Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2019

LEONARDO RAFAEL MIRANDA OLIVEIRA

Produtização de uma aplicação móvel: design participativo e desenvolvimento

Productizing a mobile application: participatory design and development

Universidade de Aveiro
2019

Departamento de Eletrónica,
Telecomunicações e Informática

LEONARDO RAFAEL
MIRANDA OLIVEIRA

Produtização de uma aplicação móvel: design participativo e desenvolvimento

Productizing a mobile application: participatory design and development

"*Simplicity is the ultimate sophistication*"

— Leonardo da Vinci

Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2019

LEONARDO RAFAEL MIRANDA OLIVEIRA

Produtização de uma aplicação móvel: design participativo e desenvolvimento

Productizing a mobile application: participatory design and development

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Doutor Diogo Nuno Pereira Gomes, Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Óscar Emanuel Chaves Mealha, Professor associado com agregação do Departamento de Comunicação e Arte da Universidade de Aveiro.

Convido o leitor a experimentar a app que nasceu deste projeto, disponível em desvaneio.com, que reflete o gosto e dedicação com que este trabalho foi desenvolvido nos últimos meses.

agradecimentos / acknowledgments

Quero agradecer...

Aos professores Diogo Gomes e Óscar Mealha, que me orientaram nesta dissertação, e que desde logo acreditaram neste projeto e me deram liberdade para fazer algo com que me identificasse.

Ao Diogo Ferreira, Ricardo Jesus, Carolina Abrantes, Jorge Hencleeday, Francisco Branco e Pedro Corado, que chateei bastante e que de alguma forma acompanharam o projeto de perto.

À Fabiana Madeira e Mariana Monteiro, do grupo no Facebook Troca e venda de livros em segunda mão, Jorge Ferreira e Márinho Fernandes, da Ror de Livros, Susana Freitas, da Sana Editora, e Pedro Cipriano e Ana Carrilho, da Editorial Divergência, por abraçarem este projeto.

Aos participantes das diversas fases de design da app e a todos os que a têm utilizado, pelos preciosos comentários e sugestões.

A todos os que me fizeram crescer, em especial os professores, começando na minha professora da primária, Fátima Branco. Este trabalho é simbólico de todo o seu contributo. Obrigado por me ensinarem como é bom fugir para a escola.

Ao Sr. Vítor, Ana, Patrícia, Rafael, Mina, Denis, Chico, André e William.

À minha mana, por todo o apoio e por me ensinar a ver as coisas por outras perspetivas.

E a ti, mãe. Pelo exemplo de força e superação. Por transformares todos os impossíveis em oportunidades. Por dares tanto com tão pouco. Por me deixares sonhar com juízo. Obrigado por me guiares. Lembras-te de quando te dizia que um dia terias um motorista para te levar a todo o lado, e que não havias de te preocupar com mais nada? Eu quero ser esse motorista.

Palavras Chave

UX, design participativo, escalabilidade, computação em nuvem, aplicação móvel, livro.

Resumo

O advento dos *smartphones* mudou a forma como comunicamos e colaboramos. Durante os últimos anos houve um aumento exponencial de aplicações móveis, representando novos desafios tecnológicos que exigem soluções eficazes. A internet está repleta de informação sobre a criação de aplicações, mas muitos aspetos são ignorados ou simplificados para apresentar uma solução simples para quem está a começar. A construção de um protótipo é relativamente simples, mas há diversos desafios a enfrentar quando se constrói uma aplicação com padrões mais elevados. Tendo em conta o público-alvo, será que vai ao encontro das suas necessidades? Será fácil de utilizar? Sabendo que potencialmente será instalada por um número crescente de utilizadores, será que é escalável? Qual será o *deployment* adequado? Como lidar com atualizações frequentes? Estas são algumas das questões que guiaram o plano de trabalhos desta dissertação. O documento parte de uma ideia para uma aplicação móvel e aborda os diversos aspetos a ter em consideração na conceção e transformação dessa aplicação num produto pronto para ser disponibilizado aos utilizadores, na Google Play Store ou na Apple App Store. Nesse sentido, este documento reporta todas as decisões tomadas durante este projeto, principalmente no que respeita a aspetos relacionados com as fases de design, desenvolvimento e *deployment*. A versão beta está atualmente disponível na Play Store, com mais de 300 utilizadores registados. Espera-se que a aplicação seja suficientemente representativa nos desafios que envolve, pelo que o processo de design participativo e respetivo desenvolvimento poderá ser adaptado a outras aplicações. Espera-se ainda que as ideias aqui discutidas possam servir de exemplo quanto ao que poderão ser os problemas a ter em conta quando se começa a desenvolver uma aplicação, quais poderão ser as soluções para esses problemas e quais serão os erros a evitar. O trabalho futuro inclui as melhorias necessárias até à versão 1.0 e a exploração mais detalhada de alguns tópicos, tais como gamificação e *A/B testing*, de modo a melhorar a interface e otimizar a experiência de utilização.

Keywords

UX, co-design, scalability, cloud computing, mobile app, book.

Abstract

The advent of smartphones changed the way we communicate and collaborate. During the last years mobile applications have experienced an exponential growth, representing new technological challenges that require effective solutions. The internet has plenty of information about building applications, but many aspects are ignored or simplified to present a simple solution to who is starting. The construction of a prototype is relatively straightforward, but there are several challenges faced when building an application with higher standards. Considering the target audience, does it meet the user needs? Is it easy to use? Knowing that it can potentially be downloaded by a growing number of users, is it scalable? What is the right deployment? How to handle frequent updates? These are some of the questions that nourished the research agenda in this dissertation. It starts from an idea for a mobile application and goes through some of the aspects to consider when conceiving and transforming that application in a product ready to be delivered to the customers, available in the Google Play Store or Apple App Store. Thus, this document reports on all the steps that were followed during this project, regarding aspects mainly related to the design, development and deployment stages. The beta version of the app is currently available in the Play Store, with more than 300 users registered. It is expected that the application is sufficiently representative of the challenges it involves, so the co-design and development process could be adapted to other applications. Hopefully, the ideas discussed here can be an example of what might be the problems to keep in mind when starting to develop an application, what might be the solutions for these problems and what will be the mistakes to avoid. The future work includes the necessary improvements to reach the version 1.0 and the further exploration of some topics, such as gamification and A/B testing to improve the user interface and optimize the user experience.

Contents

List of Figures

Glossary

ACID	Atomicity, Consistency, Isolation, Durability
AWS	Amazon Web Services
CED-UA	Conselho de Ética e Deontologia da Universidade de Aveiro
CRUD	Create, Read, Update, Delete
eCPM	effective Cost Per Mille
GAEFE	Google App Engine Flexible Environment
GAESE	Google App Engine Standard Environment
GCE	Google Compute Engine
GCP	Google Cloud Platform
GDPR	General Data Protection Regulation
GKE	Google Kubernetes Engine
HEART	Happiness, Engagement, Adoption, Retention, and Task success
IaaS	Infrastructure as a Service
MVP	Minimal Viable Product
OEM	Original Equipment Manufacturer
ORM	Object-Relational Mapping
OS	Operative System
PaaS	Platform as a Service
PWA	Progressive Web App
RPC	Remote Procedure Call
SLA	Service Level Agreement
SQL	Structured Query Language
SWOT	Strengths, Weaknesses, Opportunities and Threats
UI	User Interface
UX	User Experience
VM	Virtual Machine

CHAPTER 1 Introduction

"When something happens in your life that you know will not happen again, how do you treat it?" – Prem Rawat

During the last years smartphones have changed our lives in completely new ways. In 2008, Tomi Ahonen pointed to the cellphone as the seventh mass media, comparing it to the previous ones (respectively print, recording, cinema, radio, television and internet) and pointing out seven advantages that differentiate it from them [1]. Part of these advantages is related to the fact that a cellphone is a personal device, which means that all the experience of each cellphone is targeted to its user. We were in 2008, when the smartphones as we know them today were giving the first steps, one year after the announcement of the first iPhone. Since then, the industry evolved and the market grew accordingly. By the end of 2018, 66% of individuals[1] owned one [2]. Going back to 2012, as an example in Portugal, only 18% of the Portuguese population had one. This number raised to 67% five years later, in 2017 [3]. Although the adoption is predominant in people aged less than 25, the trend is followed by all the age groups. Even for people aged 55+, in the same period (2012-2017), the adoption rose from 6% to 38%. This means new opportunities to a large part of the population, and nearly everyone in a not so far future. People take their phones everywhere and are permanently connected. This emerging reality is a great opportunity to explore new ideas. In a world where it became common sense that everything is invented, smartphones are a good example that breaks that assumption. Everyday new apps (formally known as mobile applications) arrive to assist us in a wide range of scenarios. These apps improve the way we communicate, collaborate and access information and knowledge.

The starting point of the work described here is the proposal of an idea that aims to improve some aspect of people's daily life through an app. When planning to build that app, several questions showed up, related to what is the correct way to do it. Regarding the people who will use the app, how can we be confident that the solution being designed meets

[1]Estimated from an analysis to 52 countries (including Portugal), representing 65% of the world's population.

their needs and is easy to use? And considering that the app will be available to anyone willing to install it, how can we make it scalable, all of that with a delivery process that allows frequent updates? These and other challenges are recurrent problems that might not be considered when making a prototype, but they need to be addressed when planning to obtain a robust solution. However, this is not straightforward, mostly because apps are a recent reality. There is not enough related work validating the whole process, the process of transforming a prototype of an app into a final product, this is, to productize it. This is the motivation for this work, that aims to tackle these challenges with a research divided into two main pillars: i) the aspects related to the design process, with a focus on the interface with the user, regarding of the user experience, and ii) the development and deployment of a codebase that is able to scale from a few to thousands of users, being ready to be delivered through agile processes. These two pillars have a concern in common: efficiency and the continuous seek for improvements through an iterative process. Only with that in mind was it possible to achieve the main goal of this work: addressing efficiently these challenges to deliver a product to the customers, available in the Google Play Store and in the Apple App Store.

To achieve this, the first thing to do was to find a good idea for an app, which was already defined before the beginning of this dissertation. After, it was necessary to search for products in the market that target the same idea or part of it. The analysis was pursued for several apps in the Google Play Store and Apple App Store. The developments of this benchmarking analysis allowed to systematize the business requirements for the app and to prepare a first proof of concept of the user interface and interaction narratives, that were also defined prior to this dissertation. This proof of concept had a simple workflow of one of the primary scenarios. Starting there, the goal of this dissertation was to validate the design, development and deployment process, starting where many other hobby projects end: a prototype. Thus, this document will not focus on the specification of the project, such as the requirements, use cases and scenarios, that have been evolving over the several iterations, as a result of the followed process. It will focus on the design, development and deployment process itself, considering the need of transforming the app into a product. The approach to achieve that was by looking at the examples of companies of reference with a strong presence on the internet. The decision was to search for the published work of these companies, to understand the best practices in scenarios of apps predicted for millions of users. According to the results of this research, the design of the app went through several phases, with the realization of a participatory study, that actively involved an intentional sample of the target users. It was followed by the development and deployment of the app and, finally, all the work continued to be validated after the first public deployment, and it will continue to be actively developed according to the feedback received by the users. Fig. 1.1 summarizes what was already accomplished and what was addressed during this dissertation.

This project is illustrative of the overall production cycle of a real app, through the design, development, deployment and validation of the solution. Consequently, this document is about the lessons learned of what might be the problems to keep in mind when starting to develop an app, what might be the solutions for these problems and what will be the mistakes

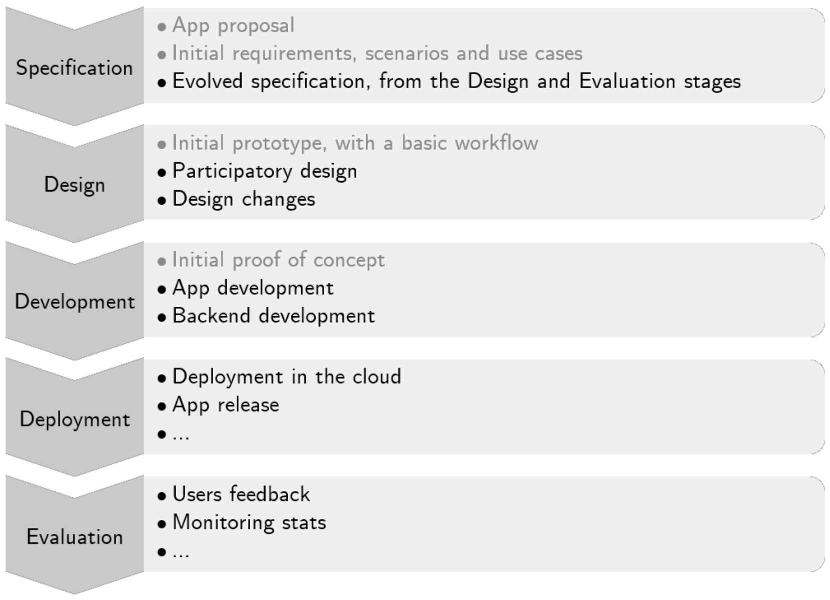

Specification
• App proposal
• Initial requirements, scenarios and use cases
• Evolved specification, from the Design and Evaluation stages
Design
• Initial prototype, with a basic workflow
• Participatory design
• Design changes
Development
• Initial proof of concept
• App development
• Backend development
Deployment
• Deployment in the cloud
• App release
• ...
Evaluation
• Users feedback
• Monitoring stats
• ...

Figure 1.1: The work prior to this dissertation, in grey, and during this dissertation, in black.

to avoid. The research and development agenda was adapted considering the limited time to accomplish all the goals, with some simplifications that did not compromise quality and with a plan for the long-term, because this is a project with future, which does not end in this work. Something that will need ongoing sustainable work and support, not just a proof of concept.

As a result, this document is organized taking into account the different stages of this project. After this introduction, Chapter 2 further discusses the challenges mainly involved in the design, development and deployment of an app, with a review of the literature. It is followed by Chapter 3, that exposes the participatory design study, Chapter 4 and Chapter 5, that reports the decisions for the development and deployment, Chapter 6, with the evaluation of the Minimal Viable Product (MVP) in production, and Chapter 7, that discusses the business model for the app and the roadmap for the following months. Finally, Chapter 8 presents some final considerations and future work.

CHAPTER 2

Background

"Those who cannot remember the past are condemned to repeat it." – *George Santayana*

As we saw, this research and development project was born from the idea of creating an app. This app needed to be transformed into a viable solution, which involved different challenges common when conceiving many other apps. This Chapter presents the research done to support these challenges, that mainly served as a starting point for its design, development and deployment. It ends with a contextualization of the proposed idea for the app, so hopefully it will help illustrating more effectively how these challenges were addressed throughout the project, as it will be described in the next Chapters.

2.1 User-centered design

The time when computers were something complicated and of difficult access for the regular user is past. In recent years technology has become more and more personal, first with the emergence of the personal computer and lately with the advent of personal devices like smartphones. Making this technology available to anyone carries the responsibility of adapting it to a wide diversity of users. Understanding the perspective of the users is essential to successfully address their needs when using an app or any product or service [4]. This is the approach of a user-centered design, where users are brought into the design process and taken into account according to their characteristics, to address several aspects, namely usability.

2.1.1 Usability

Usability is a quality attribute that measures how easy to use are the interfaces [5]. [5] emphasizes the importance of the usability on the web saying that it is *a necessary condition for survival.* In other words, if an app is hard to use, people uninstall it and try others. So, the solution to avoid that is about addressing usability, which can be improved with user testing.

2.1.2 UX

Usability is standardized (ISO 9241-11), being proposed in 1998 and revised in 2018. Apart from being documented and systematized for many years, more recently there has been the need to place the user in a more comprehensive perspective. To go beyond the common checklists used when evaluating usability (e.g., check if the user can undo an action by mistake or check if the same icon is not wrongly used to represent two different concepts [6]), designers try to find the users needs, expectations and wishes. This is the main goal of User Experience (UX). UX is about the complete path during the usage of the product (app, in our scenario), not only during the contact with it but also after or before (e.g., the expectations before the usage, the motivations). To anticipate the users needs, they are invited to collaborate in the design process.

2.1.3 Contextual interviews

One particular example of collaboration are contextual interviews [4]. In contextual interviews, users are interviewed in the environment where they perform the tasks the app is willing to cover. Imagining a bookstore, talking with the users in that context helps to better recognize their needs (e.g., what are their concerns when choosing a new book to read), mainly because it is easier to talk about something we were doing at the moment than recalling it in a different context and situation of use. These type of interviews, that are usually semi-structured (with questions previously prepared), are open to the flow of the conversation. This allows to capture not only the explicit needs but also to anticipate the implicit needs the user is not aware of or would not be able to express in other contexts. As we will see next, in Chapter 3, this was the approach taken in the participatory design study realized in the early phase of this project. Users were actively involved in the design process since the beginning, namely through contextual interviews.

2.1.4 Measuring UX

In agile environments, the design process is iterative, and ideally the result of a change in the design is measured and improved in each iteration. To complement the feedback obtained with the direct contact with the users (e.g., contextual interviews), it is useful to measure UX when not in the presence of the user. Apps have an advantage in that sense, because they can be tracked at any time, providing metrics of usage in a large scale. Even so, measuring UX, and more precisely, knowing what is relevant to measure, can be challenging. Google created a framework called Happiness, Engagement, Adoption, Retention, and Task success (HEART), that encompasses the indicators they considered of most relevance when measuring UX [7]. Some examples of metrics are i) the number of visits per week per user (engagement), ii) the time to complete a task (retention) and iii) the percentage of users that completed a task (task success). These metrics are guidelines and must be chosen according to the app's specific scenario. We will talk about how these guidelines were adapted in the context of this app, in Chapter 6, where it is described the way the solution was validated.

2.1.5 A/B testing

Other techniques that can be used are related to A/B testing. When testing a specific feature, some users can have the version A and others the version B of the app, and the results can be compared. Besides that, [8] refers the importance of distinguishing the initial effect and the settled effect, because the initial impact the user has with a new feature can be different from the feeling after getting used to it. This topic is further explored later in this dissertation, in Section 6.5.

2.1.6 User engagement

Among the dimensions covered by the HEART framework, user engagement is explored with special detail when addressing UX. It relies on the positive effects in the interactions with a product or service that affect the way users feel captivated and motivated to use it [9]. One of the strategies to achieve that is about exploring the characteristics of video games, motivated by the belief that any service that takes advantage of these characteristics can also be fun to use. This strategy, commonly called gamification, is subject of study in [10], where the researchers verify the effectiveness of using medals in an online store platform. When implementing it, users feel rewarded as they make use of the platform, and so they get more engaged.

2.2 Development and deployment

Having discussed the design, let us now focus on the development and deployment stages. Generally apps are supported by a backend, that handles things like communication and data persistence. But not all the apps need it. Perhaps, a simple game will only rely on the app itself. Also, different apps will require backends of different complexities. The following considerations are a tentative to cover some of the main aspects that need to be considered for a backend served by a web API. Later, based on this research, the decisions for this project are presented in Chapter 4 and Chapter 5.

2.2.1 Web APIs

Web APIs enable apps to talk to each other. Nowadays, most of the web APIs follow the same architectural principle: REST. More specifically they rely on HTTP and JSON. In these cases, apps consume a REST API that is made available by a server, the backend. REST has been there for years, and it is great in the way it is easy to understand by humans. Also, it respects the loose coupling between clients and servers, while taking advantage of the web infrastructure already being built on top of HTTP. However, the web is constantly evolving, and the requirements of the recent years are more demanding. REST is human-readable, but not so machine-readable. There is not a formal API contract that allows to automate client libraries, for example when deciding what parameters and types a certain endpoint accepts. REST depends on the documentation and how programmers use it to understand the behavior of each endpoint. So, it requires human efforts that could be reduced with other approaches. In addition, it does not handle well streaming, specially bi-directional streaming.

And all of this is inefficient: REST is textual, which is not optimal for networks, and it is hard get many resources in a single request [11].

To solve the machine-readable contract we have solutions like Swagger, that plays an important role standardizing the APIs of the projects that use it. Still, Swagger is verbose, we do not get solved the problem of using textual representations and we are still stuck on the issue of streaming. That is where a solution like gRPC enters, a Remote Procedure Call (RPC) framework that claims to be high-performance and universal. On top of gRPC, a solution like GraphQL, a query language for APIs, solves the problem of getting many resources in a single request. GraphQL was created by Facebook in 2012 and is used by many other companies that need stronger solutions than REST.

2.2.2 Frameworks

When following a mainstream client-server architecture with a web API, there are several decisions on the server implementation. There are plenty of alternatives of web frameworks, that serve the common use cases when developing solutions for the web, without needing to reinvent the wheel. Namely, they usually come with i) authentication, ii) an Object-Relational Mapping (ORM), that greatly facilitates data modeling and database access, tools for iii) tests and iv) logging, and v) plugins, that extend their functionalities without extra effort to develop our own implementations of things that other developers already did. Developing a web application requires many small decisions, and a web framework makes most of these decisions for us, letting the developer focusing on the implementation of the business logic of the application[1] itself. A web framework intrinsically forces a structure for the code (e.g., how the data is modeled or where the urls are defined). Some projects might have higher standards, with requirements that would force the team to develop their own framework, but for the large majority of the projects, specially the ones that are starting, the existing web frameworks are strong time savers. Also, one of the omnipresent nightmares is security, and these frameworks are developed and maintained by many people and serve many projects, so security holes are constantly being addressed. Things like SQL injection will be handled out-of-the-box. Some of the currently most used web frameworks are Spring, for Java, Laravel, for PHP, Ruby on Rails, for Ruby, or Django, for Python. There are many other solutions, some of them more minimalistic and lightweight, other more robust and hard to learn.

2.2.3 SQL vs NoSQL

When choosing a database, one of the decisions to make is whether to opt for an SQL or a NoSQL solution [12]. Relational databases use Structured Query Language (SQL), that, as the name implies, is a language to query structured data. This is one of the key differences between SQL and NoSQL databases. In SQL the modeled data follows a rigid structure, while NoSQL has a dynamic schema, which is a better fit for unstructured data. In a scenario where the data has dynamic properties, or where their structure is susceptible to frequent changes, NoSQL is flexible to support it, and there are no issues handling migrations, like

[1]Application in the backend, not to be confused with the app.

in SQL. Of course, this freedom has a cost: efficiency. SQL is more efficient and is specially good for complex queries. So, when dealing with data that is conventionally structured, SQL is the right choice. We also have to note that SQL is there for years, and is currently the standard of the industry. With slight changes from technology to technology, all the SQL databases share the same language, while the NoSQL ones have their own different solutions, that are adapted to their different structures, that might be document-based, key-value pairs, graph stores or wide-column stores [13]. NoSQL came to answer to these new use cases. The internet is full of information, that is growing and getting more complex, and it is not easy to structure it, so these new approaches are an effort to make it easier. SQL relies on its stability and robustness, gained with the maturity from the last years, while NoSQL is still young. Many NoSQL solutions are only now starting to support ACID transactions, which is critical in some systems [14]. About scalability, SQL databases grow vertically (i.e., increasing the power of each machine), while NoSQL databases grow horizontally (i.e., adding more machines). Scaling vertically has its limits, so horizontal scaling is a preferable choice when working with large amounts of data. Choosing NoSQL might mean being locked in some technologies. Firebase Cloud Firestore, from Google, or DynamoDB, from Amazon, are two examples of key-value databases that are proprietary. By using them we get dependent on their infrastructure, so migrating to other cloud provider or other database solution gets difficult.

Wrapping up, relational databases are the answer to well structured systems, specially when dealing with critical data, that needs the maturity of a widely tested solution, that has proven results from many years of development. On the other hand, non-relational databases come to answer the new use cases of today, where we deal with dynamic information, that might come in huge amounts of data. Some studies that compare performance [15] and scalability [16] can help making a choice. NoSQL is being used across big companies. For example, MongoDB is used by eBay, Google and Facebook [17]. There are also companies that decided to use NoSQL solutions but had to move back to SQL [18], or from a NoSQL solution to other NoSQL solution [19].

2.2.4 Cross-platform apps

We have been talking about the backend, but the development of this project strongly relies on the app, so there are also decisions to make in this regard. In April 2019, 75% of the worldwide smartphones were Android, while 23% were iPhones [20]. In some countries the numbers are more balanced. For example, in the same period, in the USA 53% of the devices were iPhones, while 46% were Android. Also, iOS users are more likely to spend money on in-app purchases than Android users [21], which is decisive for the revenue of a company. Clearly we cannot ignore one platform if we need to have a strong presence in the mobile market, so the problem is how we can head to these platforms taking into account all the differences that this implies in the development. In the last years, several efforts have been done in the market of mobile development, and today there is a wide range of alternatives on the table, and the native vs cross-platform argument is one of the hot topics. A native app is an app developed for a specific platform, written in the language targeted by the operating system [22]. For Android this would correspond to Java and for iOS it would correspond to Objective-C or Swift. Basically this has been the reality of mobile development since the beginning. It happens that developing native apps for different Operative Systems (OSs) means different source codes, different projects, with independent lifecycles, usually with different teams, because it forces developers to be skilled in different technologies. All of this costs time and human resources, so this issue in the market has been tackled with the emergence of cross-platform apps. Cross-platform apps allow to have a common source code written in a language that is later interpreted accordingly in the different platforms. Cross-platform solutions would ideally solve the gaps between platforms, but the gains with this fusion come associated to some trade-offs. Today we could split the existing cross-platform solutions into three different categories: hybrid, native and PWA.

Hybrid apps

The first cross-platform solutions to show up were hybrid. In these solutions the apps run in a web view, so the code is written in HTML and JavaScript. This is specially useful for who comes from the web development. Because the code is not native, the major con is the performance. Also, they can access hardware like any other native app, as long as HTML5 supports it, but sometimes it can be troublesome, because the access is not direct like in a native app. Examples of hybrid cross-platform solutions are Ionic, Cordova, Sencha or PhoneGap [23].

Native apps

Native solutions for cross-platform apps promise to offer almost the same performance as native apps [23]. Developers only have the overhead of learning a new framework, that is designed to embrace the particularities of each platform in a unique API. This type of solutions has been growing in the last years with frameworks like Xamarin, in 2011, React Native, in 2015, and Flutter, in 2017. Xamarin is maintained by Microsoft, but it is loosing the market (see Fig. 4.2). React Native is maintained and used by Facebook in their apps. Flutter is maintained by Google, and is already being their choice for some of their apps. These real-world examples from big companies are decisive to overcome the preconception that it is risky to opt for a cross-platform solution, that these apps are less capable, stable or professional. If Facebook does it with some of the most used apps in the world, every app can do it.

PWAs

Progressive Web Apps (PWAs) are a very particular case of apps. We might even not consider them as real apps. When we access a website in the smartphone and it pops a message saying something like *add to home screen*, it allows to pin this website among the remaining apps in the smartphone. The app can after be accessed by its icon, outside the browser, while still being a web page. This is how a PWA works. PWAs are useful because of the simplicity with which they conquer their space in the user' smartphone, while looking like any other real app, with a name and icon. This lower friction can be decisive to convince a user to "install" an app that otherwise they would not want to bother installing from the store. All of this without the need to distribute binaries and keeping compatibility across different versions of the OS. Still, like the hybrid apps, PWAs are slower, so not recommended for more demanding requirements. Also, the access to the hardware is limited [24].

2.2.5 Cloud computing

Now regarding the deployment, one of the concerns to keep in mind when delivering an app in a store is that it gets available to any user who is willing to install it from the moment it gets published. This means that a huge number of users might access it. And the changes can be within years or within days. So, it is important to make sure an app is scalable, being ready to support a growing number of users. More than the increase over time, it is important to be ready for peaks of utilization during a full day, because the number of users at 7pm will be different than at 5am (see Fig. 2.1). In an app that is connected to an API, exposed by a backend somewhere on the internet, this means that the needed computational resources of that backend might vary a lot, which directly affects costs. In the past, companies had their own data centers, where they would place more servers when needing to scale. These servers could take weeks to arrive and setup, which implied an unfavorable delay to react to these traffic changes [25]. Also, that static infrastructure was the same when supporting high traffic peaks and in the remaining time, meaning that most of the time the power of the computational resources was not being fully used. This suboptimization of the infrastructure represented unnecessary extra costs in the companies budgets.

Today, although this is still a reality in many companies, the phenomenon of the cloud computing changed the game. Companies are now giving up on their infrastructures and starting to rent servers in the cloud. This means that they can use the computing resources needed at each moment, being able to scale instantly. This way, they only pay for what they are using, which allows to reduce costs. Netflix is a good example of the cloud movement

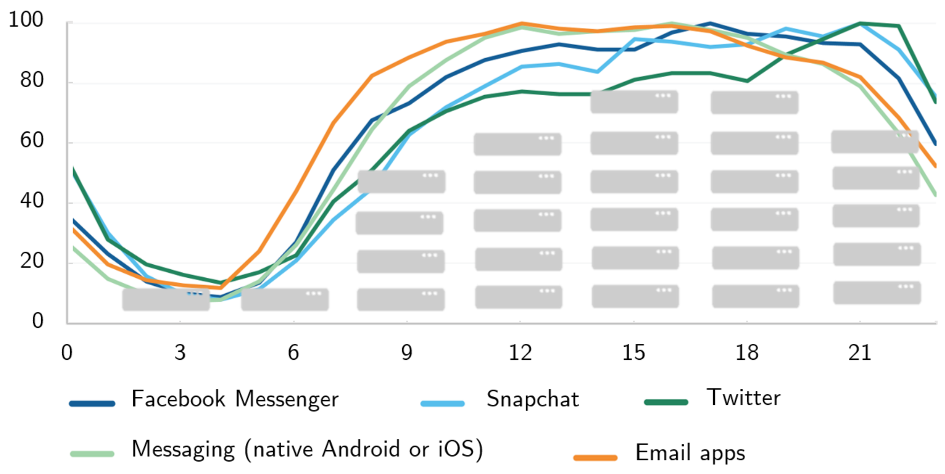


Figure 2.1: The percentage of active users during a day in several apps [26]. The stacks of servers below the curves are merely illustrative of the proportion of resources needed.

[25]. In 2008 they had their own data centers, and they were not being able to scale at the same speed as their customers were growing. So they moved to Amazon Web Services (AWS). Along with AWS, the other main players in the cloud are Google Cloud Platform (GCP) and Microsoft Azure [27]. It is important to note that the cloud is not a solution exclusive for big companies. A simple startup, that would struggle to afford its own data center, can simply start using the cloud for $0,013/hour [28].

PaaS vs IaaS

Cloud providers have different layers of services. For example, clients (companies) can opt to use the Infrastructure as a Service (IaaS) or the Platform as a Service (PaaS). In IaaS, the clients can directly configure the machines, having high flexibility in the architecture design. PaaS is more high level, so clients have to do less configurations and take advantages of some existing mechanisms, like the automatic scaling of resources. Due to its flexibility, IaaS may potentially allow a more efficient use, being less expensive. On the other hand, PaaS is an easier and faster way to start using the resources with less efforts [29].

2.2.6 Microservices

The cloud provides the necessary resources for scaling, but the way an application scales depends on the architecture of the solution [30]. Let us imagine a system with three subservices: a store, a chat to contact the sellers and a recommendation engine to suggest items to buy. These three services can be deployed in a single monolithic application. However, they are different between each other, so the resources they need are irregular across the application. Scaling a monolithic application is done as a whole, so some resources are wasted with less used services. In opposition, a microservice architecture allows the existence of independent services, rather than a monolithic set. In the given example, there would be three microservices. This would allow scaling each one according to their needs and avoiding other potential problems, like the existence of a single point of failure.

2.2.7 Continuous delivery

A microservices architecture also opens the path to continuous delivery strategies, because each new deployment only affects the correspondent microservice [30]. Continuous delivery is about automating the manual steps, that most of the time are error-prone configurations, allowing to speed up the release cycles. Paddy Power, an Irish bookmaker with an annual turnover of six thousand millions, decided to improve their release model, because the manual procedures in each step until the deployment resulted in delays in the release of new features, which affected their competitiveness in the market [31]. Each time they needed to release a new version, only the setup of the testing environment could take three weeks. After a period of integration of a continuous delivery process, they were able to be more productive, with processes compatible with agile methodologies.

2.3 The app proposal

We saw some of the topics considered of major relevance for the app being developed in this project. Now, before heading to Chapter 3, where it will be presented the participatory design study, we will take some time to discuss the motivations and the initial concept, which will help illustrating more effectively the design, development and deployment concerns, materializing the motivations already discussed in Chapter 1 respecting to the challenges raised with the evolution of the market of the apps.

2.3.1 Motivations

The idea for the app starts from the amount of books spread all over the world. Particularly, in the peoples' houses there are books that are only read once, or even never read, being left on their shelves indefinitely. These books could be more useful if, for example, they were shared with other people. Fortunately, books have been around for many years, and this is not a recent problem. That is why there are libraries. Also, it is common to find second-hand books in fairs. There are even places on the street with books in booths, where people are encouraged to leave their books in exchange for the existing ones (see Fig. 2.2).

If we look at new technologies, people use web platforms like OLX or Facebook Marketplace for the purpose of selling their books. There are even Facebook groups about second-hand books, where people publish the items they are offering for exchange, sale or donation. They generally put a picture of one or several books and expect other people to be able to find the books they are looking for. Most of the time the other users have to go picture by picture, without the ability to search, for example, by the title (see Fig. 2.3). This happens because sellers do not have an efficient way to catalog their books, so it is easier to just take a picture of a bunch of them. In another perspective, if people do not have a book in mind but find a cover with a suggestive title, they have to google it to look for more details, like the summary

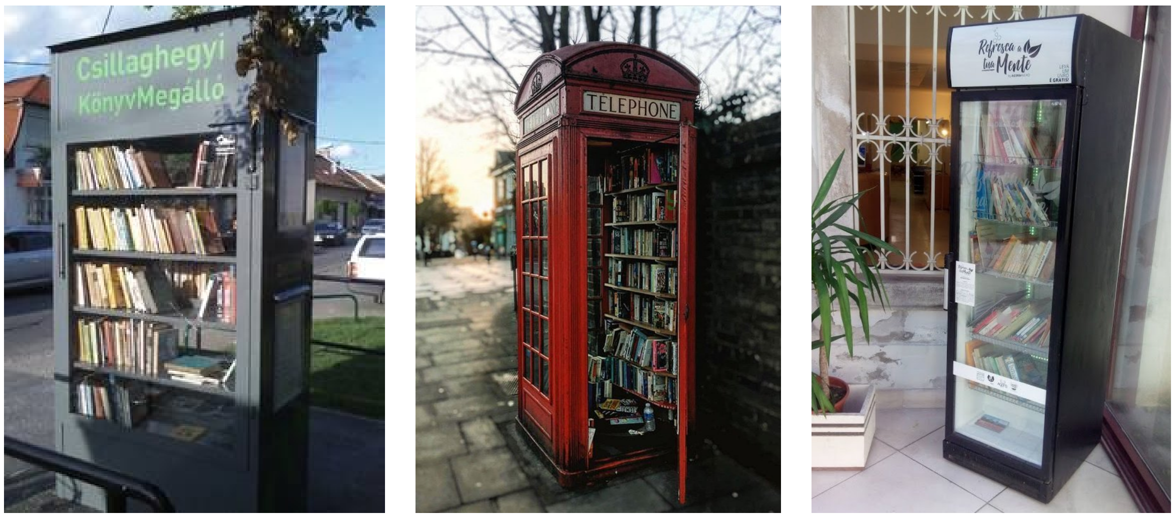


Figure 2.2: Some places where people can exchange books, from the left to the right, in Budapest, London and Aveiro. The one in Aveiro is inside the municipal library. It is literally a refrigerator and says *refresca a tua mente* (in english, *refresh your mind*).

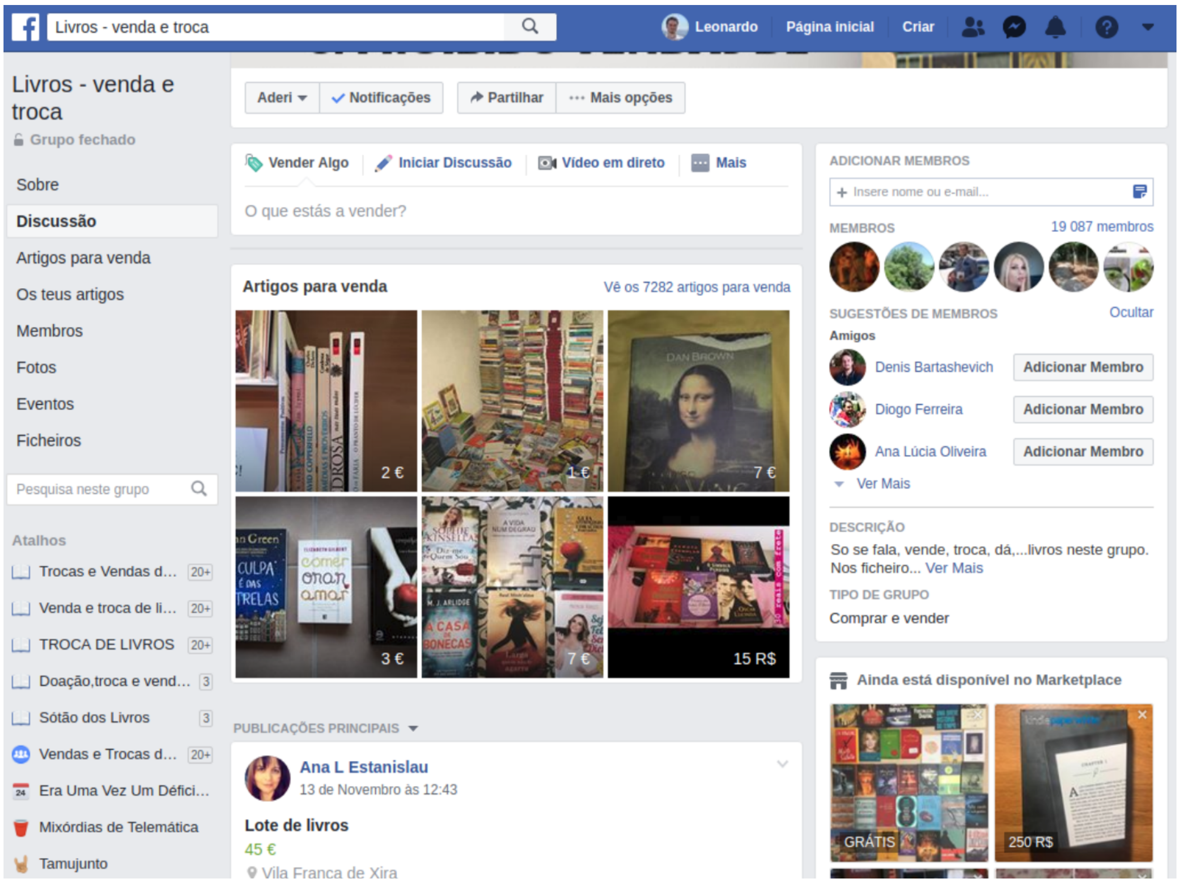


Figure 2.3: A Portuguese Facebook group with around 20.000 members, where users post books to sell. At the center it is possible to see some examples of pictures with a bunch of books, which makes it difficult to search.

or reviews. Another issue is related to the concept of these platforms, that are only targeted for sales. What if people want to donate, loan or exchange books? These are some of the aspects that are not possible to address in platforms like these, that are generic to any product. Thus, it is clear that, when it comes to books, some procedures in these shops are annoying and have a margin to improve.

2.3.2 Concept

The idea for this project was to solve that, by creating an app where people can easily organize the books they have for sale, loan, donation or exchange. Along with this, users would also be able to mark the books they plan to read, they are reading or they already read, and also to mark their favorites. So, more than a store, the app would be a tool to assist users in their cataloguing and reading process. The reading process, despite of being an individual experience, generally starts in a social perspective. People read books that are recommended by someone. Considering this, readers would also be able to look for what their friends are reading, or what are their opinions on books to find some inspiration for their next reads.

Wrapping up, the app aims to be i) a library, to assist the users in the reading process, ii) a social network, to share reading experiences (like reviews), and iii) a store, to exchange books. If we look at it in a wider perspective, the app looks to help the readers in all the book's life cycle, since i) they plan to start reading a new book, ii) the experience during reading, until the moment iii) they decide to give a new life to it, instead of keeping it in a shelf forever. As it will be possible to notice in the next sections, the aim of this app is not innovative in the features it is willing to provide, but in the way it pretends to chain them in a new paradigm, providing a complete experience to the users.

2.3.3 SWOT analysis

Having the idea for the app defined, we now discuss it using a SWOT analysis (Strengths, Weaknesses, Opportunities and Threats). This will help us defining the MVP, later discussed in Section 2.3.4.

Strengths

The app promises to offer an experience exclusive to the universe of the printed books that is currently not met by other products. As we will see next, in Section 2.3.5, the existent solutions are not so complete. But more than all its features, the app will be a cohesive package, with all its interconnected dimensions: library, social network and store.

Weaknesses

We are in the digital era, and the market of the eBooks is conquering its space. The app will not be an eBook reader, neither will it support eBooks in the near future. For example, the store, where the users exchange books between them, would only make sense for printed books. This alienates potential users who no longer fall into the universe of printed books.

Opportunities

The app wants to play a role in the sharing/circular economy. It might also be a place where bookstores, publishers and libraries will find a way to keep contact with their readers. In an ideal scenario, someone in a city willing to read a book, would be able to do it not only by knowing if a friend has it so they could lend it, but also by searching for it in the book stores and libraries nearby. All of this in a single app.

We may even dare to compare the democratization of the books on this platform with the way Uber does it with the cars or Airbnb with the accommodation. These are platforms that do not own the goods but provide the right platforms to exchange them.

Threats

The main threat to the business will be the way people will read in the future. Will printed books continue to be used as they are nowadays? Several studies point out that people still prefer printed books, even the younger generations. Millennials prefer printed books [32] and are reading more than older generations [33]. And the sales are in accordance with that. After a period of glory in the last years, when eBooks started to conquer part of the market of printed books, now it is getting stable, in some countries with slight drops in sales for eBooks [34][35]. In contrast, for example, in the UK printed books sales increased for the fourth consecutive year, in 2018 [36]. Several other studies point out that printed books are preferred for learning over eBooks [37]. Although the focus of the app is not on school books, this is also an indicator that the market for printed books is stable.

Other threats might be related to the presence of other companies that currently dominate the market. For example, Amazon owns Goodreads, probably the best platform to find book reviews. If an idea like this app succeeds, it will not be hard to adapt their business to answer to the changes in the market. And then the law of the strongest might be decisive.

2.3.4 MVP

With the scope of the app defined, and with part of its reasoning based on a SWOT analysis, it was possible to understand the priorities for the MVP, and how they would meet the needs of the target users.

Priorities

The app is primarily designed to be used socially: users show what they are reading, commenting and citing books, and exchange them with other users. The experience when using the app is maximized with a good community of users, that are readers. Still, that community is not easy to achieve. It all depends on the success of the dissemination of the app. And not only the number of users matters. It matters if they are connected to each other, for example, geographically. The store is designed to allow exchanging books with the people in the same area. Also, it can be an obstacle if a user speaks a language and all that the app has to suggest are readers that speak other language. So, it is important that the app is ready for that, being worth to be downloaded and kept if the social dimension fails. If the store or the social network do not succeed, users will always have their library, that they can manage individually. There are several apps that only do this: book cataloging. Most of them are not intuitive and have other problems that were discussed below, in Section 2.3.5. The library will be the safer bet for the app. So, it is mandatory that it is well done.

Target users

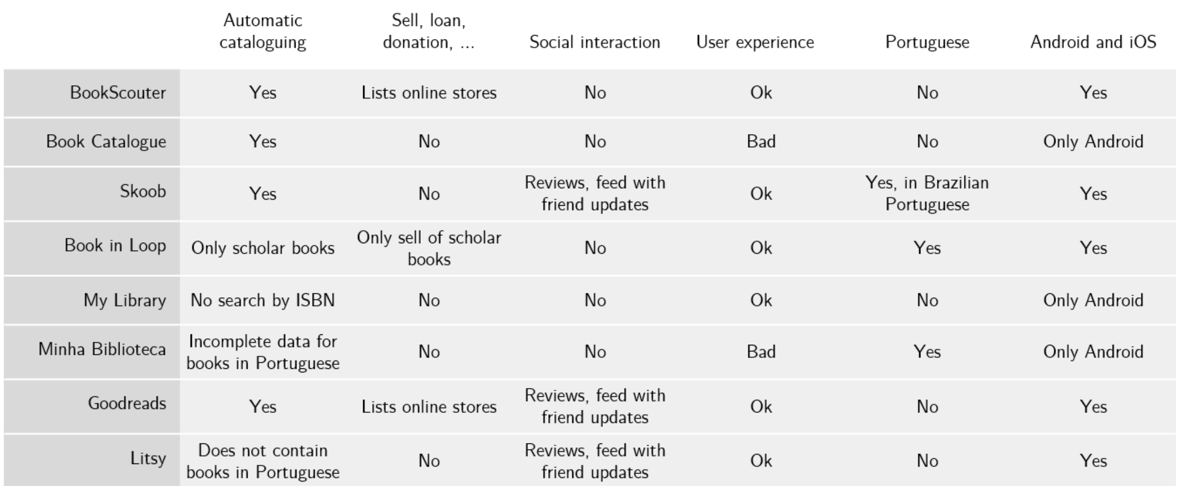
The typical users for the app will be, of course, people who like to read. Given the focus on the social dimension, the primary user is someone that likes printed books, but with an interest in the potential of new technologies as a complement to the experience of reading. Someone who looks for reading suggestions, supporting their choices on the critics from other people. Someone that, while reading, is willing to express their thoughts but does not feel fulfilled with the typical social networks, that are overcrowded of random posts, with fake news and clickbaits, where everything is discussed but nothing is learned, and a small comment can ignite arguments where the important is to offend instead of privileging healthy discussions.

Given the large scope of the app, it will also serve the needs of other users, that will make use of a subset of the app's functionalities. Secondary users might only use the library to catalogue their books or the store to sell them. This last example might be the case of most of the people that come from the Facebook groups and other general-purpose tools that allow to sell things. Thus, the app needed to be designed supporting these and other different scenarios. The challenge was interconnecting them in a consistent experience.

2.3.5 Benchmarking analysis

To validate the concept of the app, it was necessary to search and compare products in the market that target the same idea or part of it. Several apps were tested in a preliminary analysis. After verifying the irrelevance of part of them, the parameters were adjusted to improve the quality of the group of apps under analysis and, consequently, to obtain better insights. This adjust resulted in 8 apps, selected from a group of apps in the Google Play Store and Apple App Store, with at least 50.000 downloads and exclusive to books (which excludes generic shopping platforms). It is important to notice that, although this was a systematic analysis, it was still dependent on the personal perspective and user experience of the student developing this project. Fig. 2.4 summarizes the result of this analysis [2]. The apps were compared according to several parameters, that were considered the core aspects for their suitability to the concept of this project: i) automatic cataloguing, ii) sell, loan, donation and exchange mediation, iii) social interaction, iv) user experience, v) availability in Portuguese and vi) availability for Android and iOS. The automatic cataloguing was considered essential. It is what might consume more time when making an offer or searching for a new book. In general the observed apps are well prepared for that. In addition to search by title, most of them allow the scan of the book's bar code, which accelerates the process. Two of them have trouble with books in Portuguese, one only handles scholar books. The mediation of the several possibilities of exchange is a negative point across the analyzed apps. One of them allows to sell or buy scholar books, two list online stores and the remaining do not offer any option. In this parameter none of them is close to what is the goal of the app being developed. About the social interaction, three have reviews of books and have a feed where users can check for updates from friends. In these three it is included Goodreads, detained by Amazon (the biggest online store, that started by selling books, in 1994). Goodreads is the best place to find reviews of books, and no other solution is close to it. The user experience was not a special concern in the apps under analysis, which was to be expected, given the maturity of the apps under analysis. Even so, two of them have a confuse design, for example with menus with unrelated options or hard to find features. Maybe their success in the app stores

[2]The apps are available here:
BookScouter: https://play.google.com/store/apps/details?id=com.BookScouter (Android), https://apps.apple.com/us/app/bookscouter/id366508853 (iOS);
Book Catalogue: https://play.google.com/store//details?id=com.eleybourn.bookcatalogue (Android);
Skoob: https://play.google.com/store/apps/details?id=com.gaudium.skoob (Android), https://apps.apple.com/br/app/skoob-para-quem-ama-livros/id904670263 (iOS);
Book in Loop: https://play.google.com/store/apps/details?id=com.bookinloop.bil (Android), https://apps.apple.com/pt/app/book-in-loop/id1244672473 (iOS);
My Library: https://play.google.com/store/apps/details?id=org.zezi.gb (Android);
Minha Biblioteca: https://play.google.com/store/apps/details?id=com.vgm.mylibrary (Android);
Goodreads: https://play.google.com/store/apps/details?id=com.goodreads (Android); https://apps.apple.com/us/app/goodreads-book-reviews/id355833469 (iOS);
Litsy: https://play.google.com/store/apps/details?id=com.catch84.litsy (Android), https://apps.apple.com/us/app/litsy/id1037017919 (iOS).



	Automatic cataloguing	Sell, loan, donation, ...	Social interaction	User experience	Portuguese	Android and iOS
BookScouter	Yes	Lists online stores	No	Ok	No	Yes
Book Catalogue	Yes	No	No	Bad	No	Only Android
Skoob	Yes	No	Reviews, feed with friend updates	Ok	Yes, in Brazilian Portuguese	Yes
Book in Loop	Only scholar books	Only sell of scholar books	No	Ok	Yes	Yes
My Library	No search by ISBN	No	No	Ok	No	Only Android
Minha Biblioteca	Incomplete data for books in Portuguese	No	No	Bad	Yes	Only Android
Goodreads	Yes	Lists online stores	Reviews, feed with friend updates	Ok	No	Yes
Litsy	Does not contain books in Portuguese	No	Reviews, feed with friend updates	Ok	No	Yes

Figure 2.4: Comparison of the apps under analysis.

is justified because, despite the interaction problems, they present more features than other alternatives that are more usable but less complete. In what concerns to the availability in Portuguese, this is a missing point in most of the solutions under analysis. Finally, about the availability in the different OS, most of the apps are ready for Android and iOS.

Along with this last section with some outlines of the app specification, these were the topics considered of major relevance to further contextualize this dissertation. This review of the literature for the design, development and deployment process is believed to lead to solutions that are beyond other proof of concept solutions. As we will see throughout this document, different priorities were given to these topics in the course of the project in order to make it compatible with the available time.

CHAPTER 3

Participatory Design Study

"Ask the right questions if you're going to find the right answers." – Vanessa Redgrave

The insights from the preliminary phase of research, described in Section 2.3, allowed to gather the first requirements and the initial prototype of the app. At this point, these requirements were not the answer for the final solution, but a tool to ask the right questions. The next step was to ask these questions to the potential users, so a participatory study was prepared with their active involvement. The study was approved by Conselho de Ética e Deontologia da Universidade de Aveiro (CED-UA), which advised about the ethics during the process (see Appendix A). This Chapter exposes the planning, execution and results of the study.

3.1 Planning

The participatory design study consisted of an inquiry session, planned with an intentional sample of users who manifested the taste for reading and referred that they had read at least 2 books in the last 12 months as a hobby, in an attempt to select people that would represent the needs of the target audience and would have the interest to use the final product. During 30 to 60 minutes, the participants would be invited to collaborate along three different moments, described below.

3.1.1 Contextual interview

The first moment consisted of an individual conversation with the participants, with the intention of characterizing them as readers and users of smartphones. In that interview they were questioned about some points related to the proposed use cases of the app, in order to understand their suitability for the future users. This way, it consisted of a semi-structured interview, with a set of questions, with factual and opinion answers, while being open to the flow of the conversation. The contribution of the participant in a context where there would be time and freedom to a more detailed dialogue on some of the issues would allow to gather personal views, habits and expectations that would help, in a subjective way, to shape the concept of the app to the users.

Expectations

It was important to register the expectations as neutrally as possible. The participants would be previously informed that they would be collaborating in the design of an app about books. Before further contextualization, they were asked about what they thought that the app would be. At best, this approach would remind them of something that they would like to see in an app, something that would be useful to them. The participants' suggestions could also consist of features that were not yet been considered in the study.

Background

It was also relevant to properly understand the participants' background in which respects to their digital literacy. Different backgrounds could result in different visions. Also, this could be important to compare differences in performances in the moment next to the interview, where the participant would test a prototype of the app (see Section 3.1.2).

Habits

Participants were also questioned about their habits on i) reading, ii) book sharing and iii) social experiences in the context of reading, the three main areas considered in Section 2.3, when defining the initial concept for the app. For example, about book sharing, the participants could point out their issues when trying to exchange books with other people. Nothing better than to bring the participants that already faced these issues to help designing the solutions. Some of the questions were open-ended, like *What do you appreciate the most in your favorite social network?*, and other closed-ended, like *Have you ever loaned a book to someone who did not return it?*. Closed-ended questions were useful to better organize the answers statistically, like in a survey. Still, in a type of conversation where the interest was in the details, open-ended questions allowed to get qualitative insights, rather than statistical ones [38].

3.1.2 Functional prototype's test

For the second moment of the session, the participants were invited to test a functional prototype of the app in a smartphone, which was improved from the proof of concept prior to this dissertation. The smartphone would be integrated within eye tracking technology, that would allow to capture the movement and fixation time of the eyes along the screen. The images of the face and voice would also be recorded, so later when analysing the results it would be easier to recall the tests. With this, the facial expressions of the participants could help to understand some key moments, like satisfaction or confusion. When facing the prototype, the participants would execute a set of predefined tasks, while being asked to think-aloud, this is, expressing what they were thinking at each moment, so it would be easier to understand the way they performed each task or what difficulties they would face. The tasks were followed while describing a scenario. The following is an excerpt of that scenario, where the participant was in the role of Ana:

Peter, a friend of Ana, told her the last time they were together from a book he read recently. Later Ana remembers the conversation they had and decides to look for more information about the book in the app Desvaneio. She searches for the title - Memorial do Convento - and finds it (1). She reads the abstract (2) and likes it, so she decides to bookmark it with the label "to read" (3).

The segments of the scenario were mapped to several tasks (in this excerpt there were 3 of them, marked with the numbers in parentheses). For each task, any details considered relevant were noted, so that later the results could be analysed, crossing the information from the eye movement, facial expressions and speech. Other analytics, like the success of the tasks' execution and the duration, were also noted.

3.1.3 Qualitative inquiry

After completing the test, the recording was stopped and the participants filled a questionnaire where they had a set of affirmations to be indicated with the degree of agreement, according to the Likert scale. These affirmations were related to topics on the usability of the application, like *There is consistency in the displacement of the presented contents* or *The app is slow*, so the participants' considerations in this questionnaire would be confronted with the app's performance at the tests.

After this, it was important to complement it with a deeper understanding of how was the experience with the prototype. Participants were invited to freely express their opinions, and they had the prototype with them so they could continue exploring or recover any details from the test that they forgot to say. The conversation was assisted with some concrete questions to help unlocking the conversation, for example *What things should be changed in the app?* or *How do you imagine yourself using this app?*.

3.2 Execution

The planned session was executed according to the described in Section 3.1. 28 people participated in the experience, aged 18 to 56 years, 15 women and 13 men. They were informed about the goals of the study, that the collected data was anonymous, being exclusively used in the context of the study, and that their participation was volunteer, being possible at any moment to go back on that decision. Each participant was invited to read and sign a document with an informed consent. The sessions happened in the room 40.2.10 of the DigiMedia Research Center of Universidade de Aveiro[1], a laboratory of usability equipped with eye-tracking technology and a calm environment to perform the interviews. 16 tests were recorded, 2 of them had technical issues while recording, so they were excluded from the analysis. The remaining 12 were not recorded nor analysed, because it would not be necessary to gather information of so many users, so we would save some resources not recording nor

[1]digimedia.web.ua.pt

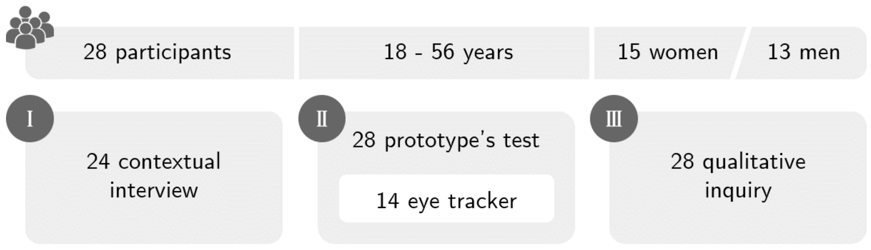


Figure 3.1: The distribution of the participants in the participatory study.

analysing the remaining ones. Nielsen states that, on average, with only 5 users we can find 84% of the usability problems in a system [39]. Still, all the 28 participants tested the prototype according to the proposed tasks, so they were able to express their opinion by the end of the tests, in the last moment of the session, the qualitative inquiry. 4 participants did not enter in the contextual interview, because they were the first ones to execute the tests to check if the tasks were well-balanced. Their tests were okay, so the results were also included, but the contextual interviews were not performed because their opinions would be biased after being already presented with a solution during the test. The participants' distribution is summarized in Fig. 3.1.

3.3 Results

The study was very fruitful in different ways, as we will see below. As a whole, it provided the necessary tools i) to understand what was correct or needing to change, ii) to discover new ideas and iii) to prioritize things, according to the importance people put in some issues. The following topics are not an exhaustive presentation of the results. Instead, it is a critical reflexion on some representative components, discussing their importance to this project and hopefully to other similar projects.

3.3.1 Contextual interview

As the first moment, interviews allowed to better understand the needs of people in the context that the app aims to cover. We will start by talking about this in more detail, followed by other topics that turn out to be all related to this core idea.

Users needs

When the participants were asked about what they imagined the app would be about, without any knowledge of what they were going to test after, the top three suggestions were i) tracking their reads, this is, registering the books they plan to read or they already read (9/24, 38%), ii) reading eBooks (9/24, 38%) and getting book recommendations (8/24, 33%). i) was already planned and included in the prototype. ii) was a nice feature, but incompatible with the current vision of the app, that is targeting printed books, so although the preference of people was clear, it was necessary to understand that it is not possible to satisfy all the uses cases and to make all the features compatible, even when the subject is the same: reading books. iii) was an use case already planned, but not in the short term, because good recommendation requires many data, and the app would not have it in its early days. Given the importance people gave to this issue, this use case was prioritized to meet this need earlier.

Different backgrounds

It was interesting to verify that different people would point out different opinions, according to their backgrounds. For example, when asked about what were the advantages of printed books and eBooks over each other, some people said that eBooks tire the eyes, because of the screen, or that they get more easily distracted because of the notifications or the temptation to do other things in the smartphone, while other people pointed out that the screens of book reading devices such as Kindle are comfortable and they even look like real paper. As we can see, different people provided opposite opinions for the same issue. The difference was on their context, in this case, their familiarity with eBook readers.

Opposite opinions

Often even the same participant manifested apparently contradictory visions. As we mentioned above, 9/24 (38%) of the participants predicted that the app would be to read eBooks, but when asked for their preference on printed books or eBooks, 20/24 (83%) said to prefer printed books. After, when asked about what would be the most used format to read books in the future, only 2/24 (8%) said printed books. For the remaining, 12/24 (50%) said eBooks, 8/24 (33%) said both and 2/24 (8%) did not provide a final answer. Analyzing these results required to distinguish people's personal preferences from their expectations for the other people. Probably getting their personal preferences was more enriching, because knowing what people want for them is more accurate than knowing what people think that other people want for them. In a future opportunity it would be better to avoid questions that tend to be this second case. So, for example, they should be asked what would be the format to read books that *they would be using in the future*, instead of what would be *the most used format* to read books in the future. By asking the former it would still be possible to answer the later, when grouping the responses of all the participants, and it would be more accurate.

Validation

Some subjects discussed during the conversations were useful to validate the opportunities for the app as a product (see Section 2.3.3). For example, 20/24 (83%) participants said that they have books at home they never read. They are not likely to read these books in the near future, and these books could be the books that other people are looking for. They could make some money reselling these books to other people that would not mind to buy second-hand books at a lower price, as many other people already do, for example, in Facebook groups to exchange books (see Section 2.3.1). Looking to another example, 12/24 (50%) people said that they already had issues with loans. In most cases they forgot who they lent a book or, even remembering, the other person would not return it and they were not comfortable asking it back. One of the app's goals is to mediate the exchange of books. People will be able to track their loans, and there is a trigger for returning them, because the loans are explicitly registered and visible to both users.

The right questions

During the interviews it was also possible to understood the importance of asking more than a *yes* or *no*. For example, when people were asked if they had books at home that they would not mind to exchange by other books with other people, 18/24 (75%) said *yes*, and 6/24 (25%) said *no*. Looking only for this answer we might be led to believe that millions of books lost in the people's houses could be being exchanged, that we are facing an excellent business opportunity. But, when justifying their answers, from who said *yes*, 13/18 (72%) pointed out that they would only exchange the books they did not like or with which they no longer identify. Who said *no* pointed out that they like to have their books with them, some people telling that they have an emotional value. With the insights of the detailed answers of the participants it was possible to look from a more moderate perspective at the opportunity of convincing people to use an app to exchange books, because the relation they have with their books can be a hitch to this.

Short-term and long-term

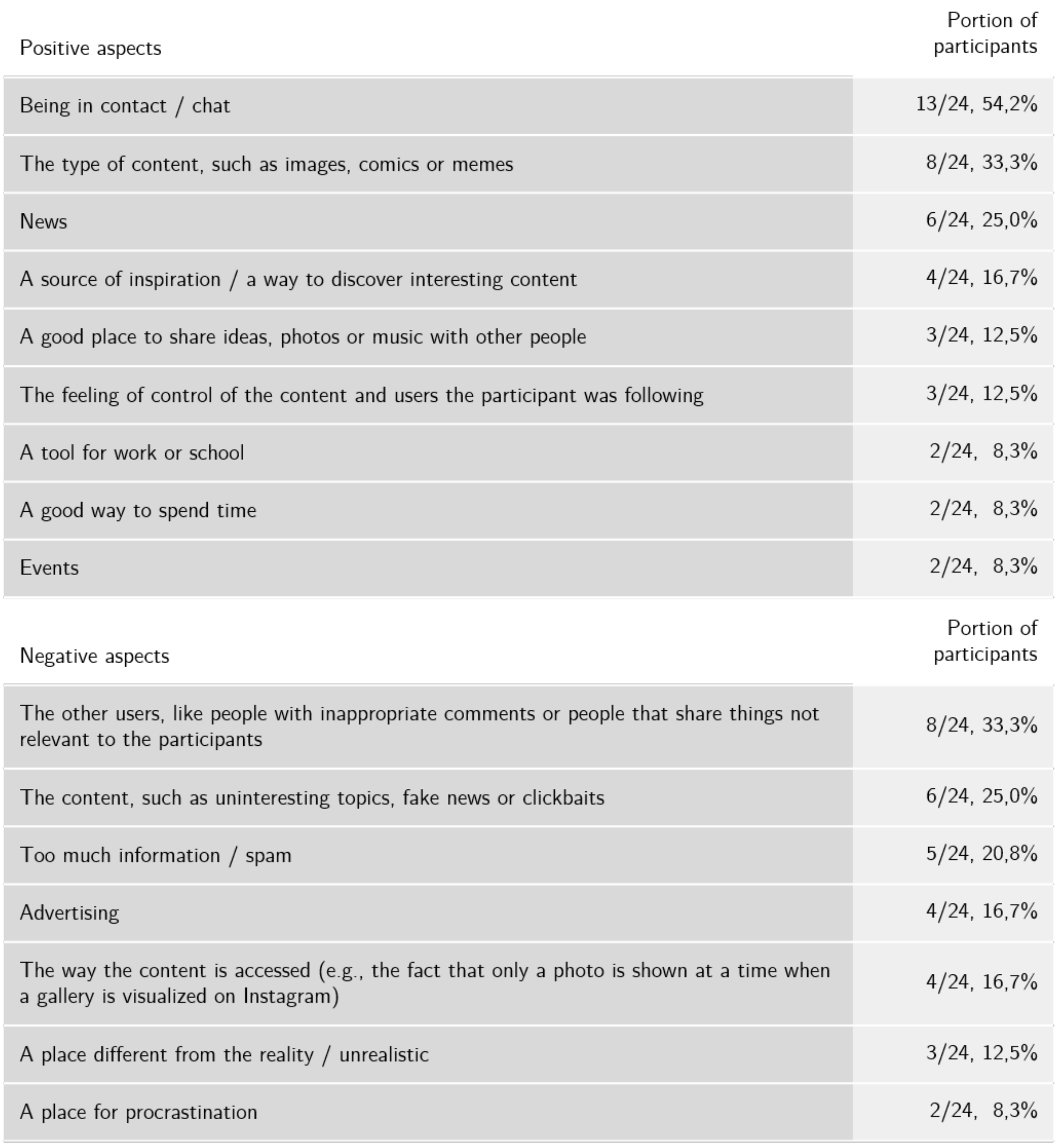
Some suggestions were considered relevant for the context of use of the app, but not viable in the short-term, needing to be better elaborated in a future reflection. For example, a participant told that, during the reading process, they use an app with a dictionary to search for the meaning of words. The app for this project already allows to track the reading progress, by marking the current page. With the observation of this participant it came the idea of providing a sort of reading mode in the app, where users could track the time they spend reading, update the current page, search for the meaning of words or do others things that would help them during the moments they are reading. The app could even have some reading challenges, like badges for reading over a defined number of hours or days in a row. This idea was kept for future reflection, so when the key features are addressed and the app is stable, this will be evaluated and considered as a possible feature. This and other insights were useful to plan interesting use cases that need further exploration and will be part of the plan for the long-term.

Secondary limitations

It was also important to check possible technical limitations, that go beyond the users preferences. Not only the opinion questions were relevant, but also simple things like asking if they own a smartphone, what is the operating system and if they have mobile data on their devices. This type of information helped to contextualize and to understand some of the opinions for the scope of this project, that is centered on the app itself. 21/24 (88%) of the participants revealed to use mobile data. This is relevant because, if they did not, it means that they would only have access to the internet in places with Wi-Fi, maybe only at home, work or a cafe. In the meanwhile, on their travels, they would not. For example, at the exact moment they were exchanging a book with someone on the street, they would not be able to rely on the internet to confirm the delivery on the app. So, it would be important to make the app available offline, for instance allowing to mark a delivery as confirmed and leaving feedback while offline, and automatically synchronizing it when online, notifying the other user. But developing the app to work offline has extra costs of implementation, so this is not a linear decision. Knowing that the strong majority of the participants have mobile data, it is fair to consider that it is not mandatory to develop features to support the app while offline, thus alleviating the costs of implementation.

Further research

Finally, these contextual interviews were a motivation for a research related to this project. Regarding the social component of the app, participants were asked about the social network they used the most: i) *What do you appreciate the most in this social network?*, ii) *What most annoys you in this social network?*, iii) *Considering the last 12 months, has there been any change in how often you use this social network? If yes, why?*. With these open questions, the participants had the possibility to freely express their opinions, pointing out one or more aspects. The aspects were after analysed and grouped as positive or negative, as shown in Fig. 3.2. 15 participants referred to Facebook and 9 referred to Instagram. Some of the aspects were characteristic of one social network. For example, *the feeling of control of the content and users the participant was following* was pointed out from the participants who chose Instagram, and *a tool for work or school* was pointed out from the participants who chose Facebook. Looking at these opinions from a broader perspective, we can observe that people value the content, the way they access it and the way they control it. This led to wonder how the way in which social networks and other sources of information are designed influences how people interact with this information, particularly with regard to their attention. This resulted in the elaboration of another study and subsequent writing of an article entitled *How to deal with mindless scrolling?*, which is further detailed in Section 8.1.



Positive aspects	Portion of participants
Being in contact / chat	13/24, 54,2%
The type of content, such as images, comics or memes	8/24, 33,3%
News	6/24, 25,0%
A source of inspiration / a way to discover interesting content	4/24, 16,7%
A good place to share ideas, photos or music with other people	3/24, 12,5%
The feeling of control of the content and users the participant was following	3/24, 12,5%
A tool for work or school	2/24, 8,3%
A good way to spend time	2/24, 8,3%
Events	2/24, 8,3%

Negative aspects	Portion of participants
The other users, like people with inappropriate comments or people that share things not relevant to the participants	8/24, 33,3%
The content, such as uninteresting topics, fake news or clickbaits	6/24, 25,0%
Too much information / spam	5/24, 20,8%
Advertising	4/24, 16,7%
The way the content is accessed (e.g., the fact that only a photo is shown at a time when a gallery is visualized on Instagram)	4/24, 16,7%
A place different from the reality / unrealistic	3/24, 12,5%
A place for procrastination	2/24, 8,3%

Figure 3.2: The aspects pointed out by the participants about the social network they used the most.

3.3.2 Functional prototype's test

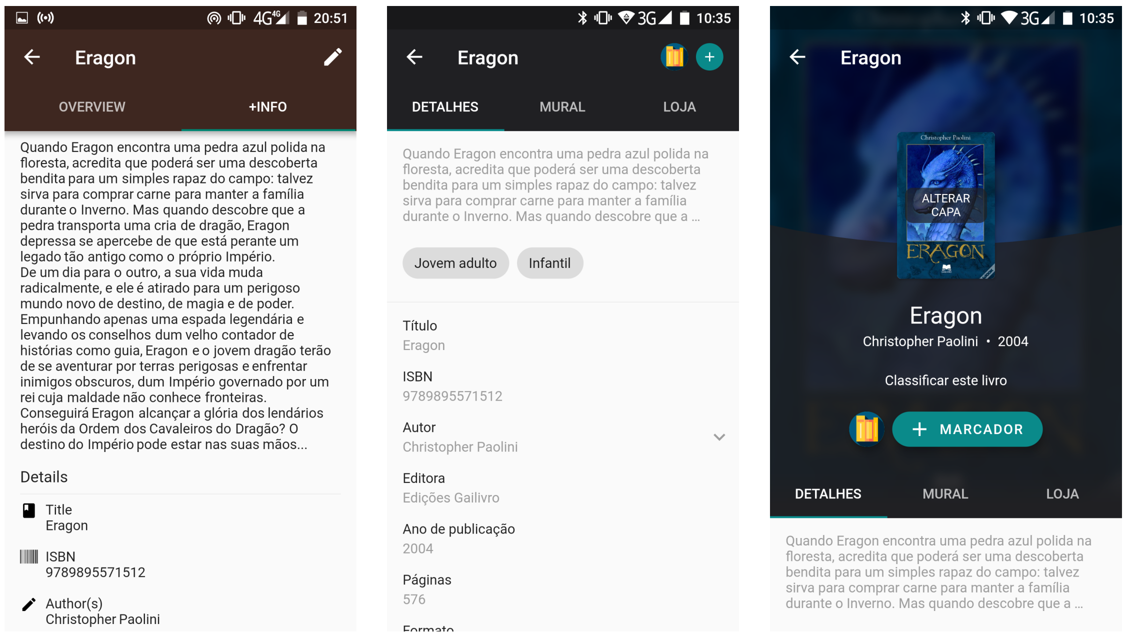
As planned, after the interviews the participants were invited to test a functional prototype of the app, as described on Section 3.1.2. The tests ran as expected. The participants were able to perform the proposed tasks successfully, with no major issues. It was possible to observe the successful application of some usability principles [6][40] and, in some cases, their absence, as we will see now. Some representative changes that were implemented are reported along the topics discussed below. Note that this is not an exhaustive but representative review.

Redundancy

A good design provides alternative paths to accomplish the same goal. When asked to quote a book, some users decided to go to the book profile, where there is a view that lists all the posts from that book, and other users went to the mural, where all the posts are. This was because they interpreted the steps of the task differently. Some thought about the object (the book) first, others thought about the place (the mural). Both alternatives were valid, and if one was missing, a portion of the users would be searching around for some more seconds until finding the option to publish the post. Sometimes redundancy might be noisy, and lead to confusion, but this is an example where it fits well.

Discoverability

Discoverability, this is, the ability to figure out what, where and how things can be done [40], can be decisive for the success of a task. At some point, after accessing the profile of a book to read its summary, participants were asked to mark it to read later. The task was apparently simple, there was a highlighted button just at the top of the profile where they could tap to add it. Still, several users had difficulty to use it. And that was because it was not visible. This happened because, when the book view was open, it was being shown in the header, but after scrolling it to read the summary, it was not showing anymore. This would also happen with any other view inside the book profile, when scrolled. Some users, after finding it, commented that they reminded the button but forgot how to go back to it, until they found it. Maybe in the next time they would know where it is, and this would not be an important issue, but bookmarking is probably the main action in a book profile, and it would not make sense to have it so inaccessible and, worse than that, forcing the users to "unscroll" every time they need to use it, leaving what they are doing at that moment. Given this, after the study the interface was changed to make this option always visible (see Fig. 3.3).



Quando Eragon encontra uma pedra azul polida na floresta, acredita que poderá ser uma descoberta bendita para um simples rapaz do campo: talvez sirva para comprar carne para manter a família durante o Inverno. Mas quando descobre que a pedra transporta uma cria de dragão, Eragon depressa se apercebe de que está perante um legado tão antigo como o próprio Império.
De um dia para o outro, a sua vida muda radicalmente, e ele é atirado para um perigoso mundo novo de destino, de magia e de poder. Empunhando apenas uma espada legendária e levando os conselhos dum velho contador de histórias como guia, Eragon e o jovem dragão terão de se aventurar por terras perigosas e enfrentar inimigos obscuros, dum Império governado por um rei cuja maldade não conhece fronteiras. Conseguirá Eragon alcançar a glória dos lendários heróis da Ordem dos Cavaleiros do Dragão? O destino do Império pode estar nas suas mãos...

Details

Title
Eragon

ISBN
9789895571512

Author(s)
Christopher Paolini

Quando Eragon encontra uma pedra azul polida na floresta, acredita que poderá ser uma descoberta bendita para um simples rapaz do campo: talvez sirva para comprar carne para manter a família durante o Inverno. Mas quando descobre que a ...

Jovem adulto
Infantil

Título
Eragon

ISBN
9789895571512

Autor
Christopher Paolini

Editora
Edições Gailivro

Ano de publicação
2004

Páginas
576

Quando Eragon encontra uma pedra azul polida na floresta, acredita que poderá ser uma descoberta bendita para um simples rapaz do campo: talvez sirva para comprar carne para manter a família durante o Inverno. Mas quando descobre que a ...

Figure 3.3: The book profile, before the changes, on the left, and after the changes, on the center. On the right, it is the view before being scrolled. Before the changes, the bookmark button is not visible when scrolled, and after the changes it gets collapsed at the top right of the screen, being always accessible.

Icons and words

This was not the only issue with the button to bookmark. The button had a plus sign followed with the word *adicionar* (in english, *add*). Some participants said that it looked like an option to add the book to a shopping cart, not to bookmark. Some said that the icon and the word were vague, and could be many things. On top of that, this misunderstanding was probably aggravated because of the existence of a store in the app, so users knew they could buy books in the app, and that option could be that. This misunderstanding illustrates well the need to take the necessary time to use the right icons and words when designing the interface. They must be self explanatory, concrete.

Constraints

It was interesting to notice that some constraints on the system helped to guide the users on their tasks. When leaving feedback to someone, after simulating a sale of a book, users had a dialog to choose form three emojis, symbolising a positive, neutral or negative opinion, and a short message describing their experience. Some people started by the message, and only after they chose the emoji, making use of the options in reverse order of what appeared in the interface (the emojis were above the text). There were even people who did not chose a face, but both were mandatory. When trying to submit the dialog, they noticed that the button was not active, and so they easily understood that something was missing, and finally chose the face and successfully submitted the dialog when the button was activated (see Fig. 3.4). The use of constraints (in this case the visual perception of the deactivated button, in grey) helps users to reduce the range of alternatives, so they clearly understand what to do

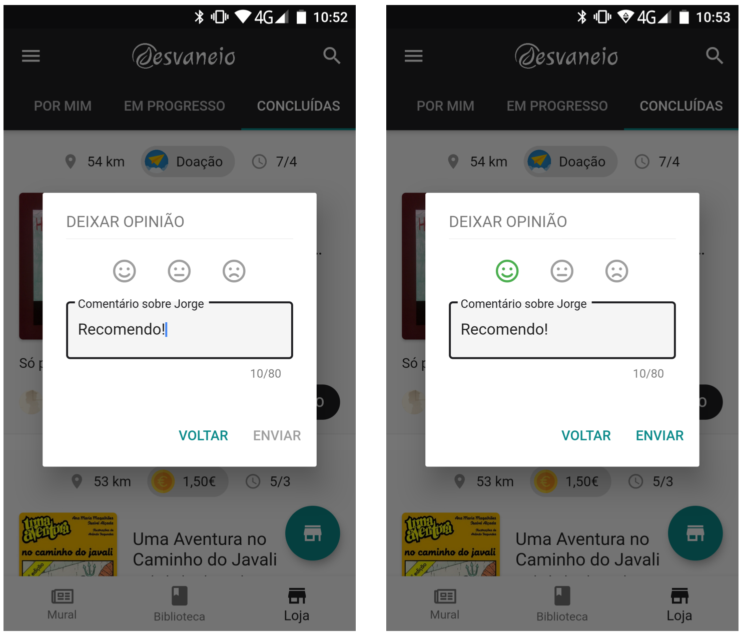


Figure 3.4: The button to submit feedback disabled before choosing an emoji, on the left, and enable after choosing it, on the right.

and fail less often. An alternative to that particular example could be having the submit button always active and having a message pointing out that a field was missing when the user tapped on it, like it is usual in long and complex forms. But for this simple situation this would lead the user to fail first when it could be easily prevented.

Still related to the topic of constraints, some users said things like *Is that it?* or *It can only be here!* while they tapped on things in the interface. When starting a post, there was even a participant that said *I am not sure what I am doing, but this option stood out*, which could be an indicator that the participant is lost, but most likely not, because they performed all the tasks without issues. In general, the participants referred that the interface was clean and intuitive, that the features were easy to find. Probably part of that minimalism was decisive to constrain actions and, consequently, it provided a more efficient experience.

Minimalism

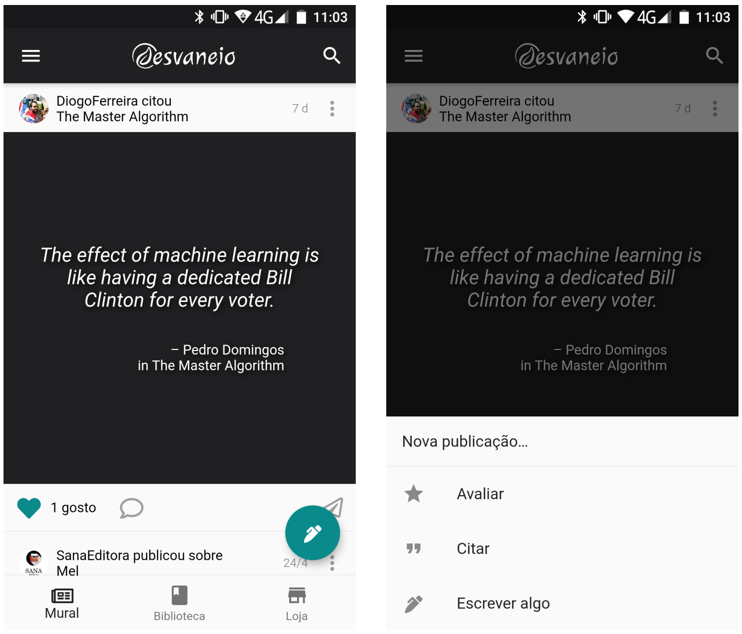
As we will see highlighted in Section 3.3.3, it was consensual among the participants that the interface was clean. Even so, there were several details that needed to be improved, and the participants talked about this. For example, the book profile had an option in the top bar (in Android it is called *app bar*) to easily allow users to suggest editions to book details. Some participants said that this option does not need so much visibility, for example *When using the app I guess that I will only need to suggest editions in exceptional circumstances, book details are not edited all the time.* What they talked about is directly linked to the practice of providing different levels of detail. The interface should show the essential options first, and the remaining second. With that in mind, this option was changed to be less highlighted.

Error prevention

During the tests the participants also experienced the problem of lack of a confirmation dialog or an undo option in an important step of the workflow in offers. After deciding to exchange a book, each user could confirm the delivery of the book. Some users touched the confirmation button when not asked to, and it immediately performed the action, with no way to go back. In the tests it was not problematic, because confirming the delivery was something planned later in the plot, but if the app was in production it would not be very comfortable if someone would do it unintentionally. Given this, later when redesigning the app this button was changed to show a confirmation message.

Recording the tests

Participants touched "everything" in the interface, even in places we would not expect them to. It is easy to lose track of this endless sequence of touches during the tests, and this is something where the ability to capture the screen during the experiences was very important. Only when playing the records it was possible to identify some patterns that were common to some participants, that were not noticed before. For example, four participants touched the book cover when trying to find the summary of a book. The book cover was just an image, with no action attached to it, and it was not expected that it had the affordance to be touched. But it did. Maybe because people expected to "open" it. When redesigning the prototype, it was changed to allow accessing the summary from there. Looking to another example, when tapping the button to create a post, it would show a dialog from the bottom with the options of types of posts to make, and with a title above saying *New post...* (see Fig. 3.5). A participant tapped the title - that once again had no event associated to it - before choosing an option, and they were asked if it went as they expected. They answered *I guessed it was supposed to write something there, but no. Maybe because of the suspension dots.* It looks like the title was being confused with some type of input. When checking the recording it was noticed that other participants did the same, and maybe it was because of a thing as simple as three dots. Other very simple examples were quite subtle and would not have been noticed if the experiences could not be revisited later. Simple issues that do not block the successful use of the system, but, as a whole, might be significant to the overall experience.



The effect of machine learning is like having a dedicated Bill Clinton for every voter.

– Pedro Domingos
in The Master Algorithm

Figure 3.5: The dialog that shows up after pressing the button to make a post. The suspension dots after *Nova publicação* (in english, *New post*), might be misleading.

Keeping it realistic

While preparing the tests, a special care was taken in the way the participants would understand the story they would be listening when asked to perform the tasks to be tested during the experiences. To have a good scenario, not only it was important to have a good *plot*, but also a good *stage*. And the stage was the app. The views were populated with illustrative content that would make sense, that would seem real. The more it would look authentic, the more the tests would be closer to the reality. Of course, the participants would still be in a white room, with a camera pointed at their eyes, but at least a good prototype would help to abstract them from these obstructions and to really image themselves doing the given tasks. Despite this effort, there were a couple of views that were not populated as they should. For example, there were no publications in the profile of a book where the participants were asked to visit. The view was empty, showing a message *There are no posts here.* Some participants reached this view when searching the place to make a new post, read the message out loud and thought they were not in the right place. Maybe it was because they saw it empty. What would be their reaction if the view was already populated? Maybe it would be more self-explanatory, because they would have examples of other posts that would be an hint for the action they were about to perform.

Unnecessary tasks

During the tests, when simulating the workflow since the creation of an offer to the exchange of a book, there was a task where participants needed to use a chat to match the day and place of the delivery. This task turned to be trivial, because people are already used to the interface of a chat like it. This reminds of the importance of prioritizing the tests to the interfaces that do not follow the standards. It is with the new that users are most easily surprised and where it is likely to be more things to fix.

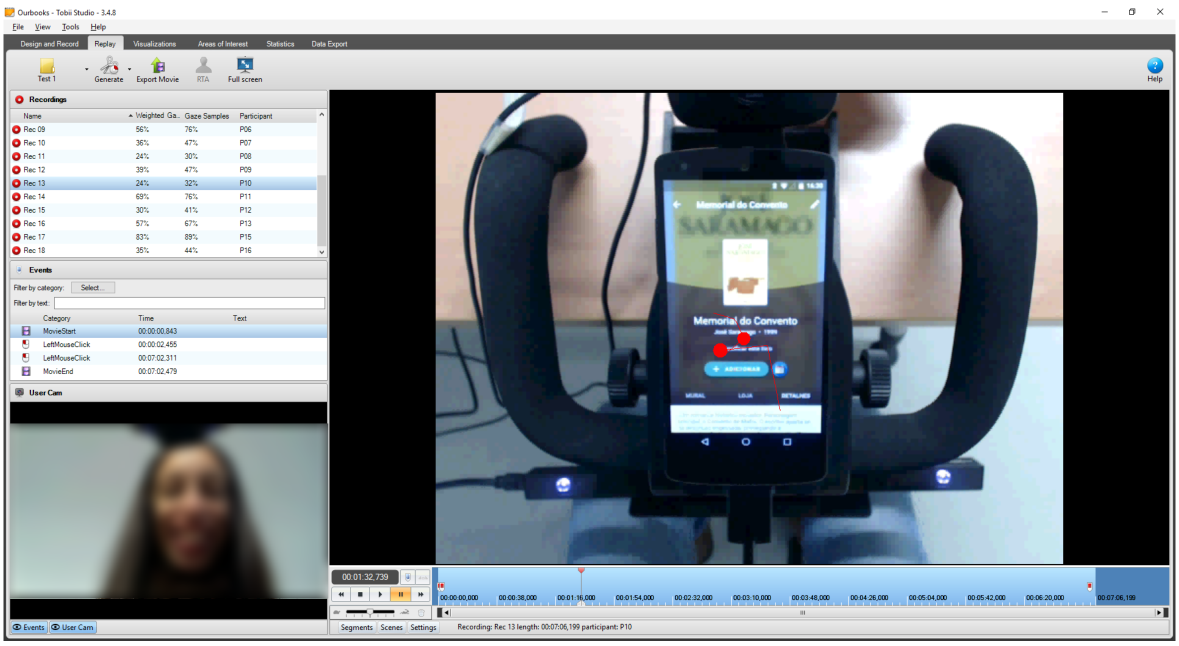
The number of testers

As referred in Section 3.2, 16 participants realized the tests with the full setup, with the eye tracking and recording. The remaining 10 also tested the prototype, but without this equipment, so the analysis to their experiences was not so granular, and the focus was on getting their opinions in the qualitative inquiry, and not so much in the usability problems. Still, the 16 participants that tested with the full setup were more than enough for the issues found in the prototype. The first 10 participants faced nearly all the problems that were found by the total 16, which is in line with the recommendations of Nielsen for usability tests, that says that with only 5 users we can find 84% of the usability problems in a system [39]. On the other hand, the amount of insights received in the qualitative inquiry was so vast that it was worth it to invest in more participants (we will talk about this inquiry next, in Section 3.3.3). If the intention was to assess the usability of the interface, a smaller number would be enough, but because the goal was to really understand how they would imagine themselves using the app, the big picture of what they liked and what they did not, all the insights were welcome and the number of participants was appropriate.

Eye tracking

The eye tracker allowed to complement the insights from the tests with the capture of the movement and fixation time of the eyes along the screen (see Fig. 3.6). Tobii Studio, the software that supported the recording, observation and interpretation process, provides different visualization possibilities, such as heat maps and gaze plots. Fig. 3.7 shows the heat maps of an offer detail's view in the first 5 seconds the participants looked at it. It is possible to notice that the main focus of attention is on the button to accept the offer (*aceitar proposta*), which is understandable given the emphasis that the button has and considering that it is related to the action that the participants had to perform at that moment. However, the participants also looked at other areas, namely the picture of the other user, that applied to the offer. It is natural that people are interested to know who they are dealing with, but maybe it is odd that they paid less attention to other important details, such as the book to be sold. Crossing this fact with the comment of a participant that said *there is a lot of information that has to be interpreted in this view and I think that the image of the other person could be reduced*, it was assumed that people's attention was being affected by the highlight given to that element, so this was taken into account when redesigning the view (see Fig. 3.8).

Regarding gaze plots, Fig. 3.9 shows an example with a book detail's view in the first 2 seconds the participants looked at it. There, it is noticeable that the attention went first to the central area, where there is text with the title, author and year (possibly because at that moment users were asked to search by the book's summary), while other areas, like the elements at the top and the tabs at the bottom, were secondary.



Recordings

Name	Weighted Ga...	Gaze Samples	Participant
Rec 09	56%	76%	P06
Rec 10	36%	47%	P07
Rec 11	24%	30%	P08
Rec 12	39%	47%	P09
Rec 13	24%	32%	P10
Rec 14	69%	76%	P11
Rec 15	30%	41%	P12
Rec 16	57%	67%	P13
Rec 17	83%	89%	P15
Rec 18	35%	44%	P16

Events

Filter by category: Select...

Filter by text:

Category	Time	Text
MovieStart	00:00:00,843	
LeftMouseClick	00:00:02,455	
LeftMouseClick	00:07:02,311	
MovieEnd	00:07:02,479	

00:01:32,739

00:00:00,000 00:00:38,000 00:01:16,000 00:01:54,000 00:02:32,000 00:03:10,000 00:03:48,000 00:04:26,000 00:05:04,000 00:05:42,000 00:06:20,000 00:07:06,199

Recording: Rec 13 length: 00:07:06,199 participant: P10

Figure 3.6: Tobii Studio replaying the test of a participant. The image of the app's screen is complemented with the eye movement (represented by the red dots) and the face and speech of the participant.

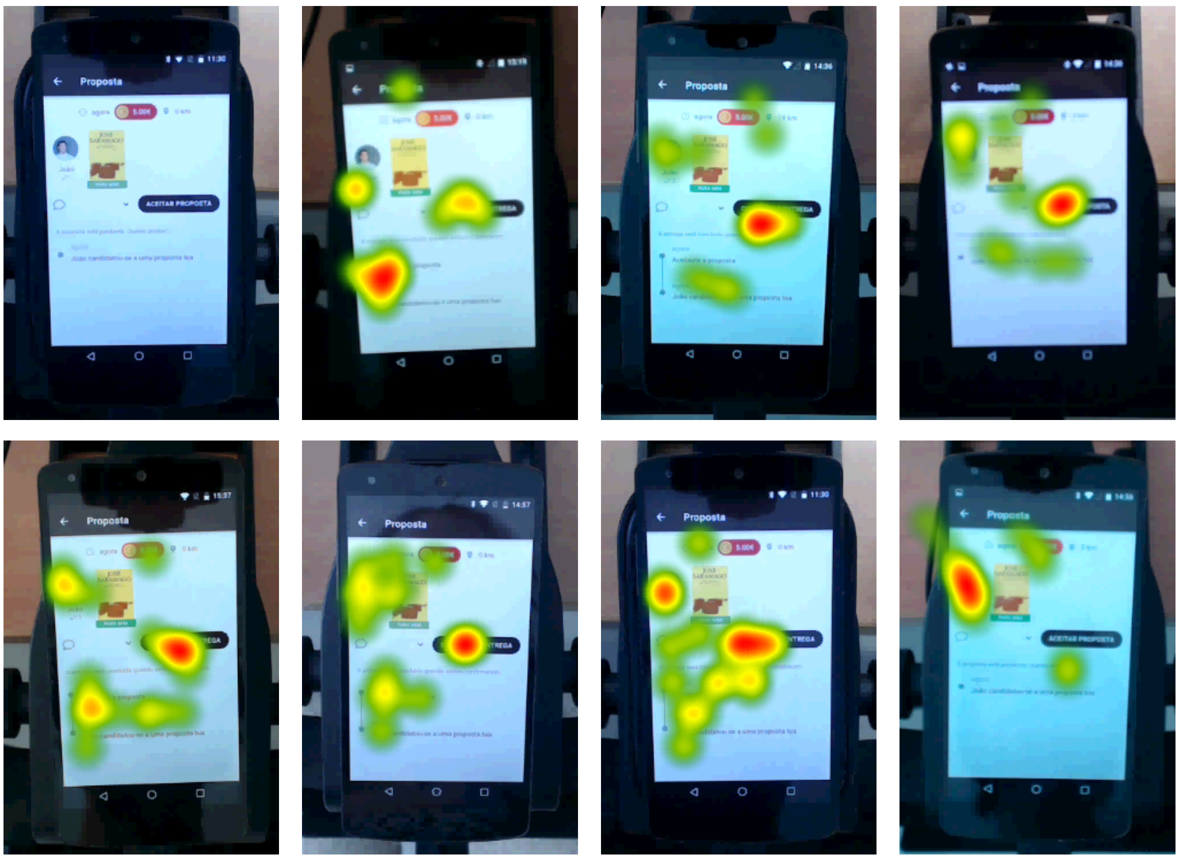
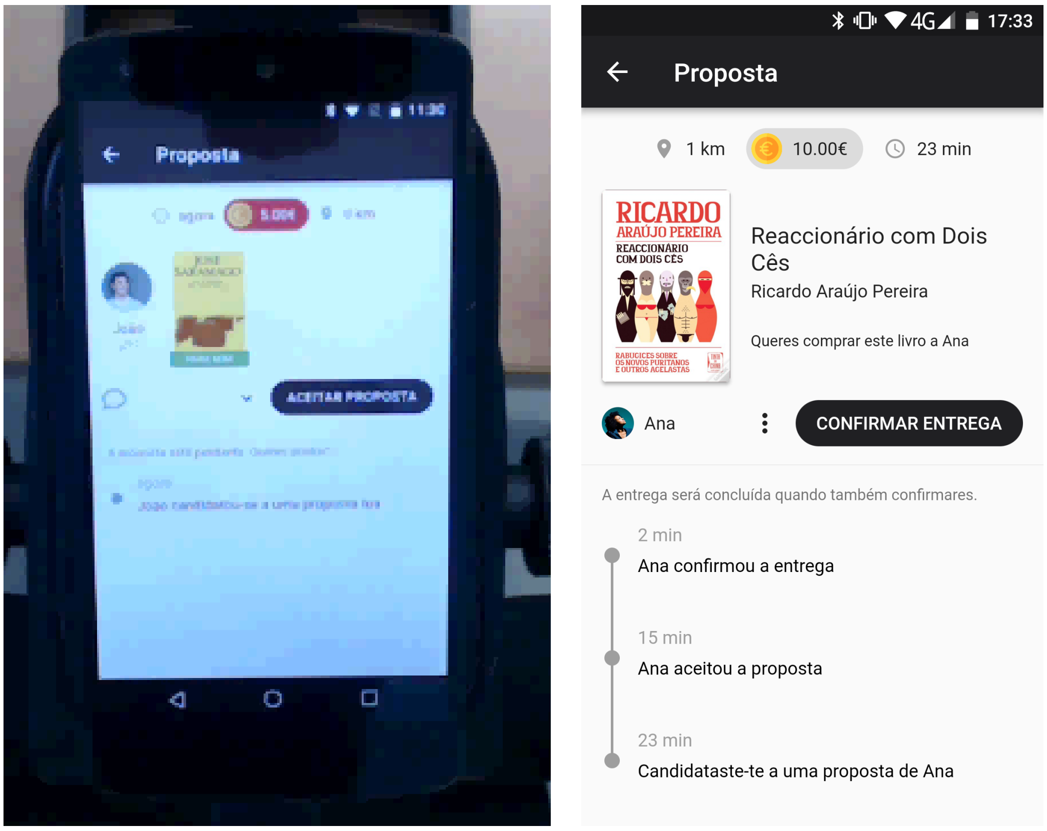


Figure 3.7: The heat maps of an offer detail's view in the first 5 seconds some of the participants looked at it. The hottest areas represent the places where the eyes were fixed longer.



Reaccionário com Dois Cês
Ricardo Araújo Pereira
Queres comprar este livro a Ana
Ana
A entrega será concluída quando também confirmares.
2 min
Ana confirmou a entrega
15 min
Ana aceitou a proposta
23 min
Candidataste-te a uma proposta de Ana

Figure 3.8: An offer detail's view during the tests, on the left, and after the redesign, on the right. There are several changes to the placement and size of the different elements.

In addition to these examples, other situations were inspected throughout the recordings, always with the insights from the remaining analysis in mind, because it would not be feasible to analyse and compare the eye tracking data from all the recordings exhaustively. In general, no anomalies were detected, which was in accordance with the fact that the tests have been completed successfully, with no significant difficulties. However, it should be noted that the interface was dynamic, with constant changes, such as the navigation between different views and scrolling, which complicated the analysis by areas of the screen. Moreover, the screen of a smartphone is smaller than, for example, the screen of a computer, where people would be more likely to get lost, so in those situations the help of the eye tracker would be even more valuable.

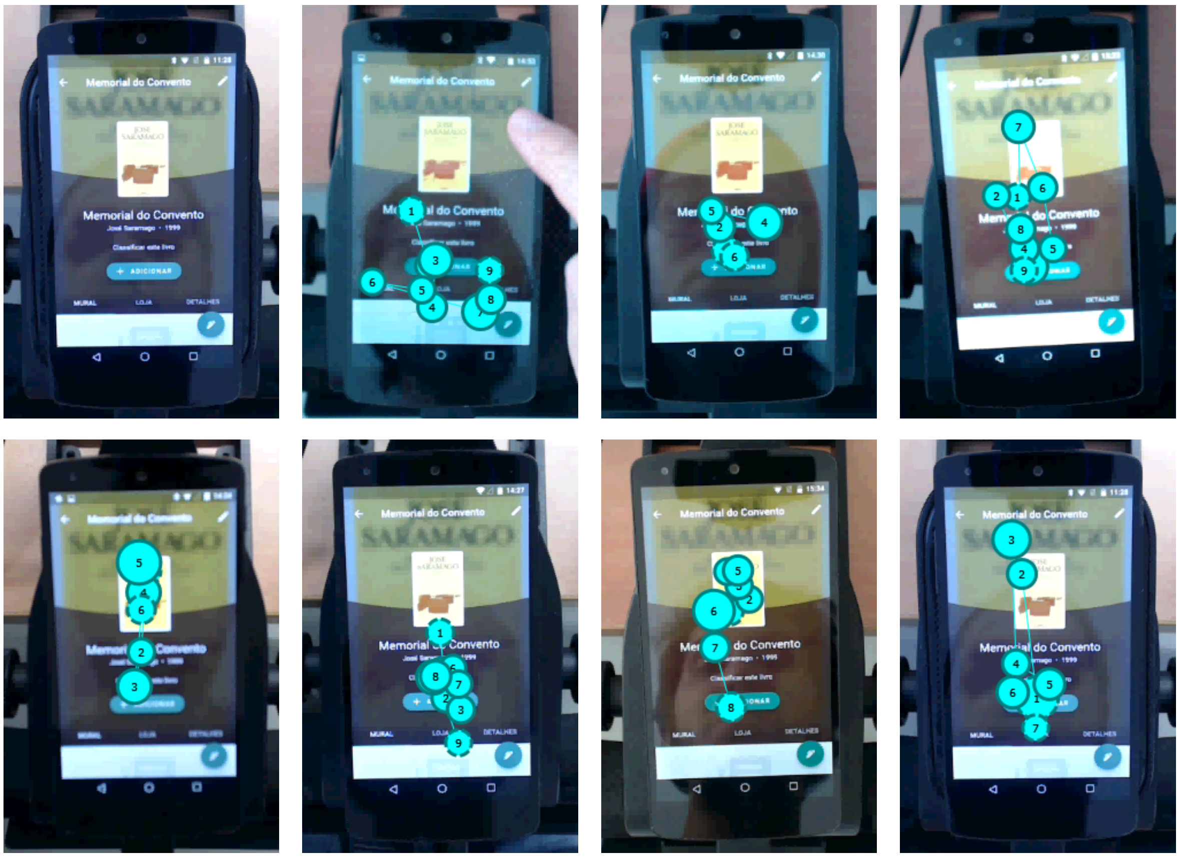


Figure 3.9: The gaze plots of a book detail's view in the first 2 seconds some of the participants looked at it. The circles represent the places where the eyes fixed and are numbered according to their movement.

3.3.3 Qualitative inquiry

The tests were specially useful to focus on the usability. After this moment it was important to complement it with a deeper understanding of how was the experience with the prototype.

Questionnaire

The participants filled a small questionnaire, as described in Section 3.1. This allowed to compare the app's performance and the users' feedback during the tests to their explicit verdict. There were no issues found on this questionnaire (see Fig. 3.10), but if they were, it would be necessary to revisit the tests in search of the possible causes.

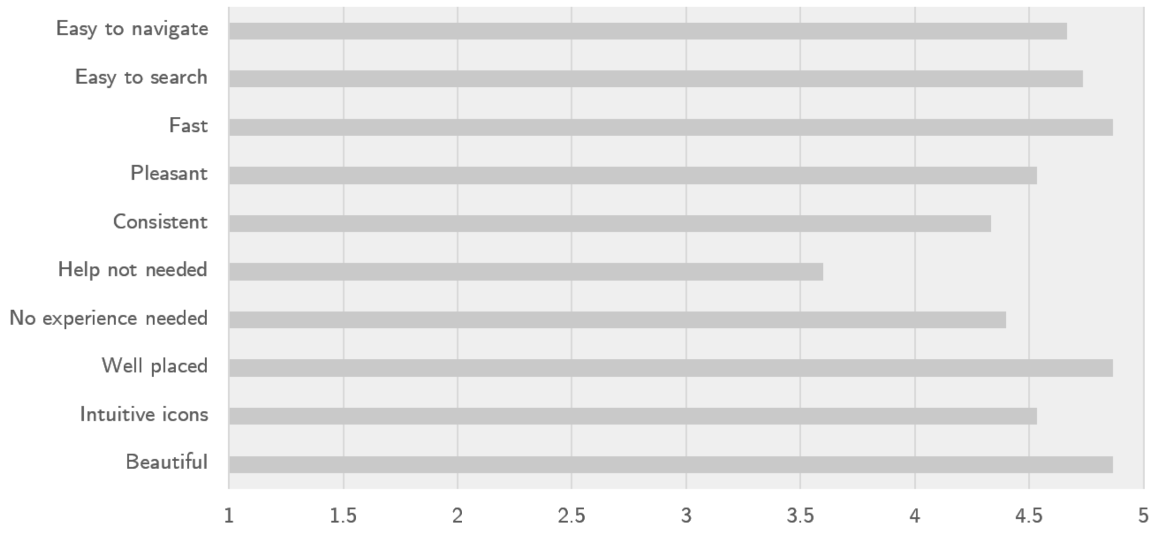


Figure 3.10: The system's usability evaluation in different aspects, using a Likert scale with a range from 1 (strongly disagree) to 5 (strongly agree). The questionnaire is available in Appendix A.

Improvements

After the questionnaire, people recognized some issues that would need to be improved if using the app in their daily lives. For example, the prototype allowed to search by title when finding for books in the store. Some users suggested to enable search by author and genre, because they would imagine themselves looking for inspiration, without a book in mind, and it would be easier if they could get it based on their favorite authors and genres.

Particular vs general

Some participants reported use cases that might be very specific to their experience as readers. For example, one participant said *It is essential to me to know who is the translator of a book, because there are bad translators that I do not trust anymore, so I won't read their books.* This information is not easy to retrieve from the existing databases, so nor the prototype nor the current version of the app shows it for the majority of the books. Therefore, for this and other features suggested by the participants it was necessary to consider their feasibility and the portion of people who manifested it.

Clarification

These conversations were also useful to clarify some use cases of the app. For example, when talking about the feature to scan the barcode of a book a participant said *This is very useful. I imagine myself in a library, taking a book, easily scanning it, and voilà, I can get all the information about it, including reviews.*

These were the three main moments planned for the sessions. The first one, the interview, was more divergent, subjective and independent from the solution that would be presented to the participants in the second moment, with the prototype's test. These tests converged the participants to a solution. During this moment it was possible to get some technical insights, which was quite effective in showing the robustness of the prototype and highlighting the issues to be corrected. In this analysis a special focus was given on what went wrong, because it was what allowed to understand the mistakes to avoid in the future and to improve the prototype. Other usability topics like i) feedback, ii) mappings, iii) consistency and iv) help were not discussed here, but would also deserve to be referred in a complete overview. This discussion went towards where it could be a contribution to be made, namely by including examples, rather than speaking exhaustively about a subject that is so well detailed in the literature (for more references on usability please see [6], [40] and [41]). The last moment allowed to access the prototype and the proposal for the app, now more conditioned by the solution that had been presented. It was very important to highlight the strong and weak points in the design of the app, allowing to gather a lot of new changes. By the end of the study, all the suggestions were analyzed, filtered and organized to be addressed in three different stages: i) the first public release, ii) the version 1.0 and iii) later in the future, in the long-term (for more information on the roadmap see Section 7.1). It would make sense to separate the first moment from the rest, so we would have two different sessions, and it would be possible to use the feedback form the first one as input of the second one. For logistic reasons, namely the short time to execute this project, it was decided to do it in a single session. The complete procedure is present in Appendix A.

CHAPTER 4

Development

"What I cannot create, I do not understand." – Richard Feynman

The participatory design study allowed to crystallize the requirements for the app. At this time, the focus was on the development. Throughout the whole process, design and development were side by side, iteratively, with the requirements from the design influencing the development. In Chapter 5 and Chapter 6 we will talk about the remaining stages in the cycle, respectively the deployment and validation of the solution, and how the insights from there acted as input to the design. For now, we will discuss some of the implementation decisions regarding the app and backend.

4.1 App

As we saw in Section 2.2.4, there are currently cross-platform solutions to keep a single code base when developing for Android and iOS. This allows to speed-up development and reduce costs. Currently, among the possible cross-platform choices, native solutions seem to be the obvious way to go. They offer a unique toolset to develop cross-platform solutions without compromising the performance. React Native and Flutter are now the two dominant technologies, and they were compared when deciding how the app would be implemented.

4.1.1 Flutter vs React Native

React Native, from Facebook, has been around for some time (since 2015), so it is more mature. Flutter, is fresher (version 1.0 was released on 5 December 2018), and takes advantage of its youth, being a framework that hopefully learned from the mistakes of the previous ones. Here are some of the points considered when exploring both to find the best fit for the app of this project [23].

Stability

React Native is hopefully more stable than Flutter, mainly because of its age, and considering that there are high standard apps using it. This includes Facebook and Instagram, and other apps outside the Facebook monopoly, like Pinterest or Skype [42]. This is a guarantee that the framework is prosperous. On the other hand, Flutter is recent and only now the first apps are starting to show up. At the moment, the best examples of apps built with Flutter are Alibaba and Google Ads, the platform from Google for advertising [43]. Considering this, React Native takes advantage.

Performance

Both React Native and Flutter rely on their efficient architecture, that maps the components written in the framework to their respective native components (see Fig. 4.1). In React Native this is done by an intermediate called a bridge, that maps the User Interface (UI) and services (e.g., location, bluetooth). Flutter removes the need of a bridge, with the usage of channels instead, to map services, that are more efficient [44]. Also, while the UI in React Native is generated using the Original Equipment Manufacturer (OEM) widgets, this is, the original widgets from the OS, Flutter uses its high-performance rendering engine to generate the UI, which allows more customization, while not compromising the performance. However, this moves the rendering to the application side, resulting in larger application sizes. Still, for the UI, the need of a bridge to the UI is also not necessary in Flutter, unlike React Native, that uses it to map to the OEM widgets.

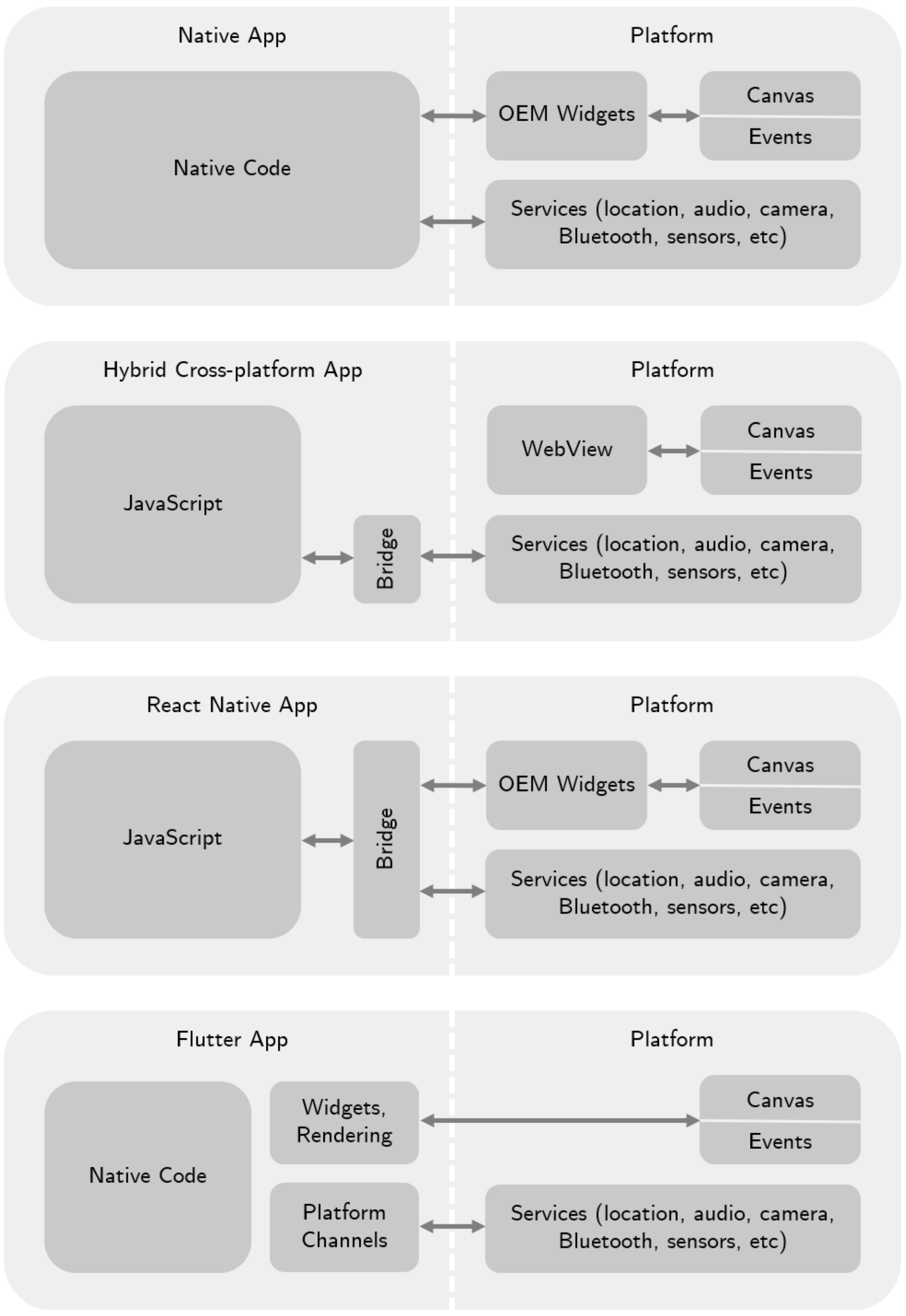

Native App
Platform
Native Code
OEM Widgets
Canvas
Events
Services (location, audio, camera, Bluetooth, sensors, etc)
Hybrid Cross-platform App
Platform
JavaScript
Bridge
WebView
Canvas
Events
Services (location, audio, camera, Bluetooth, sensors, etc)
React Native App
Platform
JavaScript
Bridge
OEM Widgets
Canvas
Events
Services (location, audio, camera, Bluetooth, sensors, etc)
Flutter App
Platform
Native Code
Widgets, Rendering
Platform Channels
Canvas
Events
Services (location, audio, camera, Bluetooth, sensors, etc)

Figure 4.1: From the top to the bottom, the architecture of i) native apps, ii) hybrid cross-platform apps, iii) React Native apps and iv) Flutter apps. Source: [44]

Productivity

Productivity is key to develop apps faster. Both React Native and Flutter are equipped with a hot reload feature, which allows the developer to instantly reflect the changes in a running app on save. React Native is programmed in Javascript, which is an huge plus for who comes from the web. In Flutter, a new language has to be learned: Dart.

DevOps

Flutter comes with a good command line interface, which allows things like upgrading the framework or building apps. The same happens with React Native. Even so, Flutter looks more complete. It even has *flutter doctor*, a tool to check and solve problems in the setup of the development environment. With Flutter, it is straightforward to follow their documentation to deploy apps with fastlane [45]. On the other hand, React Native only provides a manual process for deploying, which is not so satisfactory [46]. Flutter official documentation comes with material on tests and continuous integration [47]. It even has a tool to build, test and deliver Flutter apps, developed purposely for Flutter, called Codemagic. React Native does not.

Community and popularity

Both React Native and Flutter are experiencing an impressive momentum. There is a lot of information about these technologies across the several platforms where developers share their ideas, problems and solutions, such as GitHub, Stack Overflow and Medium. Figure 4.2 shows the popularity of these frameworks in Google Trends, compared to Xamarin. During the last months, in the context of this dissertation, the evolution of these platforms on Stack Overflow and GitHub was also recorded, which resulted in the registration of the number of questions on Stack Overflow and the number of stars on their main repositories on GitHub (see Fig. 4.3). We can observe Flutter gaining popularity over React Native.

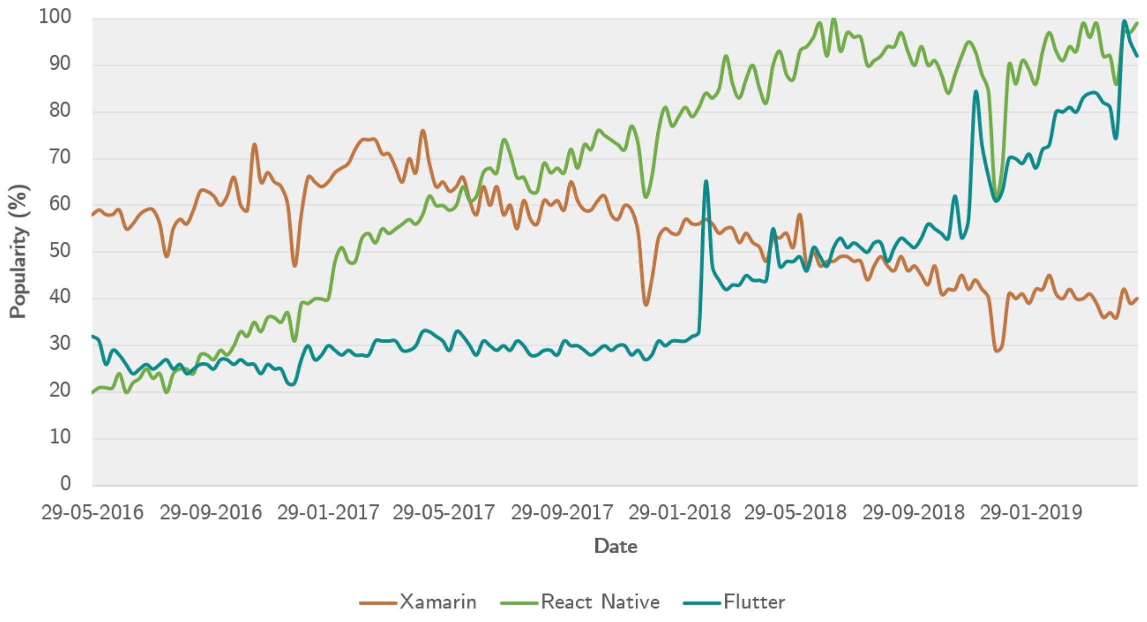


Figure 4.2: The popularity of Xamarin, React Native and Flutter over the last three years (from 23 May 2016 - to 23 May 2019), worldwide, according to Google Trends. Xamarin is progressively loosing ground and Flutter is the one that has grown the most in the recent months.

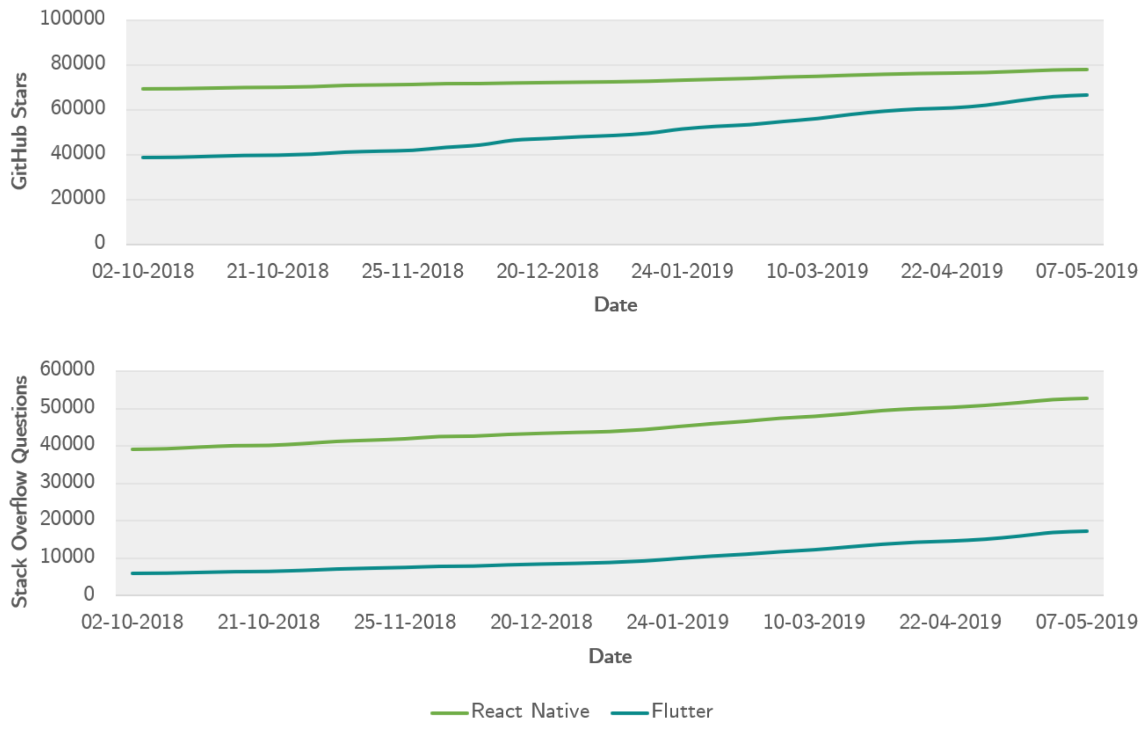


Figure 4.3: The number of questions on Stack Overflow and the number of stars on on GitHub of React Native and Flutter, from 2 October 2018 to 7 June 2019.

Plan for the future

Based on the characteristics we observed above, native cross-platform apps are promising solutions for the mobile. Still, who is planning the next step, the incoming years? On 5 December 2018, during the live event where Flutter 1.0 was released, Google not only delivered its first stable version, but also talked about the next years. And they are planning a future where Flutter might run across all the devices, not only on mobile, but also web and desktop. They are already developing the necessary tools for that. By the end of the conference, they minimized the screen with the slides that were supporting their presentation, that were developed in a desktop program built with a version of Flutter for Desktop, on MacOS [48]. This moment was symbolic of their vision for Flutter: not another mobile development framework, but ideally a solution for cross-platform development for all the devices that have a screen. About React Native, it already has a version for the web, called React, that actually came before React Native, in 2013. Besides that, around 5 months after the Flutter event, Microsoft announced React Native for Windows, during Microsoft Build 2019. This indicates that, in the long-term, React will also expand their focus to the remaining platforms. Before that, React Native already had non-official spin-offs for desktop, such as Proton Native and React Desktop, both with more than 8.000 stars on GitHub, on 7 June 2019.

Contrary to what would be expected, Flutter seems to have an healthier presence than React Native. And all it starts by the official documentation, that is full of examples. There is even an official YouTube channel that has several tutorials. React Native is out for 4 years, and yet it seems easier to start learning Flutter than React Native. In Flutter things look more up-to-date than in React Native. React Native is backed by its maturity, but Flutter is changing the game. And change is essential for evolution. There are already cases of apps that moved from React Native to Flutter [44] [49]. After weighting in accordance with the criteria described above, the choice for the app to develop in the project ended up being Flutter. Still, it would also be a good choice to opt for React Native. Today native cross-platforms are the way to go in mobile development, and both Flutter and React Native are expanding their scope to other platforms, which is a plus in the long-term.

4.1.2 Advantages of Flutter

After choosing Flutter to develop the app for this project, it was possible to have a deeper understanding of the potentialities of this framework. Prior to this work I had the possibility to develop a couple of simple native Android apps. When moving to Flutter I noticed several differences in the way it softens the development. Until now we had the big picture of the advantages of using a cross-platform solution (Section 2.2.4 and Section 4.1.1). Now we have an attempt to extract some low-level characteristics, more related to the coding part, that are specific to Flutter. Most of them should be equivalent in React Native.

High-level

A framework that embraces two different platforms (Android and iOS) necessarily introduces complexity. Each platform has specificities that need to be unified in a single API, in order to make possible to share the same source code for the two target platforms. Despite this, this single API results in an abstraction that simplifies many other aspects. Then, programming in Flutter means not worrying about things that otherwise would need to be handled when programming natively. For example, in Android we often need to be careful about the compatibility of some features with different versions of Android, resulting in inconvenient if-clauses (if running version X, do this, else do that). With Flutter all we need to know is that it compiles the code to be compatible to Android Jelly Bean (4.1) or newer, and iOS 8 or newer [50].

Another example of this abstraction is related to the activity lifecycle in Android [51]. When the screen rotates, the view needs to be readjusted to the new orientation, so the activity is destroyed and created again. With this, the programmer needs to save the state of the app before the activity is destroyed and load it when recreated, otherwise the progress in the view is lost when the screen rotates. In Flutter, we do not need to worry about preserving the state when the screen rotates, because it handles it transparently.

Powerful language

Flutter is programmed in Dart, a language that is designed to make the most of the web and mobile. Comparing to Java (Android), it is more flexible and compact. For example, it supports dynamic variables, with type inference, and has methods with optional arguments. There is no need to explicitly define getters and setters, and the parameters in the constructors are automatically mapped to the variables, without verbose assignments (see Fig. 4.4). Also, functional programming is a plus in Dart, something that is not in the essence of the imperative style of Java. Even though streams and lambda functions were introduced in Java 8, bringing some functional characteristics, Android is only compatible with Java 8 starting on Android 7.0 [52].

User interfaces are full of events and asynchronous tasks, like pressing a button and showing a loading indicator until a content is ready to be displayed. In Dart, with the reserved keywords *await* and *async* it is possible to handle these requirements in an elegant way, eliminating the need to deal with nested callbacks and other tricks to manage the flow of dynamic interfaces, with complex effects and animations that are usual in mobile apps and websites. So, Dart is much more suitable for these use cases. It is worth to note that most of these characteristics are shared with JavaScript, that is also optimized for the needs of the web. JavaScript is the language in which React Native, the main competitor of Flutter (see 4.1.1), is programmed.

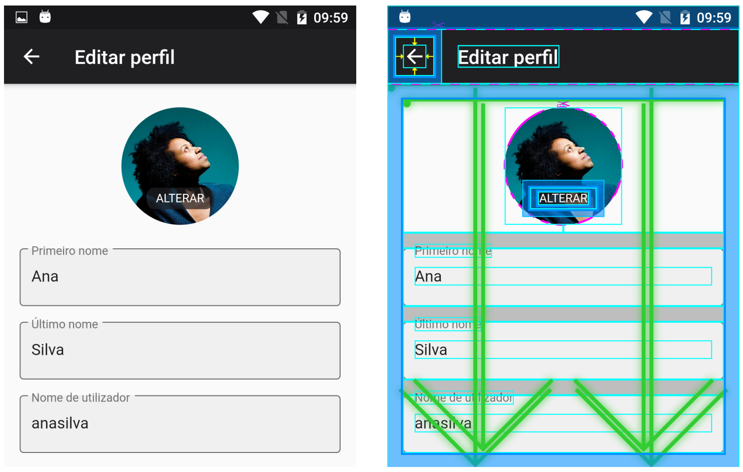
```
// Java
public class Book {
  String isbn;
  String title;
  List<String> authors;

  public Book(String isbn, String title, List<String> authors) {
    this.isbn = isbn;
    this.title = title;
    this.authors = authors;
  }
}

// Dart
class Book {
  String isbn;
  String title;
  List<String> authors;

  Book(this.isbn, this.title, this.authors);
}
```

Figure 4.4: A simple example of code in Java, above, and Dart, below. In Dart constructors are less verbose, as well as in several other situations, although the syntaxes of the two languages are very similar.

Easier design

In Android the design and functionality are kept separately. There are XML files to define the templates and Java files to define the behavior. This is positive in the way that it matches the conceptual differences, defining a clear separation of concerns from the UI and logic. Also, hopefully it promotes better structured source code. But it brings some adversities. First, it is necessary to map the components in both files, so XMLs are full of identifiers that are referred in the Java files. Also, XML files are static, and often the interface is dynamic, so even if the default UI is defined in the XML files, the Java files still need to handle it according to what is happening at runtime. So, there is an effort to keep both design and logic separated and dependent at the same time. In Flutter, these two are defined together, through a compact and elegant way. All the components in the UI are widgets, that are characterized by their visual and functionality. Widgets are nested together, and can preserve states and be exchanged at runtime according to the defined logic. Fig. 4.5 shows an example of a widget made by combining multiple widgets.

Flutter is paying particular attention to the fully native experience in an app developed in Flutter. Besides performance, it is also important the match of the design with the OS standards (such as animation effects in lists, icons or buttons), that are different in Android and iOS. Flutter has a complete set of UI components with the design of the respective platforms, so developers do not need to make them by their own or rely on any third party packages for the majority of their needs. The same does not happen with React Native, that does not have as much components out-of-the-box as Flutter.


09:59

Editar perfil

ALTERAR

Primeiro nome

Ana

Último nome

Silva

Nome de utilizador

anasilva

```
Widget changePictureView = GestureDetector(
  onTap: changePicture,
  child: Stack(
    alignment: Alignment.bottomCenter,
    children: <Widget>[
      UserPictureView(
        user: user,
      ),
      Positioned(
        bottom: 8.0,
        child: TransparentLabel(
          text: Translations.of(context).actionChange,
        ),
      ),
    ],
  ),
);
```

Figure 4.5: The view to change the user picture, which is a widget made of multiple nested widgets (using the *child* attribute). Above, it is the whole screen to edit the profile, with and without the Debug Paint tool. Below it is the correspondent code (just of the view to change the user picture, not the entire screen). The view is a *Stack*, that is overlapping a list of two *children*: the picture itself (*UserPictureView*) and a label (*TransparentLabel*) with the text *alterar* (in english, *change*). Note that TransparentLabel is inside a *Positioned*, which is also a widget that is forcing a margin at the bottom. Both *UserPictureView* and *TransparentLabel* are custom widgets that are being reused. The former receives a user object and draws its profile picture inside a circle and the later displays a given text inside a colored box with some transparency. Finally, the *Stack* that holds them is a child of *GestureDetector*, a widget that detects all the gestures inside it. In this situation it is listening to tap events to trigger *changePicture*, a function that handles the change of the profile picture.

Packages

Dart has a repository called Dart Packages, where the packages developed by the community are officially maintained. This repository scores each package with criteria like code analysis, degree of maintenance and popularity, which is particularly useful when choosing a package to use. The developers of the packages are encouraged to describe them, add setup instructions, a demonstration example and a changelog. These requirements also influence the score of the plugins. Having a single official repository reduces headaches searching for plugins in different places. It can be argued that this not new. For example, we already have the Mvn Repository, for Java, or the Python Package Repository, for Python. But they do not work in the same way, specially in terms of unnecessary complexity. Dart Packages is much more simple and self-explanatory, answering the question *How do I use this package?* straight to the point. One can verify it just by visiting Dart Packages (https://pub.dartlang.org/), Mvn repository (https://mvnrepository.com/) and Python Package Repository (https://pypi.org/).

4.2 Backend

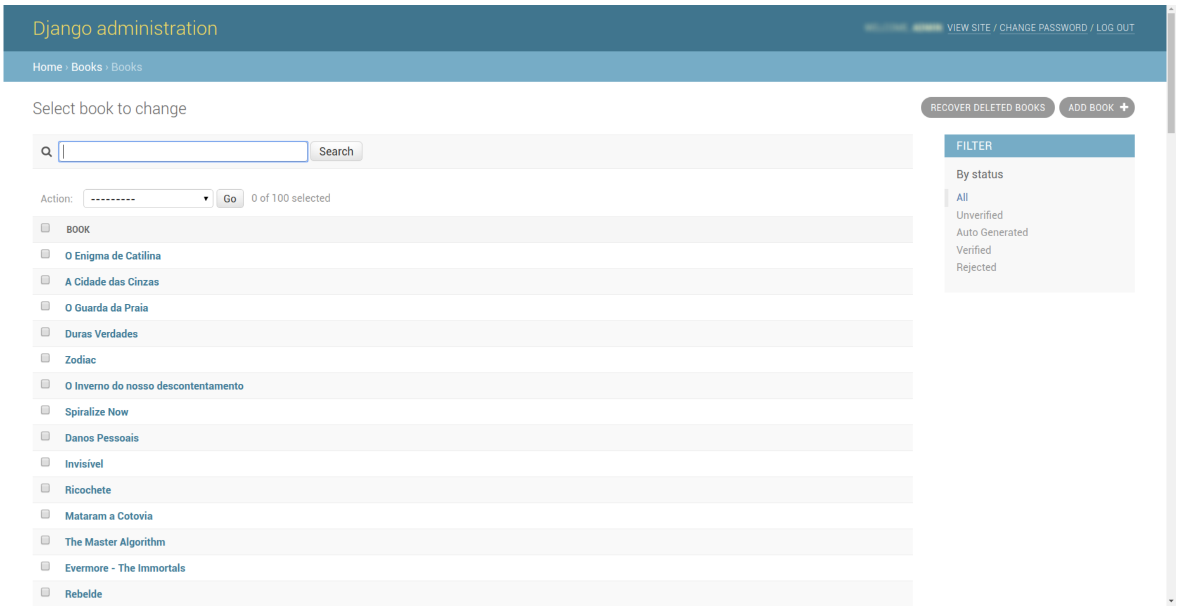
The app is supported by a backend, that handles things like communication and data persistence. These were the decisions when considering the app's backend, served by a REST API.

4.2.1 REST API

As we saw in Section 2.2.1, there are several issues faced when dealing with REST APIs, and currently there are other solutions used by high standard companies. In this project the choice was to build a REST API, mainly because it is still the standard way to build web APIs, and the time would not allow to explore something bolder. Nevertheless it is important to notice that REST is not the ultimate solution for web APIs, and that the game is changing in this area. As an example, one of the inconvenient consequences of choosing it for this project is related to the problem of getting many resources in a single request. When consuming the API in the app side, it was needed to make several requests for each view, even for simple ones. For example, when displaying an offer to sell a book, it was necessary to request i) the offer, ii) the user that made the offer and iii) the book, in three different requests. When loading a list of offers, the number of requests increases accordingly. This issue would not be faced if there was other way to retrieve multiple resources in a single request.

4.2.2 Using Django

Having defined that the system architecture would follow the mainstream client-server paradigm with a REST API, it was time to choose the solution for the server. One of the decisive criteria for choosing one is the language in which it is programmed. For this project the choice was Django, mainly because of the previous experience and familiarity with the framework and the language. As already mentioned in Section 2.2.2, web frameworks usually come with tools that accelerate development like authentication, an ORM and plugins. Django is no exception, having them out-of-the-box. Some examples of plugins that were useful



Django administration

[redacted] VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Books › Books

Select book to change

RECOVER DELETED BOOKS ADD BOOK

Search

Action: --------- Go 0 of 100 selected

	BOOK
	O Enigma de Catilina
	A Cidade das Cinzas
	O Guarda da Praia
	Duras Verdades
	Zodiac
	O Inverno do nosso descontentamento
	Spiralize Now
	Danos Pessoais
	Invisivel
	Ricochete
	Mataram a Cotovia
	The Master Algorithm
	Evermore - The Immortals
	Rebelde

FILTER

By status

- All
- Unverified
- Auto Generated
- Verified
- Rejected

Figure 4.6: Django Admin listing the books catalogued in the system, allowing to perform CRUD operations, that are integrated with the text search and filter tools.

for this project are Django REST Framework, django-reversion and django-friendship. Django is primarily targeted for server-side generated websites, so it does not produce a web API by default. Django REST Framework enables it. For some of the entities in the data model it was needed an history of changes, as it is the case of the editions to a post. Django-reversion allows version control. For the social component of the project it was necessary to handle users connections, like friends and follows. Django-friendship comes with this. These examples illustrate how the use of plugins speeds up the development of a web application, just by joining existing pieces with the necessary data modeling and logic from project to project.

Django also comes with an administration panel, an interface that automatically maps the entities into lists, tables and forms, allowing to easily perform Create, Read, Update, Delete (CRUD) operations, which can be useful for some administrative tasks. The tabular interface is not so user friendly, but it is an acceptable solution for something that requires almost no configuration and is not aimed to be used by the end users (see Fig. 4.6).

This integrated ecosystem provided by Django also has its cons. For example, one might not want to use the integrated authentication system. When developing a microservices architecture (see Section 2.2.6), the application could rely on an external authentication service. When doing it, the existing authentication system will be there, but not being used anymore. The same applies to other features that one might want to split in several services. Django is monolithic, and heading to a micro-services solution might end in solutions that do not seem to be taking full advantage of the potential Django would give otherwise, in monolithic applications [53].

Django powers many projects, from small to big ones. Instagram is one of the largest deployments using this framework [54]. They use it because of its simplicity, but it comes with a trade-off: efficiency. They always have this concern in mind, considering that they have millions of users. There might be other alternatives, more efficient, but less simple. Simplicity is key to a project that is beginning, like this one. Also, we will not be serving millions of users, and there is always the possibility of tweaking Django to accommodate any needed changes, or even migrating to other solution if it becomes necessary in the future.

4.2.3 Choosing a database

Django ORM is flexible, being ready to work with several database engines, including relational (SQL), like MySQL and PostgreSQL, and non-relational (NoSQL), like MongoDB. It also has connectors to search engines, like Elasticsearch, and supports other type of technologies for data persistence. As Instagram points out in [55], it is a good principle to go with proven and solid technologies when we can, and in Section 2.2.3 we saw that NoSQL databases still need some time to become more stable, so the choice for this project was for a relational database. In this project it was used PostgreSQL, mainly because it is open-source and has a full-text search engine from which Django takes advantage, such as sorting results by relevance according to a ranking function, like how close are two search terms [56]. With PostgreSQL, some of these search features could be used. Django also supports MySQL, and it would also be a good option, but without the full-text search capabilities.

4.3 Development process

With the technological choices made, the development of the project as a whole was divided into two main tasks: the app and the backend. They were developed in parallel and incrementally, according to the goals defined for the MVP (see Section 2.3.4). In each increment the solution was validated and the remaining steps were adjusted to the new insights. It would be exaggerated to say that it was followed an agile methodology, due to the project size and team dimension (one person), but all the work was done with these principles in mind. Only like this it would be possible to consider the user / client as the central entity in the process. This iterativity was more evident when the first prototype was developed for the participatory design study (see Chapter 3) and after the release of the first public version of the app in the Play Store, as we will see in Chapter 6.

4.3.1 Tests

Tests are necessary because we all make mistakes. And some mistakes can be expensive. In this project, imagining that there is an error in the workflow of an offer, somewhere from its creation to the moment a user leaves feedback to another, that would be expensive. Users are relying in the system to exchange books for money (even if the money transaction by itself is by their own), so any gap in the process would directly influence their trust in the app. Another example is related to the visibility of the information in the app. Users can choose to keep their books private. If someone is reading *Fifty Shades of Grey* and does not want anyone to know about it, it is extremely important to make sure that this information is hidden from all the other users in all the circumstances. This can be more tricky than it seems, because the app has several views that deal with book lists, and those views are different depending on who is accessing them, so just a missing line of code to filter out the private books could result in an embarrassing bug.

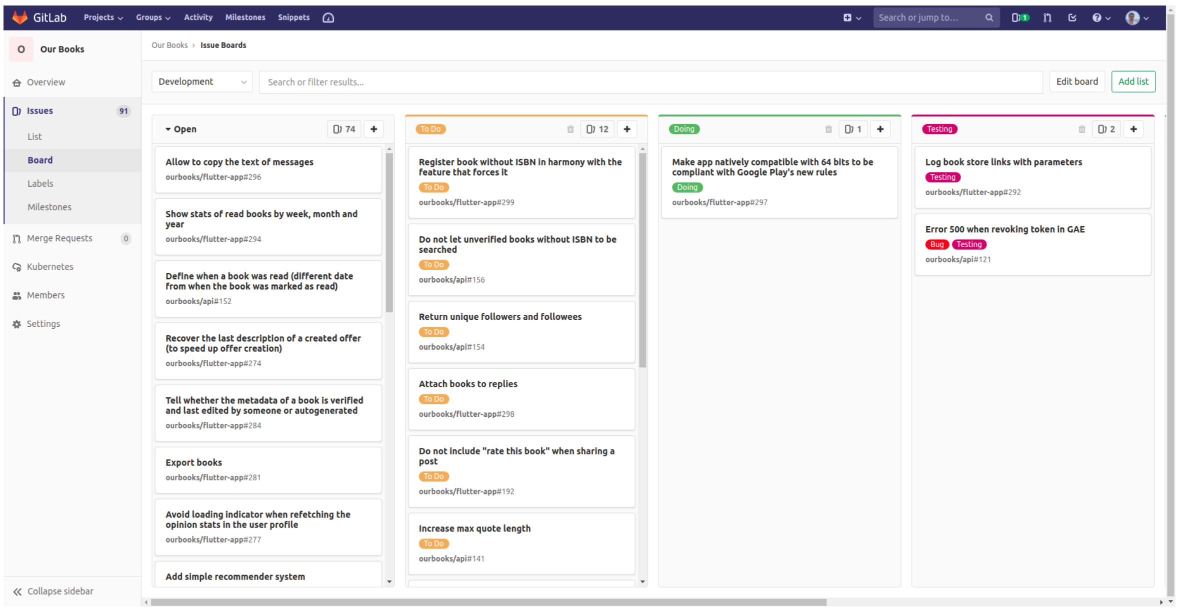
Tests come to help preventing these scenarios. They give the developer the confidence that their code is healthy. Testing does not only make sure the code we just did is working, but that any future changes to that code are covered. Even a small change can result in unpredictable behaviors. Imagining that a user reports a bug and it is fixed in a future release without a test, and for some reason a change unpredictably reintroduces that bug, that situation would not be tolerable. So, tests are essential and should not be neglected in real-world solutions, specially when they involve people. Even so, one must also recognize that it is impractical to test all the possibilities. It would cost more time in the development process. Also, given all the iterativity of the design and development process, some features come and go, so it is important to decide what is essential.

In this project some use cases were prioritized and tested in accordance to that. The main focus was on the backend, because it is the core of the system, where the data is maintained and manipulated. Several unit and integration tests were written, which allowed to reproduce the complete workflows of some use cases, as it is the example of a sale of a book. For the app, initially it was planned to be the same, but testing it is trickier. First, because it involves testing an interface, and second, because the interface has been evolving from version with version, which would force to rewrite the tests to keep up with the changes. Therefore, initially a couple of tests were written, just to give it a try, but they are not representative, so they are not reliable to make sure the app is working. In the future, when there are more resources and the interface is more stable, it will be done.

4.3.2 Issue tracking

We have been talking about features, bugs and iterative increments, but how they were managed? There are several tools that help managing projects, like Pivotal Tracker, Trello or Asana, providing ways to manage tasks, prioritizing and estimating the time to complete them and distributing them throughout the team. In alternative to these tools, that generally come with paid plans, git hosts also provide their solutions. Although they are a simplified version of these tools, with less features, they are more close to the code. They have the advantage of being directly linked to the version control of the repository, allowing to link the tasks to the changes in commits and merges. For this project, the source code is hosted on GitLab, and they provide a tool called *Boards*, which allows to manage issues in a visual manner. They were organized in lists, according to their current status (see Fig. 4.7). Also, it is possible to define milestones and to track their progress over the time. They have been used to control the development of each release (see Fig. 4.8). When releasing a new version, the summary of the changes is documented and added to the changelog (see Fig. 4.9).

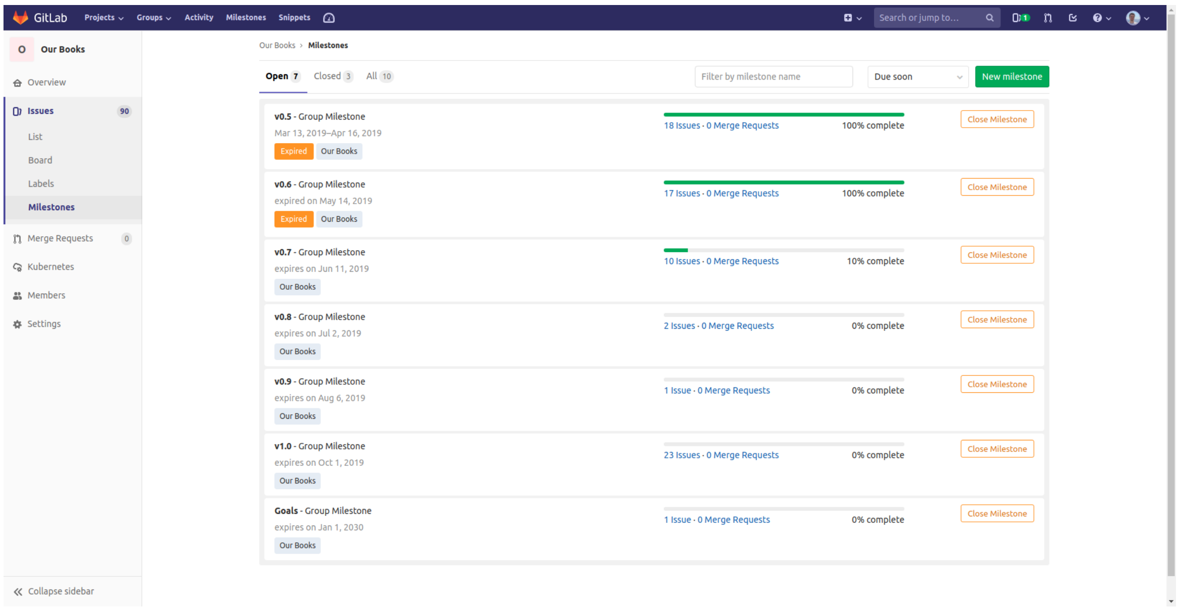
These were the main topics taken into account for the development phase. Other apps may face the same questions but not the same solutions. It all depends on the requirements of each project. Because of that, it were presented the various possibilities and their pros and cons. Hopefully, the decisions for this project may be the solutions for other projects. As it


Our Books › Issue Boards

Development

Open (74)	To Do (12)	Doing (1)	Testing (2)
Allow to copy the text of messages — ourbooks/flutter-app#296	Register book without ISBN in harmony with the feature that forces it — To Do — ourbooks/flutter-app#299	Make app natively compatible with 64 bits to be compliant with Google Play's new rules — Doing — ourbooks/flutter-app#297	Log book store links with parameters — Testing — ourbooks/flutter-app#292
Show stats of read books by week, month and year — ourbooks/flutter-app#294	Do not let unverified books without ISBN to be searched — To Do — ourbooks/api#156		Error 500 when revoking token in GAE — Bug, Testing — ourbooks/api#121
Define when a book was read (different date from when the book was marked as read) — ourbooks/api#152	Return unique followers and followees — To Do — ourbooks/api#154		
Recover the last description of a created offer (to speed up offer creation) — ourbooks/flutter-app#274	Attach books to replies — To Do — ourbooks/flutter-app#298		
Tell whether the metadata of a book is verified and last edited by someone or autogenerated — ourbooks/flutter-app#284	Do not include "rate this book" when sharing a post — To Do — ourbooks/flutter-app#192		
Export books — ourbooks/flutter-app#281	Increase max quote length — To Do — ourbooks/api#141		
Avoid loading indicator when refetching the opinion stats in the user profile — ourbooks/flutter-app#277			
Add simple recommender system			

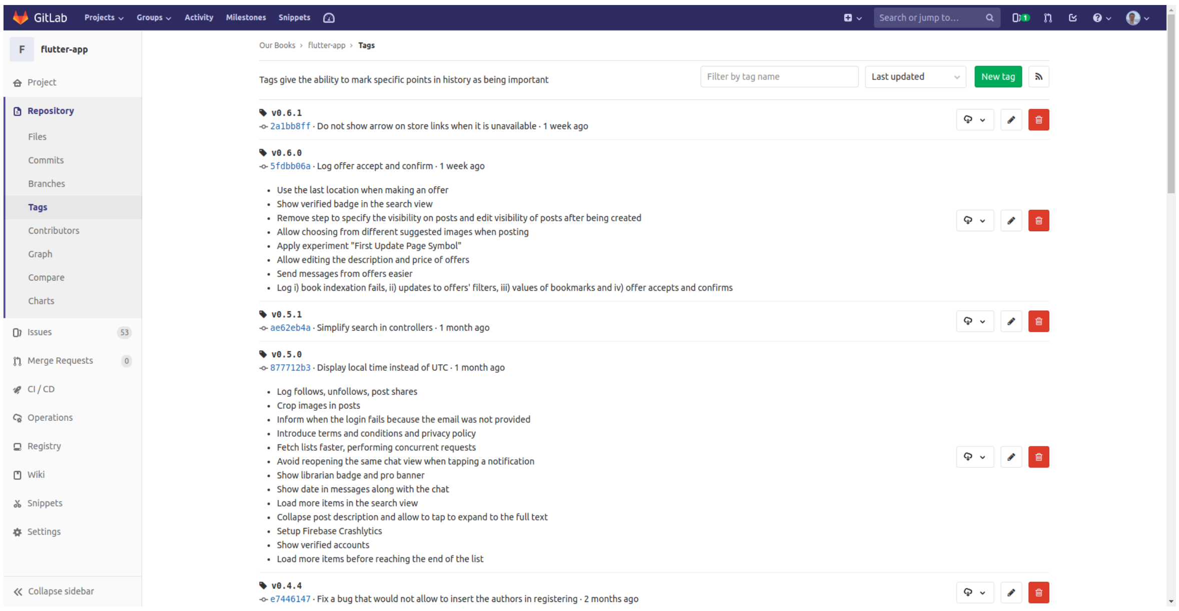
Figure 4.7: The boards on GitLab, on 21 May 2019, with several lists: open (not scheduled yet), to do (ready to be started soon), doing (currently being addressed), testing, stale (considering to be closed without being addressed) and done.


Our Books › Milestones

Open 7 Closed 3 All 10

Milestone	Dates	Labels	Issues	Progress
v0.5 - Group Milestone	Mar 13, 2019–Apr 16, 2019	Expired, Our Books	18 Issues · 0 Merge Requests	100% complete
v0.6 - Group Milestone	expired on May 14, 2019	Expired, Our Books	17 Issues · 0 Merge Requests	100% complete
v0.7 - Group Milestone	expires on Jun 11, 2019	Our Books	10 Issues · 0 Merge Requests	10% complete
v0.8 - Group Milestone	expires on Jul 2, 2019	Our Books	2 Issues · 0 Merge Requests	0% complete
v0.9 - Group Milestone	expires on Aug 6, 2019	Our Books	1 Issue · 0 Merge Requests	0% complete
v1.0 - Group Milestone	expires on Oct 1, 2019	Our Books	23 Issues · 0 Merge Requests	0% complete
Goals - Group Milestone	expires on Jan 1, 2030	Our Books	1 Issue · 0 Merge Requests	0% complete

Figure 4.8: The progress of the defined milestones on GitLab, on 21 May 2019. Version 0.5 and 0.6 were completed, 0.7 was in progress and the remaining ones up to 1.0 were scheduled.



Tags give the ability to mark specific points in history as being important

v0.6.1
2a1bb8ff · Do not show arrow on store links when it is unavailable · 1 week ago

v0.6.0
5fdbb06a · Log offer accept and confirm · 1 week ago

- Use the last location when making an offer
- Show verified badge in the search view
- Remove step to specify the visibility on posts and edit visibility of posts after being created
- Allow choosing from different suggested images when posting
- Apply experiment "First Update Page Symbol"
- Allow editing the description and price of offers
- Send messages from offers easier
- Log i) book indexation fails, ii) updates to offers' filters, iii) values of bookmarks and iv) offer accepts and confirms

v0.5.1
ae62eb4a · Simplify search in controllers · 1 month ago

v0.5.0
877712b3 · Display local time instead of UTC · 1 month ago

- Log follows, unfollows, post shares
- Crop images in posts
- Inform when the login fails because the email was not provided
- Introduce terms and conditions and privacy policy
- Fetch lists faster, performing concurrent requests
- Avoid reopening the same chat view when tapping a notification
- Show librarian badge and pro banner
- Show date in messages along with the chat
- Load more items in the search view
- Collapse post description and allow to tap to expand to the full text
- Setup Firebase Crashlytics
- Show verified accounts
- Load more items before reaching the end of the list

v0.4.4
e7446147 · Fix a bug that would not allow to insert the authors in registering · 2 months ago

Figure 4.9: The history of all the versions of the app on its GitLab repository. The minor versions (X.Y.0) are documented with a summary of the changes.

may be noticed, there was a slight difference in the weighting of the app and the backend. For the app, the mindset was to build something fresher and bolder when choosing technologies and approaches. For the backend, it happened the opposite, with a more moderated vision, biased to older and proven technologies. The rationale behind this was that the app is just the outermost layer, and it can evolve with a different flexibility, without affecting the remaining implementation. The backend is the core of the solution, so the way it is modeled and any change to it may have a significant impact in the whole system. The app can be changed at any time, even in the future when there are other forms of interaction, maybe on other platforms different from smartphones, while the backend will now undergo the inertia of the currently modeled solution, and a weak solution could spoil everything.

CHAPTER 5

Deployment

"The whole is greater than the sum of its parts." – Aristotle

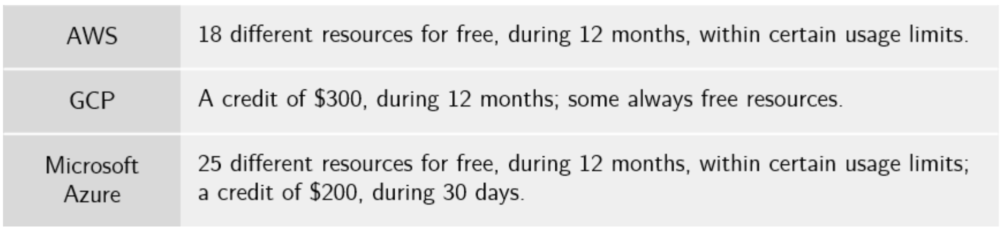
When the app got ready for the first public contact with the users, it was necessary to prepare its deployment. Not a basic setup for demo purposes somewhere in a VM in a server, but a robust deployment that could be fairly ready for the future. This entails taking into account issues such as scalability, monitoring and automation. In this Chapter we explore the choices and the tools used to achieve the desired system architecture.

5.1 Choosing a cloud provider

One of the important decisions to take was where the backend would be deployed. In Section 2.2.5 we already talked about the cloud phenomenon and its importance. The decision for this project was to follow this path, choosing between the three main cloud providers: AWS, GCP and Microsoft Azure. These three platforms provide resources for the different layers of services (IaaS and PaaS), also having an integrated ecosystem that is prepared to answer common needs, such as for scalability, monitoring and automation.

In this regard, we are immediately faced with a question which may be decisive in the choice of the provider. Taking monitoring as an example, each one offers its proprietary technologies. AWS has Amazon CloudWatch, GCP has Google Stackdriver and Microsoft Azure has Azure Monitor. When opting by a PaaS, it is useful to rely on the provider's solutions to maximize the integration. But this integration comes associated with a vendor lock-in. In this example, we get locked to their monitoring tools, and the same happens in several other aspects across the whole deployment.

Many other aspects matter when choosing a cloud provider, such as Service Level Agreements (SLAs), data privacy policies, client support, and pricing. This last point was particularly important for this project, so the investment to produce the app could be minimized. These three providers have starter plans that are free, so developers can use them to better



AWS	18 different resources for free, during 12 months, within certain usage limits.
GCP	A credit of $300, during 12 months; some always free resources.
Microsoft Azure	25 different resources for free, during 12 months, within certain usage limits; a credit of $200, during 30 days.

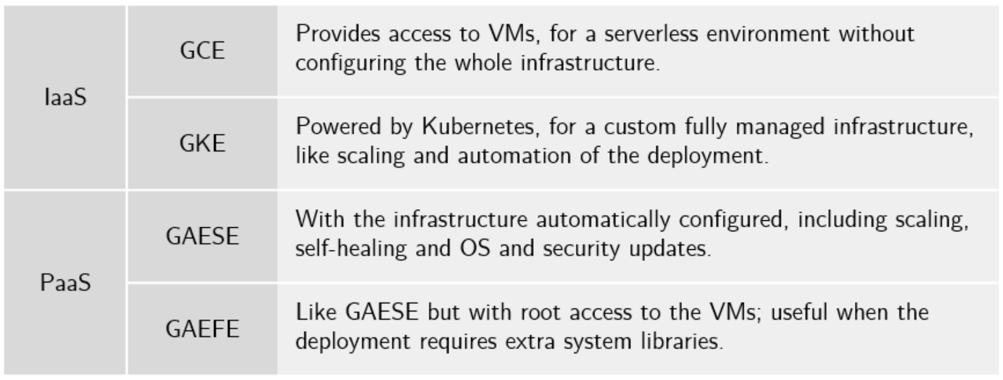
Figure 5.1: The free starter plans of the considered cloud providers.

explore the existent tools. This is also useful for small projects or projects that are now starting, taking advantage of a free deployment for a period of time. AWS offers 18 different resources for free, during 12 months, within certain usage limits. GCP does something similar, offering a credit of $300 that can be used during 12 months. With that credit, developers can choose more freely how to access the resources, because the limit is global (the credit itself) and not by product or service, like with Amazon. Also, GCP has a tier of resources that are always free, even after the first 12 months. Microsoft Azure offers 25 different resources for free, during 12 months, also within certain usage limits, and a credit of $200, that is valid for 30 days. Fig. 5.1 summarizes the three different plans. Comparing them, GCP was what seemed to present the best offer, both for the first 12 months and for being the only one that provides always free resources beyond that period.

5.2 Deployment on GCP

5.2.1 Server

GCP has four main hosting options to deploy Django, that are documented in [57]. For an IaaS, there is Google Compute Engine (GCE) and Google Kubernetes Engine (GKE). GCE is a service to using Virtual Machines (VMs). This is useful for who needs a serverless environment without configuring the whole infrastructure. For a fully managed infrastructure, there is GKE, powered by Kubernetes, that, as a container orchestration solution, gives the flexibility of managing things like scaling and automation of the deployment. As we saw in Section 2.2.5, an IaaS is powerful but requires more efforts on the configuration than a PaaS, which was the choice for this project. For a PaaS, GCP has Google App Engine Flexible Environment (GAEFE) and Google App Engine Standard Environment (GAESE). These two environments let the deployment process be focused on the application itself (Django, in our case), while handling transparently all the remaining management of the infrastructure. This includes automatic scaling, self-healing (checking if instances are working fine and recovering from any problems, for example, replacing an instance by a new one) and applying OS or security updates. The difference between GAEFE and GAESE is that GAEFE gives root access to the instances, which are VMs from GCE, while GAESE does not. This means that GAEFE can be configured with custom Docker runtimes, while GAESE has a static default environment, with minimal configuration. GAESE can be used if the application to deploy requires no extra system libraries or Python libraries, otherwise GAEFE is the way to go.



IaaS	GCE	Provides access to VMs, for a serverless environment without configuring the whole infrastructure.
	GKE	Powered by Kubernetes, for a custom fully managed infrastructure, like scaling and automation of the deployment.
PaaS	GAESE	With the infrastructure automatically configured, including scaling, self-healing and OS and security updates.
	GAEFE	Like GAESE but with root access to the VMs; useful when the deployment requires extra system libraries.

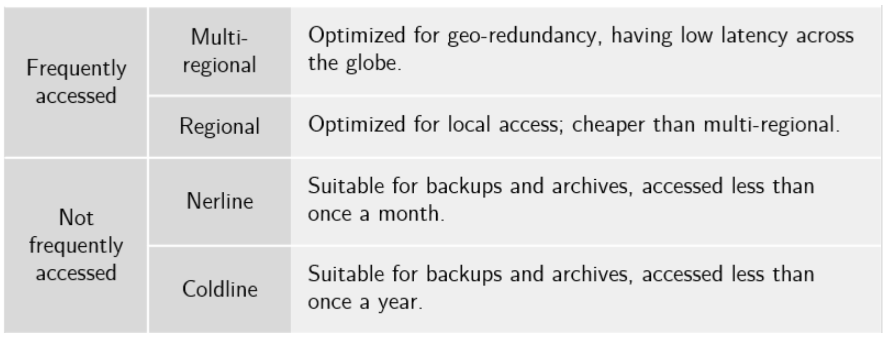
Figure 5.2: The four main hosting options to deploy Django on GCP. Source: [57]

Fig. 5.2 summarizes the four different options. In this project it was possible to keep the source code close to the default libraries, so it enabled the use of GAESE.

Beyond the minimum configuration, GAESE was also preferable over GAEFE because of the pricing. GCP offers the first 28 instance hours per day. Each instance hour costs \$0,05, corresponding to a daily save of \$1,40. For example, if we were running on average 2 instances a day, the first 14 hours would be for free. GAEFE is not included in the free plan. Another useful feature is that GAEFE allows to scale to 0 instances when there is no traffic for several minutes (the way this time is managed by the engine is not clearly documented [58]), while GAESE requires that at least 1 instance is running all the time. This is not problematic for applications that are regularly accessed, but for projects that have only a few users by day, there is a visible difference in the costs. Also, for the deployment of this project, it was followed a microservices approach. For example, there is a microservice that handles the search and indexation of new books, also getting the information of prices in online stores. This service in not used as often as the core of the application, so being able to scale to 0 instances (and consequently spending and paying nothing for these resources) is a plus. Because of that, the choice for this project was GAESE.

5.2.2 Databases

Having the server defined, it was necessary to connect it to the databases. Because of the microservices architecture, each service has its own database. GCP supports MySQL and PostgreSQL. As decided in Section 4.2.3, it was used PostgreSQL. It can be created SQL instances, with different specifications, such as CPU, memory and storage. For each instance, it can after be configured any desired databases. This means that the service is accounted for the instances, and not the databases inside them. The platform allows to manage the connectivity (internal or external), users, automatic backups and scalability (e.g., creating read replicas). For this project it was necessary to keep more than a database, that ideally should be independent from each other. The option was to create all the databases inside the same instance, to save costs. Even so, the databases were managed independently, the only difference was being inside the same instance, and competing for the same resources, which is not the desired when scaling. The plan is that, in the future, these databases are migrated to



Frequently accessed	Multi-regional	Optimized for geo-redundancy, having low latency across the globe.
	Regional	Optimized for local access; cheaper than multi-regional.
Not frequently accessed	Nerline	Suitable for backups and archives, accessed less than once a month.
	Coldline	Suitable for backups and archives, accessed less than once a year.

Figure 5.3: The four storage solutions in GCP. Source: [59]

different instances when needing to scale.

5.2.3 Storage

For the storage, in this project essentially images, GCP has four solutions: multi-regional, regional, nearline and coldline [59]. Multi-regional is optimized for geo-redundancy, having low latency across all the globe. Regional is optimized for local access, and it is cheaper than multi-regional because of that. These two solutions are targeted for storage that is frequently accessed. For storage that is not frequently accessed, such as backups or archives, there is nearline and coldline, for data accessed less than once a month and less than once a year, respectively. These two solutions are the less expensive. Fig. 5.3 summarizes the four different solutions. The choice was for multi-regional, because of the redundancy and latency. Despite being the most expensive, storage is not significant in the overall project budget. For example, multi-regional costs \$0,026 per GB/month [60], while the cheapest option of an SQL instance (with a shared CPU of 0,6 GB RAM) costs \$7,67 per month [61].

5.2.4 Logging

Stackdriver was used to gather all the server logs. Stackdriver is a proprietary solution from Google that is automatically associated when deploying on GAESE. Still, it was necessary to configure Django to communicate the logs to Stackdriver, because the generic ones captured from the outside by GAESE where not informative enough, mainly internal server errors (500), unauthorized requests (401) and forbidden requests (403), and other internal logs that were previously written in the development phase. Stackdriver has a free quota of 50 GB per month, retained for 30 days, included in the free plan. Despite being a generous quota, it is important to keep logs compact and useful, and not logging everything just because we can. Too many logs introduces noise and can be costly when scaling the application. A single user makes several requests while using the app, that uses the API, each request resulting in a log, that is textual and verbose, so it is not so hard to achieve the 50 GB when multiplying it for some hundreds of users.

5.2.5 Network

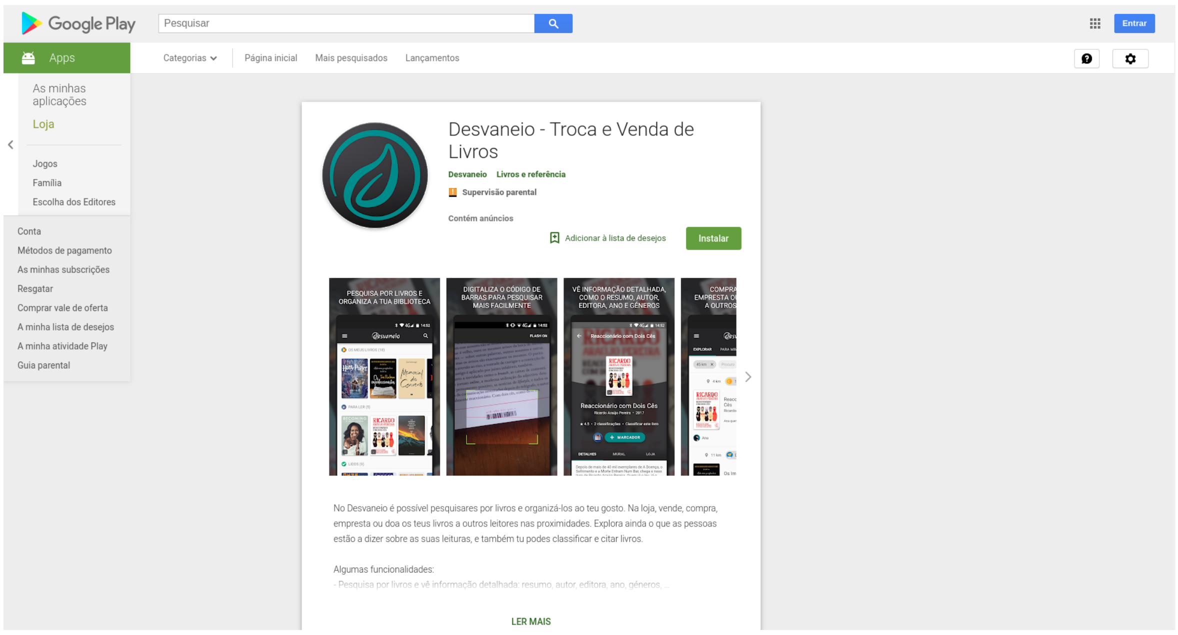
All of these components could communicate between them with different network configurations, that affect the traffic and performance, and also the pricing. For example, communications between the databases and the server side are free if using the internal network and in the same continent, and paid otherwise. This forces the databases to be deployed in the same continent as the servers to take advantage of that. But while database instances are available on the USA, Europe, Asia and Australia, the free version of GAESE is only available on the USA. So, the option was to make the deployment on the USA, even if it slight influences the latency in Portugal, where this app will be primarily used in its early days. GAESE also does not provide a way to access the internal network, so it forces the traffic to the database to be external, and consequently charged. Concerning storage, the traffic is mediated by GAESE when uploading, also using the external network, but it is charged by *upload* operation, not the traffic itself. When retrieving the objects from the storage, it happens the same, with the *retrieve* operations being charged independently from the traffic.

5.2.6 Security

GAESE has HTTPS out-of-the-box, so all the communications are secure. The remaining deployment is backed by GCP, which means that all the issues related to the OS like any security updates are automatically handled. From the application side, it is configured to be prepared for brute force and denial-of-service attacks.

5.3 Publishing the app

The beta version of the app was published in the Google Play Store (see Fig. 5.4), as it will be further detailed in Chapter 6. For that, it was necessary to create a developer account, which costs a lifelong fee of $25. The app was developed both for Android and iOS (see Section 4.1), but App Store requires to pay $99 per year, which was not compatible with the effort of saving costs. Hence, the iOS version will be introduced in a later stage of the roadmap, when the app is able to return the initial investment, as we will see in Chapter 7. We will now talk briefly about the tools on the Play Console (from where all the process of publishing the app is managed) and what was used from it in the context of this project.



Desvaneio - Troca e Venda de Livros

No Desvaneio é possível pesquisares por livros e organizá-los ao teu gosto. Na loja, vende, compra, empresta ou doa os teus livros a outros leitores nas proximidades. Explora ainda o que as pessoas estão a dizer sobre as suas leituras, e também tu podes classificar e citar livros.

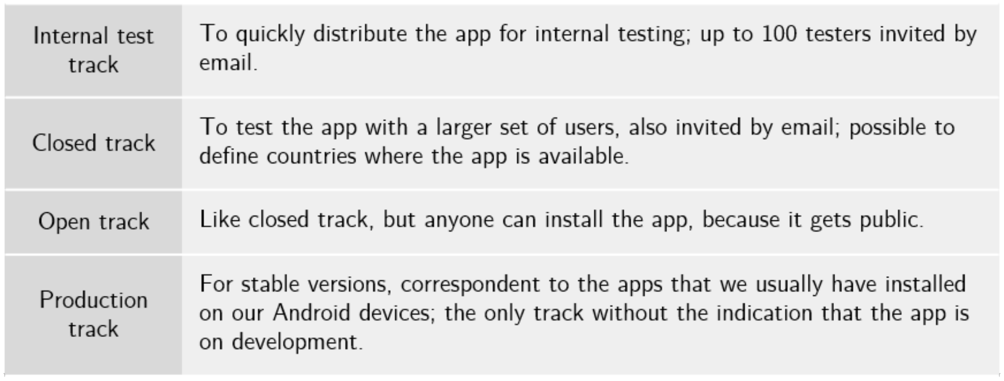
Algumas funcionalidades:
- Pesquisa por livros e vê informação detalhada: resumo, autor, editora, ano, géneros, ...

Figure 5.4: The app, available in the Google Play Store. It can be found by searching *Desvaneio* or directly accessed here on https://play.google.com/store/apps/details?id=com.desvaneio.flutterapp.

5.3.1 Release management

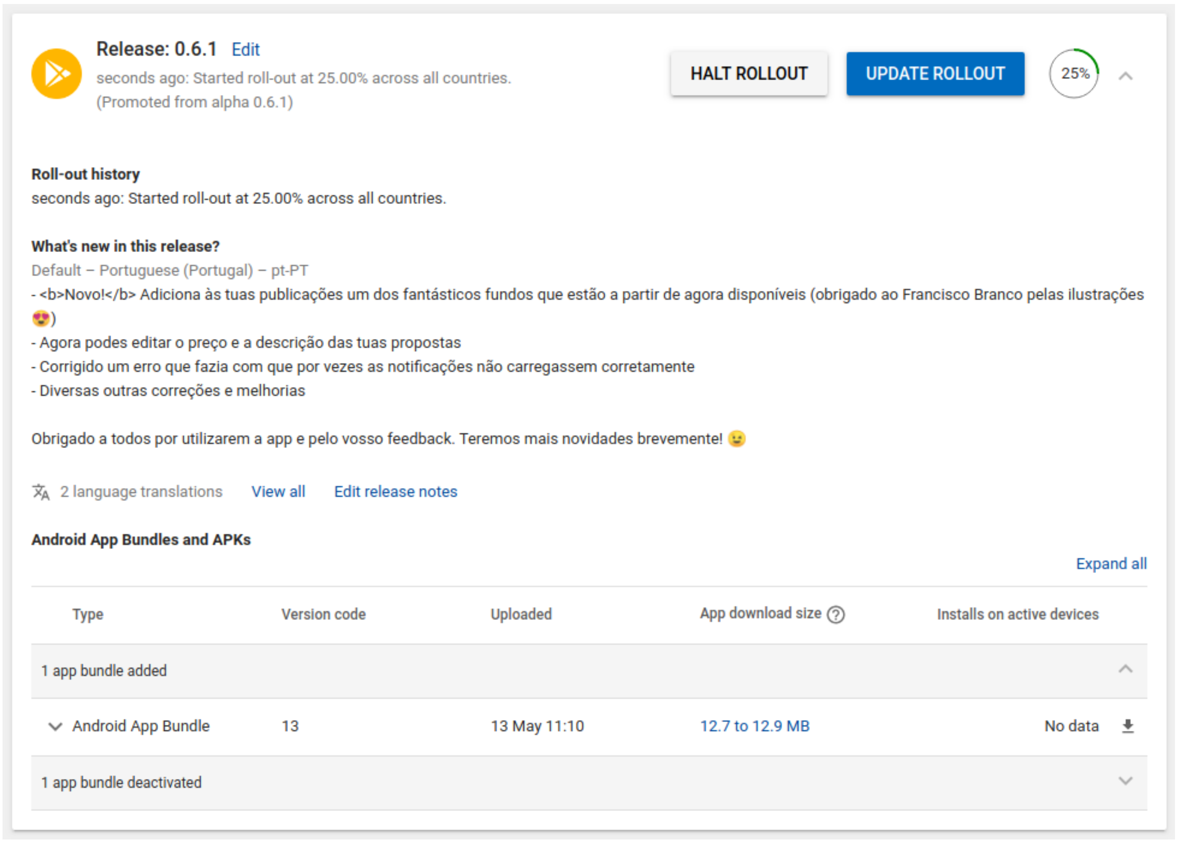
The Play Console allows to manage different layers of releases. These different layers allow to easily stage each new version, with different groups of users, generally from a smaller group to everyone. This allows to progressively release new versions, while being able to detect any problems early. Starting from the narrowest to the least restricted they are: internal test track, closed track, open track and production track. Internal test tracks allow to quickly distribute the app for internal testing, with up to 100 testers, that are invited by email. Closed track, also called alpha, allows to test the app with a larger set of users, also invited by email. In this track and the remaining ones (open track and production track) releases are not instantaneous, and might take some time to be available to everyone, generally up to 48 hours. Open track, also called beta, is like closed track, but anyone can install the app. In these three tracks, users are warned that these versions are under development, and that it can be unstable. Finally, the production track is where the last stable version of the app resides. This is the track correspondent to the apps that we usually have installed on our Android devices. Only the versions on this track can be publicly rated by the users. The closed track, open track and production track also allow to define the countries where the app is available. Fig. 5.5 summarizes the different tracks.

In addition, all of the tracks allow to make progressive rollouts. Progressive rollouts are a feature that make it possible to define a percentage of the users that receive an update (see Fig. 5.6). These tools extend the possibilities of configuration to control how the new versions are released. Instead of exposing all of the users to a version that can have issues, it is better to progressively deploy it. These issues can be users disliking a new feature or crashes. With



Internal test track	To quickly distribute the app for internal testing; up to 100 testers invited by email.
Closed track	To test the app with a larger set of users, also invited by email; possible to define countries where the app is available.
Open track	Like closed track, but anyone can install the app, because it gets public.
Production track	For stable versions, correspondent to the apps that we usually have installed on our Android devices; the only track without the indication that the app is on development.

Figure 5.5: The available tracks in the Play Console.



Release: 0.6.1 Edit
seconds ago: Started roll-out at 25.00% across all countries.
(Promoted from alpha 0.6.1)

Roll-out history
seconds ago: Started roll-out at 25.00% across all countries.

What's new in this release?
Default – Portuguese (Portugal) – pt-PT
- <b>Novo!</b> Adiciona às tuas publicações um dos fantásticos fundos que estão a partir de agora disponíveis (obrigado ao Francisco Branco pelas ilustrações 😍)
- Agora podes editar o preço e a descrição das tuas propostas
- Corrigido um erro que fazia com que por vezes as notificações não carregassem corretamente
- Diversas outras correções e melhorias

Obrigado a todos por utilizarem a app e pelo vosso feedback. Teremos mais novidades brevemente! 😉

2 language translations View all Edit release notes

Android App Bundles and APKs

Expand all

Type	Version code	Uploaded	App download size	Installs on active devices
1 app bundle added				
Android App Bundle	13	13 May 11:10	12.7 to 12.9 MB	No data
1 app bundle deactivated				

Figure 5.6: The overview of the release of version 0.6.1 in the Play Console, on 15 May 2019. In the upper left corner it can be updated the rollout progress, which was 25% at that moment.

the enormous diversity of Android devices and versions out there, apps are very prone to manifest issues when being deployed, even after being carefully tested.

In this project all of the tracks have been used for each release, with exception to the production track. At the date of this document, the app is in Beta. Also, the app was made available only in Portugal. This allowed to test and improve it in a more controlled environment, before releasing the version 1.0, that will be described in Section 7.1.3.

5.3.2 Other tools

The Play Console has other tools that help developers to manage the deployment of the app. It produces reports about metrics like crashes or the average ratings, compared against the app versions, Android version, country, users' language and other factors that might influence the performance of these indicators. This is very important to better understand the feedback from the users, which is decisive in the dimension of the user experience.

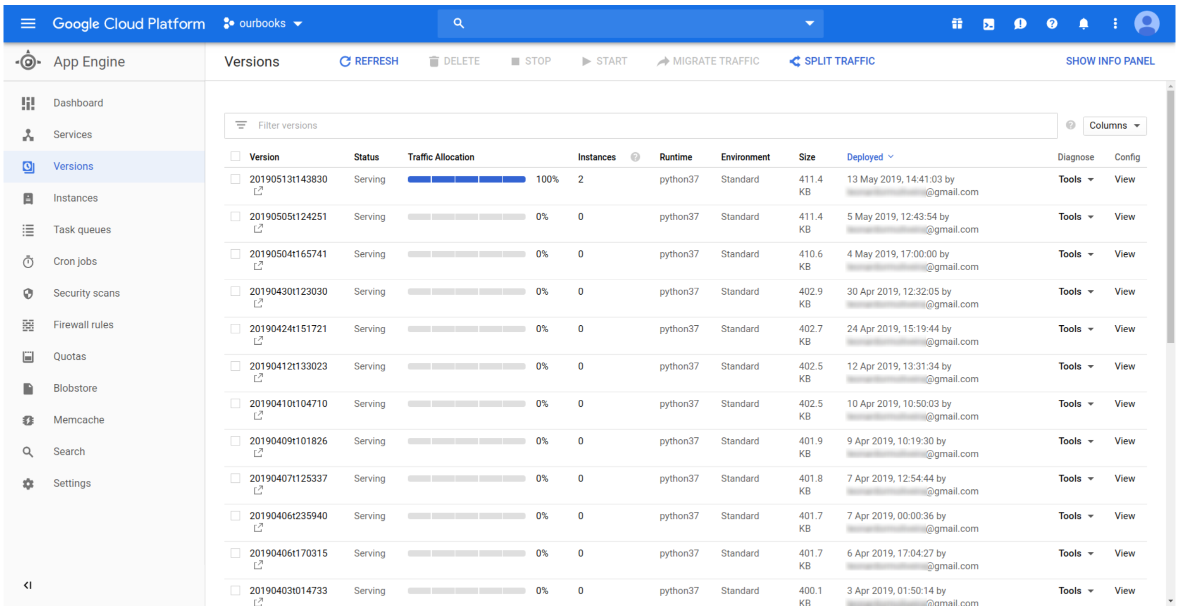
5.4 Scalability

The two main points in the application deployment that need special attention to make it ready to scale are the server and databases. As we saw above, GAESE scales dynamically. The automatic scaling can be controlled horizontally, defining a minimum or maximum number of instances, and vertically, changing the classes of the instances, that can go from 128 MB RAM, 600MHz to 1024 MB RAM, 4,8GHz [62]. About the databases, their instances can be scaled vertically, also with the possibility of creating read replicas.

5.5 Automation

GCP has gcloud, a command-line tool to speed-up some common tasks, allowing to automate all the processes in the platform, some of them not possible in the browser, on its web console. Apart from the initial configuration to setup all the deployment, each new version for the application to be deployed in the GAESE can be released with a single command: *gcloud app deploy*. So, the current process to do it is running a script that pulls the last version of the master branch, from the repository, and runs the deployment command. GCP automatically handles the upload of the new version and splits the traffic to it when it gets ready. The versions can later be managed in the dashboard. If necessary, changes can be rolled back to previous versions (see Fig. 5.7). This script also has other intermediate steps that use some existent Django command line tools, like checking and applying any database migration or updating the translation files.

About the app, Flutter has a guide to use fastline [45] in the official documentation, a widely used tool to release mobile apps. There is also Codemagic, a platform dedicated to Flutter for the same purposes. These tools allow to automate the build, run tests, assigning app versions and other repetitive tasks when releasing a new version of the app, all of that synchronized with its repository, that acts as a trigger on each update. At the moment the process of deployment is done manually, building each new version using a Flutter command and uploading the generated output to the Play Console. For future work, this process will be automated using one of these tools.



Version	Status	Traffic Allocation	Instances	Runtime	Environment	Size	Deployed	Diagnose	Config
20190513t143830	Serving	100%	2	python37	Standard	411.4 KB	13 May 2019, 14:41:03 by [redacted]@gmail.com	Tools	View
20190505t124251	Serving	0%	0	python37	Standard	411.4 KB	5 May 2019, 12:43:54 by [redacted]@gmail.com	Tools	View
20190504t165741	Serving	0%	0	python37	Standard	410.6 KB	4 May 2019, 17:00:00 by [redacted]@gmail.com	Tools	View
20190430t123030	Serving	0%	0	python37	Standard	402.9 KB	30 Apr 2019, 12:32:05 by [redacted]@gmail.com	Tools	View
20190424t151721	Serving	0%	0	python37	Standard	402.7 KB	24 Apr 2019, 15:19:44 by [redacted]@gmail.com	Tools	View
20190412t133023	Serving	0%	0	python37	Standard	402.5 KB	12 Apr 2019, 13:31:34 by [redacted]@gmail.com	Tools	View
20190410t104710	Serving	0%	0	python37	Standard	402.5 KB	10 Apr 2019, 10:50:03 by [redacted]@gmail.com	Tools	View
20190409t101826	Serving	0%	0	python37	Standard	401.9 KB	9 Apr 2019, 10:19:30 by [redacted]@gmail.com	Tools	View
20190407t125337	Serving	0%	0	python37	Standard	401.8 KB	7 Apr 2019, 12:54:44 by [redacted]@gmail.com	Tools	View
20190406t235940	Serving	0%	0	python37	Standard	401.7 KB	7 Apr 2019, 00:00:36 by [redacted]@gmail.com	Tools	View
20190406t170315	Serving	0%	0	python37	Standard	401.7 KB	6 Apr 2019, 17:04:27 by [redacted]@gmail.com	Tools	View
20190403t014733	Serving	0%	0	python37	Standard	400.1 KB	3 Apr 2019, 01:50:14 by [redacted]@gmail.com	Tools	View

Figure 5.7: The history of deployed versions of the server, on 17 May 2019. These deployments happened transparently to the app, which continued to use the different versions of the API without having to be updated.

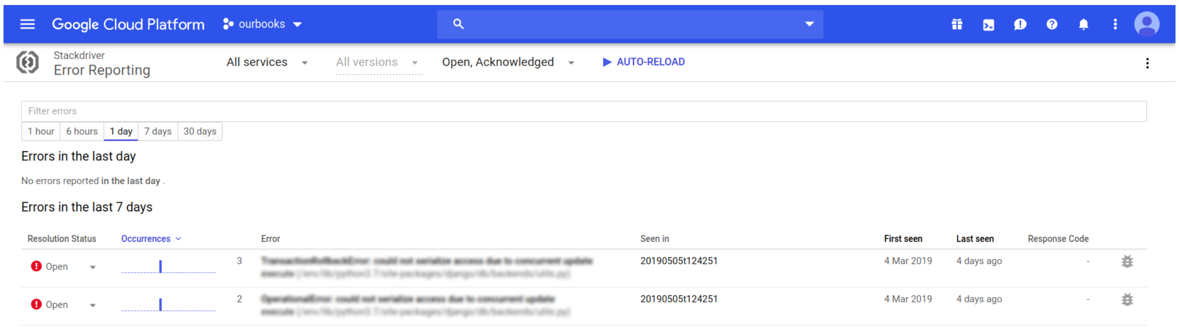
5.6 Monitoring

Lack of communication is a problem in many situations in our lives. Often something is wrong and we do not have feedback about that. Or the opposite, when we not aware of was is being done right. Monitoring can help with that, allowing to react in accordance to what is observed. In this project it was important to monitor the different components of the system, both in the app and in the backend. Also, it was important to complement that monitoring with the explicit feedback from the users, providing an easy way to report issues and suggestions.

5.6.1 Stackdriver

We already introduced Stackdriver in Section 5.2.4, when talking about logging. Logs are important to register the relevant events. With Stackdriver, we get logged all the interactions with the server, which allows to monitor them. They are filtered by priority (debug, info, warming, error and critical), and it can be created metrics to track them according to their parameters, allowing to get statistics of those observations over time. They can also be created alerts to report an unusual value specified in those metrics.

Stackdriver has also a tool called Error Reporting, that acts on the logs correspondent to errors, grouping them and counting their occurrences, so it it easier to understand the origin of the problems. They can be managed (first they are open, after they can be marked as acknowledged, resolved or muted) and can be linked to the correspondent issue being solved in the repository of the source code (see Fig. 5.8).



Errors in the last day

No errors reported **in the last day** .

Errors in the last 7 days

Resolution Status	Occurrences		Error	Seen in	First seen	Last seen	Response Code
Open		3	[redacted]	20190505t124251	4 Mar 2019	4 days ago	-
Open		2	[redacted]	20190505t124251	4 Mar 2019	4 days ago	-

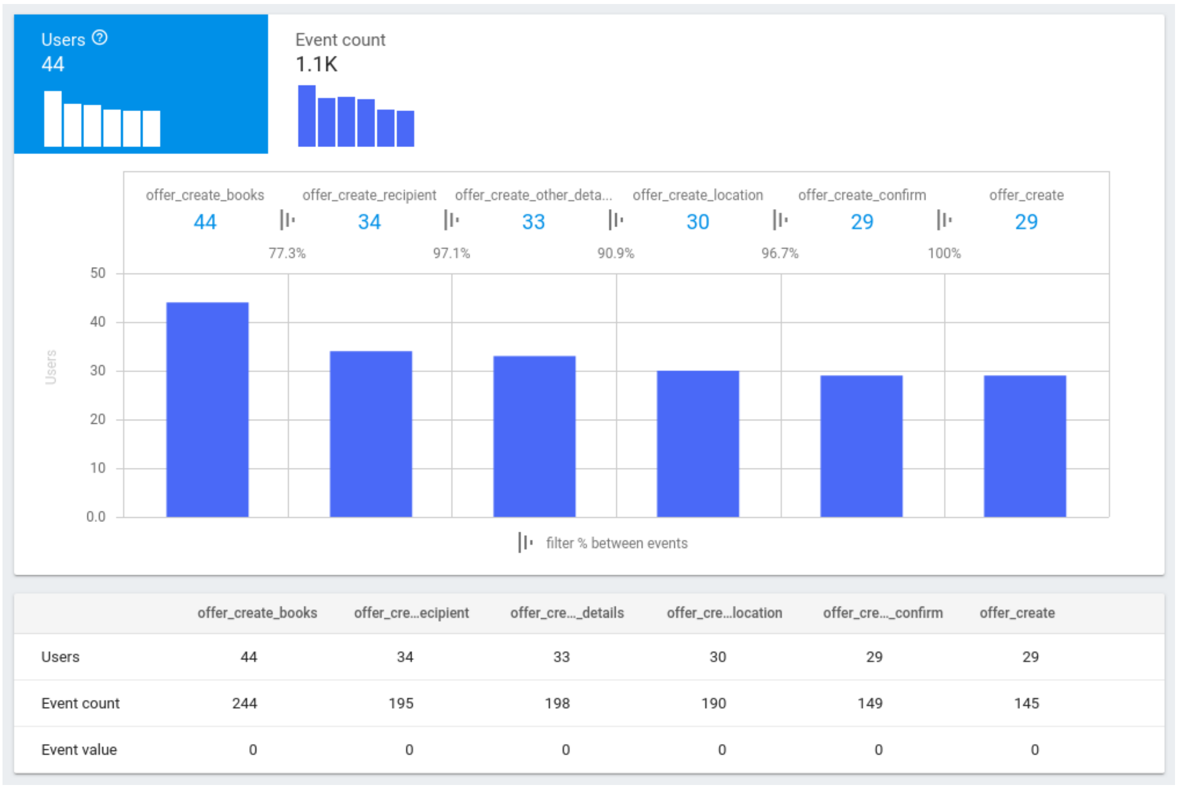
Figure 5.8: The Error Reporting dashboard, showing the open occurrences on 11 May 2019.

5.6.2 Firebase

Firebase is a platform developed by Google that supports apps in many aspects, such as data persistence, notifcations and monitoring. Firebase Analytics allows to monitor apps through a very complete web interface. Out-of-the-box it is possible to know the number of active users daily, weekly and monthly, the average time spent in the app, the number of users on the app at each moment, the distribution of the users by app version and country and other important metrics that help to understand the user engagement. In addiction, it is possible to log custom events. For example, in this app it has been logged when a book barcode is scanned, a bookmark is created or a user is followed. Later it is possible to understand how the users are taking advantage of the different features in the app. Of course, for privacy reasons, this information is always grouped and anonymous.

Firebase allows to create funnels, that are visualizations of the completion rate of a series of steps (events). For this app it has been used to monitor some processes, for example, the creation of an offer. This process involves i) selecting the book, ii) specifying the target, iii) adding the price and an optional description, iii) choosing the location and, finally, iv) confirming its publication. Along the whole process there can be a problem that make the users to abandon it. Also, it can be useful to filter users properties to understand the segments of the user base that complete the process at a higher rate. Fig. 5.9 shows the funnel of the offer creation. From one step to another, it shows the percentage of the users that continued. In this case he have a bigger drop from the first step to the last one. This might correspond to users that start the creation of an offer just by curiosity or mistake and abandon it immediately. If this drop was in the middle of the process there would be a problem to solve, and the effect of the changes could be observed filtering and comparing the statistics of the new app version and the previous one, for example.

Firebase also has Remote Config, a tool to create simple configurations through the definition of variables that are interpreted in the app at runtime, allowing to adjust its behavior without the need to release a new version. This was used in this app for several configurations, namely for the enabling of advertising, that is disabled by default (see Section 7.1.5). If at any moment there was a huge increase of the users on the app, increasing traffic and increasing the expenses from the resources used in the deployment when scaling, it would



	offer_create_books	offer_cre...ecipient	offer_cre..._details	offer_cre...location	offer_cre..._confirm	offer_create
Users	44	34	33	30	29	29
Event count	244	195	198	190	149	145
Event value	0	0	0	0	0	0

Figure 5.9: The funnel for the process of making an offer to exchange a book, showing the progress of 44 users that initiated it. From the left to the right, it shows the drops in each step, ending with 29 users. The plot is from 11 May 2019, for the last 28 days, and groups the actions of every user, which means that if someone created 10 offers, it only shows the progress as a single process. This can be useful to check if the users completed the process at least once. The dashboard also allows to show all the events, counting all the actions from the users, and allowing to check how the users performed in all the offers they created.

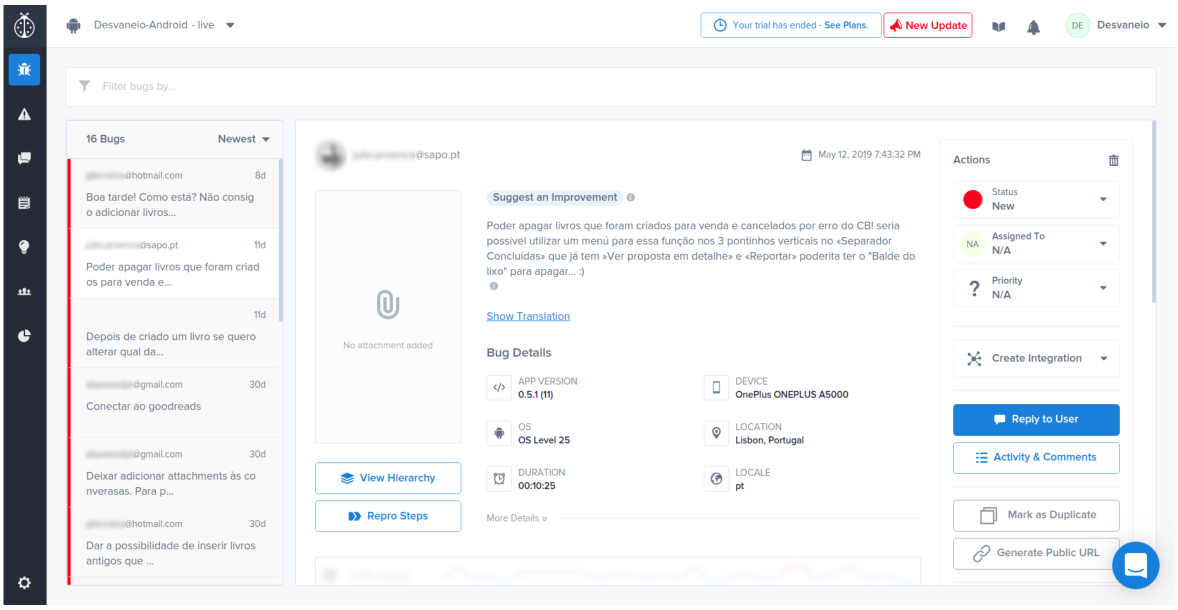
be possible to change this configuration in Remote Config and some ads would start being displayed.

There is also the need to monitor crashes. Play Console already tracks them, and Firebase also does it, but with a more contextualized view of the occurrences. With Flutter, it was also possible to use a plugin to report non-fatal bugs, that do not make the app to crash but interfere with its normal behavior. Detecting bugs is important, but quickly understanding how they are affecting the users engagement is even more informative, hence the importance of this integration with Firebase.

Firebase has much more potentialities. This an illustration of how it can help in monitoring. Later, in Section 6.5 we will see how it can be used for A/B testing.

5.6.3 Instabug

We have been talking about information that is implicitly collected. What about the users' explicit feedback? This is specially important in the initial phase of any app, where there will be many things to fix. Instabug is a tool that can be integrated with an app to allow users providing that feedback. They can simply shake the phone to report problems or send



[redacted]@hotmail.com 8d
Boa tarde! Como está? Não consig o adicionar livros...

[redacted]@sapo.pt 11d
Poder apagar livros que foram criad os para venda e...

11d
Depois de criado um livro se quero alterar qual da...

[redacted]@gmail.com 30d
Conectar ao goodreads

[redacted]@gmail.com 30d
Deixar adicionar attachments às co nverasas. Para p...

[redacted]@hotmail.com 30d
Dar a possibilidade de inserir livros antigos que ...

[redacted]@sapo.pt

May 12, 2019 7:43:32 PM

No attachment added

Suggest an Improvement

Poder apagar livros que foram criados para venda e cancelados por erro do CB! seria possivel utilizar um menú para essa função nos 3 pontinhos verticais no «Separador Concluídas» que já tem »Ver proposta em detalhe» e «Reportar» poderita ter o "Balde do lixo" para apagar... :)

Show Translation

Bug Details

APP VERSION: 0.5.1 (11)
DEVICE: OnePlus ONEPLUS A5000
OS: OS Level 25
LOCATION: Lisbon, Portugal
DURATION: 00:10:25
LOCALE: pt

Actions
Status: New
Assigned To: N/A
Priority: N/A

Figure 5.10: The Instabug dashboard, showing a suggestion of improvement among a list of other open issues.

suggestions, being possible to attach a screenshot or a voice note. The paid plans of Instabug have extra features, like a live chat and surveys. Bugs and suggestions are after managed in a dashboard, that complements the users' information with details of the device, like the app version or the Android version. Fig. 5.10 shows the dashboard for this app.

5.7 Architecture

The different components exposed in this chapter resulted in the deployment depicted in Fig. 5.11. The app relies on the API provided by the server, that is powered by GAESE, while also being connected to Firebase and Instabug. The server application, which provides all the core functionalities, relies for now in the book search service, that is in charge of searching and indexing new books and keeping them updated with the prices in different online stores. The architecture is ready to be expanded with other microservices that will be developed later. The application is connected to Cloud Storage, Cloud SQL and Stackdriver.

The current architecture allowed to make a consistent deployment, leveraging the necessary GCP tools to save time, being ready to scale and having the necessary tools for effective monitoring. For now GAESE is sufficient, scaling and supporting self-healing out-the-box. One day, if the project grows and the investment justifies it, it can be used a custom orchestration system, such as Kubernetes, for a more personalized resource management, that can be optimized, consequently reducing the operation costs.

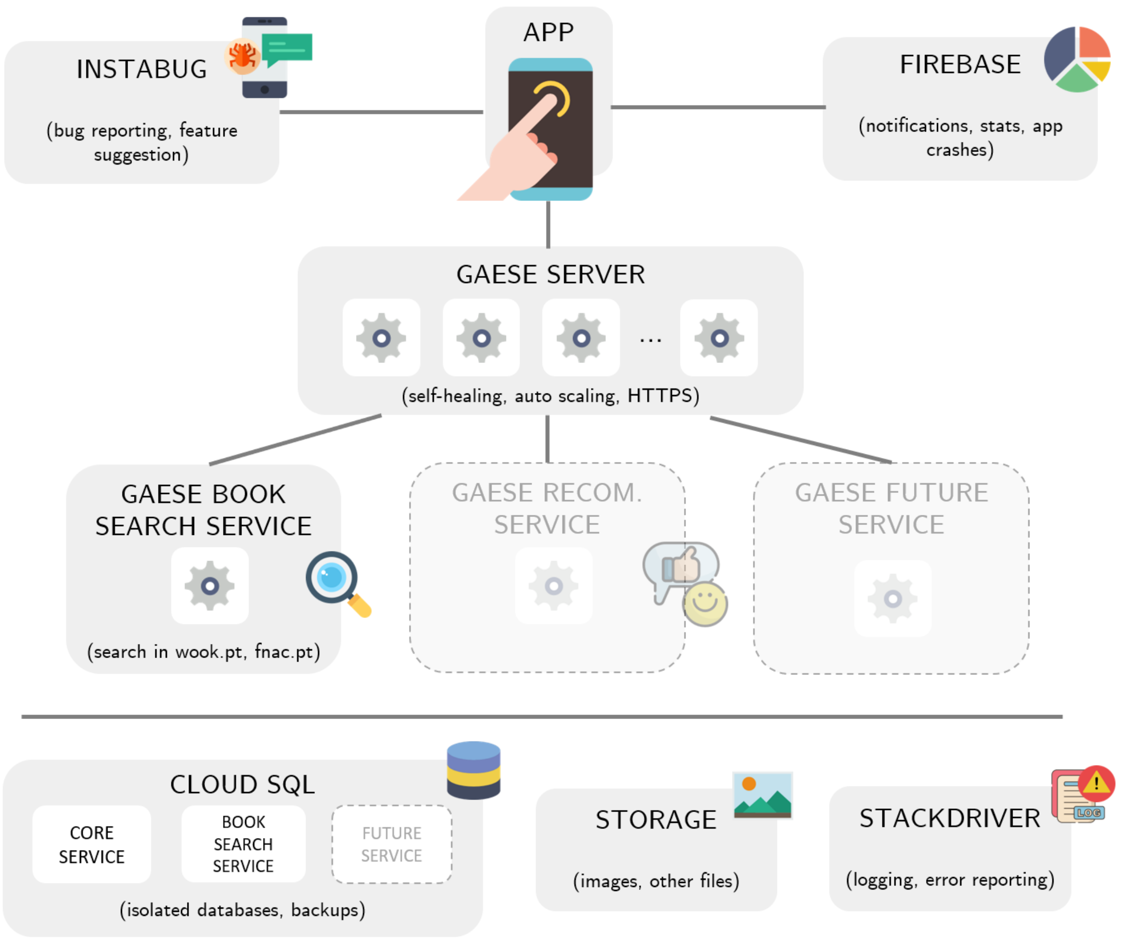

INSTABUG
(bug reporting, feature suggestion)
APP
FIREBASE
(notifications, stats, app crashes)
GAESE SERVER
...
(self-healing, auto scaling, HTTPS)
GAESE BOOK SEARCH SERVICE
(search in wook.pt, fnac.pt)
GAESE RECOM. SERVICE
GAESE FUTURE SERVICE
CLOUD SQL
CORE SERVICE
BOOK SEARCH SERVICE
FUTURE SERVICE
(isolated databases, backups)
STORAGE
(images, other files)
STACKDRIVER
(logging, error reporting)
LOG

Figure 5.11: The overall deployment. The dashed components are not deployed yet.

CHAPTER 6

Evaluation of the MVP in Production

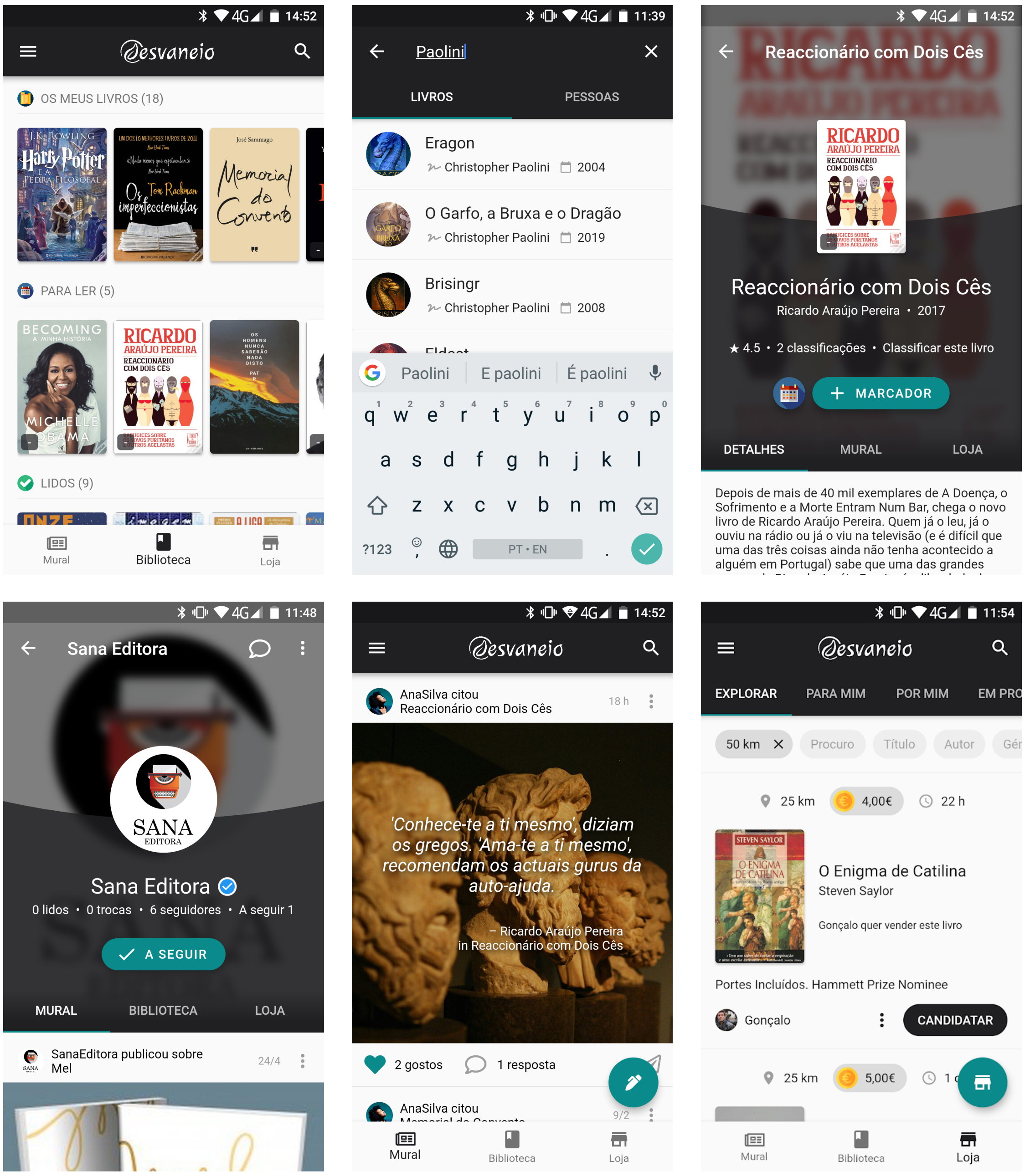
"Na cama que farás, nela te deitarás." – Portuguese proverb

When the deployment got ready, it was finally time to launch the first public version of the app [1]. The beta was released on 18 March 2019. With the first people available to use the app, it was possible to enter a new phase of design and development, with the feedback of a product being tested in real life. This Chapter reports the main moments since the app went into production.

6.1 Some screenshots

Before continuing analysing the MVP, some screenshots from the app in production might help creating a more complete mental model of it. Fig. 6.1 shows some representative views.

[1]The app can be found by searching *Desvaneio* on the Play Store, or directly accessed here: https://play.google.com/store/apps/details?id=com.desvaneio.flutterapp



Depois de mais de 40 mil exemplares de A Doença, o Sofrimento e a Morte Entram Num Bar, chega o novo livro de Ricardo Araújo Pereira. Quem já o leu, já o ouviu na rádio ou já o viu na televisão (e é difícil que uma das três coisas ainda não tenha acontecido a alguém em Portugal) sabe que uma das grandes

'Conhece-te a ti mesmo', diziam os gregos. 'Ama-te a ti mesmo', recomendam os actuais gurus da auto-ajuda.

– Ricardo Araújo Pereira
in Reaccionário com Dois Cês

Portes Incluídos. Hammett Prize Nominee

Figure 6.1: Some screenshots of the deployed MVP, from the top to the bottom and the left to the right: i) the users library, where they can organize their books; ii) the search view, in this case finding a book by its author; iii) the book profile, where the summary, genres and other details can be found; iv) a user profile, in this case from a publisher, that has a verified account, highlighted with a blue badge; v) the feed, showing a quote of a book; vi) the store, listing all the offers in a range of 50km.

6.2 The first users

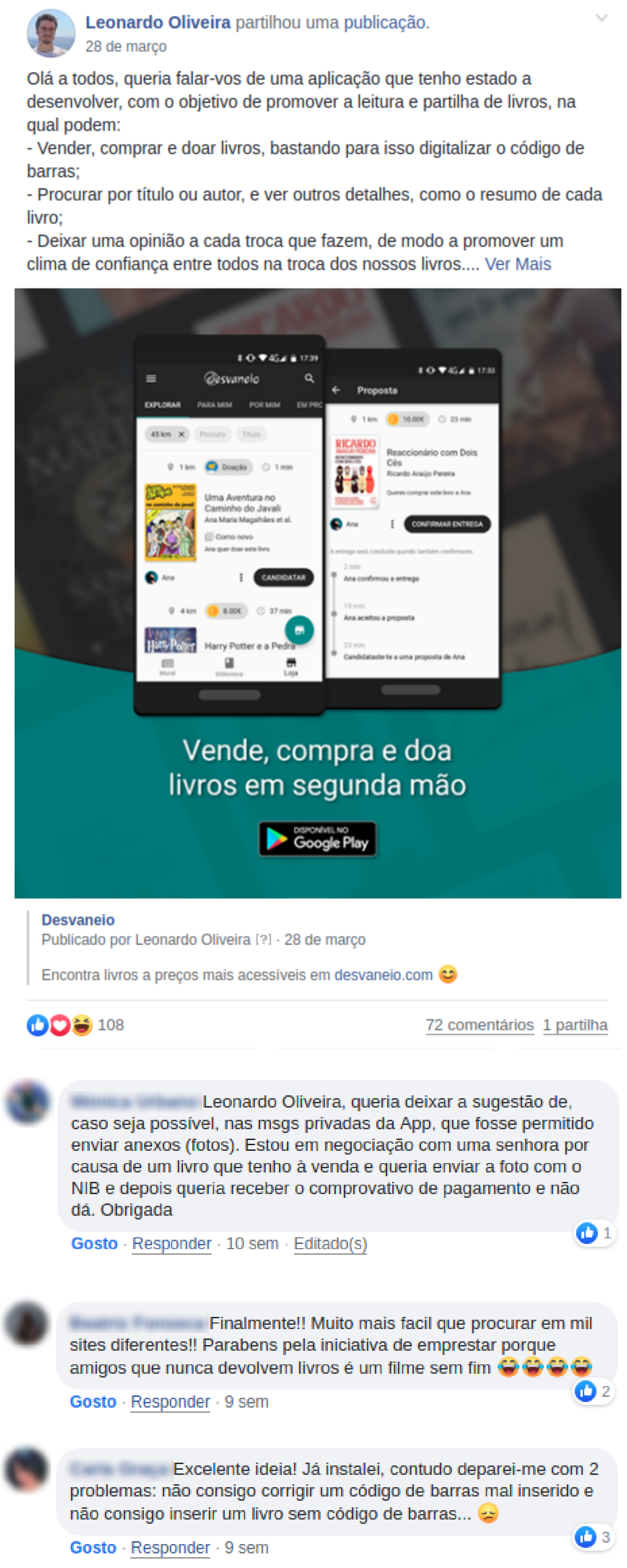
As highlighted in Section 5.3.1, the idea when releasing the app for the first time was to do it in a controlled environment. This way it would be possible to test and improve the app without any surprises that could affect the stability of the process. For this, beta was released only in the Google Play Store, limited to the portuguese store.

In the first days, a Facebook and Instagram pages were created[2] and the app was announced to the people that participated in the participatory design study (see Chapter 3), family and friends, and other people somehow involved in the project. This was important to confirm that everything was running smoothly, for example making sure the app would not crash when exposed to a greater variety of devices.

This group of users was appropriate, most of them being part of the target audience of the app. Still, it was not enough. It would take more people to give some momentum to the app, so the first posts would be published and the first books would be exchanged. So, it was necessary to reach more people within the target audience, while keeping a controlled environment. Then we went back to where it all begun, and from where part of the formulation of the initial idea for the store in the app emerged: the Facebook groups. A group of sale and exchange of second-hand books in Portugal, with around 5.000 members, was considered a good chance. There were several other groups with the same purposes, with more or less members, (there was one with around 20.000), but this was the one considered to have the best balance between its size and a good community. The administrators of this group were contacted and kindly allowed to publish a post announcing the app (see Fig 6.2). The post got popular, with more than 150 reactions (likes and comments), and resulting in almost 50 new installations in a single day.

After this moment, the user base continued to grow over the time, with other moments of dissemination through partnerships (see Section 6.4).

[2]The Facebook and Instagram pages can be accessed on desvaneio.com


Vende, compra e doa livros em segunda mão
DISPONÍVEL NO Google Play

Leonardo Oliveira partilhou uma publicação.
28 de março

Olá a todos, queria falar-vos de uma aplicação que tenho estado a desenvolver, com o objetivo de promover a leitura e partilha de livros, na qual podem:
- Vender, comprar e doar livros, bastando para isso digitalizar o código de barras;
- Procurar por título ou autor, e ver outros detalhes, como o resumo de cada livro;
- Deixar uma opinião a cada troca que fazem, de modo a promover um clima de confiança entre todos na troca dos nossos livros.... Ver Mais

Desvaneio
Publicado por Leonardo Oliveira [?] · 28 de março

Encontra livros a preços mais acessíveis em desvaneio.com 😊

108 · 72 comentários · 1 partilha

[redacted] Leonardo Oliveira, queria deixar a sugestão de, caso seja possível, nas msgs privadas da App, que fosse permitido enviar anexos (fotos). Estou em negociação com uma senhora por causa de um livro que tenho à venda e queria enviar a foto com o NIB e depois queria receber o comprovativo de pagamento e não dá. Obrigada
Gosto · Responder · 10 sem · Editado(s) 👍 1

[redacted] Finalmente!! Muito mais facil que procurar em mil sites diferentes!! Parabens pela iniciativa de emprestar porque amigos que nunca devolvem livros é um filme sem fim 😂😂😂😂
Gosto · Responder · 9 sem 👍 2

[redacted] Excelente ideia! Já instalei, contudo deparei-me com 2 problemas: não consigo corrigir um código de barras mal inserido e não consigo inserir um livro sem código de barras... 😔
Gosto · Responder · 9 sem 👍 3

Figure 6.2: The post announcing the app on the Facebook group for sale and exchange of second-hand books in Portugal. At the bottom there are comments from some of the users with their feedback. The screenshots were taken on 7 June 2019.

6.3 Feedback and changes

The first users were crucial for the birth of the app. They reported several problems that were not detected in the design and development stages. For that, they had several ways to communicate. The app had a menu to provide feedback, powered by Instabug (see Section 5.6.3), that could also be easily used by shaking the device. Although this was the main channel of communication, people also did it using the chat in the app (talking directly to the developer), and the Facebook and Instagram pages. Until 6 June 2019, 27 people reported more than 97 bugs, improvements and new features.

Most of the suggested improvements were caused by problems of who was actually using the app with a real need, which was not so easy to be reproduced at the design stage. For example, a user got unsatisfied because they did not have a way to edit the description of an offer after creating it. The solution for that was cancelling it and creating a new one, which was not practical. Another example was a user that, after making a considerable number of offers, started to notice that some steps could be accelerated, like the definition of the location. Every time an offer was created the location had to be selected, while it would be easier if it kept the place from the last created offer, allowing to change it if necessary.

Most of the reported bugs were small details in the interface that would not interfere with the overall experience. Yet at first they seemed to have no end in sight. As the days went by, it was relieving to realize that the problems started to converge. From 18 March 2019, the day the app got published, to 5 June 2019, 120 changes were committed across all the repositories of the different components being developed, resulting in 19 new deployed versions in the backed and 4 new versions of the in the Play Store. Due to the release process of the backend, it allowed to easily deploy new changes transparently, without interfering with the use of the app. The same does not apply to the app, where each new version corresponds to a new download from the store. This could be annoying for users, and for this reason its updates were less frequent.

6.4 New use cases

The app brought together a wide range of users, with different visions and needs. Some users just wanted to manage their books, others were more interested in selling or buying. There were even some organizations that got registered: Ror de Livros, a bookstore that sells new and second-hand books, and Sana Editora and Editorial Divergência, two publishers. These organizations are partners of the project, and are currently collaborating in the definition of new use cases. For example, Ror de Livros wanted a way to be distinguished from the other sellers, because they are a store, with physical presence, that issues invoices. Users have the rating system to help them deciding if they can trust a seller, but an organization should have a special distinction. Because of that, they now have a badge, that tells they are a verified seller, so buyers can see it when buying their books (the same happened later with Sana Editora and Editorial Divergência, as it is possible to see in Fig. 6.1).

About the publishers, it would be useful if they were able to manage the information of their books, allowing them to edit their records. In this regard, the necessary efforts have been made to allow this. Also, it was verified that some of the books automatically indexed from the web have inaccuracies. Because of this, there is new a role in the app: *librarian*. Users with this role are allowed to edit any book, while the remaining users can only edit books cataloged by themselves. This new role allows to resolve any faults in the books details while preventing any abusive use of the app.

6.5 A/B testing

Firebase has different tools to assist developers with their apps (see Section 5.6.2). Among these tools it supports A/B testing. For that, it can be created experiments that define the app behavior using server-side configuration parameters, making use of Firebase Remote Config (also discussed in Section 5.6.2). This remote configuration allows to run the experiments using different variables in the same version of the app, running at the same time with different configurations for different users, according to the group they belong to. This is very powerful, because it allows to have a common environment for all the users, which would not be possible if doing the experiment with different versions of the app or in different time periods. When creating an experiment, it can be defined the i) percentage of users that will be targeted by it, ii) the versions of the app that will run it, iii) optionally an activation event (any action that the users perform in the app that make them eligible for the experiment), iv) the primary metric to track, v) any additional metrics, for example to track the engagement, and finally v) the values of the parameters for the Remote Config that each group being tested will have, that will configure in runtime the different behaviors of their apps.

So far, it was possible to run an experiment for this app, that was primarily created to better understand how this tools works. It was noticed that only 12.5% (6/48) of the books that were marked as being read had the indication of the current reading page. There is the hypothesis of the option to do it is not sufficiently visible, so users do not easily discover it. Currently it is displayed a small indicator at the bottom of the cover of the books marked to be read or being read, in the users library, that allows to quickly update the page. However, when the page is not set, it is shown a dash, symbolizing that the value is not defined. Possibly this symbol is not being noticed, or is not self-descriptive, so it was decided to create an A/B test to compare it with a play symbol (see Fig. 6.3). It turns out that the results were inconclusive, because the users continued to not make regular use of this feature. Even so, this experiment allowed to test the A/B testing tool without a drastic change. In the future it will be created more experiments to try new features, that might have a greater impact in the overall user experience.

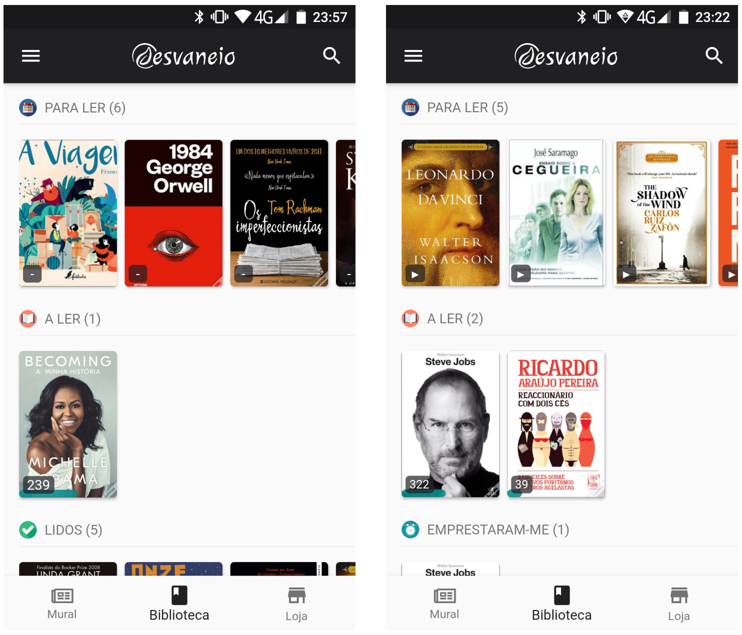


Figure 6.3: Two users with different versions of the A/B test. The books that are being read show the current page on the lower left corner of the cover. The books that still are not being read show a placeholder symbol so users can tap it when they start reading and update the page. The difference in the two experiments is that, on the left, the symbol is a dash and, on the right, the symbol is a play button.

6.6 Usage statistics

Users have been gradually adopting the app. They have been able to discover it primarily from the Facebook and Instagram pages created to spread the app. As we saw above in Section 6.2, the Facebook page was shared in a Facebook group of sale and exchange of second-hand books, which corresponded to the moment when more people joined the app, and also in the partners' networks (Ror de Livros, Sana Editora and Editorial Divergência). This dissemination originated several other interactions from different people (likes, comments and shares), which made it possible for the app to reach more people. Fig. 6.4 shows the evolution of the active installs (the number of devices that have the app installed on each day, which excludes the uninstalls) over the time, labeled with the moments the app was shared.

The app has been actively used, and this has been quite enriching for the reasons we saw above. On 5 June 2019, 312 users were registered. On average, there were 17 daily active users, 60 weekly active users and 222 monthly active users, with a daily user engagement of 3min24s. 2047 books were added to the users libraries. 389 offers were created, resulting in 49 applications (users that manifested the interest of without necessarily proceeding with the exchange) and in the exchange of 11 books. Users followed each other 566 times, sent 505 messages, created 96 posts and liked 176 times.

Meeting the users needs is key to the success of a product, and the strength of this dissertation is the way it goes beyond a simple proof of concept. The feedback received from the users with the app in production has been an important input for the next iterations of

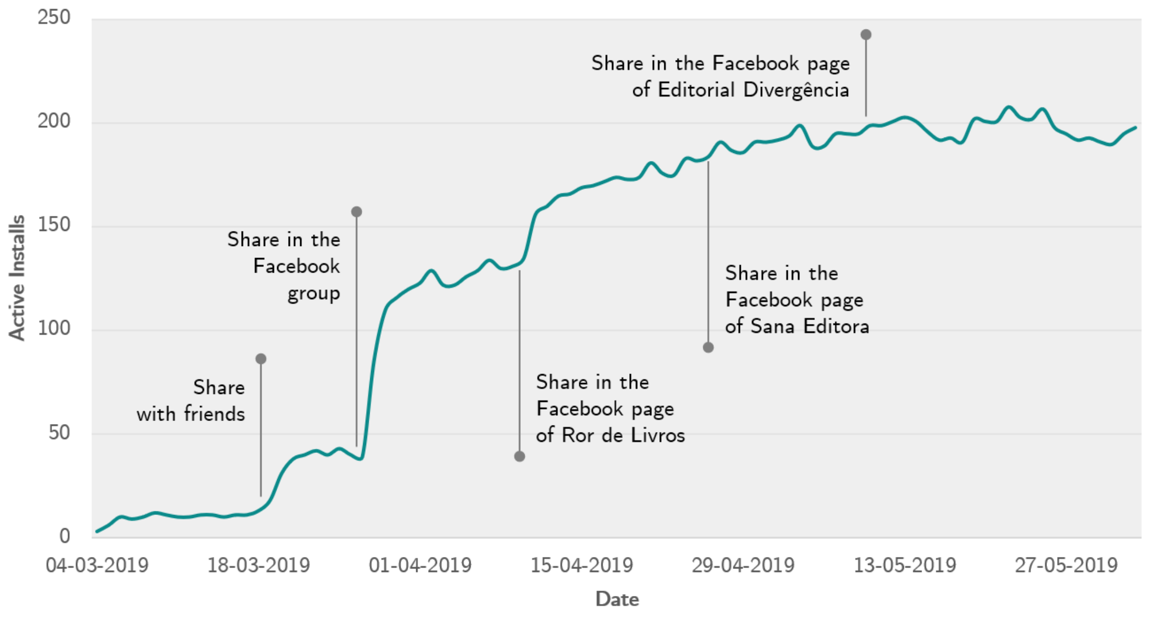


Figure 6.4: The app's active installs over the time.

the design. This was only made possible due to the way the development and deployment were conducted, with all the integrated tools prepared to collect the necessary insights and to provide a channel to receive the users feedback.

CHAPTER 7

Preliminar Business Model

"Hoping for the best, but expecting the worst." – Alphaville

The app was planned to be sustainable. We have been addressing several aspects related to sustainability, mostly in the decisions taken in the design, development and deployment stages. Yet we did not discussed how it will be made profitable, or at least able to cover the costs. The idea in the plan for the future of the app is to continue investing on its development in the following months. By the end of this period, this might be a project in expansion, with many users using the app, or it might continue like today, with a small group of users. The app will need to be ready for both scenarios, because if the project fails the users that already joined the app deserve to continue having it available, and if it succeeds, the expenses of scaling must be covered. In this Chapter it is discussed the business model of the project.

7.1 Roadmap

Let us consider a plan for a year from the day of the release of the first public beta version, on 18 March 2019. Until now several versions were already released, addressing bug fixes and new features (see Section 6.3). Also, some partnerships were made possible (see Section 6.4). The plan is to keep working on it taking into account the main topics discussed below.

7.1.1 Core features

It has been given priority to the core features, that defined the MVP. The MVP is now being used and is ready for the different use-cases, has discussed in Section 2.3.4. This includes attending the possibility that the app does not have many users, as it is desirable to boost its most social components. The next months will be essential to keep polishing the core features, so the changes that the concept of the app succeeds are higher. The focus has been on realizing what has led people to download the app and what keeps them, and improving the app in this sense.

7.1.2 Expansion

For now the app is available in the Play Store, conditioned to a beta release in Portugal. This has been a good ecosystem to make the necessary improvements to the core features in contact with the first users. The next step will be releasing it in a second country, probably the United Kingdom or the United States of America. If the decision goes to the USA, it might reach 5x more people than in the UK, considering their population. Both the countries have a good culture of reading. The USA and the UK are the 22nd and 25th countries where people read most, respectively [63]. Making the app available in a second country will be a different challenge, because at this point there will be two different communities of users, which has direct implications in the app, adding a different degree of complexity to it. For example, the books published in english must be as successfully indexed as the ones in portuguese. Also, when displaying suggested users to someone in the USA, it might be more interesting to show first the people that speak the same language, and not someone commenting books in portuguese. About the store, it is based on the proximity of the users, showing the offers of other people around near a given radius, so it might not be problematic. Still, it will be needed to deal with two different currencies. This conditioning of different books and different people cohabiting in the same app will be attended when expanding to a second country. After that, it will be easier to make the app available in the remaining countries.

7.1.3 Version 1.0

The release of the first stable version is the more anticipated moment. It will happen at the same time as the release of the iOS version (see Section 7.1.4), so users will have for the first time the possibility of taking advantage of the whole platform, in the two main mobile stores. Up to the version 1.0, the app will have some non-essential but important features, such as book recommendation and the ability to edit the user profile, and features to keep it compatible with edge cases, such as exporting data about books or deleting accounts. These two last examples are also essential to have the app in accordance with the EU General Data Protection Regulation (GDPR). These changes are planned and have been developed in different milestones, corresponding to the releases until version 1.0 (see Fig. 7.1).

7.1.4 iOS release

The release of the iOS version will happen in a moment when the app is stable, with the release of version 1.0 for Android, which will be compatible with the expected different problems that will be manifested in the iOS deployment, so there will be a period to meet these potential problems. Because the codebase is the same for Android and iOS (see Section 4.1), there will not be a significant development effort to make it ready for iOS. Even so, it will be needed to change some icons and interactions to adapt the app to the iOS experience. The deployment for iOS will cost the annual $99 necessary to have a developer account on the App Store, so this moment will be preceded by the enabling of advertising in the app, to allow supporting it (see Section 7.4).

7.1.5 Advertising

Advertising will be one of the sources of incoming. Google has an advertising platform that allows to show ads in exchange for money. This solution is widely used for the web and mobile [64]. For apps, there is Google AdMob, that provides different solutions targeted to mobile devices, such as i) interstitials, that are full-screen ads generally used in natural break points in the flow (e.g., the end of a game level), ii) or rewarded ads, where the users explicitly choose to view it, generally a video, and at the end they receive some reward that can be used in the context of the app. At this moment there are interstitials placed in the app, but deactivated by a remote configuration (see Section 5.6.2). When the time comes, they will be activated. For now it would not be useful to have them activated, because i) the app is giving the first steps, and nobody likes advertising, so this would contribute negatively to the first impression people have of it, ii) because premium accounts are still not introduced, so users would not have an alternative to disable the ads even if they did not mind to pay for it, and iii) because it would not have a significant return, since there is not enough traffic in the app to make it profitable.

7.1.6 Premium accounts

Along with advertising, premium accounts will be another source of incoming. The goal for the app is to make it free to everyone, with the option to pay a periodic subscription to access extra features, in a premium plan. These extra features are being carefully chosen, so the experience of who is using the free version is not affected. Users that make a conventional use of the app shall be satisfied with the free version, while the users that want to use it more professionally will have the option to do it with a premium account. The two main areas to be monetized are the library and the store, for who wants to manage a small library or to sell books on a larger scale. For example, users with a premium plan will have the possibility to add custom bookmarks (which might be useful to organize the books by location, for example) and an higher limit of active offers to sale books (currently, for regular users the limit is 10). At the moment, these extra features are already being developed and tested with a small group of users.

This is the roadmap for a period of one year. Fig. 7.1 shows a timeline with the main milestones for the next months. The long-term is also being planned, so there will be a logical continuity at the end of this period. This includes i) several features, such as a reading mode (see Section 3.3.1) or making the app available while offline (see Section 3.3.1), ii) new contexts of usage, such a web version of the service to be accessible and more efficiently used in a computer browser, and iii) different markets, as we will now see in Section 7.2.

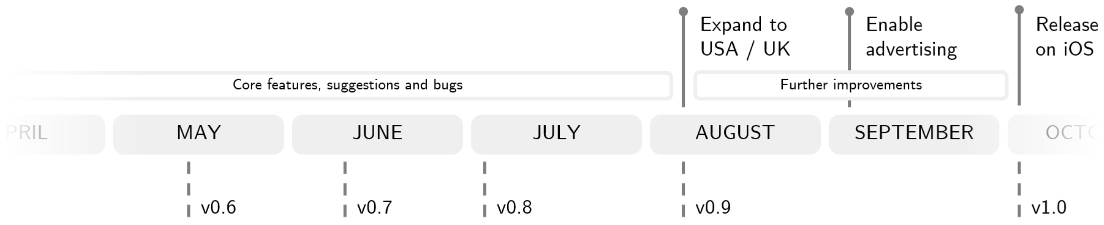
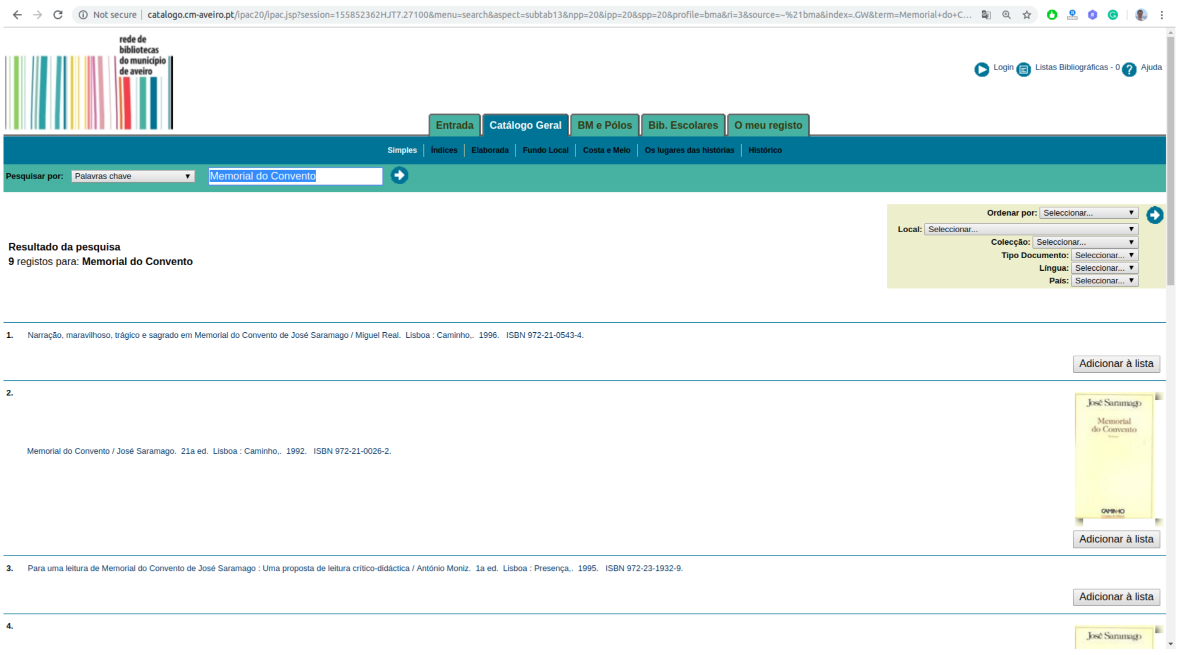


Figure 7.1: The roadmap of the following months. 0.7 and 0.8 will still be abundant in changes, 0.9 and 1.0 will be more conservative, to be able to converge to a stable version. After the 1.0 release, the first months will be specially focused on the iOS version, that despite taking advantage of the stability of the improvements from the last months, it may bring platform specific issues. After that, it will be time to invest in the long-term features and on some marketing (see Section 7.3).

7.2 Future partnerships

In this short time since the release of the first public version of the app it was already possible to establish some partnerships and explore new use cases (see Section 6.4). The plan is to continue expanding the vision for the app to a broader market, and not only the personal usage. An ideal scenario would be an app where everything related to printed books would be there. Imagining someone in a city willing to read a book, it would be possible not only to know if a friend has it so they could lend it, but also to search for it in the book stores and libraries nearby. Two good markets to integrate in the app would be bookstores and libraries, both for the same reason: their online presence. Generally bookstores do not have the necessary dimension to keep an online website, so they at most rely on the social media, like Facebook, to keep a page for their business. In the case of libraries, some have a catalog of their books online, but they have not been able to follow the recent technological changes. Taking as example the municipal libraries in Portugal, they do not have apps. Searching for *biblioteca municipal* (in english, *municipal library*) in the Play Store will return no relevant results. Also, their online catalogs are completely outdated (see Fig. 7.2).

The app has several opportunities to explore here, that are now clearer than when performing the SWOT analysis in Section 2.3.3. Out-of-the-box, it is already possible to catalog and sale books, but still it needs to be adapted to meet the necessary requirements of these more demanding use cases. Achieving this, the app will be the place to get readers closer to the institutions in an interconnected and harmonic environment that is not possible in other general-purpose platforms.



Not secure | catalogo.cm-aveiro.pt/ipac20/ipac.jsp?session=155852362HJT7.27100&menu=search&aspect=subtab13&npp=20&ipp=20&spp=20&profile=bma&ri=3&source=~%21bma&index=.GW&term=Memorial+do+C...

Pesquisar por: Palavras chave | Memorial do Convento

Ordenar por: Seleccionar...
Local: Seleccionar...
Colecção: Seleccionar...
Tipo Documento: Seleccionar...
Língua: Seleccionar...
País: Seleccionar...

Resultado da pesquisa
9 registos para: **Memorial do Convento**

1. Narração, maravilhoso, trágico e sagrado em Memorial do Convento de José Saramago / Miguel Real. Lisboa : Caminho,. 1996. ISBN 972-21-0543-4.

Adicionar à lista

2. Memorial do Convento / José Saramago. 21a ed. Lisboa : Caminho,. 1992. ISBN 972-21-0026-2.

Adicionar à lista

3. Para uma leitura de Memorial do Convento de José Saramago : Uma proposta de leitura crítico-didáctica / António Moniz. 1a ed. Lisboa : Presença,. 1995. ISBN 972-23-1932-9.

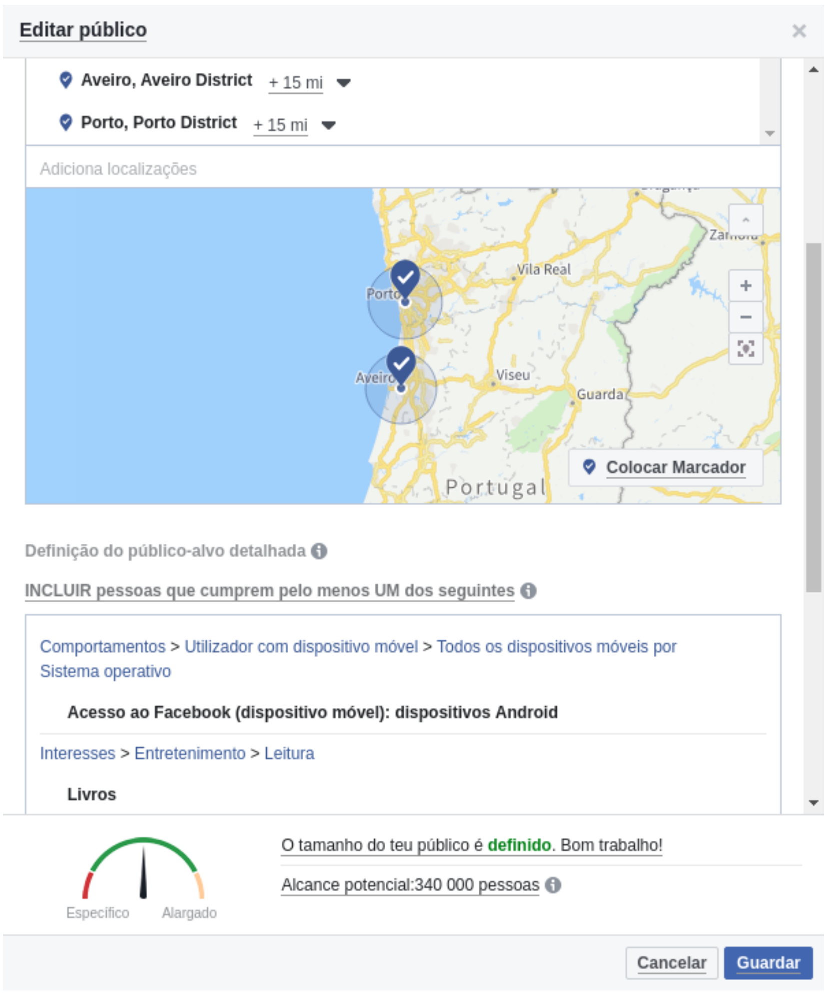
Adicionar à lista

4.

Figure 7.2: The online catalogue of the municipal library of Aveiro, on 22 May 2019, showing the search results for *Memorial do Convento*. Its design is rudimentary and the connection is not secure (HTTPS is not being used), which is worrisome considering that there is a login system where the users can manage their requisitions. The website can be accessed on catalogo.cm-aveiro.pt.

7.3 Marketing

At some point it might be interesting to advertise the app in the social media, to give it a boost. Facebook is the perfect place for direct advertising, so it would allow to reach the target audience of the app. It was performed a simple simulation to check how it could be done: selecting everyone with 18-35 years old, in a radius of 15mi (~24km) from Aveiro and Porto (choosing a small area instead of the whole country might be useful to create a community of users that will exchange their books close to each other), that are accessing an Android smartphone or tablet and have interest for books, it returns a potential reach of 340.000 people (see Figure 7.3). Defining a budget of €3,5 for a day of advertising, Facebook estimates the reach of 910 to 3.400 people (it will depend on how users interact with the advertised post).


Editar público

Aveiro, Aveiro District + 15 mi

Porto, Porto District + 15 mi

Adiciona localizações

Definição do público-alvo detalhada

INCLUIR pessoas que cumprem pelo menos UM dos seguintes

Comportamentos > Utilizador com dispositivo móvel > Todos os dispositivos móveis por Sistema operativo

Acesso ao Facebook (dispositivo móvel): dispositivos Android

Interesses > Entretenimento > Leitura

Livros

O tamanho do teu público é definido. Bom trabalho!

Alcance potencial:340 000 pessoas

Figure 7.3: A simulation to advertise a post on Facebook, targeting the audience that might be interested in the app, on 29 May 2019.

7.4 Balance

Until now the project had the investment of i) a person designing and developing the app, ii) the volunteer participation of several people, iii) the resources provided by the university, namely the room to perform the participatory study (see Chapter 3), and iv) the payment of the developer account fee of $25. This was compatible with the initial stage, but will not be viable in the future, because the deployment resources will start to be payed after the free trial of one year, and the same for the human resources, specially when planning to have a team working on it. In order to anticipate the profit of the app, it was estimated a balance between the revenue and expenses, depending on the evolution of the user base. It was considered a range from 1 user to 1.000.000 users, and several assumptions have been made. The calculations were made taking as unit a day (24 hours), considering that on each day 10% of the users would access the app.

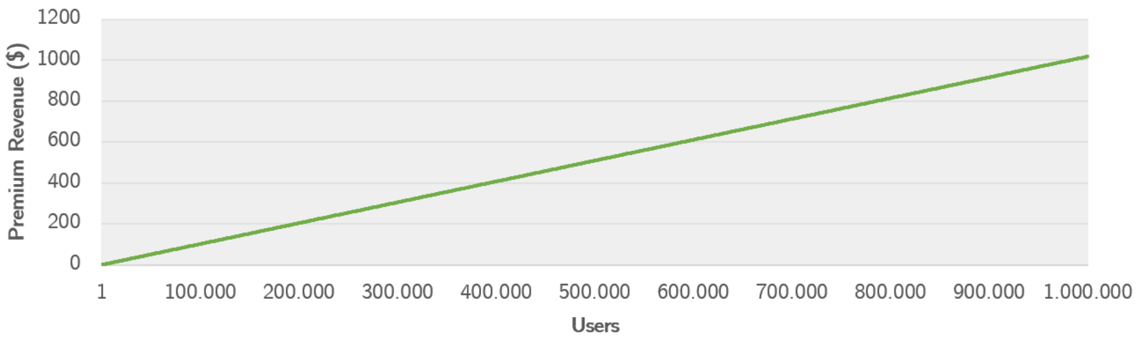


Figure 7.4: The daily premium revenue as a function of the number of users.

7.4.1 Revenue

There will be profit from premium accounts and adverting. Supposing that 3% of the n users will have a premium subscription, that costs 1 euro (1,12 dollars) each month, the daily gain will be (see Fig. 7.4):

$$n * 0,03_{users\_with\_premium} * daily\_gain\_by\_user$$

where $daily\_gain\_by\_user$ is given by

$$\frac{0,7_{google\_play\_revenue\_split} * 1,12}{30_{days\_in\_a\_month}}$$

For the advertising, supposing that each user active on that day sees 3 ads (not necessarily all in a unique session, it can happen at any moment of the day), and that the effective Cost Per Mille (eCPM)[1] is $3,50 [66], if we exclude the premium users, that are not exposed to advertising, we have a daily gain of (see Fig. 7.5):

$$n * 0,1_{daily\_active\_users} * 0,03_{users\_without\_premium} * 3_{ad\_impressions} * \frac{3,5_{ecmp}}{1000}$$

In the estimate of the revenue it is not being accounted the different performances of the ads on iOS and Android (ads displayed on iOS will be more valuable [66]) and other possible sources of income, like partnerships.

[1]eCPM is a term used to express the average revenue per thousand of ads impressions [65].

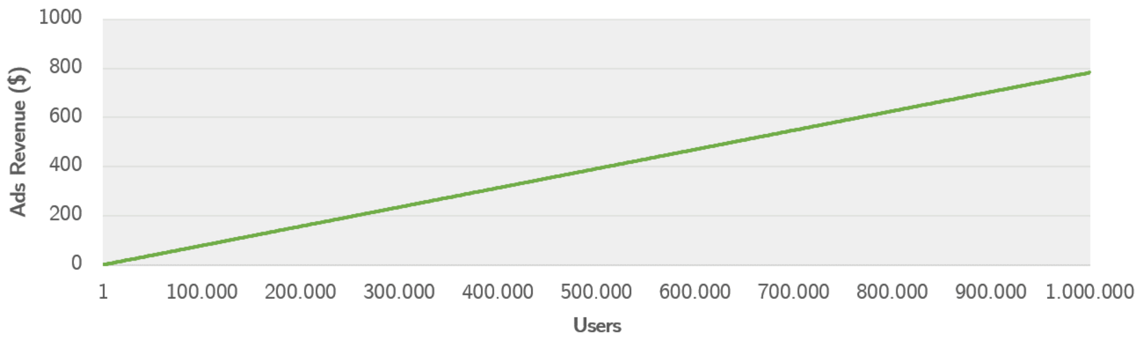


Figure 7.5: The daily ads revenue as a function of the number of users.

7.4.2 Expenses

Let us consider the server, running on GAESE (see Section 5.2.1), and the database (see Section 5.2.2), and ignore the storage, that is not significant in the overall expenses (see Section 5.2.3).

For the server, for simplification, it was considered the instances of class F2 (256 MB RAM, 1,2GHz), but there are instance types that are more powerful and could be used to scale vertically. This scenario is restricted to horizontal scaling. Each F2 instance costs $0,10 per hour (the equivalent to 2 instance hours), and the first $1,40 of each day are included in the free plan (correspondent to 28 instance hours, mapped in 14h). Supposing that, on average, each instance is able to handle 10 request per second, this gives 864.000 requests per day. Now, supposing that each user that is in the app during a day, on average, performs 500 requests to the server (for example loading a list of 10 offers involves at most 1 request for the list, 1 request for each of the different 10 seller profiles and 1 request for each of the 10 different books, in a total of 21 requests). Given this, each instance can handle, on average 864.000 / 500 = 1.728 users per day. This assumes that the traffic is homogeneous throughout the whole day, which is not what happens, but it is also valid for a different distribution, because if there are more instances allocated at a given moment, there will be less instances at another moment, if we consider the same total traffic for a day (recall Fig. 2.1). This would give a daily cost of (see Fig. 7.6):

$$2,4_{per\_instance} * \left\lceil \frac{n * 0,1_{daily\_active\_users}}{1728_{users\_in\_a\_vm}} \right\rceil - 1,40_{for\_free}$$

There is only an integer number of instances, that is why ceil is applied. This assumes that there is always at least one instance running, that is ready to receive a request at any time. For few users, this is overestimated, because they will not be using the full power of an instance. For example, the traffic will be condensed at the day, and the instances are able to scale to 0 at night, when nobody is accessing the app. For more users, this simplification is okay, because it is expected that the times of least traffic will have a number of users different from zero.

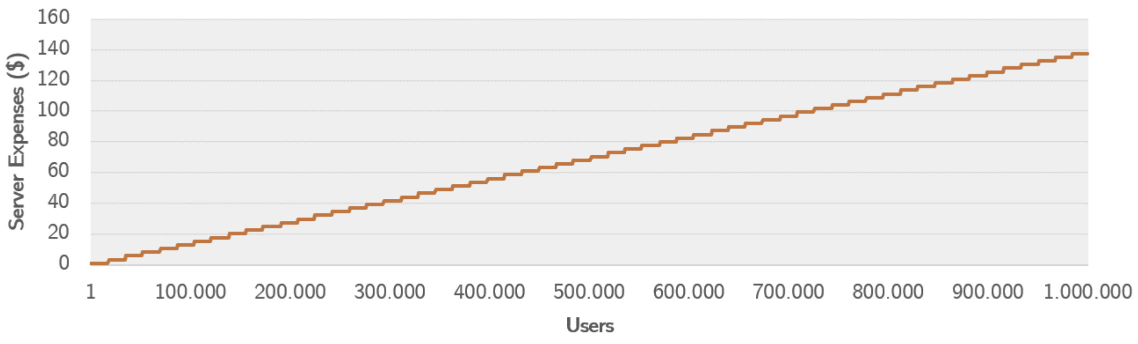


Figure 7.6: The daily server expenses as a function of the number of users.

For the database, again for the sake of simplification, it was considered only the CPU power (being ignored memory, storage and network usage). It is the resource that most affects the price of the instance. Still, it is hard to predict the needs for this resource, since many factors can influence it, like the size of the database, the type of queries or how caching is used. For less demanding requirements, there are different instance types with a predefined package of CPU and RAM, that have a shared CPU. For more demanding requirements, the instances can be built according to the desired personalized specifications, while having a dedicated CPU. In this estimation, let us suppose that each CPU supports 100 connections per second, which gives 8.640.000 requests per day. Let us also suppose that it used a shared CPU instance up to 100 connections and dedicated CPUs for more than 100 connections. Using a shared CPU instance costs \$0.252 per day, while each dedicated CPU (that has the option for high availability or not, and it will be considered high availability enabled) costs \$0.9912 per day [67]. Also, considering again an average of 500 daily requests performed by an user, this gives 8.640.000 / 500 = 17.280 users per day per CPU. Again, this assumes that the users are evenly distributed along the entire day, which will not happen. But the database resources will not scale up and down like what happens with the server, so in reality the database must be prepared for the traffic peaks and probably this estimation is undervalued. To compensate it, let us consider a factor of 3 times the estimated number of users. This would give a daily cost of (see Fig. 7.7):

$$\begin{cases} 0,252 & \text{if } n * 0,1_{daily\_active\_users} * 3_{cfactor} \leq 17280 \\ 0,9912_{per\_cpu} * \lceil \frac{n*0,1_{daily\_active\_users}*3_{cfactor}}{17280_{users\_per\_cpu}} \rceil & \text{otherwise} \end{cases}$$

Finally, it must also be considered the annual \$99 to keep the app on the app store. It is not significant if the app will have a lot of users, but otherwise it will be. This is an analysis restricted to the deployment costs. Other expenses are not being addressed here, and some are actually the more significant, such as human resources. Marketing, taxes, development material (such as PCs and smartphones), a workspace and other type of expenses would also have to be considered.

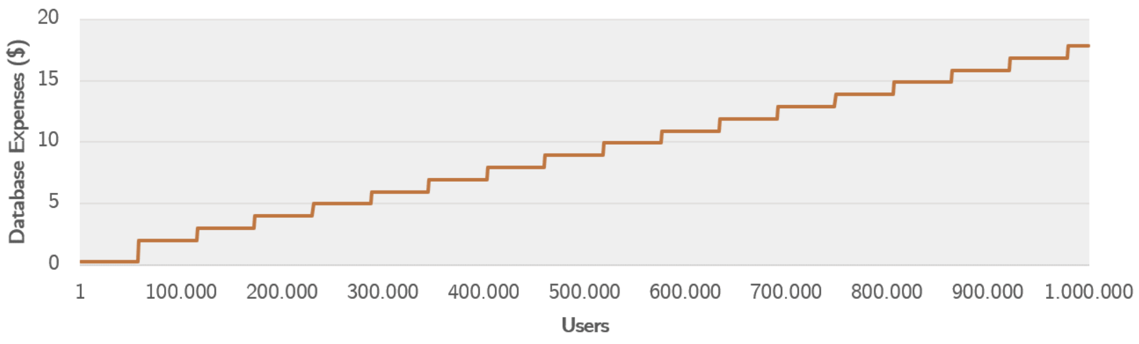
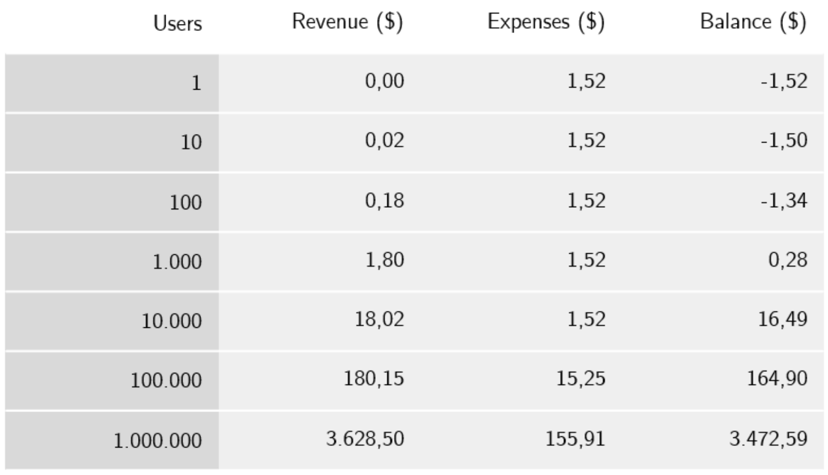


Figure 7.7: The daily database expenses as a function of the number of users.



Users	Revenue ($)	Expenses ($)	Balance ($)
1	0,00	1,52	-1,52
10	0,02	1,52	-1,50
100	0,18	1,52	-1,34
1.000	1,80	1,52	0,28
10.000	18,02	1,52	16,49
100.000	180,15	15,25	164,90
1.000.000	3.628,50	155,91	3.472,59

Figure 7.8: The daily balance of the revenue and expenses for different orders of magnitude of users.

Given this, Fig. 7.8 shows the daily balance of the revenue and expenses for different orders of magnitude of users. The break even point happens around 850 users, when the earnings start to cover the costs (see Fig. 7.9).

This preliminary business model, with a plan for the delivery, partnerships, marketing, revenue and expenses, is an attempt to prepare the following months of the app in the market. The focus of this dissertation is on the design and development process, but since it is about productizing an app, it was important to portray the process contextualized with its business model. The aspects highlighted here are closely related to other topics addressed in the previous Chapters, such as i) the design and development, with the features being delivered according to the planned roadmap, and ii) the deployment, with the need of estimating the necessary resources and their costs.

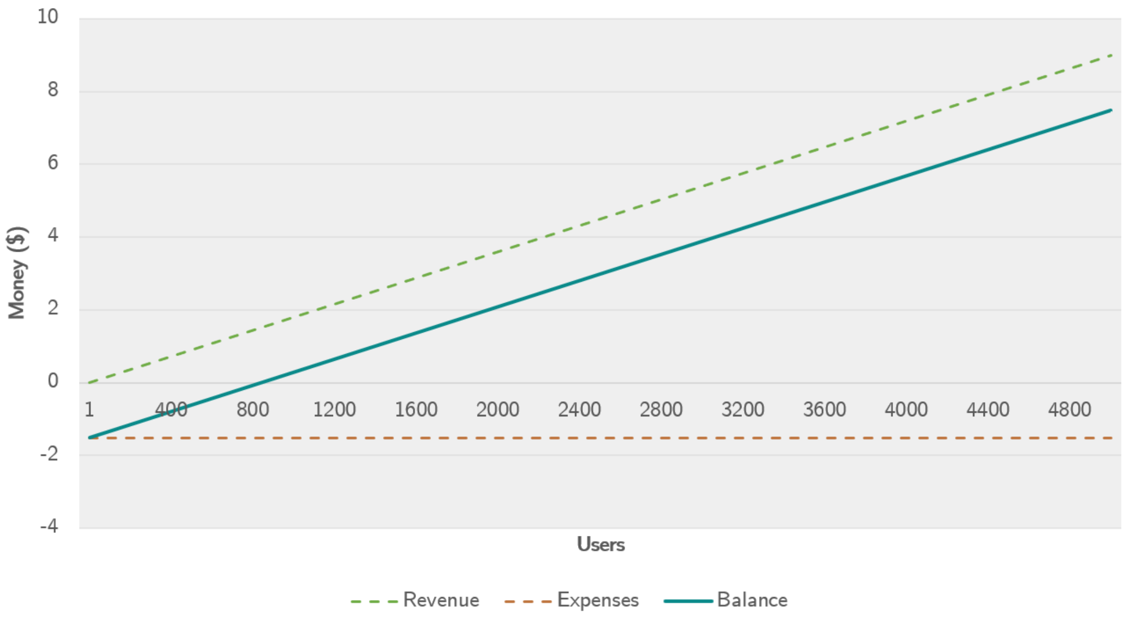


Figure 7.9: The daily balance of the revenue and expenses up to 5000 users. The break even point happens around 850 users.

CHAPTER 8

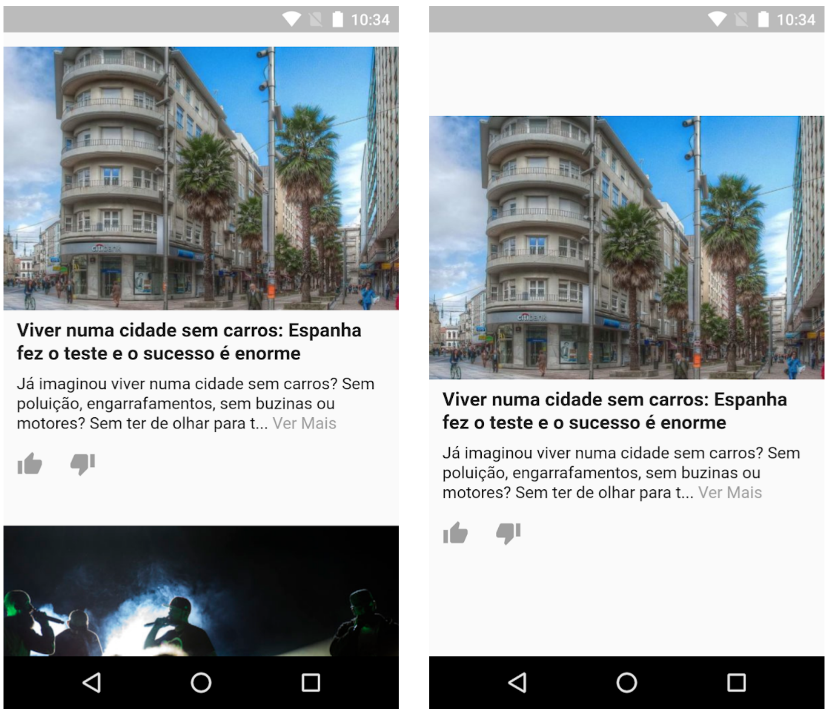
Final Considerations

"All endings are also beginnings. We just don't know it at the time." – Mitch Albom

This work has been challenging in the way it involves different areas, all together for the same purpose - a product. Not a proof of concept. Like any product, it is not a finished work, so its development will continue beyond this dissertation. This Chapter concludes this document, with a reflexion on what was accomplished, the outcomes of a complementary work to this dissertation and some perspectives for the future.

8.1 Complementary work

As a result of the possibility of attending one subject of any area of study in the university, as part of the syllabus of Informatics Engineering, during the period of this dissertation it was attended the subject of Interaction Design and Analysis, from the PhD in Information and Communication in Digital Platforms, in DeCA. The choice for this subject was motivated by the interest of learning more about the interaction design, which was also one of the motivations for this dissertation. Consequently, the outcomes from this subject were important for this work in such a way that the participatory design study, described in Chapter 3, resulted in the elaboration of another study, that was proposed and developed in this subject, with a subsequent writing of an article entitled *How to deal with mindless scrolling?*. The study questioned how the way in which social networks and other sources of information are designed influences how people interact with this information, particularly with regard to their attention (see Section 3.3.1). Everyday, we are overwhelmed with enormous amounts of information, most of the time, consumed among apparently infinite lists, whether in social networks or news portals. This can lead us to lose the focus of what is relevant to read, and we might end up mindless scrolling, this is, scrolling the content without paying attention to it. This study sought for solutions that might help dealing with mindless scrolling. Then, it focused on a particular aspect of how users navigate from publication to publication, when scrolling,


10:34

Viver numa cidade sem carros: Espanha fez o teste e o sucesso é enorme

Já imaginou viver numa cidade sem carros? Sem poluição, engarrafamentos, sem buzinas ou motores? Sem ter de olhar para t... Ver Mais

10:34

Viver numa cidade sem carros: Espanha fez o teste e o sucesso é enorme

Já imaginou viver numa cidade sem carros? Sem poluição, engarrafamentos, sem buzinas ou motores? Sem ter de olhar para t... Ver Mais

Figure 8.1: The interfaces of versions A, on the left, and B, on the right.

designing a test for two different interfaces. Interface A was prepared with a regular scrolling list and interface B was prepared with a list that only scrolls one publication at a time (see Fig. 8.1). The results showed an increase of the time spent in each publication of 89% from interface A to B, which indicates a clear tendency to help improving the users focus, thus suggesting that this may be an effective approach to deal with mindless scrolling. The article is available in Appendix B.

8.2 Other considerations

The focus on this document was to describe and discuss the most significant parts of the project along the dissertation. During this path, other topics were not highlighted. Here are some considerations about them.

8.2.1 GDPR

The project was born during the moment of greater agitation about the EU GDPR, that started to be enforced on 25 May 2018. It was mandatory to consider it during the process, specially because of the social component of the app, so it was developed taking into account the implications of this regulation. The privacy policy that users agree with when registering on the app is also in accordance with it (https://desvaneio.com/privacy-policy). Still, as described in Section 7.1.3, there are some features planned to be addressed up to version 1.0, when it will be fully compliant with GDPR. The care with people's personal data was also reinforced early in the project, during the participatory design study, when the process was revised by CED-UA (see Chapter 3), that advised about the ethics of the study.

8.2.2 Implementation details

This document abstracted the implementation details. Still, the project has involved making many decisions to create something to last in the long-term. Because this was not a proof of concept, but something to continue being used in the future, the implementation was made with the responsibility of not ignoring details that otherwise could be neglected in a prototype. Some examples of these topics are the i) data model, that is complex and extensive, given the amount of features, ii) the structure of the API, including how the URLs and methods were organized, grouping its functionalities consistently, iii) some followed programming patterns, that helped to improve the code organization, and hence its readability, or iv) the implications of reactive programming, that was extremely useful to answer the needs of a dynamic UI, and which allowed to change something in a view and updating the remaining views with the same information, without the need to manually refresh them. These and other implementation details do not contribute substantially to the objectives of this dissertation, which was focused on the process, but still it was a significant part of the time invested in the project.

8.3 Future work

The development of this project opened many doors for future work. Besides what is planned in Chapter 7, with all the changes to be made in the following months, there will be three main areas of focus that will be addressed in the long-term, with the natural evolution of the app: A/B testing, gamification and automation.

About A/B testing, until now it was possible to perform a small experiment just to get familiar with Firebase (see Section 6.5). Later, with a larger user base, it will be possible to take more advantage of this tool. It is planned to be run an experiment regarding the results of the article done in parallel with this dissertation, which evaluates an alternative to deal with mindless scrolling (see Section 8.1). The laboratory test done there will be adapted to the real scenario of people using the feed of the app.

Gamification is known to improve the user experience (see Section 2.1.6). The app has a large margin of progression towards this. For example, users can be challenged to read more and share their reading progress to their contacts. This was not considered essential for the MVP, but now that the app is getting stable it will certainly be a priority. As there was no opportunity to conveniently explore this topic, it will require further research before heading to concrete changes in the app functionality.

Finally, there is work to do on automation, specially for the release process of the app in the Play Store. Currently it is compiled and uploaded manually, but tools like fastlane or Codemagic can automate the process, saving time and avoiding possible mistakes common on these repetitive tasks.

8.4 Conclusions

Throughout this project it was expected to learn what would be necessary to transform a prototype into a product, facing issues that do not arise when developing a proof of concept.

The proposal of this work was to deliver a real app to the customers, with the satisfaction of all the quality requirements that it implies. This has been accomplished, with the release of the beta version on the Play Store, on 18 March 2019, which has already reached more than 300 users, and with a sustainable plan to the version 1.0 (see Chapter 7).

It was ambitious to create something in a period of a year, in a multidisciplinary project, taking into account the different concerns of different roles, for instance, the UX designer and developer, finding trade-offs when addressing some conflicts of interest. Meanwhile, time was always a conditioning factor. If it had been possible it would have been interesting to cover in this dissertation what will be done in the following months, specially the validation of the solution with a larger audience. Still, as a work in progress, hopefully for several years, it would not be expected to finish it by the end of this dissertation.

Therefore, it is expected that the app is sufficiently representative in the challenges it involves, so the process could be adapted to other apps. Hopefully, the ideas discussed here can be an example of what might be the problems to keep in mind when starting to develop an app, what might be the solutions for these problems and what will be the mistakes to avoid.

Appendix A

This appendix consists of the documentation sent to CED-UA when asking about the ethics of the Participatory Design Study (see Chapter 3), according to the regulation on https://www.ua.pt/ced/page/17836. In total there are 5 documents: i) the CED-UA form filled out with the description of the project, ii) the description of the procedure of the study, iii) a declaration from the leaders of the study, in this case the mentors of the project, and the declarations of the directors of iv) DETI and v) DeCA.

universidade de aveiro theoria poiesis praxis

Leonardo Oliveira

[illegible]

Formulário para pedido de parecer sobre questões éticas no âmbito de projetos de investigação científica

1. **Caracterização do projeto**

As aplicações móveis têm tido um crescimento exponencial desde que surgiram lojas como a Apple Play Store ou a Google Play Store. O desenvolvimento de aplicações tem sido estudado em diversos moldes, desde as linguagens de programação à experiência de utilização (UX), cobrindo aspetos relacionados com plataformas, modelos de negócio e HCI.

Hoje em dia, os alunos são expostos em diversas unidade curriculares a estes aspetos do desenvolvimento de aplicações, mas não chegam a ir além de simples provas de conceito. Nesta dissertação é proposto ao aluno seguir de forma completa os diversos passos para trazer uma aplicação móvel para o mercado: desde a ideia até ao produto, disponibilizado numa loja de aplicações, com clientes. Mais do que o trabalho de desenvolvimento da aplicação, esta dissertação foca-se nos aspetos menos explorados na conceção do serviço que suporta a aplicação móvel, atendendo às questões de UX e avaliação de usabilidade de um *Minimum Viable Product* (MVP), com um público específico, aliado à validação de um modelo de negócio que possa suportar a infraestrutura e os custos de desenvolvimento da aplicação.

Neste sentido, queremos atender às questões éticas inerentes ao estudo proposto neste pedido de parecer, que envolve a participação dos utilizadores no desenvolvimento do MVP.

1.1 Designação do projeto
"Concepção e desenvolvimento participativo de uma mobile app para a mediação da leitura de livros"

1.2 Data de início

11 Fev 2019

1.3 Data de conclusão

12 Julho 2019

1.4 Financiamento

a) Não se aplica porque o projeto tem enquadramento no trabalho de dissertação de mestrado do estudante de Engenharia Informática, Leonardo Oliveira.

1.5 Objetivos (3000 caracteres)

Na sequência do trabalho de dissertação do estudante, pretende-se proceder a um estudo que se enquadra na fase de design e prototipagem de uma aplicação móvel, numa altura em que é de extrema relevância a colaboração de potenciais futuros utilizadores que possam validar a solução encontrada. A solução consiste num protótipo funcional de uma aplicação no qual, de um modo geral, é possível os utilizadores organizarem os seus livros, trocá-los com outros leitores, comentá-los, entre outras possibilidades. Pretende-se pedir a colaboração de 30 a 40 participantes numa sessão

Leonardo Oliveira

com duração prevista de 30 minutos a 1 hora. Os participantes serão recolhidos através de uma amostra intencional formada a partir daquele que se prevê ser o público-alvo da aplicação - pessoas que têm hábitos de leitura lúdica.

1.6 Metodologias experimentais e planificação (5000 caracteres)

A planificação experimental encontra-se detralhadamente explicada no documento em anexo “Procedimento_Design_Participativo_2019FEV13.pdf”.

1.7 Questões éticas sobre as quais se pretende o parecer do CED (3000 caracteres)

Será preciso o parecer do CED sobre a dimensão de participação direta dos diversos sujeitos que entram no processo de concepção e avaliação do design de interação e das interfaces do produto (mobile app) em causa. Parecer que deve recair sobre o documento anexo que informa sobre “Procedimento proposto para a dimensão Design Participativo”.

1.8 Legislação que, eventualmente, possa estar na origem do pedido (1000 caracteres)

Não existe nenhuma legislação que o exija este pedido a não ser a conduta ética e deontológica da equipa de investigação que considera necessário dar conhecimento e obter o parecer da Comissão de Ética da UA, onde se enquadra este trabalho.

1.9 Estudos semelhantes publicados em revistas internacionais. Indicar, nomeadamente, trabalhos com implicações de natureza ética (2000 caracteres)

2. **Equipa**

2.1 Investigador responsável
Diogo Gomes, co-IR: Óscar Mealha e
Leonardo Oliveira

2.2 Categoria
Professor Auxiliar, co-IR Professor Associado com agregação e
estudante Mestrado em Engenharia Informática

2.3 Entidade de acolhimento no qual se integra

Instituto de Telecomunicações/DETI e DigiMedia/DeCA

2.4 Contributo do projeto para os objetivos estratégicos da entidade de acolhimento (1000 caracteres). - Declarações anexas.

Este projeto contribui para a área estratégica de interação e media digital, neste caso para mediar o contexto de leitura de livros. Trata-se um projeto inovador no contexto da engenharia informática e na área de ciências e tecnologias da comunicação e é nutrido cientificamente de forma transdisciplinar pela participação direta destas duas grandes áreas disciplinares, sem prejuízo para a contaminação que tem recebido, detalhadamente de outras.

2.5 Experiência do investigador responsável na área do projecto, incluindo até 5 artigos científicos em revistas de circulação internacional (3000 caracteres)

- A video engine supported by social buzz to automatically create TV summaries. Pedro Almeida, Jorge Ferraz de Abreu, Rita Oliveira, and Diogo Gomes. Multimedia Tools and Applications, pages 1--19, feb 2018.
- Modelling patterns in continuous streams of data. R Jesus, M Antunes, D G Gomes, and R Aguiar. Open Journal of Big Data, 4(1):0, dec 2017.

- Mealha, Ó. (2016). Citizen-driven dashboards in smart ecosystems: a framework. *Interaction Design and Architecture(s) Journal - IxD&A | Special Issue on: Special Issue on: Smart Learning Ecosystems and Regional Development*, *31*, 32–42.
- Rodrigues, R., Veloso, A., & Mealha, O. (2016). Influence of the graphical layout of television news on the viewers: An eye tracking study. *Revista Da Obercom - (OBS*) e-Journal*, *10*(1), 67–82. Retrieved from http://www.obercom.pt/content/8.cp3

2.6 Outros investigadores com participação relevante, incluindo dados biográficos e publicações (3000 caracteres)

- Simon Attfield, Gabriella Kazai, and Mounia Lalmas. Towards a science of user engagement (Position Paper). WSDM Workshop on User Modelling for Web Applications, 2011.
- Irene Au, Richard Boardman, Robin Jeffries, Patrick Larvie, Antonella Pavese, Jens Riegelsberger, Kerry Rodden, and Molly Stevens. User experience at google: focus on the user and all else will follow. In CHI'08 Extended Abstracts on Human Factors in Computing Systems, pages 3681–3686. ACM, 2008.

3. **Assinatura do Orientador Responsável do Trabalho de Investigação**

(IR/Orientador)

Diogo Gomes

Diogo Gomes

(co-IR/co-Orientador)

[illegible]

Oscar Mealha

(Estudante Mestrado Engenharia Informática)

Leonardo Oliveira

Leonardo Oliveira

Procedimento proposto para a dimensão Design Participativo

Concepção e desenvolvimento participativo de uma mobile APP para a mediação e leitura de livros

Na sequência do trabalho de dissertação do aluno Leonardo Oliveira (leonardooliveira@ua.pt), do mestrado em engenharia informática da Universidade de Aveiro, sob orientação dos professores Diogo Gomes (dgomes@ua.pt) e Óscar Mealha (oem@ua.pt), pretende-se proceder a um estudo que se enquadra na fase de design e prototipagem de uma aplicação móvel, numa altura em que é de extrema relevância a colaboração de potenciais futuros utilizadores que possam validar a solução encontrada. A solução consiste num protótipo funcional de uma aplicação no qual, de um modo geral, é possível os utilizadores organizarem os seus livros, trocá-los com outros leitores, comentá-los, entre outras possibilidades. Pretende-se pedir a colaboração de 30 a 40 participantes numa sessão com duração prevista de 30 minutos a 1 hora. Os participantes serão recolhidos através de uma amostra intencional formada a partir daquele que se prevê ser o público-alvo da aplicação - pessoas que têm hábitos de leitura lúdica. As sessões ocorrerão no laboratório 40.2.10, que possui um equipamento com a tecnologia de eye tracking. Para cada sessão estão planeados diversos momentos cujos objetivos são descritos ao longo deste documento.

1. Consentimento informado

Inicialmente, é explicado ao participante o contexto da sessão, nomeadamente os objetivos do estudo. O participante é informado que os dados recolhidos durante a sessão são anónimos, sendo exclusivamente utilizados no contexto do estudo, e que a sua participação é voluntária, sendo possível a qualquer momento recuar nessa decisão. O participante é então convidado a ler e assinar um documento com um consentimento informado (**Anexo I**).

2. Entrevista semiestruturada

Seguidamente haverá uma conversa com o participante, na qual se pretende caracterizar o seu enquadramento enquanto leitor e utilizador de aplicações móveis. São também abordados pontos que vão ao encontro dos casos de utilização da aplicação, com o objetivo de perceber a sua adequação para futuros utilizadores, havendo lugar a recolha de dados quantitativos, valorativos de opinião e dados qualitativos. A contribuição do utilizador, num contexto em que haverá espaço para desenvolver um diálogo mais detalhado em relação a algumas das questões expostas, permitirá recolher opiniões, hábitos e expectativas que ajudarão, de uma forma subjetiva, a ajustar o conceito da aplicação, de modo a aproximá-la das necessidades dos futuros utilizadores. O guião utilizado durante a entrevista está presente no **Anexo II**.

2.1. Informação sociodemográfica

É recolhida a idade e o sexo. Outros aspetos sociodemográficos foram considerados irrelevantes para o contexto deste estudo.

2.2. Expectativas

Estando o participante já elucidado quanto ao assunto da sessão, de forma a perceber as suas expectativas quanto ao tema do estudo, ser-lhe-á perguntado o que imagina que a aplicação faz, quais as suas funcionalidades. Na melhor das hipóteses, esta abordagem fará surgir à memória algo que o participante gostaria de ver numa aplicação, algo que lhe seria útil. A sugestão do participante poderá ainda consistir numa funcionalidade que ainda não tenha sido considerada no estudo, ou então o participante poderá não se lembrar de nada em concreto.

2.3. Literacia digital

A aplicação proposta tem uma vertente digital implícita. A prática na utilização de um smartphone poderá justificar as maiores ou menores dificuldades a executar a experiência com o protótipo da aplicação, numa fase mais adiante na sessão.

2.4. Hábitos de leitura lúdica

O principal público-alvo da aplicação são as pessoas com hábitos de leitura lúdica. Perceber esses hábitos permitirá confrontá-los com as opiniões do participante em relação ao protótipo apresentado. Coloca-se a hipótese de que participantes com hábitos de leitura demonstrem opiniões mais favoráveis em relação à relevância da aplicação.

2.5. Hábitos de partilha de livros

A aplicação envolvida neste estudo coloca-se como uma solução para as potenciais dificuldades que surgem quando as pessoas pretendem trocar livros entre si (seja empréstimo, venda, doação ou troca por troca). Será relevante relacionar os participantes que já enfrentaram essas dificuldades com as suas opiniões quanto ao protótipo proposto.

2.6. Hábitos sociais de leitura lúdica

Existe também uma componente de interação social que a aplicação proposta pretende explorar. Perceber os hábitos de interação social das pessoas na partilha de opiniões sobre as suas leituras permitirá avaliar a viabilidade da existência da componente de interação social na aplicação.

Estes são os tópicos previstos para a entrevista a realizar-se. Tratando-se de uma conversa aberta, dialógica, é natural que alguns temas sejam mais elaborados no decurso da entrevista semiestruturada. No caso de, no contexto da entrevista, serem abordados outros assuntos que não estejam aqui mencionados, o investigador terá o cuidado de os filtrar e não os incluir no contexto de investigação aqui exposto.

3. Experiência

Após a entrevista, o participante é convidado a realizar uma experiência realizada com um smartphone com o protótipo da aplicação. O smartphone está equipado num equipamento com a tecnologia de eye tracking, que permite recolher o movimento e o tempo de fixação do olhar ao longo do ecrã. A imagem do rosto e a voz são igualmente captados. É-lhe descrito um conjunto de tarefas a executar numa aplicação, sob a forma de um cenário/história (**Anexo III**). Ao mesmo tempo que o participante executa as tarefas descritas, é utilizada a técnica *think-aloud*, sendo-lhe pedido que expresse o que está a pensar a cada momento, para ser possível compreender melhor a forma como este executa cada tarefa ou quais as dificuldades que encontra. Ao longo da experiência o investigador vai tomando notas de detalhes que considere relevantes para, numa fase posterior, os dados do movimento dos olhos, expressão facial e discurso possam ser cruzados. Para tal, o investigador irá preenchendo uma folha que segue o modelo do **Anexo IV**.

Após a conclusão da experiência são colocadas algumas questões ao utilizador, de forma a obter uma opinião sobre o protótipo, percebendo os pontos mais positivos e menos positivos (**Anexo V**).

O participante preencherá ainda um questionário que avalia de modo quantitativo a usabilidade do sistema. No questionário é apresentado um conjunto de afirmações para serem assinaladas com o grau de concordância, segundo a escala de Likert (**Anexo VI**). As considerações do participante neste questionário serão confrontadas com o desempenho da aplicação no momento da experiência, previamente realizada.

4. Fim da sessão

A sessão termina com o investigador a deixar ao participante o seu contacto, para que este possa falar mais tarde sobre qualquer outro assunto que ache relevante, como por exemplo o envio de sugestões de melhorias ou novas funcionalidades.

Anexo I - Consentimento informado

Como aluno do mestrado em engenharia informática da Universidade de Aveiro, quero convidá-lo(a) a participar numa sessão relativa ao trabalho de dissertação que estou a realizar sob orientação dos professores Diogo Gomes (dgomes@ua.pt) e Óscar Mealha (oem@ua.pt). A sessão insere-se num estudo de design participativo para o posterior desenvolvimento de uma aplicação móvel, estando previsto ter uma duração de 30 minutos a 1 hora, num conjunto de três partes.

Primeiro, começaremos com uma conversa na qual colocarei algumas questões sobre hábitos de leitura e de utilização de aplicações móveis. Essas questões ajudar-me-ão a perceber as opiniões dos participantes relativamente ao enquadramento da aplicação móvel a desenvolver.

De seguida, iremos testar a primeira versão da interface da aplicação móvel e algumas funcionalidades em prova de conceito. Através da tecnologia de eye tracking, o movimento e o tempo de fixação do olhar serão gravados e as métricas utilizadas permitirão analisar a interação humano-computador.

Finalmente, para compreender dados de cariz qualitativo, iremos terminar a sessão com uma breve conversa pós-experiência. Desta forma espero comparar as expectativas que cada utilizador tem para com a interface prototipada e as suas funcionalidades.

Lembro que este é um teste à interface e não ao seu utilizador e que todas as informações de caráter pessoal são estritamente anónimas, o direito à privacidade é totalmente garantido. A sua opinião e contributo empírico é extremamente relevante para melhorar a interface e as funcionalidades do protótipo em estudo.

Obrigado pela sua valiosa participação.

CONSENTIMENTO INFORMADO

Compreendi a explicação que me foi dada sobre a investigação que está a ser realizada e que as informações recolhidas são anónimas.

Eu entendo que os resultados do estudo podem ser publicados em revistas científicas, apresentados em reuniões / eventos científicos e usados em outras atividades de investigação, sem qualquer violação de confidencialidade/anonimato.

Ao participar nesta atividade, autorizo o uso de dados anónimos para a finalidade da investigação que lhe está associada e mencionada acima.

Nome do participante: ____

O participante

____

_____ de _______________ de 2018

O investigador

____

_____ de _______________ de 2018

Anexo II: Guião de entrevista semiestruturada

Informação sociodemográfica

Idade: Sexo:

Expectativas

Hoje vou-lhe apresentar uma aplicação sobre livros. O que lhe vem à cabeça? O que imagina que essa aplicação faz?

Literacia digital

- Possui um smartphone?
  Se sim, qual o sistema operativo?
  Há quanto tempo (meses / anos)?
  Já teve outro smartphone com outro sistema operativo? Qual?
  O seu tarifário inclui dados móveis?
- Utiliza alguma rede social para efeitos lúdicos no smartphone?
  Se sim, qual?
  Há quanto tempo (meses / anos)?
  Com que frequência (diária / semanal / mensal / anual)? Quantas horas?
  O que mais gosta nessa rede social?
  O que mais o aborrece nessa rede social?
- No caso de ter Facebook e Instagram, utiliza mais o Facebook ou o Instagram? Porquê?

Pensando nos últimos meses 12 meses, acha que tem usado o Facebook com cada vez mais ou menos frequência? Porquê?

Hábitos de leitura lúdica

- Faz uma leitura lúdica de livros?
- Se sim, com que frequência (diária / semanal / mensal / anual)? Quantos? Quer deixar um justificação para a sua resposta?
- Quando escolhe um livro para leitura lúdica, quais são os critérios/fatores que tem em consideração?
- Está ligado a algum tipo de atividade lúdica ou profissional que esteja relacionada com livros? Qual?
- Com que frequência vai a uma biblioteca requisitar um livro (semanal / mensal / anual)? Quer deixar um justificação para a sua resposta?
- Tem alguma forma de registar/catalogar/guardar informação dos livros que lê, os que já leu e/ou os que possui? Como?

Se não, sente a necessidade desse registo/catalogação/arquivo?

- Em que proporção recorre a livros impressos ou ebooks para leitura lúdica (percentagem)?
  Havendo diferença de frequência entre os dois tipos, qual o motivo?

- No futuro, qual é que acha que será o formato preferencial de um livro para efeitos de leitura lúdica? Porquê?

Hábitos de partilha de livros

- Costuma <u>emprestar</u> ou receber emprestado livros de amigos, família ou outras pessoas? Com que frequência (semanal / mensal / anual)?
  Tem alguma dificuldade na gestão dos empréstimos?

  Já lhe ocorreram situações de não saber a quem emprestou ou quem lhe emprestou?
- Já fez empréstimo de livros, e para o efeito procurou alguma ferramenta digital para tal? Qual?

- Tem livros em casa que nunca leu?
- Tem livros em casa que gostaria de <u>trocar</u> por outros livros com outras pessoas?
- Já trocou livros com outras pessoas? Como?

- Já vendeu ou comprou livros a outras pessoas? Como?
- Já fez empréstimo/troca/venda de livros, e para o efeito procurou alguma ferramenta digital para tal? Qual?

  Se sim, o uso dessa ferramenta correu bem? Porquê? Se algo correu mal, porquê?

  Se não, utilizaria uma plataforma desse tipo para vender ou comprar um livro? Quais as preocupações? Confiaria num desconhecido para lhe enviar um livro por correio?

Hábitos sociais de leitura lúdica

- Costuma comentar sozinho ou com outras pessoas os livros que lê ludicamente?

- Já publicou alguma coisa sobre alguma leitura lúdica? Onde?

- Usa alguma rede social relacionada com livros? Qual?

- Conhece a aplicação Goodreads?
  Se sim, já utilizou? Já comentou algum livro lá? O que mais gosta desta aplicação (rede social)? O que mais o incomoda?

Anexo III: Guião do cenário e tarefas pretendidas

As tarefas pretendidas estão numeradas ao longo do cenário descrito.

Cenário

O Pedro, amigo da Ana, falou-lhe da última vez que estiveram juntos de um livro que leu recentemente. Mais tarde a Ana lembra-se da conversa que tiveram e decide procurar por mais informação sobre o livro na app Desvaneio. Pesquisa pelo título - Memorial do Convento [1] - e encontra. Lê o resumo [2] e acha interessante, pelo que decide guardar a obra com o marcador "Para ler" [3].

Repara entretanto que existe uma secção da loja online onde pode encontrar exemplares de livros que outros utilizadores disponibilizam para vender, emprestar, doar ou trocar [4]. Como não encontra lá nenhuma proposta, decide criar uma para comprar o livro [5]. A proposta é pública, no valor de 5€. Ana coloca a sua localização atual e, depois de verificar que tudo está como pretendido, confirma a criação da proposta. Uns minutos depois, outro utilizador chamada João candidata-se a essa proposta. Ana aceita [6]. Combinam a hora e o local por mensagem. Ana envia uma mensagem a João dizendo "Estamos combinados?" [7] e ele responde afirmativamente. Mais tarde encontram-se para realizar a compra. Quando se encontram, os dois confirmam a entrega na aplicação [8]. Despedem-se, seguindo cada um o seu caminho. Mais tarde, Ana classifica positivamente a troca [9], escrevendo "Livro em bom estado", e contribuindo assim para o sistema de classificação, que ajuda a manter um clima de confiança entre leitores na troca dos seus livros.

Já durante a leitura, Ana identifica-se bastante com uma passagem inspiradora do livro - "Podemos fugir de tudo, não de nós próprios.". Decide partilhá-la com os outros utilizadores com quem está conectada. Para tal, cria uma nova publicação, escolhendo uma imagem artística da galeria de fotos que serve de pano de fundo à citação [10]. Depois de a tornar visível para as suas conexões, publica-a. A citação fica associada ao livro, na app, pois não há pior coisa que uma citação perdida algures sem referência ao autor.

Tarefas

1. Pesquisa pelo livro
    - Vai ao menu de pesquisa
    - Escreve "Memorial do Convento"
    - Escolhe o resultado pretendido
2. Encontra o resumo
    - Navega para o separador "Detalhes"
3. Marca "Para ler"
    - No header, vai a "Adicionar"
    - Escolhe a opção "Para ler"
4. Encontra loja
    - Navega para o separador "Loja"
5. Cria proposta
    - Pressiona o botão circular ao fundo
    - Escolhe a opção "Comprar"
    - Avança, no botão "Seguinte"
    - Avança, no botão "Seguinte"
    - Coloca o preço de 5€
    - Avança, no botão "Seguinte"
    - Vai ao botão "Escolher lugar..."

- Escolhe “Localização atual”
    - Avança, no botão “Seguinte”
    - Confirma criação, no botão “Confirmar”
6. Aceita candidatura
    - Pressiona o botão “Aceitar”
7. Combina local e hora
    - Pressiona o botão de chat
    - Pressiona a caixa de texto para o efeito
    - Escreve “Estamos combinados?”
    - Pressiona o botão de enviar
8. Confirma transação
    - Pressiona o botão “Confirmar”
9. Deixa opinião
    - Pressiona o botão “Deixar opinião”
    - Escolhe a figura com um sorriso
    - Escreve “*Livro em bom estado*”
    - Pressiona o botão “Deixar opinião”
10. Publica citação
    - Vai à secção “Feed”
    - Pressiona o botão circular ao fundo
    - Escolhe a opção “Citar”
    - Vai a “Escolher livro”
    - Escolhe o primeiro resultado (já está nas pesquisas recentes)
    - Avança, no botão “Seguinte”
    - Escreve “*Podemos fugir de tudo, não de nós próprios.”*
    - Avança, no botão “Seguinte”
    - Escolhe o icon da galeria
    - Escolhe uma imagem
    - Avança, no botão “Seguinte”
    - Avança, no botão “Seguinte”
    - Publica, no botão “Publicar”

Anexo IV: Folha de registo durante a experiência

Tarefa	Sucesso (Sim, +/-, Não)	Início e fim (hh:mm)	Comentários do participante (P) Notas do investigador (I)
1 Pesquisa pelo livro		___ : ___ ___ : ___	
2 Encontra o resumo		___ : ___ ___ : ___	
3 Marca “Para ler”		___ : ___ ___ : ___	
4 Encontra loja		___ : ___ ___ : ___	
5 Cria proposta		___ : ___ ___ : ___	
6 Aceita candidatura		___ : ___ ___ : ___	
7 Combina local e hora		___ : ___ ___ : ___	
8 Confirma transação		___ : ___ ___ : ___	
9 Deixa opinião		___ : ___ ___ : ___	
10 Publica citação		___ : ___ ___ : ___	

Para cada tarefa, que está numerada, o investigador aponta:

- O grau de sucesso na sua execução. “Sim” se concluiu com ajuda, “+/-” se precisou de alguma sugestão para concluir, “Não” se precisou de instruções explícitas de como proceder.
- O momento em que se iniciou e foi concluída. Seria possível inferir o momento em que se iniciou com base nos tempos de conclusão das restantes tarefas, mas poderá acontecer haver pausas entre tarefas, caso o participante ou investigador sintam necessidade de discutir algum detalhe.
- Comentários mais significativos que o participante faça durante o processo de *think-aloud* ou qualquer outra nota, como por exemplo algum detalhe que seja relevante confrontar com os dados de eye tracking no momento da análise posterior à experiência. Por exemplo, no momento de escolher uma imagem para uma publicação, se o participante demorou a escolher entre as duas opções disponíveis para o efeito (tirar fotografia ou escolher na galeria), será importante perceber se o seu olhar esteve entre essas duas opções (dando a entender que esteve indeciso) ou se na verdade esteve perdido a olhar para outro detalhe na interface (dando a entender que não sabia o que fazer ou que não encontrava o sítio onde se escolhe uma imagem).

Anexo V: Guia do questionário pós-experiência

O que mais gostou nesta experiência? Porquê?

O que é que menos gostou? Porquê?

O que é que acha que devo acrescentar/alterar/melhorar na aplicação que acabou de usar?

Quem imagina a usar esta aplicação?

Utilizaria esta aplicação? Que funcionalidades? Porquê?

Anexo VI: Questionário relativo à usabilidade

Relativamente à experiência que acabou de realizar, por favor assinale o grau de concordância, de 1 (discordo totalmente) a 5 (concordo totalmente), em cada uma das afirmações seguintes.

	1 Discordo totalmente	2 Discordo parcialmente	3 Nem discordo nem concordo	4 Concordo parcialmente	5 Concordo totalmente
É fácil orientar-me na app.					
Encontro facilmente o que procuro na app.					
A app é lenta.					
A app é agradável de utilizar.					
A app tem algumas características irritantes.					
Existe consistência na disposição e nos conteúdos apresentados.					
Sinto necessidade de ajuda em algumas funcionalidades.					
A utilização requer conhecimentos mais aprofundados ou experiência anterior.					
O tamanho dos caracteres no ecrã torna-os fáceis de ler.					
A informação mais importante possui um bom destaque.					
A quantidade de informação apresentada por ecrã é adequada.					
A disposição da informação apresentada por ecrã é adequada.					
Os ícones apresentados são intuitivos.					
O aspeto gráfico é atrativo.					
É fácil navegar na app.					

universidade de aveiro theoria poiesis praxis

DECLARAÇÃO

Declaramos que todo o processo de concepção e design participativo inerente a este trabalho de mestrado intitulado, "***Concepção e desenvolvimento participativo de uma mobile app para a mediação da leitura de livros***", processa e manipula dados e/ou informações dos participantes de forma integralmente anonima. Garante desta forma total confidencialidade dos elementos recolhidos no decurso da investigação e a sua destruição no final da mesma.

Universidade de Aveiro, 13 Fevereiro de 2019

Diogo Gomes

Diogo Gomes
(Professor Auxiliar, IR/Orientador)

[illegible]

Óscar Mealha
(Professor Associado com Agregação, co-IR/co-Orientador)

Leonardo Oliveira

Leonardo Oliveira
(Estudante Mestrado em Engenharia Informática)

universidade de aveiro

deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

Declaração

Para os devidos efeitos, e na qualidade de Diretor do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, declaro que tenho conhecimento e apoio *o* projecto de dissertação de mestrado em Eng. Informática de "Concepção e Desenvolvimento participativo de uma mobile App, para a mediação da leitura de livros", a realizar sob a orientação do Prof. Doutor Diogo Nuno Pereira Gomes (DETI/UA) e do Prof. Óscar Emanuel Chaves Mealha (DECA/UA).

Universidade de Aveiro, 13 de Fevereiro de 2019

[illegible]

UNIVERSIDADE DE AVEIRO
Departamento de Electrónica
Telecomunicações e Informática
3810-193 AVEIRO

Prof. Dr. Nuno Borges de Carvalho
Director do DETI

universidade de aveiro theoria poiesis praxis

DECLARAÇÃO

Este projeto de Mestrado em Engenharia Informática do estudante Leonardo Oliveira, com orientação conjunta do Diogo Gomes (DETI/IT) e Óscar Mealha (DeCA/DigiMedia), intitulado ***Concepção e desenvolvimento participativo de uma mobile app para a mediação da leitura de livros***" contribui para a área estratégica de interação e media digital, neste caso para mediar o contexto de leitura de livros. Trata-se um projeto inovador no contexto da engenharia informática e na área de ciências e tecnologias da comunicação e é nutrido cientificamente de forma transdisciplinar pela participação direta destas duas grandes áreas disciplinares, sem prejuízo para a contaminação que tem recebido, detalhadamente de outras. O projeto reune todas as condições para ter o apoio e enquadramento científico-pedagógico do DeCA.

Universidade de Aveiro, 13 Fevereiro de 2019

[illegible]

Professor Doutor Rui Raposo

Professor Auxiliar do Departamento de Comunicação e Arte

Diretor do DeCA

Appendix B

This appendix contains the provisional version of the article entitled *How to deal with mindless scrolling?*.

How to deal with mindless scrolling?

Leonardo Oliveira (DETI), Mário Vairinhos (DeCA / DigiMedia), Óscar Mealha (DeCA / DigiMedia), and Diogo Gomes (DETI / IT)

Abstract—People make use of smartphones like never before. Everyday, we are overwhelmed with enormous amounts of information, most of the time, shown as an endless list of items, whether in social networks or news portals. This can lead us to lose the focus of what is relevant to read, and we might end up ”mindless scrolling”, i.e. scrolling the content without a focused attention to it. In this study we compare a set of representative apps, seeking for solutions that might help dealing with mindless scrolling. This study also focuses on a particular aspect of how users navigate from publication to publication, when scrolling, a situation that suggested testing two different interface proposals (A and B). Interface A was prepared with a regular/linear scrolling list of items and interface B was prepared with a list of items that only scrolls one item on screen at a time. The results show an increase of the time spent in each publication of 89% from interface A to B, which indicates a clear evidence of improvement on the user's focus. This may be an effective media interaction and interface design principle to deal with mindless scrolling. Future work is being planned to confirm with significance these findings in a controlled experiment with a real app, deployed in the Google Play Store with hundreds of users.

Index Terms—mindless scrolling, UX, Facebook, infinite scrolling, feeds, lists, scroll, swipe.

1 Introduction

SMARTPHONES are rapidly spreading worldwide. By the end of 2018, 66% of individuals[1] owned one [3]. Going back to 2012, for example in Portugal, only 18% of the Portuguese population had one. This number raised to 67% five years later, in 2017 [4]. Although the adoption is predominant in people aged less than 25, the trend is followed by all the age groups. Even for people aged 55+, in the same period (2012-2017), the adoption rose from 6% to 38%. A large part of the population, and nearly everyone in a not so distant future, take their phones everywhere and are permanently connected. Most of the time with a smartphone is dedicated to social networking [1]. People access social networks and news portals, being exposed to a huge amount of information. All of this in lists, apparently infinite, scrolled continuously from one content to another. The way of consuming that information can be tedious sometimes, and people might end up mindless scrolling, ie. scrolling the content without paying attention to it. This can be a significative point in the user experience, because the user will not retain the information he is exposed to, and may feel it was a waste of time. If we think of any social network (eg. Facebook, Instagram or Twitter), lists are always there ready to be scrolled. For many years this has been the standard solution to show timelines of contents, and before smartphones it already was like this when using computers, with the only difference being the use of the mouse to scroll instead of a finger.

Social networks are widely investigated in several studies, with focus on aspects like the way people interact and the time they spend, sometimes with negative consequences, resulting in addictions [8] [6] or distractions that penalize their productivity [9]. With that in mind, the phenomenon of mindless scrolling was the starting point for this work, that investigates alternatives to navigate over contents, looking to improve the way people focus on each content and remember it. With that goal in mind, after this introduction, the problem is further contextualized in Section 2, followed by the research procedure, in Section 3. Next, Section 4 reports on a comparison study of several mobile apps to better identify the mindless scroll issue. This worked as a preliminary study phase to prepare an A/B test with 14 participants on two different versions of a mobile prototype. The results of the tests are presented and discussed in Section 5. The document ends with the conclusions and future work, in Section 6.

1. Estimated from an analysis to 52 countries (including Portugal), representing 65% of the world's population.

2 Background

The motivation for this work emerged from an earlier study developed in the context of the participatory design of an app with a social network targeted to people willing to share their reading experiences. The study was performed in the last quarter of 2018, with 24 participants aged 19 to 56 years, 14 women and 10 men, coming from a convenience sample of people who declared to read at least two books a year as an hobby. 22 of them had a smartphone for at least three years, 2 using iOS and 20 using Android. All the participants made use of at least one social network. The study consisted of individual contextual interviews [10], that were semi-structured (with questions previously prepared), and open to the flow of the conversation. These interviews allowed to anticipate the needs users would have when using the app with the social network to share their reading experiences. Among other questions, participants were asked about the social network they used the most: i) *What do you appreciate the most in this social network?*, ii) *What most annoys you in this social network?*, iii) *Considering the last 12 months, has there been any change in how often you use this social network? If yes, why?* With these open questions, the participants had the possibility to freely express their opinions, pointing out one or more aspects. The positive aspects were: being in contact

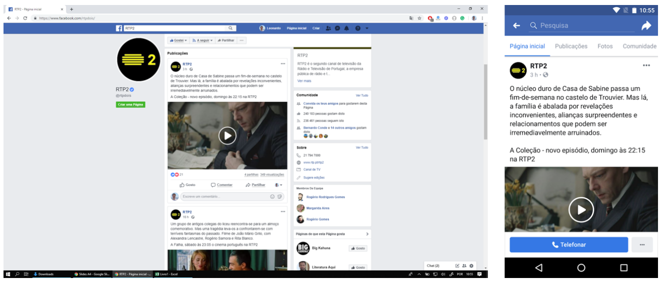
/ chat (13 participants); the type of content, such as images, comics or memes (8); news (6); a source of inspiration / a way to discover interesting content (4); a good place to share ideas, photos or music with other people (3); the feeling of control of the content and users the participant was following (3); a tool for work or school (2); a good way to spend time (2); events (2). The negative aspects were: the other users, like people with inappropriate comments or people that share things not relevant to the participants (8); the content, such as uninteresting topics, fake news or clickbaits (6); too much information / spam (5); advertising (4); the way the content is accessed (4), eg. the fact that only a photo is shown at a time when a gallery is visualized on Instagram; an ecosystem different from the reality / unrealistic (3); a place for procrastination (2). Note that the positive and negative aspects were referent to the social network each participant used the most. 15 participants referred to Facebook and 9 referred to Instagram. Some of the aspects were characteristic of one social network. For example, the feeling of control of the content and users the participant was following was pointed out from the participants who chose Instagram, and a tool for work or school was pointed out from the participants who chose Facebook. If looking at these opinions from a broader perspective, we can observe that people value the content, the way they access it and the way they control it.

The reduced screen size in smartphones is challenging when designing interfaces. There, the importance of the content leads to different approaches that aim to improve in efficiency and satisfaction the way the content is displayed and controlled, and how the navigation happens, which is the case of the publications in a feed of a social network. [13] studied the impact of controls to adjust the prioritization of the users feeds, confirming that they would feel more satisfied with the content when being able to use the controls, whether they worked or not. The placebo effect would be compensated with improvised solutions, like scrolling faster to reach the relevant content. With a special focus on the navigation aspect, [7] explored the power of scrollable grids in improving navigation and visual search, while [14] compared two alternatives to switch between open web pages in a browser: i) horizontal swiping pages and ii) vertical stacked views, that were scrolled, introduced by the browser Google Chrome. In a scenario where the navigation is inside documents, [12] compared flick and ring, two different scrolling techniques. It is not possible to think about documents and navigating without relating to the real world. In the real world documents are not scrolled. Instead, there are pages, that are flipped. This affordance of paper document is not generally used in the digital world. [11] proposed Flipper, a digital navigation technique to read documents based on the physical document flipping.

This inconsistency from physical document flipping to digital document scrolling is a path to be explored in the scenario of publications in feeds. The contents in apps of news or social networks are usually displayed through lists. People scroll these lists to go from content to content. These contents result from huge amounts of information, and can be presented in apparently infinite lists. People do not have the time to check everything, and not everything interests to people, so they still have to go through these lists to filter what is relevant to them. It is not difficult to end up distracted or frustrated, scrolling without retaining the information they are exposed to, while losing information that would be interesting to look deeper. The term *mindless scrolling* will be used in this manuscript to describe this phenomenon. When mindless scrolling, people might miss content that would be interesting to them. Their experience while mindless scrolling might also be affected with a feelling of waste of time. However, this does not have to be interpreted as something negative. In the study conducted prior to this experiment, mentioned above, some people pointed out that their favorite social network is good to spend their time. Maybe mindless scrolling contributes positively to the engagement the users feel when accessing their social networks. Also, it is reasonable to consider that it is advantageous to companies to maximize the time users are lost through the infinity of posts that are available, while spamming them with advertising [2].

Besides lists, there are other patterns used to organize content in smartphones, such as grids, swimlanes or carousels. Still, lists are the standard way to present sets of contents [5]. Before smartphones, computers already used them, and they existed before computers. We can take the example of phone books or dictionaries. Lists are useful because they can offer a notion of relevance (the most important contents come first), ordering (eg. alphabetical order), flow (we can remember of a content near to another, and while scrolling our eyes' movement naturally goes from one content to another). Let us focus on the idea of flow. In computers this is made possible due to the screen size. Several contents can be displayed at the same time. In smartphones, the screen is smaller, and some contents that need more space to be displayed might need the whole screen, such as posts on social networks (see Fig. 1). For instance, several tweets can be displayed at the same time in Twitter, due to their reduced size, but not in Facebook. In a computer, the user can walk to the next post while still seeing the current post, but in a smartphone, seeing the next post involves completely scrolling out the current one. Maybe, there is also an extra effort to figure out where the post starts and ends. If the user is scrolling from post to post until finding something that interests him, he is repeatedly i) starting scrolling to fit the next post in the screen and ii) briefly stopping to check if it is relevant. Why should the user bother about this repetitive and potentially distracting task of scrolling and stopping, scrolling and stopping, if only a post can be displayed at a time? This is something that might be influenced by the reduced screen size. So, if the screen size is a significant difference from a computer to a smartphone, why is the scrolling mechanism equivalent? It is slightly different, because in a computer we have a mouse, and in a smartphone we use our fingers, but the concept is the same. This is the reason why we are interested in investigating the smartphone's scenario in this work.

We can also put the question of the flow in reverse: when scrolling from post to post, are n't the posts around distracting, even if their position in the list is part of the flow? What if only one post was shown at a time, even if there is room for several in the screen? Maybe the user would be more focused in each post, not being tempted to continue scrolling without giving enough attention to the



RTP2
3 h

O núcleo duro de Casa de Sabine passa um fim-de-semana no castelo de Trouvier. Mas lá, a família é abalada por revelações inconvenientes, alianças surpreendentes e relacionamentos que podem ser irremediavelmente arruinados.

A Coleção - novo episódio, domingo às 22:15 na RTP2

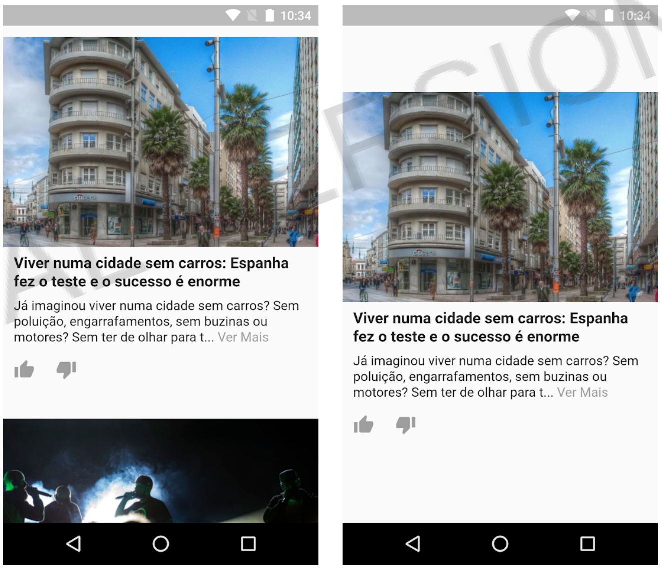
Fig. 1. The same post seen in a computer screen (1920x1080px), on the left, and in a smartphone (1080x1920px), on the right, published on Facebook.

current post. We seek to find answers to these conjectures throughout this work.

3 Methodology

We started by identifying the techniques used in different apps that show feeds to their users, so it was possible to have a comprehensive scenario of related apps and situations. Given the many available apps that have feeds, we choose some representative ones. The analysis was performed on apps with feeds, with a special focus on social networks and news portals: Facebook, Instagram, Twitter, Pinterest and 9GAG. Other apps like Reddit, Tumblr, Snapchat, Linkedin, YouTube or Flickr were also considered, but were left out when selecting a representative group. The analysis was done according to their versions on Google Play Store on February 2019. These applications have a standard interface across Android and iOS, so the characteristics pointed out for the Android versions also apply to the iOS ones. The interfaces of the apps were inspected according to several criteria that were considered as having influence in the way users successfully consume their feeds, retaining the information they are exposed to. This analysis can be found in Section 4.

The insights from the apps' inspection opened the possibilities for further investigation considering the aspects that might influence mindless scrolling. We decided to focus on the characteristics of the navigation itself. We prepared two versions of an interface that displays a list of 15 posts, to make an A/B test with participants. The posts were news from NiT, a portuguese portal with publications about culture and lifestyle. We choose NiT to increase the chances of compatibility of the publications with the participants' interests. Specific topics, like politics or sports, would be a valid choice too, but would require to find participants with these interests. The target user for that interface would be anyone making use of their free time to read some soft news, like they would do on their preferred social network. The interface was kept minimalistic. The two versions had in common an area with a set of posts, with no more menus, sections or headers. Each post had a picture, a title with two lines, a summary of three lines and two buttons to like or dislike. The summary could be expanded when tapped, showing the complete content, with a length of 800 to 1300 characters, excluding spaces. Some posts were reduced from the original publication on NiT to fit in the limit of 1300 characters. When collapsed (only showing the three lines' summary), all the posts had the same height. When expanded, the interface was kept in the same view, with the height of the post being increased to show the complete content. The news were selected with the intention to be distinct, to avoid repeating themes. The effort to standardize the structure of the posts was to fairly allow comparing them when analyzing the results of the tests. The difference between the two interfaces (A and B) were in the navigation between the posts. The interface A presented the typical list, with the usual scrolling capabilities. The interface B presented a sort of a vertical slideshow, with a post in each slide (see Fig. 2). A slide would fit the entire screen, so only a post could be seen at each time. To see the next or previous post, the user would swipe up or down, respectively. The main differences from version A to B are: i) only shows one post at a time; ii) each scroll action[2] only moves forward or backward one post at a time; iii) as a consequence of the two previous ones, when swapping, the post is automatically placed in the screen, with no need to control when to stop swiping.



Viver numa cidade sem carros: Espanha fez o teste e o sucesso é enorme

Já imaginou viver numa cidade sem carros? Sem poluição, engarrafamentos, sem buzinas ou motores? Sem ter de olhar para t... Ver Mais

Viver numa cidade sem carros: Espanha fez o teste e o sucesso é enorme

Já imaginou viver numa cidade sem carros? Sem poluição, engarrafamentos, sem buzinas ou motores? Sem ter de olhar para t... Ver Mais

Fig. 2. The interfaces of versions A, on the left, and B, on the right.

The two versions of the interface were submitted to tests with volunteer participants. Each version was tested by 7 participants, 4 females, 3 males, aged 19-24, in a total of 14 participants. All the participants were students at the University of Aveiro and realized the test in a silent place on the facilities, where they would feel comfortable (the library or a study room). In each test, the participant was given a Google Nexus 5, running Android 6, with the version of the app correspondent to the test. The participant was invited to check out the feed until reaching the last publication. After, the participant was asked to estimate the number of publications that were in the feed and, after, to recall the topics they remembered. Also, the screen was recorded during the time the participant was reading, using an app called DU Recorder, so it was possible to track the time each

2. Consider a scroll action as the complete movement from the moment the user puts the finger down, to the moment the finger leaves the screen.

topic was being read and the interactions with the posts to read more, like it or dislike it. With the metrics from the recall and time spent we expected to find differences in the way of navigating in the two interfaces, related to the susceptibility to mindless scrolling. The results and discussion can be found in Section 5.

4 Analysis of Existing Apps

Taking as example some of the existing apps that are representative of the possibilities to show and interact with feeds, we could have a deeper understanding of the generality of the solutions being used. A special focus was given to the examples of Facebook, Instagram, Twitter, Pinterest, Flipboard and 9GAG. The aspects taken into consideration were related to the way the content is accessed, in an attempt to match them with the needs manifested by the users in the study described in Section 2.

4.1 Following topics and marking preferences

The control over the content that is displayed to the user, ie. the ability to see what they want to see and ignore what they do not want to see, is key to their satisfaction when accessing the feeds in these type of apps. Because it is not possible to know if we are really interested in something until we see it, generally there are mechanisms to allow users to show their preferences, so the content is suggested according to that. Facebook, Instagram, Twitter, Pinterest and Flipboard allow users to follow other accounts (people, pages, organizations, ...), meaning that their publications are grouped and displayed to the users. Instagram, Pinterest and Flipboard go a bit further, also allowing to follow topics, which allows users to access their interests independently of who is publishing them. Facebook, Instagram, Twitter, Pinterest and Flipboard have options to hide publications, a mechanism that they use to show less publications similar to the hidden ones. Also, Instagram lets the user mark a preference to show more publications similar to a specific publication. All of the apps allow to report on inappropriate content.

4.2 Feed personalization

Users can also personalize their feeds in each utilization. Facebook and Twitter let the user to show the popular publications or the most recent ones first. Still, when choosing the later option, the content is somehow pre-processed, so the publications are not shown in chronological order. Also, in Facebook the option to show the more recent ones first is not easy to find, and most of the users will not be aware of that possibility.

4.3 Feed dynamics

Let us consider an ad-hoc nomenclature to characterize three types of feeds according to the way they change over time: i) the dynamic ones, ii) the pseudo-static ones and iii) the static ones. The dynamic ones change every time users refresh it, the static ones keep the same order of the publications over time and the pseudo-static ones seem to be static but the publications might change their places in each refresh. This might influence the notion the users have of the system. Perhaps a dynamic feed seems to be infinite, with a lot of information, and at some point users cannot control what they already saw or not. Considering that nomenclature, Facebook and Twitter are dynamic, Instagram, Flipboard and Pinterest are pseudo-static and 9GAG is static. Instagram has a feature that helps the user to control the publications already visited. It shows a separator to inform when the user has seen all the posts from the last two days.

4.4 Levels of detail

In general, the apps seem to have two levels of detail, resulting in two feed sections: one with the publications the user is following and other with other topics that might be of interest to the user, in a section where the user can explore it. In Twitter it is the section *trends*, in Instagram it is the section *explore*, in Pinterest it is the *home* and in 9GAG it is the section *fresh*. When users have less time, they may only check the first section, and when they have more time available, they may explore the suggestions in the second section.

4.5 Filtering

Filtering the information if also important. All of the apps allow to do that in some way. Facebook, Instagram and Twitter have hashtags, that are keywords optionally marked in each publication about a topic. Users can list the publications with a particular hashtag. Instagram, Pinterest, Flipboard and 9GAG also place the publications in wider categories, allowing to filter publications by generic topics.

4.6 Retrieval

It is not unusual to find an interesting publication while visiting a feed and days after, in a tentative to retrieve that publication, it is hard or nearly impossible to find it again. This commonly happens because of the enormous amount of publications available. Facebook and Twitter allow text search for publications. Also, Facebook, Instagram, Twitter, Pinterest and Flipboard allow to save publications. Later, users can find them in their list of saved publications.

4.7 Navigation

About the way users navigate from publication to publication in their feeds, there are some different solutions. The most common way is a scrollable list of publications, which is the example of Facebook, Instagram, Twitter, Pinterest and 9GAG. Also, in their sections to explore more content, Instagram and Pinterest show a grid of publications with their images, in a compact way. Users have more publications in the screen at a time, and can scroll through the publications faster. This might be more engaging for the users, because it is faster, more dynamic, and more visual, but goes in the opposite direction of what is being studied in this work, that tries to find ways to deal with mindless scrolling. On the other hand, Flipboard presents only a publication at a time, forcing the user to a new scroll action each time they want to go to the next publication. The same concept is used in some sections of Facebook, when showing news in publications. In these sections, there is a sort of slideshow, and the user slides left or right when moving to the next or previous publication. This allows the user to focus on a publication at a time. Flipboard follows the convention of keeping the scrolling in the vertical direction, while these sections of Facebook are horizontal, which forces the explicit indication of the direction of the scroll with a signifier, a *read more* button to move to the next publication. Another good example of that focus is Tinder[3]. Tinder is a dating app, allowing its users to find nearby people. There, they make use of a pile of cards, where each card corresponds to a person's profile. If the user is interested, swipes right, if not, swipes left. Then, the card immediately below shows up. Following this workflow, the user is focused in one person at a time, which is considerably different from scrolling a list. The case of Flipboard is equivalent, but with news instead of user profiles.

These type of approaches, that seem to go beyond the traditional scrolling lists, are interesting examples of solutions that might considerably influence mindless scrolling. This is in line with the different aspects analysed in this section across these apps, that might be influencing mindless scrolling. Several other aspects may influence the way the user is focused when visualizing a feed, not only related to the interface but also psychological, like how relaxed, tired or frustrated the user is. In order to limit the scope of this work, navigation between publications attains a special focus, which is the essence of the experiment described in Section 3 and discussed in Section 5, where the results are explored.

5 Results and Discussion

The results from the A/B test were extracted according to the described in Section 3. The values were grouped by the correspondent versions of the experience (version A, with a regular scroll list, and version B, with a list that only scrolls one publication at a time). On average, each participant took 1min41s to check the feed on version A and 3min35s on version B, which corresponds to 6.8s and 14.3s by publication, respectively. This corresponds to an increase of 110% in the time spent in each of the 15 publications (from now on we will call it fixation time), from version A to B. The distribution of the average fixation times is depicted in Fig. 3. If we ignore the first three publications (in a tentative to cancel any bias when adapting to the interface at the beginning of the test, and the last publication, to cancel any bias by the conscience of the participant that the end of the feed was reached), the average fixation time for the remaining 11 publications is 7.1s for version A and 13.4 for version B. This way, the increase slightly dropped to 89%. We consider that the accounting of the 11 publications is fairer than the total, so we will continue this analysis using this subset. The average number of expanded publications in each experience (the participant tapped on *read more*) was 4.3 out of 11 (version A) and 5.4 (version B). This means that, on average each participant read 1.1 more publications on

3. This app is not in the set of apps under analysis because it does not follow the concept of reading publications, like in a news portal or other conventional social network.

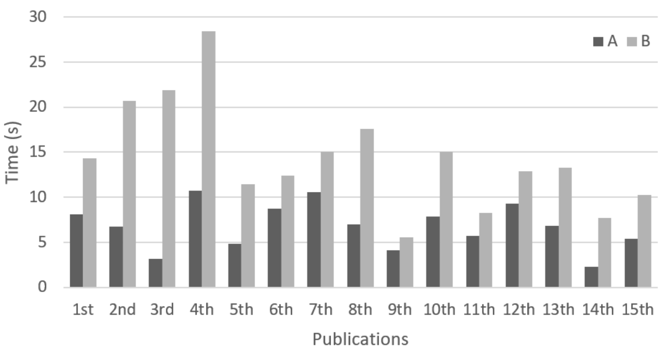


Fig. 3. The average fixation times for each version.

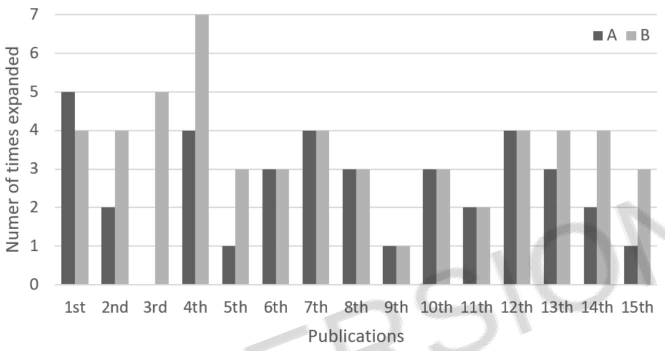


Fig. 4. The number of times each publication was expanded in the total of the 7 experiments for each version.

version B than on version A. Fig. 4 shows the distribution of the number of times each publication was expanded in the total of the 7 experiments for each version (please note that the number of times a publication was expanded, as shown in Fig. 4, is different from the number of publications a participant expanded, corresponding to the averages 4.3 and 5.4).

Considering only the publications expanded by the participants, the fixation times were different. The averages were 14.0s (A) and 21.4s (B), resulting in a difference of 53% from A to B. This indicates that, when having interest for a particular new, the gap between the two versions was lower. Still, the tendency for higher fixation times in version B persisted.

When the participants were asked to estimate the number of publications they went through, on average they diverged from the real number (15 publications) for 3.1 (A) and 2.0 (B), while the standard deviation for the guesses was 3.9 (A) and 3.6 (B). When they were asked to recall the topics of the publications, on average they remembered 6.1 (A) and 6.9 (B). This proximity indicates that, even though the fixation time was considerably different in the two versions, the participants were still conscious of what they were reading.

About the usage of the buttons to show reactions (like or dislike), no indication was given to the participants, to avoid influencing their behavior. Three participants asked if they were supposed to use them and five of them made use of it. Given the scarcity of data, we opted for not analysing this component.

When asked how interesting was the content of the publications, in a scale from 1 (nothing interesting) to 5 (totally interesting), the averages were 4.0 (A) and 4.3 (B). These scores indicate that the participants considered the publications interesting, which is a good indicator. A lack of a match of the participants to their interests would bias their reading experience.

When asked how pleasant it was to use the interface, in a scale from 1 (nothing pleasant) to 5 (totally pleasant), the averages were 4.3 (A) and 4.1 (B). There seems to be no significative difference in the satisfaction between the two interfaces. Even so, at the end of each experiment, when the other interface was showed to the participant, the preferences were to version A. 7 participants preferred the interface A, 4 preferred the interface B and 3 did not manifest a preference. Some of the reasons for A were the familiarity and the ability to discover the next publication. Some of the reasons for B were the way it allows to focus on each publication and the way it helps distinguishing the start and the end of the publications.

The differences in the results for each version of the interface are significative. In the perspective of the fixation time, version B indicates a tendency to keep the readers more focused. Despite the huge differences in the average fixation times, the value of the differences is not the most important, but the tendency it represents. Interface B seems to have positive effects in controlling mindless scrolling. The participants remembered the topics of the publications similarly in both versions, so their attention was not affected. Still, their focus was different, both for all publications (increase of 89%) and for the publications they were interested and decided to expand (increase of 53%). This meets the comments of some participants, that said they would feel more tempted to go to the next publication before being done with the current publication if is was already visible in the screen, which happened with version A.

6 Conclusions and Future Work

Several aspects might influence the way we are focused when accessing a feed. In this work we explored some differences in the navigation between publications, performing an A/B test with two different versions of an interface. Users recalled in the same way the topics of the publications they crossed by in the tested interfaces, so the perception that they had of the publications was almost unaffected. Still, in an interface where the user is forced to stop in each publication and where there are no publications showing around, the results indicate that the user is less tempted to mindless scrolling, with a considerable increase of the fixation times (in this experiment it was 89%).

The results are a good indicator that the differences in the tested versions have influence in mindless scrolling. However, this pilot study reveals promising results that should be thoroughly tested with a more significant sample. As future work, these two versions will be adapted to a real app, prepared to support A/B testing, where several metrics will be compared to assess the focus, like the fixation time, and also the engagement, including the number of likes / comments, proportion of likes / comments per posts visualized or the number of visits to the app during a period of days.

References

[1] Distribution of time spent on mobile apps in the United States. https://www.statista.com/statistics/319652/us-mobile-app-time-distribution-category/, 2017. [Online; accessed 19-February-2019].

[2] Facebook Advertising Platform: The money making engine behind Facebook's growth. https://digit.hbs.org/submission/facebook-advertising-platform-the-money-making-engine-behind-facebooks-growth/, 2017. [Online; accessed 21-February-2019].

[3] Smartphone penetration to reach 66% in 2018. https://www.zenithmedia.com/smartphone-penetration-reach-66-2018/, 2017. [Online; accessed 20-February-2019].

[4] Consumer Barometer with Google. https://www.consumerbarometer.com/en/, 2018. [Online; accessed 28-November-2018].

[5] Alan Cooper, Robert Reimann, David Cronin, and Christopher Noessel. *About Face: The Essentials of Interaction Design*. Wiley, Indianapolis, IN, 4 edition, 2014.

[6] A. Kaun and F. Stiernstedt. Facebook time: Technological and institutional affordances for media memories. *New Media and Society*, 16(7):1154–1168, 2014.

[7] Ken Pfeuffer and Yang Li. Analysis and modeling of grid performance on touchscreen mobile devices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 288:1–288:12, New York, NY, USA, 2018. ACM.

[8] Aneta Przepiorka and Agata Blachnio. Time perspective in internet and facebook addiction. *Computers in Human Behavior*, 60:13 – 18, 2016.

[9] L. Rosen and A. Gazzaley. *The Distracted Mind: Ancient Brains in a High-Tech World*. Penguin Random House, 2016.

[10] M. Stickdorn and J. Schneider. *This is Service Design Thinking: Basics, Tools, Cases*. BIS Publ., 2012.

[11] Liyang Sun and François Guimbretière. Flipper: A new method of digital document navigation. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 2001–2004, New York, NY, USA, 2005. ACM.

[12] Huawei Tu, Feng Wang, Feng Tian, and Xiangshi Ren. A comparison of flick and ring document scrolling in touch-based mobile phones. In *Proceedings of the 10th Asia Pacific Conference on Computer Human Interaction*, APCHI '12, pages 29–34, New York, NY, USA, 2012. ACM.

[13] Kristen Vaccaro, Dylan Huang, Motahhare Eslami, Christian Sandvig, Kevin Hamilton, and Karrie Karahalios. The illusion of control: Placebo effects of control settings. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 16:1–16:13, New York, NY, USA, 2018. ACM.

[14] Andrew Warr and Ed H. Chi. Swipe vs. scroll: web page switching on mobile browsers. In *In Proc. of CHI2013*, pages 2171–2174, 2013.

References

[1] T. Ahonen, *Mobile as 7th of the Mass Media*. Futuretext, 2008.

[2] *Smartphone penetration to reach 66% in 2018*, https://www.zenithmedia.com/smartphone-penetration-reach-66-2018/, [Online; accessed 28 November 2018], 2017.

[3] *Consumer Barometer with Google*, https://www.consumerbarometer.com/en/, [Online; accessed 28 November 2018], 2018.

[4] M. Stickdorn and J. Schneider, *This is Service Design Thinking: Basics, Tools, Cases*. BIS Publ., 2012, ISBN: 9789063692797.

[5] J. Nielsen, *Usability 101: Introduction to Usability*, https://www.nngroup.com/articles/usability-101-introduction-to-usability/, [Online; accessed 28 November 2018], 2012.

[6] ——, *10 Usability Heuristics for User Interface Design*, https://www.nngroup.com/articles/ten-usability-heuristics/, [Online; accessed 28 November 2018], 1995.

[7] K. Rodden, H. Hutchinson, and X. Fu, «Measuring the user experience on a large scale», in *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, New York, New York, USA: ACM Press, 2010, p. 2395, ISBN: 9781605589299. DOI: `10.1145/1753326.1753687`. [Online]. Available: `http://portal.acm.org/citation.cfm?doid=1753326.1753687`.

[8] I. Au, R. Boardman, R. Jeffries, P. Larvie, A. Pavese, J. Riegelsberger, K. Rodden, and M. Stevens, «User experience at google: focus on the user and all else will follow», in *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, ACM, 2008, pp. 3681–3686.

[9] S. Attfield, G. Kazai, and M. Lalmas, «Towards a science of user engagement (Position Paper)», *WSDM Workshop on User Modelling for Web Applications*, 2011. DOI: `978-1-4503-0493-1/11/02`. [Online]. Available: `http://www.dcs.gla.ac.uk/%7B~%7Dmounia/Papers/engagement.pdf`.

[10] J. Hamari, «Do badges increase user activity? A field experiment on the effects of gamification», *Computers in Human Behavior*, vol. 71, pp. 469–478, 2017, ISSN: 0747-5632. DOI: `10.1016/J.CHB.2015.03.036`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0747563215002265`.

[11] A. Shreve, *Intro to gRPC: A Modern Toolkit for Microservice Communication*, https://www.youtube.com/watch?v=RoXT_Rkg8LA, [Online; accessed 26 April 2019], 2017.

[12] Xplenty, *The SQL vs NoSQL Difference: MySQL vs MongoDB*, https://medium.com/xplenty-blog/the-sql-vs-nosql-difference-mysql-vs-mongodb-32c9980e67b2, [Online; accessed 26 April 2019], 2017.

[13] *Types of NoSQL Databases*, https://www.mongodb.com/scale/types-of-nosql-databases, [Online; accessed 24 May 2019].

[14] M. Keep and A. Cabral, *MongoDB Multi-Document ACID Transactions are GA*, https://www.mongodb.com/blog/post/mongodb-multi-document-acid-transactions-general-availability, [Online; accessed 26 April 2019], 2018.

[15] Y. Li and S. Manoharan, «A performance comparison of SQL and NoSQL databases», in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, 2013, pp. 15–19. DOI: `10.1109/PACRIM.2013.6625441`.

[16] R. Cattell, «Scalable SQL and NoSQL Data Stores», *SIGMOD Rec.*, vol. 39, no. 4, pp. 12–27, 2011, ISSN: 0163-5808. DOI: `10.1145/1978915.1978919`. [Online]. Available: `http://doi.acm.org/10.1145/1978915.1978919`.

[17] *Our Customers | MongoDB*, https://www.mongodb.com/who-uses-mongodb, [Online; accessed 24 May 2019].

[18] S. Mei, *Why You Should Never Use MongoDB*, http://www.sarahmei.com/blog/2013/11/11/why-you-should-never-use-mongodb/, [Online; accessed 26 April 2019], 2013.

[19] B. Jamin, *Reasons Not To Use Firebase*, https://crisp.chat/blog/why-you-should-never-use-firebase-realtime-database/, [Online; accessed 26 April 2019], 2016.

[20] *Mobile Operating System Market Share Worldwide*, http://gs.statcounter.com/os-market-share/mobile, [Online; accessed 25 April 2019], 2019.

[21] R. Nelson, *Global App Revenue Grew 23% in 2018 to More Than $71 Billion on iOS and Google Play*, https://sensortower.com/blog/app-revenue-and-downloads-2018, [Online; accessed 25 April 2019], 2019.

[22] M. Rouse, *Native app*, https://searchsoftwarequality.techtarget.com/definition/native-application-native-app, [Online; accessed 23 May 2019], 2018.

[23] P. Rap, *React Native or Flutter: Which should I choose?*, https://medium.com/flutter-community/react-native-or-flutter-which-should-i-choose-48567ae2e5e1, [Online; accessed 25 April 2019], 2019.

[24] M. Firtman, *Google Play Store now open for Progressive Web Apps*, https://vaadin.com/pwa/learn/pros-and-cons, [Online; accessed 25 April 2019], 2019.

[25] T. Macaulay, *Ten years on: How Netflix completed a historic cloud migration with AWS*, https://www.computerworlduk.com/cloud-computing/how-netflix-moved-cloud-become-global-internet-tv-network-3683479/, [Online; accessed 28 November 2018], 2018.

[26] *A Day in the Life of Messaging Apps: Usage Demographics*, https://www.vertoanalytics.com/a-day-in-the-life-of-messaging-apps/, [Online; accessed 23 November 2018], 2015.

[27] L. Dignan, *Top cloud providers 2018: How AWS, Microsoft, Google, IBM, Oracle, Alibaba stack up*, https://www.zdnet.com/article/top-cloud-providers-2018-how-aws-microsoft-google-ibm-oracle-alibaba-stack-up/, [Online; accessed 13 December 2018], 2018.

[28] *Deploy an End-to-End IoT Application*, https://aws.amazon.com/getting-started/projects/deploy-iot-application/services-costs/, [Online; accessed 13 December 2018], 2018.

[29] S. Watts and M. Raza, *SaaS vs PaaS vs IaaS: What's The Difference and How To Choose*, https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/, [Online; accessed 26 June 2019], 2019.

[30] M. Villamizar, O. Garcés, L. Ochoa, H. Castro, L. Salamanca, M. Verano, R. Casallas, S. Gil, C. Valencia, A. Zambrano, and M. Lang, «Cost comparison of running web applications in the cloud using monolithic, microservice, and AWS Lambda architectures», *Service Oriented Computing and Applications*, vol. 11, no. 2, pp. 233–247, 2017, ISSN: 1863-2386. DOI: `10.1007/s11761-017-0208-y`. [Online]. Available: `http://link.springer.com/10.1007/s11761-017-0208-y`.

[31] L. Chen, «Continuous Delivery: Huge Benefits, but Challenges Too», *IEEE Software*, vol. 32, no. 2, pp. 50–54, 2015. DOI: `10.1109/MS.2015.27`. [Online]. Available: `http://ieeexplore.ieee.org/document/7006384/`.

[32] S. V. Helm, V. Ligon, T. Stovall, and S. Van Riper, «Consumer interpretations of digital ownership in the book market», *Electronic Markets*, vol. 28, no. 2, pp. 177–189, May 2018, ISSN: 1422-8890. DOI: `10.1007/s12525-018-0293-6`. [Online]. Available: `https://doi.org/10.1007/s12525-018-0293-6`.

[33] K. Zickuhr and L. Rainie, *Younger Americans and Public Libraries*, https://www.pewinternet.org/2014/09/10/younger-americans-and-public-libraries/, [Online; accessed 21 May 2019], 2014.

[34] S. Cain, *This article is more than 2 years old Ebook sales continue to fall as younger generations drive appetite for print*, https://www.theguardian.com/books/2017/mar/14/ebook-sales-continue-to-fall-nielsen-survey-uk-book-sales, [Online; accessed 21 May 2019], 2017.

[35] M. Kozlowski, *eBook sales decline by 4.9% in January 2019*, https://goodereader.com/blog/e-book-news/ebook-sales-decline-by-4-9-in-january-2019, [Online; accessed 21 May 2019], 2019.

[36] A. Flood, *'Leading the entertainment pack': UK print book sales rise again*, https://www.theguardian.com/books/2019/jan/03/leading-the-entertainment-pack-uk-print-book-sales-rise-again, [Online; accessed 21 May 2019], 2019.

[37] P. Delgado, C. Vargas, R. Ackerman, and L. Salmerón, «Don't throw away your printed books: A meta-analysis on the effects of reading media on reading comprehension», *Educational Research Review*, vol. 25, pp. 23–38, 2018, ISSN: 1747-938X. DOI: `https://doi.org/10.1016/j.edurev.2018.09.003`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S1747938X18300101`.

[38] S. Farrel, *Open-Ended vs. Closed-Ended Questions in User Research*, https://www.nngroup.com/articles/open-ended-questions/, [Online; accessed 17 April 2019], 2016.

[39] J. Nielsen, *Why You Only Need to Test with 5 Users*, https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/, [Online; accessed 17 April 2019], 2000.

[40] D. A. Norman, *The Design of Everyday Things*. New York, NY, USA: Basic Books, Inc., 2002, ISBN: 9780465067107.

[41] A. Cooper, R. Reimann, D. Cronin, and C. Noessel, *About Face: The Essentials of Interaction Design*, 4th ed. Indianapolis, IN: Wiley, 2014, pp. 508–528, ISBN: 978-1-118-76657-6.

[42] *Who's using React Native?*, https://facebook.github.io/react-native/showcase, [Online; accessed 25 April 2019], 2019.

[43] *Made with Flutter*, https://flutter.dev/showcase, [Online; accessed 25 April 2019], 2019.

[44] N. Cao, *Why I move to Flutter*, https://medium.com/@nhancv/why-i-move-to-flutter-34c4005b96ef, [Online; accessed 25 April 2019], 2018.

[45] *Continuous Delivery using fastlane with Flutter*, https://flutter.dev/docs/deployment/fastlane-cd, [Online; accessed 25 April 2019], 2019.

[46] *Running On Device - React Native*, https://facebook.github.io/react-native/docs/running-on-device, [Online; accessed 25 April 2019], 2019.

[47] *Testing Flutter apps*, https://flutter.dev/docs/testing, [Online; accessed 25 April 2019], 2019.

[48] *Flutter Live - Flutter Announcements and Updates (Livestream)*, https://youtu.be/NQ5HVyqg1Qc?t=9341, [Online; accessed 26 April 2019], 2018.

[49] GeekyAnts, *We Rebuilt a React Native App with Flutter*, https://blog.geekyants.com/we-rebuilt-a-react-native-app-with-flutter-4160f0499a82, [Online; accessed 25 April 2019], 2018.

[50] *What devices and OS versions does Flutter run on?*, https://flutter.dev/docs/resources/faq#what-devices-and-os-versions-does-flutter-run-on, [Online; accessed 24 April 2019], 2019.

[51] *Understand the Activity Lifecycle*, https://developer.android.com/guide/components/activities/activity-lifecycle, [Online; accessed 26 April 2019].

[52] *Use Java 8 language features*, https://developer.android.com/studio/write/java8-support, [Online; accessed 26 April 2019].

[53] B. Agrawal, *How we broke up our Monolithic Django Service into microservices*, https://medium.com/greedygame-media/how-we-broke-up-our-monolithic-django-service-into-microservices-8ad6ff4db9d4, [Online; accessed 26 April 2019], 2017.

[54] *Web Service Efficiency at Instagram with Python*, https://instagram-engineering.com/web-service-efficiency-at-instagram-with-python-4976d078e366, [Online; accessed 26 April 2019], 2016.

[55] *What Powers Instagram: Hundreds of Instances, Dozens of Technologies*, https://instagram-engineering.com/what-powers-instagram-hundreds-of-instances-dozens-of-technologies-adf2e22da2ad, [Online; accessed 26 April 2019], 2011.

[56] *Full text search*, https://docs.djangoproject.com/en/2.2/ref/contrib/postgres/search/, [Online; accessed 26 April 2019].

[57] *Getting Started With Django*, https://cloud.google.com/python/django/, [Online; accessed 13 May 2019].

[58] *How Instances are Managed*, https://cloud.google.com/appengine/docs/standard/python/how-instances-are-managed, [Online; accessed 25 May 2019].

[59] *Google Cloud Storage*, https://cloud.google.com/storage/, [Online; accessed 13 May 2019].

[60] *Cloud Storage Pricing*, https://cloud.google.com/storage/pricing-summary/, [Online; accessed 13 May 2019].

[61] *Cloud SQL Pricing*, https://cloud.google.com/sql/pricing, [Online; accessed 13 May 2019].

[62] *The App Engine Standard Environment*, https://cloud.google.com/appengine/docs/standard/, [Online; accessed 13 May 2019].

[63] A. Pariona, *Which Countries Read The Most?*, https://www.worldatlas.com/articles/the-countries-that-read-the-most.html, [Online; accessed 21 May 2019], 2017.

[64] A. Dogtiev, *App Revenues (2017)*, http://www.businessofapps.com/data/app-revenues/, [Online; accessed 25 April 2019], 2019.

[65] T. Chow, *eCPM – what exactly is that?*, https://adsense.googleblog.com/2006/02/ecpm-what-exactly-is-that.html, [Online; accessed 25 April 2019], 2016.

[66] A. Dogtiev, *Mobile App Advertising Rates (2018)*, http://www.businessofapps.com/ads/research/mobile-app-advertising-cpm-rates/, [Online; accessed 22 May 2019], 2019.

[67] *Cloud SQL for PostgreSQL pricing*, https://cloud.google.com/sql/docs/postgres/pricing, [Online; accessed 17 May 2019], 2019.