bring new points of view, in terms of widening the usage of the application.

A transcription of the feedback given by the users can be found in appendix F. Most of the interactions are clear for most of the users, although some of them remain confusing. The most confusing interactions have been corrected in the final version of the app.

After the testing, the users were asked to fill a survey related to the user experience while testing the app. The survey consisted of 8 questions to be answered in a *Likert scale* ranging from 1 (Strongly disagree) to 7 (Strongly agree), and 2 open questions, as follows:

1. The interface is intuitive. It is easy to know what to do.
2. The interface is friendly.
3. The icons are unexpected and hard to understand.
4. The app is useful.
5. I trust the information the app publishes.
6. The app covers the tasks I expected it to cover.
7. The app makes me feel I lose privacy.
8. It is easy to make mistakes while doing tasks with the app.
9. If you had any dificulties carrying out a task, please tell us:
10. Would you add anything to the app? If so, what?

The results of this survey are detailed in table 6.1.

Analizing this results it is inferred that users have not found major problems while using the app. They find it quite intuitive, friendly and easy to use. So, in terms of user experience, the results are positive.

	1	2	3	4	5	6	7	8	9	Average
1	6	7	6	6	6	6	6	6	6	6.11111
2	7	7	7	6	7	7	6	6	6	6.55556
3	1	1	1	2	2	6	2	3	2	2.22222
4	6	6	7	6	7	7	7	6	7	6.55556
5	5	5	6	6	6	6	7	7	5	5.88889
6	7	6	7	6	7	6	7	7	7	6.66667
7	1	4	3	2	1	2	2	1	1	1.88889
8	2	2	6	2	1	1	2	2	2	2.22222

TABLE 6.1: Survey results. Each row represents a question and each column, a user.

When asked about privacy in question 7, an overall positive feedback was received. The most positive one in the whole survey. This represents a success, since privacy was one of the main issues in the development of the app. Providing the users with a safe environment, in which they feel safe is crucial for the good functioning of this initiative.

Question 5, on the other hand, has the least positive result, although it is undoubtedly good. This question asks about the level of confidence the users have with the information they can look up in the app, and it is understandable that some of them are a little hesitant to fully trust the information. Most of the information published in the app comes from other users that post it. We believe that the fact that users have to pay to publish information in the app will make this information reliable at the end of the day, since no user would want to use money to upload irrelevant or false information. In any case, only with time will an active community be built and, when that happens, the app will be a trustworthy source of information.

6.2 Blockchain and payments

One of the main issues that was faced during the development of the app was the fact that transactions in Ethereum need *gas* to be issued, thus costing money to the issuer. This seemed like a major problem, because asking for help - which is the main goal of this app - should never be a luxury item one has to pay for.

In this initial phase, no external funding exists for the app, so all the money that is spent should come either from the creators or the users.

The final outcome was to keep some of the services free, while others do require Ether to be used. In particular, publishing points in the map requires a payment, but every other feature does not. This is, a user can ask for help to nearby people and contacts and check both the map and the alerts history without having to pay.

This has a positive effect. On the one hand, the basic services of the app, which are asking for help and looking up information remain free for the user. On the other hand, publishing has a cost, which can disuade possible "vandals" from publishing unnecessary information in the app, while aware people or collectives will still have the possibility to spread they knowledge if they will.

Chapter 7

Conclusions and Future Work

Regarding all the previous work explained in prior chapters, and specially the results analyzed in chapter 6, conclussions are given and some suggestions are laid out to be carried out in the future.

7.1 Conclussions

The outcome of this project has been a functioning application that provides functionality, mainly, for:

- Sending and receiving alerts in case of emergency.
- Storing information about dangerous places and areas.
- Storing information about offensive posters or advertisements.
- Displaying this information in a map and also in a list.

Given the results analyzed in chapter 6, it can be concluded that the main goals the project attempted to achieve were met in the following way.

7.1.1 Functionality

The main objective of this project was creating a system that provided:

1. Functionality to ask for help and receive notifications from other people asking for help.
2. Functionality to report dangerous spots (and offensive posters), and check the safest places and areas around a certain location.

Both of these functionalities aimed for a better and more secure environment for women to feel safe in the streets.

Both of these functionalities have been succesfully developed and are currently working. As suggested by user feedback, it provides a feeling of safety.

Even one of the functionalities was develoed further than initially intended. At first, notificationas were only intended to be sent through the app and, in the end, integration with Telegram has been developed, thus providing an extended functionality.

The application provides a set of functionalities that was not present in any other application in the *state of the art* (Chapter 4, section 4.2). None of the applications analyzed provided both an *alert system* and a *map* with dangerous spots. So the application is innovative in that sense. It is also the only one that uses *blockchain* for storing the data.

7.1.2 Blockchain

Another core matter in this project was taking advantage of the possibilities blockchain offers as a technology. At first, it was proposed that blockchain could be used in this context, to enhance the opportunities of developing a system of this nature.

The now existing system has been developed based on these ideas. The system is working and blockchain provides the anonimity, privacy and solidity that was intended from the beginning.

Users are aware of their privacy and anonimity, and *blockchain* intrinsically provides this. Therefore, the users have no need to trust the application or its creators, but the technology itself. Since the users' personal data is not stored in any database — only in the user's phone — no personal data management has to be done, and no privacy is lost in the process.

On top of that, information sent to the *blockchain* is permanently stored, and can not be modified nor deleted. So everytime a user wants to access the information published, they will be able to do so. This, for instance, can be useful in cases of criminal report: the user can provide irrefutable proof that the alert was sent in a certain point of time.

7.2 Limitations and future work

The system has still some limitations, some of them related to the implementation. And there is work that can be done in order to improve this. Namely:

- *Develop a system for the posters.* At this point, a user can upload a poster, but nothing else can be done. It would be useful to have a more developed system that actually allowed users to show support and do actual collective denunciations.
- *Develop a more general system to help interaction among users.* A sort of social network that allows users to create events or initiatives, and support them. And, in case of events, detail the place, time, date, and all information necessary, and let users confirm their attendance.
- *Develop a good funding system*, to avoid payments in the app.
- *Make the transaction system and the interaction with Ethereum transparent to the user.* At this moment, if a user wants to publish something in the app, they have to buy Ether with their wallet or transfer the money from a preexisting one. This process is not intuitive at all for a person that is not used to interacting with cryptocurrencies. It makes some functionalities more obscure and less intuitive for the user.
- *Get rid of the wallets using* metatransactions. This is, having a central wallet for all the system, so users do not have to worry about managing their own. The payments are centralized, thus simplifying the interaction for the users. Payments are not made by the users, but by the central wallet. The users issue the transactions and the central wallet manages them, uploading them to the *blockchain*.

  This does not contradict the fact that the system is decentralized. Only the wallet and the payments are centralized. The rest of the system and the data storage, which are the main concerns, are decentralized. Using this system, it is easier to make the interactionas straightforward for the user, and also to test different ways of funding, since money is only handled by the central unit.

- *Improve the alert system in terms of location.* At the moment, two users are said to be "near" if they share the same *zip code*. But this is not a good measure of distance. A possible improvement could be dividing the world with a fine grid, and assuming two users are "near" if they are in the same square. And doing all the other calculations following the same idea.
- *Add more functionality to the map.* For example, a button to show the current location or a search bar to look for specific locations.
- *Improve the alert history.* It could be useful to give the option to show only the user's alerts or all the alerts in the area. More filters can be added to narrow the search.

There are other limitations that are due to the technology itself. Some of them include:

- If a user publishes useless or false information and it is stored in the blockchain, it will be permanently stored along with the truthful information. An arbitration system could be implemented, to ensure the information is trustworthy before sending it to the blockchain. But this is also problematic because it elliminates part of the decentralization, and because it is not always possible to check whether a certain piece of information is truthful or not.
- The data that can be sent via *Firebase* has a limitation in size. Therefore, heavy information fields, such as a profile picture, can not be sent in the alerts.

Appendix A

Document of authorization. Template.

AUTORIZACIÓN ENTREVISTA
(Vídeo)

Sr/Sra _____,
con DNI _____ autorizo a _____
_____ a grabar en vídeo las respuestas dadas en la entrevista para la fase de captura de requisitos del TFG. El contenido de los vídeos no se hará público. Los vídeos se emplearán para analizar las respuestas dadas por la persona entrevistada con mayor nivel de detalle.

Fdo.

AUTORIZACIÓN ENTREVISTA
(Audio)

Sr/Sra _____,
con DNI _____ autorizo a _____
_____ a grabar en audio las respuestas dadas en la entrevista para la fase de evaluación del TFG. El contenido de los audios no se hará público. Los audios se emplearán para analizar las respuestas dadas por la persona entrevistada con mayor nivel de detalle.

Fdo.

Appendix B

Interview questions

We are developing a mobile app to help and prevent sexual harassment and abuse in the streets. Therefore, we will ask you some questions about these issues and set out some case study scenarios. Since this is a very sensitive subject, feel free to reject answering some questions, if you don't feel comfortable with them.

B.1 Questions oriented to work flows

- Have you ever felt insecure in the street? If so, why? How did wou react?
- When you go out in the evening/night, do you consider if you are going to pass through a conflictive area? How would you obtain that information? (Or how do you obtain it, in case you do). Do you think the time of the day affects this fact? How?
- Imagine you go out and a guy is disturbing you: he tells you some things and doesn't leave you alone. What would you do?
- Now imagine you go out and see a girl in trouble, or who needs help. What would you do?
- Have you ever seen any poster or advertisement that you thought was offensive towards your gender? Did you do anything about it?
  If so, what?
  If not, why? Who do you think would be able to do something?
- Have you ever experienced an emergency situation in the street?
- Do you usually have GPS activated in your phone? Why?
  If not, when do you turn it on?

B.2 Questions oriented to attitudes

- Do you think there are places in Madrid that are dangerous if you pass through them without being with a group of people? What places? Why? How do you know that place is not safe (or how did you find out)? Is it dangerous any time of the day? When?
- (In case you are not from Madrid) Do you think there are places in your city/town that are dangerous if you pass through them without being with a group of people? What places? Why? How do you know that place is not safe (or how did you find out)? Is it dangerous any time of the day? When?
- If you move to a new city you don't know, how would you find out about this (or how did you find out when you did)?
- Do you think your gender affects the areas you choose to circulate?
  If you belonged to the opposite one, would you think in different areas? Explain to what extent.

B.3 Questions oriented to objectives

- Imagine you have to go (or move) to a new place you don't know. What information source would you choose to orient yourself? [guide, internet, asking other people...]
- In case it is a digital source, would you rather look it up from your mobile phone or your computer?
- In case you are in an emergency situation in the street, what is the first thing you would do?

  In case you would alert someone, who would it be? How would you contact them? What if they are not nearby?

B.4 Questions oriented to the system

- Would you be comfortable sharing your location anonymously if that allowed you to help and be helped faster?
- Imagine you can know if there is somebody sear you (geographically) that needs instant help, would you go even if you didn't know the person?
- Dou you think collecting lots of offensive posters would help removing them from the streets?
- When you go out, do you always carry your mobile phone? When do you not carry it?
- If your clothes have no pockets, what do you do with your mobile phone?
- Do you know the "Puntos Violeta"? (If they don't know, explain)

  If so, do you know where are they located? How do you know?

Appendix C

Some unclassified factoids

C.1 Person C. Factoids

C.1.1 Attitudes

- *C* considera que Madrid hay sitios inseguros, pero no conoce lo suficiente la ciudad como para afirmarlo.
- *C* cree que los lugares son más peligrosos por la noche porque hay menos gente.
- *C* considera que en su ciudad hay barrios peligrosos.
- *C* opina que un sitio en el que venden droga es más peligroso que una calle principal.
- En general, *C* no se siente demasiado insegura en ningún barrio de su ciudad.
- *C* suele evitar los barrios peligrosos en su ciudad.
- *C* sabe que un barrio es peligroso porque hay menos gente y por el aspecto de las casa y las personas.
- *C* considera que en general todo es más peligroso de noche.
- *C* se entera de que un sitio es peligroso preguntando a gente conocida.
- *C* cree que las zonas peligrosas son peligrosas para cualquier persona, pero el hecho de ser mujer lo acentúa.

C.1.2 Work flows

- *C* nunca se ha sentido insegura en la calle.
- *C* no suele ir sola por sitios peligrosos de noche.
- Cuando *C* sale de noche, suele tener cuidado con los lugares por los que pasa.
- *C* toma una ruta más larga, si eso supone pasar por zonas más concurridas.
- Si a *C* un chico la molesta, lo ignora hasta que se vaya. Si no funcionara, le diría algo.
- Si *C* ve que una chica necesita ayuda, cree que la ayudaría, pero como nunca se ha encontrado en la situación, no sabe. O llamaría a la policía.
- *C* ha visto carteles ofensivos hacia su género.
- *C* ha visto cartas de bares con mujeres ligeras de ropa.
- Si *C* está con gente cuando ve un cartel ofensivo lo comenta con la gente que está con ella.
- Si *C* está sola cuando ve un cartel ofensivo, no hace nada.
- *C* nunca ha estado en una situación de emergencia en la calle.
- *C* suele tener desactivado el GPS en el teléfono.
- *C* activa el GPS en el teléfono para ciertas aplicaciones, por ejemplo para los mapas.

C.1.3 Goals

- Si *C* no sabe cómo llegar a un sitio, busca la dirección por internet, luego busca cómo llegar y después va. (Busca todo antes de irse)
- *C* busca información sobre cómo llegar a un lugar en el ordenador antes de irse.
- Cuando *C* usa Google Maps suele ser en el móvil porque se pierde por la calle.
- *C* no usa mucho Google Maps porque conoce bien su ciudad y en Madrid los planos de metro son muy buenos.
- Si *C* se encontrara en una situación de emergencia en la calle, ayudaría si puede y si no llamaría a la policía o a la ambulancia.
- Si *C* se encuentra en una emergencia, llamaría a sus padres. O a una ambulancia, según la situación.

C.1.4 System

- *C* estaría dispuesta a compartir su ubicación a fin de ayudar y ser ayudada, si es de manera anónima.
- Si *C* sabe que alguien cerca de ella necesita ayuda, acudiría, aunque no conozca a la persona.
- *C* cree que acumular quejas o denuncias sobre un anuncio ofensivo podría ayudar, porque la opinión de muchos tiene más fuerza que la de uno solo.
- *C* casi siempre lleva el teléfono consigo cuando sale a la calle.
- Cuando *C* vivía con su familia, a veces no llevaba el teléfono consigo. Por ejemplo al ir a clase.
- Desde que *C* vive sola, siempre lleva el teléfono consigo.
- *C* sabe lo que son los puntos violeta.
- A *C* le parece que los puntos violeta son una buena idea.
- *C* cree que al entrar al recinto del evento, los puntos violeta deberían estar indicados en un mapa o similar. Si no, presuntaría a alguien dónde están.

Appendix D

Persona framework

Basic App User Age : 16 - 45 years old Gender : Female	
Description	Young person who allways carries her smartphone. She has a job or is still a student. She lives in a city and has at some point felt unsafe walking alone on the street.
Training	• High School Diploma or BSc. • Not specific training in informatics. • She knows how to use a smartphone at a basic level: look up maps, information, etc.
Technology usage	• She always carries her smartphone with her. • She usually looks up maps and paths when she goes out. • She normally has the GPS off in her phone, but she turns it on when she thinks it is usefull or necessary. • She would turn her GPS on, if that allowed her to help or be helped.
Problems	• She has felt insecure in the street more than once. • When she circulates the streets at night, she feels safer if she is not alone, to avoid problems. • She tends to avoid lonely streets at night. • When she gets to a new city, she does not know what areas are the the most dangerous. • If she comes across an emergency situation in the street, she does not know who to turn to.
Goals	• She wants to know the unsafe areas of a city before going out, to be able to avoid them. • She wants to feel safe when she is in the street, whatever time it is. • She wants somebody close to her to help her if she needs it. • She wants to help whoever in need. • She wants harassment problems in our country to lessen. • She does not want to depend on others to go out and go back home at night.
Additional information	• She has heard about "Puntos Violeta", and she believes it is a good idea that should be generalized.

Appendix E

Primary persona

Sara Pereira

Basic user

"Better alone than in bad company"

Personal information

Description : **Sara Pereira Álvarez**, *25 y/o*, Portugalete, Vizcaya, Spain.
Sara studied Political Sciences in UPV, and she has recently moved to Madrid to study a Master's Degree in Political Analysis in UCM (Somosaguas). She shares a flat with two university colleagues.

Work : **Student**.
Sara does not work, and she studies her Master full-time. She attends her classes in the afternoon, so every day she leaves home at noon and she is at university untill 21:00. She always carries her Samsung Galaxy A5 with her.

Routine : **Home and university**.
Since Sara's classes are in the afternoon, she uses the mornings going to the supermarket and stocking up on food. She tends to stay at home most of the mornings, studying. She goes to university at noon, and usually arrives back around 22:15. Some weekends, she goes out for dinner with her friends.

Goals, desires and motivations

Sara is not used to living in a city as big and crowded as Madrid, so she is worried about her own safety in some contexts. Every day, she has to travel in the underground at least a couple of hours and, since her classes end late, during the return trip the wagon is quite lonely, so that unsettles her a bit. In addition, when she has dinner out with her friends and goes back home, late, from time to time she finds herself moving around streets she does not know, ignoring if there is a better path she could follow. She would like to have a way to find the safest path. Furthermore, once she was in a bar and a guy adressed her violently and her friends helped her. She doesn't know what she would have done if she had been alone.

Knowledge and skills

Sara handles digital applications with skill on a daily basis. She is a person with higher education and, although she does not have advanced knowledge in informatics, she has a good command of the basic tools she handles eveery day.

Appendix F

User Feedback and interactions

F.1 Introduction

During the testing of the final version of the app, the users were asked to complete several interactions using the app. These interactions, along with the users' answers are here described in detail. This is, the difficulties they found, comments, suggestions, and everything useful for the design process.

F.2 Interactions

F.2.1 First interaction

Case study

When the app is first opened, the register screen is shown. What would you do to register and start the app?

Expected answer

The user is expected to enter a password and then press the *Registrarse* button (or the *Done* button in the keyboard. When, right after that, the app asks for permission to access the GPS location, the user is expected to give the permission.

User answers

User 1 She writes a password and presses the *Done* button in the keyboard. She is surprised by the fact that the keyboard is not hiding automatically. She gives the permission without hesitation ("Obviously, or else it will not be able to provide your location").

User 2 He introduces a password and presses the *Done* button in the keyboard. He gives the permission without hesitation. He thinks the design is lovely.

User 3 She presses the *Registrarse* button. Then, she presses the *Contraseña* text field, makes the password visible and writes one. It is too short. She has to close the keyboard manually to see the error. Once she reads it, she makes the password longer and presses the *Done* button in the keyboard. She gives the permission without hesitation.

User 4 She presses the *Register* button, and she does not understand why it is not working. Later, she enters a password and presses the *Done* button in the keyboard. She gives the permission without hesitation.

User 5 She enters a password and presses the *Done* button in the keyboard. She gives the permission, after some doubt.

User 6 She presses the *Registrarse* button ("Oh, I thought you should press Register first"). Then she enters a password and presses the *Done* button in the keyboard. She gives the permission without hesitation.

User 7 He enters a password and presses the *Done* button. He gives the permission, with some doubt.

User 8 He presses the *Registrarse* button. Then he presses the snackbar and opens the information. Eventually, he introduces a password and presses the *Done* button. He gives the permission without hesitation.

F.2.2 Second interaction

Case study

Imagine one day you go out in the evening and meet someone. At first you are comfortable, but at some point you want to leave. You tell the person and then they stop being as nice as before, and they even become violent. You are afraid to leave on your own, so you want to use the app to ask for help. How would you do it?

Expected answer

The user is expected to press the megaphone icon in the *Home* tab.

User answers

User 1 She presses the megaphone icon quickly. When the alert is sent, she is puzzled by the fact that the error message "Ha habido un error en la transacción" is displayed.

User 2 He presses the megaphone icon quickly. He realizes the message "Ha habido un error en la transacción" is displayed.

User 3 She presses the megaphone button quickly.

User 4 She presses the megaphone button quickly.

User 5 She goes to the *Ajustes* tab and checks the alert range. She moves through the rest of the tabs and eventually, in the *Home* tab, she presses the megaphone button.

User 6 She presses the megaphone button quickly. She sees the message "Ha habido un error en la transacción" and she thinks the alert has not successfully been sent.

User 7 He presses the camera button. Then, he moves to the *Avisos* tab. Then he moves to *Mapa*, where he presses the button + and he points at the *Alerta* option. Eventually, he goes back to the *Home* tab and presses the megaphone button ("It looked like it was the background").

User 8 He moves briefly through the tabs and quickly goes back to the *Home* tab and presses the megaphon icon. He mentions that there has been an error, when he sees the message "Ha habido un error en la transacción".

F.2.3 Third interaction

Case study

By doing the previous interaction, the alert is sent to people geographically next to you. But you would like to notify some of your friends or family, since it makes you feel safer. What would you do to configure your contacts in the app?

Expected answer

The user is expected to open the *Perfil* tab and press the edit button next to the title "Grupos de Telegram".

User answers

User 1 She changes for a second to the *Ajustes* tab, but before it has even loaded, she goes to *Perfil*. Then she moves through all the tabs. After telling her that the message will be sent via Telegram, she goes back to *Perfil* quickly and presses the edit button. She explains that she does not use Telegram.

User 2 He moves to the *Ajustes* tab first and tight after that he moves to *Perfil*, where he quickly presses the edit button next to the title "Grupos de Telegram". He enters a group name and a random id and creates a contact.

User 3 She moves quickly to the *Perfil* tab, without hesitation, and sees the section "Grupos de Telegram". She presses the edit button next to the title.

User 4 She moves first to the *Ajustes* tab. Then she moves through the tabs, one by one until she reaches *Perfil*. She points at the section "Grupos de Telegram" and she presses the button by the title.

User 5 She moves to the *Ajustes* tab and immediately to *Avisos*, where she stops. She keeps moving through all the tabs. When she is told that the message is sent via Telegram, she opens the *Perfil* tab and quickly presses the edit button next to the title "Grupos de Telegram". She explains that she does not use Telegram.

User 6 She moves to the *Perfil* tab, finds the section "Grupos de Telegram" and she presses the edit button next to the title.

User 7 He moves to *Perfil* and presses the edit button next to the title "Grupos de Telegram".

User 8 He moves to the *Perfil* tab and presses the edit button next to the title "Grupos de Telegram".

F.2.4 Fourth interaction

Case study

It seems that nobody is coming and you keep waiting, so you want to make the range of the alert bigger, even if that means help will come from a further place. How would you do that?

Expected answer

The user is expected to open the *Ajustes* tab and there adjust the range by moving the corresponding bar.

User answers

User 1 She opens the *Ajustes* tab and moves the bar to the rightmost option.

User 2 He goes quickly to the *Ajustes* tab and moves the bar to the rightmost option.

User 3 She moves to the *Map* tab and says that she would choose the area there. After being told that that is not the way, she moves to the *Avisos* tab, then to *Home* and immediately to *Perfil*. After looking through the tab for a few seconds, she moves to the *Ajustes* tab and quickly sees the bar and moves it to the rightmost option.

User 4 She moves quickly to the *Ajustes* tab and modifies the range. She asks how the ranges work.

User 5 She moves quickly to the *Ajustes* tab and moves the bar to the rightmost option.

User 6 She moves to the *Mapa* tab, zooms in and interact with the pins in the map and with the + button. Then she goes back to the *Perfil* tab. She moves through all the tabs until she reaches the *Ajustes* tab, where she quickly moves the bar to the rightmost option.

User 7 He opens the *Ajustes* tab quickly and moves the bar to "Medio".

User 8 He opens the *Ajustes* tab and moves the bar to "Medio".

F.2.5 Fifth interaction

Case study

You walk next to a bar or a bus shelter and you notice a poster that seems inappropriate, because it may be offensive to some minority group, or for some other reason. You want to share it in the app to file a collective complaint. How would you do that using the app?

Expected answer

There are two possible options that are considered valid in this case:

- The user is expected to press the camera button in the *Home* tab and, in the dialogue that appears, they select the camera.
- The user is expected to open the *Map* tab, press the + icon, and select *Cartel*.

User answers

User 1 She moves quickly to the *Map* tab. She presses the + icon and then points at *Cartel*.

User 2 He moves to the *Map*, zooms in a particular area, presses the + button, selects *Cartel* and fills in the fields. He has difficulties while filling in the *Nombre* field because when he presses enter, a new line is introduced, instead of changing to the next field. Right after that, he presses the *Publicar* button.

User 3 She moves quickly to the *Mapa* tab. After thisking for a couple of seconds, she presses the + button, selects *Cartel* and she tells that she would fill in the data.

User 4 She moves to the *Home* tab and presses the camera button. She fills in the fields. She has difficulties while filling in the *Nombre* field because when she presses enter, a new line in introduced, instead of changing to the next field. Right after that, she presses the *Publicar* button.

User 5 In the *Home* tab, she presses the camera button. She tells that she could also do it at the *Map* tab, so she changes to the *Mapa* tab, presses the + button and selects *Cartel*.

User 6 She moves to the *Home* tab and presses the camera button. She fills in the data. She has difficulties while filling in the *Nombre* field, because when she presses enter, a new line is introduced, instead of changing to the next field. After hiding the keyboard, she presses the *Publicar* button.

User 7 In the *Home* tab, he presses the camera button and tells that he woud do it there.

User 8 He moves to the *Home* tab and presses the camera button. He says that he would do it there.

F.2.6 Sixth interaction

Case study

You would like to know if there has been any alert situation next to where you are, or if someone has reported any incidents. How would you find this information in the app?

Expected answer

The user is expected to move to the *Map* tab and interact with it in some way.

User answers

User 1 She presses the + button and points at *Alertas*. After being told that that is for adding new alerts, she points at the map ("and that's it").

User 2 He stays at the *Mapa* tab and presses one of the pins in the map.

User 3 she stays in the *Mapa* tab and presses some of the pins in the map.

User 4 She changes quickly to the map and moves around it. She asks for the meaning of the pins in the map.

User 5 She tells that she would find it in the map.

User 6 She changes quickly to the *Mapa* tab and moves around it.

User 7 He moves quickly to the *Mapa* tab.

User 8 He moves quicly to the *Mapa* tab.

F.2.7 Seventh interaction

Case study

You have already checked your area, but now you would like to see if there are parts of Madrid that are more dangerous than others. What would you do to get this information to be showing in the map?

Expected answer

The user is expected to stay in the *Mapa* tab and press the button in the bottom left corner.

User answers

User 1 She presses some of the pins that are locate in the map. Then she presses the icon in the bottom left corner.

User 2 He quickly pressess the button in the bottom left corner. He likes how the resulting map looks like.

User 3 She quickly presses the button in the bottom left corner. She likes the functionality a lot, and also how it looks.

User 4 She quickly presses the button in the bottom left corner. She believes red is an intuitive colour for representing the dangerous areas. She asks how many pins make an area become red. She doubts whether green or yellow represents the safest areas.

User 5 She quickly presses the button in the bottom left corner. She presses some of the pins in the map, and the corresponding information is displayed.

User 6 She presses the + button and then she selects *Alerta*. Then she does the same with *Cartel*. then she moves to *Ajustes*, then to *Avisos* and she keeps searching in the rest of the tabs. After a while, she states that the functionality should be available in the map, so she goes back to the *Mapa* tab and she presses the button in the bottom left corner. She says that the icon is not very intuitive, so she did not want to press it ("I didn't know what the arrows were and I didn't want to press it, just in case").

User 7 He quickly presses the button in the bottom left corner. He zooms out to have a better look at the areas.

User 8 He quickly presses the button in the bottom left corner and he moves around the map.

F.2.8 Eighth interaction

Case study

You are no longer interested in the area where the alerts are, but you want to find the most recent ones. How would you look it up?

Expected answer

The user is expected to move to the *Avisos* tab.

User answers

User 1 She quickly moves to the *Avisos* tab.

User 2 After a couple of seconds, he moves to the *Avisos* tab.

User 3 She quickly moves to the *Avisos* tab.

User 4 She quickly moves to the *Avisos* tab.

User 5 She quickly moves to the *Avisos* tab.

User 6 She hesitates for a second and then moves to the *Avisos* tab.

User 7 He quickly moves to the *Avisos* tab.

User 8 He quickly moves to the *Avisos* tab.

F.2.9 Ninth interaction

Case study

Scrolling through the alerts, you find one that catches your attention for some reason and want to check more information about it. How would you do that?

Expected answer

The user is expected to move to the *Avisos* tab and press one of the alerts that are displayed.

User answers

User 1 She quickly presses the title of one of the alerts and the information is displayed.

User 2 He quickly presses the title of one of the alerts and the information is displayed. He presses the title again to hide it.

User 3 She quickly presses the title of one of the alerts.

User 4 She quickly presses the title of one of the alerts. She then presses the map icon and the location of the alert is shown.

User 5 She quickly presses the title of one of the alerts. She presses the title again to hide the information.

User 6 She quickly scrolls through the alerts and she presses the title of one of them.

User 7 He presses the title of one of the alerts. He presses it again to hide the information.

User 8 He quickly presses the title of one of the alerts and the information is displayed.

F.2.10 Tenth interaction

Case study

When you send an alert, the app sends your GPS location, along with other personal information that you can configure. Where would you configure that information?

Expected answer

The user is expected to move to the *Perfil* tab and possibly modify some information.

User answers

User 1 She quickly moves to the *Perfil* tab and tells that she could modify her name, age or clothing.

User 2 He moves for a second to the *Ajustes* tab and immediately after he moves to the *Perfil* tab and points out the information he would change, while pressing the edit button.

User 3 She moves quickly to the *Perfil* tab and modifies the clothing.

User 4 She quickly moves to the *Perfil* tab.

User 5 She quickly moves to the *Perfil* tab. She presses the edit button to edit the data.

User 6 She quickly moves to the *Perfil* tab and poinits at the fields.

User 7 He quickly moves to the *Perfil* tab. He presses the edit button to modify the presonal data. He also modifies the clothing.

User 8 He quickly moves to the *Perfil* tab.

Additional questions

Question 1 What do you think the field *Santo y seña* is?

- **User 1:** A warning that you are asking for help, so that it does not seem obvious.
- **User 2:** In case you do not want to tell your name, a way for people to know who you are.
- **User 3:** Something the other person tells you so you can identify they are the one coming to help. And then your answer. (She finds it a good idea)
- **User 4:** N/A
- **User 5:** N/A
- **User 6:** Like a message, so no one else would understand.
- **User 7:** Something like a sign. To indicate the person that comes what happened.
- **User 8:** A message that is sent when you send the alert.

Question 2 Can you think of any information you would like to share, but is not allowed by the app? Or is there any information you can share using the app but you would not want to share?

- **User 1:** N/A
- **User 2:** He believes he would share his name, not his age nor the clothing (it seems like too much effort). He thinks it could also be usefull to add some new fields, such as race or hair color.
- **User 3:** She thinks the information you can share in the app is fine. She thinks the *Santo y seña* is a very good idea.
- **User 4:** She thinks everything is fine. She believes that sharing your photo would not be a good idea ("you may obtain the opposite effect").
- **User 5:** She thinks that the information you can share in the app is fine. She believes that, if you are wearing a very common outfit, it could be useful to some other object you are carrying, that may be more differentiating. She thinks hair color could be added, although almost everybody has the same hair color. Maybe there could be an open field, so the user could write whatever they will.
- **User 6:** She thinks the profile picture is useful. She can not think of anything else to add.
- **User 7:** He suggests adding as information, for example, if you are with your family, that you are with your children. Or maybe if you have any disability. He thinks the rest of the fields are proper.
- **User 8:** He would add a physical description, not only your clothing.

F.2.11 Eleventh interaction

Case study

As we said in the beginning, ths app uses cryptocurrencies to work. So, to be able to use it, you must have an account associated to the app. How would you check, within the app, your account and the money you have associated to it?

Expected answer

The user is expected to move to the *Ajustes* tab and point at the information.

User answers

User 1 She moves to the *Ajustes* tab and points at the field *Campo disponible*.

User 2 He moves to the *Ajustes* tab and points at the fields, pressing them, so the adress is copied to the clipboard. He finds it useful.

User 3 She moves to the *Ajustes* tab and points at the information.

User 4 She quickly moves to the *Ajustes* tab and points at the information.

User 5 N/A

User 6 She quickly moves to the *Ajustes* and points at the information.

User 7 He quickly moves to the *Ajustes* tab and points at the information.

User 8 He moves to the *Ajustes* and points at the information.

F.3 Final considerations

At the end of the interview, the users are asked what they think of the app, if they have found it intuitive or not and whether they would add, remove or change anything in it. In general, any kind of suggestion concerning the application. The users answers are listed below.

F.3.1 User answers

User 1 It would be nice if the range of the alert increased automatically if the alert is not reaching anyone. Apart of that, everything is nice, as long as the field "Santo y seña" is explained. Everything else is very good, the app is easy to use.

User 2 He likes the app a lot. It would be nice if the information that is displayed when you press the title of one of the alerts in the *Avisos* tab could also be displayed by clicking the pins in the map. To show what he means, he looks for a pin in the map, and he can not tell the alerts and the posters apart. He also believes the user should be able to change the profile picture. He mentions that the main button is pretty and that he likes how the bar in the *Ajustes* tab works.

User 3 She likes it a lot. She asks if one should pay to publish posters and alerts in the map, and she believes some people who are involved in this kind of topics would have no problem in paying. She mentions that she likes the color division in the map a lot.

User 4 She asks why the range can be set to "Nadie" (nobody). She thinks the app is nice. She says that, even though some of the functionalities require paying, it is a good thing that asking for help is free. She thinks adding pins to the map is a voluntary thing.

User 5 She thinks everything is pretty good. She thinks it is weird that the contacts are in the *Perfil* tab. ("I don't know any other application that has the contacts in the profile"). She believes it would be nice to have the option to modify the profile picture - otherwise, it is a nonsense to have a random picture. She asks why the range can be set to "Nobody".

User 6 She thinks the icon with the two arrows, in the button for painting the areas in the map, is not intuitive. Everything else looks good to her, and pretty intuitive.

User 7 He likes the functionality that allows the user to paint the dangerous areas in the map. He says he thought the megaphone looked like a watermark or a background. The profile section looks good to him, and perhaps he would add a menu for the clothing to be more customizable. The map is his favourite part of the app, specially the functionality for painting the areas. He thinks the app is simple to use.

User 8 He says he would add a physical description in the profile, something as an open field so each user could write whatever they will. He also suggests that, given the "Santo y seña" section is not veery intuitive, it could be nice to add a help button, that shows a explanation of the meaning of the field.

Appendix G

Heuristic evaluations

PLANTILLA DE EVALUACIÓN HEURÍSTICA

DATOS PERSONALES Y DE EVALUACIÓN
Nombre: Raquel Hervás
Sexo: M
Fecha de evaluación: 12/02/2019
Hora inicio evaluación: 12:25
Hora fin evaluación:

IMPRESIONES POSITIVAS DE LA APLICACIÓN
- **Visualmente está muy bien y es muy intuitiva de usar**
- **Sigue bastante bien las directrices de Material Design tanto en colores como en controles**
- **La parte del mapa visualiza muy bien la información que se quiere mostrar**
- **La funcionalidad principal, la del SOS, está bien en la pantalla principal y con un botón grande**

PROBLEMAS DETECTADOS
(Copiar y pegar si fuera necesario añadir más puntos)
- **Nombre: Feedback SOS**
  - Breve descripción: Cuando se pulsa el botón de SOS no se muestra ningún feedback, y el usuario podría dudar de si ha ocurrido algo.
  - Heurística violada: N01
  - Dónde se ha encontrado: En la pestaña de Home
  - Grado de severidad: 10

- **Nombre: Activación del SOS**
  - Breve descripción: Aunque es la funcionalidad principal y se quiere que sea rápido de activar, si hay un error de pulsación las consecuencias son graves (se preocupa todo el mundo), así que se debería estudiar que la acción fuera menos propensa a errores.
  - Heurística violada: N05
  - Dónde se ha encontrado:En la pestaña de Home
  - Grado de severidad:5

- **Nombre: Ajustes divididos en dos partes (1)**
  - Breve descripción: Hay ajustes que se hacen en la pestaña de perfil, y otros que se hacen en los ajustes. No acabo de entender por qué no está todo junto, ya que al final son “ajustes” tanto del usuario como de aplicación.
  - Heurística violada: N06 (porque el usuario puede dudar al ir a cambiar la configuración sobre a dónde tiene que ir)
  - Dónde se ha encontrado: Pestañas de Perfil y Ajustes
  - Grado de severidad: 7

- **Nombre: Ajustes divididos en dos partes (2)**
  - Breve descripción: Mismo que el anterior pero viola también la heurística de eficiencia porque si el usuario tiene que estar buscando tardará más en activar las opciones.

    - Heurística violada: N07
    - Dónde se ha encontrado:Pestañas de Perfil y Ajustes
    - Grado de severidad:3 (por temas de eficiencia lo veo menos grave).

- **Nombre: Mapa para registrar evento de otro lugar**
    - Breve descripción: Cuando un usuario abre el mapa suele esperar que se muestre o bien la situación general en distintas zonas del mapa, o que se active alguna función en el punto en el que está. Me queda raro que también en el mapa se puedan dar de alta eventos que han ocurrido en otro lugar distinto del que estoy.
    - Heurística violada: N04
    - Dónde se ha encontrado:En la opción del Mapa
    - Grado de severidad:6

- **Nombre: Avisos vs. Mapa**
    - Breve descripción: La información de los avisos existentes está de alguna manera dividida en dos: en el mapa se ven las zonas más peligrosas, y en los avisos se ve como lista más especializada. ¿Se podría de alguna manera conectar las dos?
    - Heurística violada: N07
    - Dónde se ha encontrado: Pestañas de Avisos y Mapa
    - Grado de severidad: 7

- **Nombre: Filtrado y ordenación de los Avisos**
    - Breve descripción: En los avisos, aunque están ordenados por recientes, sería útil poder ordenarlos por otros campos o incluso filtrarlos por zonas.
    - Heurística violada: N07
    - Dónde se ha encontrado: Pestaña de Avisos
    - Grado de severidad: 8

- **Nombre: Información de los Avisos (1)**
    - Breve descripción:En los Avisos, para cosas como lo de los carteles o para saber si alguien ya los ha reportado, estaría bien que hubiera una foto o algo así que los identificara rápidamente.
    - Heurística violada:N01
    - Dónde se ha encontrado:Pestaña de Avisos
    - Grado de severidad:6

- **Nombre: Información de los Avisos (2)**
    - Breve descripción:En los Avisos, para poder mirar si alguien ya ha reportado algo, tal y como está se viola la heurística de eficiencia porque sería muy pesado ir mirándolo todo.
    - Heurística violada:N07
    - Dónde se ha encontrado:Pestaña de Avisos
    - Grado de severidad:6

- **Nombre: Info del Wallet**

    - Breve descripción:La información sobre el wallet que hay en los Ajustes es ilegible para cualquier usuario que no esté familiarizado con el tema.
    - Heurística violada:N10
    - Dónde se ha encontrado:Pestaña de Ajustes
    - Grado de severidad: 9

- **Nombre: Consulta de Avisos**
    - Breve descripción: Aunque en la aplicación se llaman Avisos, en la evaluación se han llamado registros. Ser consistentes, buscando una palabra que se use en el mundo normal. Avisos puede sugerir notificaciones de la aplicación, por ejemplo.
    - Heurística violada: N02
    - Dónde se ha encontrado:Pestaña de Avisos
    - Grado de severidad: 3

PLANTILLA DE EVALUACIÓN HEURÍSTICA

DATOS PERSONALES Y DE EVALUACIÓN
Nombre: DAVID LLOP VILA
Sexo: M (V)
Fecha de evaluación: 12/12/19
Hora inicio evaluación: 13:20
Hora fin evaluación: 13:56

IMPRESIONES POSITIVAS DE LA APLICACIÓN

-
-
-

PROBLEMAS DETECTADOS
(Copiar y pegar si fuera necesario añadir más puntos)

- **Nombre:**
  - Breve descripción: SOS es demasiado asustadizo. Ventana no se entiende.
  - Heurística violada: 2
  - Dónde se ha encontrado: Pantalla princ
  - Grado de severidad: 9

- **Nombre:**
  - Breve descripción: Cuando das a ayuda mantener el control
  - Heurística violada: 3
  - Dónde se ha encontrado:
  - Grado de severidad: 7

- **Nombre:**
  - Breve descripción: En avisos falta el (+)
  - Heurística violada:
  - Dónde se ha encontrado:
  - Grado de severidad: 7

- **Nombre:**
  - Breve descripción: Al enviar aviso, pedir confirmación.
  - Heurística violada: Prevención errores
  - Dónde se ha encontrado: Añadir alerta
  - Grado de severidad: 1

- **Nombre:**
  - Breve descripción: El botón de zonas no da información de qué se trata
  - Heurística violada: 2
  - Dónde se ha encontrado: Mapa
  - Grado de severidad: 5

- **Nombre:**
  - Breve descripción:
  - Heurística violada:
  - Dónde se ha encontrado:
  - Grado de severidad:

- **Nombre:**
  - Breve descripción:
  - Heurística violada:
  - Dónde se ha encontrado:
  - Grado de severidad:

- **Nombre:**
  - Breve descripción:
  - Heurística violada:
  - Dónde se ha encontrado:
  - Grado de severidad:

- **Nombre:**
  - Breve descripción:
  - Heurística violada:
  - Dónde se ha encontrado:
  - Grado de severidad:

- **Nombre:**
  - Breve descripción:
  - Heurística violada:
  - Dónde se ha encontrado:
  - Grado de severidad:

Bibliography

[1] Tamara Adlin and John Pruitt. *The essential persona lifecycle: your guide to building and using personas*. OCLC: 845715995. Amsterdam: Elsevier/Morgan Kaufmann, 2010. 224 pp. ISBN: 978-0-12-381418-0.

[2] MobileSoftware AS. *BSafe – Never walk alone*. URL: https://getbsafe.com/ (visited on 05/17/2019).

[3] *bSafe you App*. Technology Safety. URL: https://www.techsafety.org/bsafe (visited on 05/17/2019).

[4] *Build software better, together*. GitHub. URL: https://github.com (visited on 05/15/2019).

[5] Mikel Cid. *Cuando tu vida corre peligro y un botón te puede ayudar: así funciona Alpify*. Xataka. Apr. 27, 2016. URL: https://www.xataka.com/aplicaciones/cuando-tu-vida-corre-peligro-y-un-boton-te-puede-ayudar-asi-funciona-alpify (visited on 05/17/2019).

[6] *Circle of 6*. Circle of 6. URL: https://www.circleof6app.com:443/ (visited on 05/17/2019).

[7] Alan Cooper et al., eds. *About Face: the essentials of interaction design ; [the completely updated classic on creating delightful user experiences]*. 4. ed. OCLC: 892639516. Indianapolis, Ind: Wiley, 2014. 690 pp. ISBN: 978-1-118-76657-6 978-1-118-76640-8 978-1-118-76658-3.

[8] *Delitos sexuales según sexo(28750)*. URL: http://www.ine.es/jaxiT3/Datos.htm?t=28750 (visited on 05/16/2019).

[9] *Download Android Studio and SDK tools*. URL: https://developer.android.com/studio (visited on 05/15/2019).

[10] *Firebase*. Firebase. URL: https://firebase.google.com/ (visited on 05/15/2019).

[11] *Firebase Cloud Messaging*. Firebase. URL: https://firebase.google.com/docs/cloud-messaging (visited on 05/15/2019).

[12] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. *An introduction to mathematical cryptography*. 2 ed. Undergraduate texts in mathematics. OCLC: 891676484. New York, NY: Springer, 2014. 538 pp. ISBN: 978-1-4939-1710-5 978-1-4939-1711-2.

[13] *Hollaback! Together We Have the Power to End Harassment*. Hollaback! Together We Have the Power to End Harassment. URL: https://www.ihollaback.org/ (visited on 05/17/2019).

[14] *Home - Pencil Project*. URL: https://pencil.evolus.vn/ (visited on 05/15/2019).

[15] *Infura - Scalable Blockchain Infrastructure*. URL: https://infura.io (visited on 05/15/2019).

[16] *Instituto Nacional de Estadistica. (Spanish Statistical Office)*. URL: http://ine.es/ (visited on 05/16/2019).

[17] Protocol Labs. *IPFS is the Distributed Web*. IPFS. URL: https://ipfs.io/ (visited on 05/18/2019).

[18] *Lightweight Java and Android library for integration with Ethereum clients: web3j/web3j*. original-date: 2016-09-04T05:48:49Z. May 15, 2019. URL: https://github.com/web3j/web3j (visited on 05/15/2019).

[19] *Mapbox*. URL: https://www.mapbox.com/ (visited on 05/15/2019).

[20] *Marvel - The design platform for digital products*. URL: https://marvelapp.com/ (visited on 05/15/2019).

[21] *MetaMask*. URL: https://metamask.io/ (visited on 05/15/2019).

[22] Arvind Narayanan. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton: Princeton University Press, 2016. ISBN: 978-0-691-17169-2.

[23] Jakob Nielsen. "Enhancing the Explanatory Power of Usability Heuristics". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '94. event-place: Boston, Massachusetts, USA. New York, NY, USA: ACM, 1994, pp. 152–158. ISBN: 0-89791-650-6. DOI: 10.1145/191666.191729. URL: http://doi.acm.org/10.1145/191666.191729.

[24] *NO MORE APP*. lanzanos.com. URL: http://www.lanzanos.com/proyectos/no-more-app/ (visited on 05/17/2019).

[25] *Online Vector Graphic Design App & Icon Image Editor - Gravit Designer*. URL: https://www.designer.io/ (visited on 05/15/2019).

[26] *Overleaf, Editor de LaTeX online*. URL: https://www.overleaf.com (visited on 05/16/2019).

[27] Mike Potel. "MVP: Model-View-Presenter The Taligent Programming Model for C++ and Java". In: (), p. 14.

[28] *Remix - Solidity IDE*. URL: https://remix.ethereum.org/#optimize=false&version=soljson-v0.5.1+commit.c8a2cb62.js (visited on 05/15/2019).

[29] *Rinkeby: Authenticated Faucet*. URL: https://faucet.rinkeby.io/ (visited on 05/31/2019).

[30] *Rinkeby: Ethereum Testnet*. URL: https://www.rinkeby.io/#stats (visited on 05/15/2019).

[31] *Safe365 - Localizador familiar para cuidar de los tuyos*. URL: https://safe365.com/es (visited on 05/17/2019).

[32] Safecity. *Safecity*. URL: https://safecity.in/ (visited on 05/17/2019).

[33] *Telegram*. URL: http://telegram.com.es/ (visited on 05/15/2019).

[34] *Telegram APIs*. URL: https://core.telegram.org/ (visited on 05/15/2019).

[35] *TeXworks*. URL: http://www.tug.org/texworks/ (visited on 05/16/2019).

[36] *TeXworks*. GitHub. URL: https://github.com/TeXworks (visited on 05/16/2019).

[37] *Ushahidi*. Ushahidi. URL: https://www.ushahidi.com/ (visited on 05/17/2019).

[38] *Welcome to Remix documentation! — Remix, Ethereum-IDE 1 documentation*. URL: https://remix.readthedocs.io/en/latest/ (visited on 05/15/2019).

[39] *Zotero | Your personal research assistant*. URL: https://www.zotero.org/ (visited on 05/15/2019).