

Giulia de Andrade Cardieri

**PWA-EU: Extending the PWA approach for  
promoting End-User Empowerment**

**Sorocaba, SP**

**February 26th, 2019**



Giulia de Andrade Cardieri

# **PWA-EU: Extending the PWA approach for promoting End-User Empowerment**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação (PPGCC-So) da Universidade Federal de São Carlos como parte dos requisitos exigidos para a obtenção do título de Mestre em Ciência da Computação. Linha de pesquisa: Interação Humano-Computador.

Universidade Federal de São Carlos – UFSCar

Centro de Ciências em Gestão e Tecnologia – CCGT

Programa de Pós-Graduação em Ciência da Computação – PPGCC-So

Supervisor: Profa. Dra. Luciana Martinez Zaina

Sorocaba, SP

February 26th, 2019

---

de Andrade Cardieri, Giulia

PWA-EU: Extending the PWA approach for promoting End-User Empowerment/ Giulia de Andrade Cardieri. – 2019.

209 f. : 30 cm.

Dissertação (Mestrado) – Universidade Federal de São Carlos – UFSCar

Centro de Ciências em Gestão e Tecnologia – CCGT

Programa de Pós-Graduação em Ciência da Computação – PPGCC-So.

Supervisor: Profa. Dra. Luciana Martinez Zaina

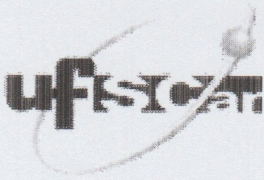
Banca examinadora: Profa. Dra. Luciana Martinez Zaina, Prof. Dr. Vânia Paula de Almeida Neris, Prof. Dr. André Pimenta Freire

Bibliografia

1. PWA. 2. Interface Adaptation. 3. Meta-design. I. End-user Development. I. Mobile Apps II. Universidade Federal de São Carlos. III. PWA-EU: Extending PWAs for End-User Empowerment

---





# UNIVERSIDADE FEDERAL DE SÃO CARLOS

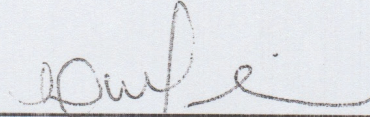
Centro de Ciências em Gestão e Tecnologia  
Programa de Pós-Graduação em Ciência da Computação

---

## Folha de Aprovação

---

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado da candidata Giulia de Andrade Cardieri, realizada em 26/02/2019:

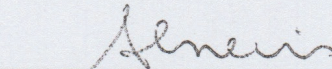


---

Profa. Dra. Luciana Aparecida Martinez Zaina  
UFSCar

---

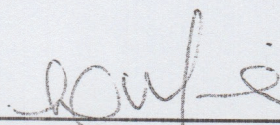
Prof. Dr. André Pimenta Freire  
UFLA



---

Profa. Dra. Vânia Paula de Almeida Neris  
UFSCar

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) André Pimenta Freire, Vânia Paula de Almeida Neris e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ão) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.



---

Profa. Dra. Luciana Aparecida Martinez Zaina

# Acknowledgements

I am very grateful to:

My advisor, Dra. Luciana Zaina for her amazing support, patience, and guidance. You helped me to learn and achieve great things over the last two years.

My parents, for their incredible support, patience, and incentive prior and during my decision to pursue this degree.

My boyfriend, Gabriel, for being with me on the good and bad moments of this journey, and also for being interested in my research without understanding computer science.

My friends, for helping me to keep my focus and motivation during the last years.

My colleagues from the UXLERIS research group for their support and cooperation.

The post-graduate computer science program at UFSCar Sorocaba, for offering the necessary resources to keep this research going.

Everyone who was a volunteer in one of my studies.

My cat, Mila, and my dog, Lola for making me calm while I planned, analyzed and wrote the results from all experimental studies.

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) for the financial support during this research project - Finance Code 001.





# Abstract

Progressive Web App (PWA) is a new approach to the development of mobile applications (apps) which was proposed by Google in 2015. It combines technology resources of both web and native apps. Meta-design is an End-User Development (EUD) approach from which end-users participate actively of a system's design process. Yet PWAs are a recent technology and the impacts of associating EUD and PWAs has been little explored. In this research, we present an investigation about end-users interactions when they act as co-designers on PWA apps. As the traditional PWA approach is limited regarding users acting as co-designers, we propose the PWA-EU approach, an extension of the traditional PWA architecture that includes meta-design concepts. We evaluated our approach concerning both end-users and developers perspectives. With regards to the end-users, we built a mobile app based on the PWA-EU approach and conducted a study in which a group of participants acted as co-designers of the app, and another group with participants interacting as non-designers. On the study with developers, we grouped 23 participants according to their professional background. Moreover, each developer had to implement the three architectural parts from the PWA-EU proposal and one from the traditional PWA approach. The findings suggested PWA-EU is a useful architecture for developers with some experience and for end-user empowerment on mobile apps, which brought significant contributions to the PWA and EUD fields.

**Key-words:** Progressive Web Apps, Interface Adaptation, Meta-design, End-user Development, Mobile Apps



# List of Figures

Figure 1 – Research cycle overview . . . . .	22
Figure 2 – PWA features. Source: created by the author . . . . .	25
Figure 3 – Service Worker Life cycle. Source: adapted from (SHEPPARD, 2017) . . . . .	27
Figure 4 – PWA Application Shell. Source: (Google Developers, 2018b) . . . . .	27
Figure 5 – User interface from PWA and Native applications - Trivago . . . . .	37
Figure 6 – User interface from Web and Native applications - TripAdvisor . . . . .	38
Figure 7 – SAM indexes scale . . . . .	40
Figure 8 – PWA-EU architectural scheme . . . . .	57
Figure 9 – UDM Example in JSON . . . . .	58
Figure 10 – UDM in Calendar App . . . . .	61
Figure 11 – Adaptation Engine in Calendar App as Content Display Formats . . . . .	61
Figure 12 – Adaptation Engine in Calendar App as UI elements . . . . .	62
Figure 13 – Task 0 - Co-designers . . . . .	68
Figure 14 – Task 1 - Include events . . . . .	69
Figure 15 – Task 2 - Browse events . . . . .	70
Figure 16 – Participants communication breakdown tags divided by co-designers/non-designers, tech/non-tech and co-designers from both tech/non-tech. . . . .	72
Figure 17 – The cancel event modal window . . . . .	72
Figure 18 – The main screen when the list option was selected . . . . .	73
Figure 19 – The mark as finalized modal window . . . . .	74
Figure 20 – Materialize’s Datepicker - the default option . . . . .	75
Figure 21 – UX perception of with Task 0 . . . . .	77
Figure 22 – The selection of User Preferences on Task 0 - Interaction Methods . . . . .	78
Figure 23 – Materialize’s Timepicker - the default option . . . . .	79
Figure 24 – Boxplots with SAM results categorized by participants role . . . . .	80
Figure 25 – Boxplots with SAM results categorized by participants experience . . . . .	80
Figure 26 – Parei Onde? main UIs . . . . .	87
Figure 27 – Medians from participants previous knowledge, HTML5, CSS, JS, jQuery, LocalStorage(LS), Bootstrap, Handlebars.js(Hjs), JSON, SPA, RWD, PWA . . . . .	90
Figure 28 – Complete UDM example . . . . .	92
Figure 29 – User Design Model (UDM) code analysis . . . . .	92
Figure 30 – UDM code from P20 . . . . .	94
Figure 31 – UDM code from P12 . . . . .	94
Figure 32 – Management Layer code analysis . . . . .	94
Figure 33 – Management Layer code example . . . . .	96

Figure 34 – Management Layer code from P14 . . . . . 96  
Figure 35 – Connection Layer code example . . . . . 97  
Figure 36 – Connection Layer code analysis . . . . . 98  
Figure 37 – Connection Layer code from P4 . . . . . 99  
Figure 38 – Connection Layer code from P7 . . . . . 99  
Figure 39 – Connection Layer code from P18 . . . . . 99  
Figure 40 – Connection Layer code from P23 . . . . . 100  
Figure 41 – Adaptation Engine code analysis . . . . . 100  
Figure 42 – P1, P7, P14, P15 and P16 UIs from "My Design" section . . . . . 101  
Figure 43 – P18's and P23's UIs from "My Design" section . . . . . 102  
Figure 44 – Questions of acceptance of the proposal . . . . . 103

# List of Tables

Table 1 – Related work comparison . . . . .	31
Table 2 – Selected UI elements and general issues regarding each . . . . .	36
Table 3 – Emotion heuristics . . . . .	40
Table 4 – Results obtained with open coding technique . . . . .	43
Table 5 – PWA-native evaluation - positive emotions in bold; PI (Pleasure index); AI (Arousal index); DI (Dominance index) median results . . . . .	44
Table 6 – Web-native coding summary - positive emotions in bold; PI (Pleasure index); AI (Arousal index); DI (Dominance index) median results . . . . .	49
Table 7 – Participants profile - group, technical profile (T. Profile), age, education and professional/academic (P/A) experience. . . . .	71
Table 8 – SAM results. PI - Pleasure Index, AI - Arousal Index, DI - Dominance Index . . . . .	76
Table 9 – Participants profile - Likert scale: I have never heard of it(1), I have heard of it(2), I have some theoretical knowledge, but no practical(3), I have good theoretical and practical knowledge(4), I have deep theoretical and practical knowledge(5) . . . . .	86
Table 10 – Questions from the post-study questionnaire . . . . .	89
Table 11 – Results from analysis divided by participant. TAM caption available at Figure 44. Code caption - Non-existent(1), Incomplete or with errors(2) and Complete with no errors(3) . . . . .	91
Table 12 – Results from Fisher’s exact test statistical analysis . . . . .	104



# List of abbreviations and acronyms

CSS	Cascading Style Sheets
DSL	Domain Specific Language
EUD	End-User Development
HCI	Human-Computer Interaction
Hjs	Handlebars.js
HTML	Hyper Text Markup Language
LS	LocalStorage
JS	JavaScript
JSON	JavaScript Object Notation
PWA	Progressive Web App
RWD	Responsive Web Design
SAM	Self Assessment Manikin
SPA	Single Page Application
SW	Service Workers
UI	User Interface
UDM	User Design Model
UX	User Experience





# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>19</b>
1.1	Research Goals	20
1.2	Methodology	21
1.3	Contributions	21
1.4	Organization	22
<b>2</b>	<b>FUNDAMENTALS AND RELATED WORK</b>	<b>23</b>
2.1	UI Elements and Interface Adaptation	23
2.2	Responsive Web Design, Native and Hybrid Applications	24
2.3	Progressive Web Apps	25
2.4	End-User Development and Meta-design	28
2.5	Related Work	30
<b>3</b>	<b>ANALYZING USER EXPERIENCE ON MOBILE WEB, NATIVE AND PWA APPLICATIONS</b>	<b>35</b>
3.1	Planning	35
3.2	Conduction	41
3.3	Analysis	41
3.4	Results and discussion	43
3.4.1	PWA-native comparison	43
3.4.2	Web-native comparison	48
3.5	Threats to validity	52
3.6	Conclusion	52
<b>4</b>	<b>THE PWA-EU APPROACH</b>	<b>55</b>
4.1	Architecture Proposal	55
4.1.1	PWA-EU Architectural Scheme	57
4.1.2	Privacy	59
4.1.3	Single-page Application Frameworks	59
4.2	Proof of Concept	59
4.3	Final Considerations	62
<b>5</b>	<b>ANALYZING PWA-EU FROM AN END-USER PERSPECTIVE</b>	<b>65</b>
5.1	Planning	65
5.2	Conduction	66
5.3	Analysis	68

<b>5.4</b>	<b>Findings</b>	<b>70</b>
5.4.1	Profile	70
5.4.2	Communication Breakdowns	71
5.4.3	UX Perception	76
<b>5.5</b>	<b>Discussion</b>	<b>80</b>
<b>5.6</b>	<b>Limitations of the study</b>	<b>82</b>
<b>5.7</b>	<b>Conclusion and future work</b>	<b>82</b>
<b>6</b>	<b>ANALYZING PWA-EU FROM THE PERSPECTIVE OF DEVELOPERS</b>	<b>85</b>
<b>6.1</b>	<b>Planning</b>	<b>85</b>
<b>6.2</b>	<b>Conduction</b>	<b>88</b>
<b>6.3</b>	<b>Analysis</b>	<b>90</b>
<b>6.4</b>	<b>Results</b>	<b>92</b>
6.4.1	User Design Model (UDM)	92
6.4.2	Management Layer	94
6.4.3	Connection Layer	97
6.4.4	Adaptation Engine	100
6.4.5	Participants' Feedback	102
<b>6.5</b>	<b>Discussion</b>	<b>105</b>
<b>6.6</b>	<b>Threats to validity</b>	<b>106</b>
<b>6.7</b>	<b>Final Considerations</b>	<b>107</b>
<b>7</b>	<b>CONCLUSION</b>	<b>109</b>
<b>7.1</b>	<b>Contributions</b>	<b>110</b>
<b>7.2</b>	<b>Limitations and future work</b>	<b>112</b>
	<b>Bibliography</b>	<b>115</b>
	<b>APPENDIX A – TCLE 1</b>	<b>122</b>
	<b>APPENDIX B – PROFILE QUESTIONNAIRE 1</b>	<b>124</b>
	<b>APPENDIX C – TASKS/SAM QUESTIONNAIRE - TYPE A</b>	<b>130</b>
	<b>APPENDIX D – TASKS/SAM QUESTIONNAIRE - TYPE B</b>	<b>160</b>
	<b>APPENDIX E – TCLE 2</b>	<b>190</b>
	<b>APPENDIX F – PROFILE QUESTIONNAIRE 2</b>	<b>192</b>
	<b>APPENDIX G – PROFILE QUESTIONNAIRE 3</b>	<b>198</b>

<b>APPENDIX H – DEVELOPER TASKS GUIDE . . . . .</b>	<b>204</b>
<b>APPENDIX I – POST-STUDY QUESTIONNAIRE - TAM . . . . .</b>	<b>208</b>



# 1 Introduction

The challenges of designing interfaces and interactions for different applications platforms are not a new topic and have been discussed in many works (BUENO; ZAINA, 2016; GHIANI et al., 2014; MANCA et al., 2013; NEBELING; SPEICHER; NORRIE, 2013; PROENÇA; NERIS, 2017; GULLÀ et al., 2015; BUENO; ZAINA, 2017). The interest has been boosted by the increase in the number of different devices for user interaction. The number of individuals using web applications on mobile devices has increased when compared with native applications. According to a Comscore research (COMSCORE, 2017), mobile internet access has already overtaken desktop internet around the globe. More than half of the minutes spent on the internet comes from mobile access. In countries such as Brazil, India, and Spain this number reaches over 70% (COMSCORE, 2017).

The user interface (UI) is the part of a software that allows people to see, hear, touch, talk to, and interact with. The design of a suitable interface should provide a mix of well-designed input and output mechanisms that satisfy the user's needs, capabilities, and limitations in the most effective way (GALITZ, 2007). UI elements are used to provide a communication channel between user and system.

Web mobile apps are popular and their coverage goes from smaller devices such as smartphones to bigger ones, smart tv for instance. Additionally, browsers and devices have their own constraints that should be analyzed by developers and designers. This wide range of means to access apps brings the need to adapt interfaces on different resources, dimensions and contexts of use (BUENO; ZAINA, 2016; GHIANI et al., 2014; MANCA et al., 2013). The first solution for developing applications that run on different devices and platforms were hybrid mobile apps. In this approach, apps are created with a combination of web technologies, such as HTML, Cascading Style Sheets (CSS), and JavaScript (JS) codes. After some transformations, the code is hosted inside a native app that uses a kind of mobile browser to run them (CHARLAND; LEROUX, 2011). In a different approach, Responsive Web Design (RWD) uses the original web app. However, it presumes application development and design should adapt its behavior to the user and environment. Adaptation is automatically made from the device screen width, orientation and platform using CSS media queries (BRYANT; JONES, 2012).

Having RWD as a foundation, Progressive Web App (PWA) is a recent approach proposed by Google that combines both web and native apps technology resources, still in a different way from the hybrid mobile approach. PWAs are initially presented as web mobile apps with RWD which after the first user interaction, progressively, become more complete and similar to native apps (PETELE, 2016). This progressive feature provides an

adaptive nature to PWAs since their UIs become more complete according to the growth of user interactions. As PWAs is a recent approach, not many empirical investigations have been carried out, especially focusing on the perspective of User Experience (UX). Conversely, Google conducted a series of case studies with companies that adopted this approach such as Twitter, Forbes, OLX and Aliexpress ([Google Developers, 2018a](#)).

The technologies previously described have its focus on automatic adaptations that do not consider end-users' perspective. End-user development (EUD) focuses on allowing users who are not primarily interested in software to create, modify and extend a system according to their needs. Meta-design is a EUD approach in which the end-user is an active participant in a system's design process. Its main goal is to create an environment that empowers end-users to do their choices. Users can participate actively in application development and are not restricted to using them ([FISCHER, 2009](#)). In this scenario, user preferences and needs are addressed by user's intervention modifying and evolving the system ([PROENÇA; NERIS, 2017](#)).

The first EUD tools focused only on desktop applications ([PATERNÒ, 2013](#)). Yet, in recent years a considerable amount of EUD related studies have taken into account web and mobile apps as their platform ([PROENÇA; NERIS, 2017](#); [COSTABILE et al., 2008](#); [PROENÇA; NERIS, 2017](#); [DANADO; PATERNÒ, 2014](#); [BOSETTI et al., 2017](#); [NAMOUN et al., 2016](#)). Still, most studies focus only on the process of understanding software development based on EUD. With that in mind, we can point out a gap related to end-users' opinion on how they adopt EUD tools to build and modify apps. Even though Namoun et al. ([NAMOUN et al., 2016](#)) investigated factors that influence end-users participation on mobile app EUD activities, the impact PWAs have on these aspects was not taken into account.

## 1.1 Research Goals

The main goal of this master's research project was to propose an approach that allows end-user empowerment on PWA apps, named PWA-EU. The motivation for conducting this research was linked to PWA's adaptive nature and its great potential to be associated with meta-design. Both approaches allow apps to become more complete while the user builds a relationship with it. Nonetheless, the traditional PWA approach does not empower users and takes their preferences into account. This limitation led to the proposal of the PWA-EU approach, an extension of the traditional PWA architecture that includes meta-design in a way that users can include their preferences to adapt interfaces during run-time. With that in mind, this master's project aimed to answer the following research question:

**How can we join PWA and meta-design in a way that users become co-**

## designers of a PWA app?

As the specific goals of this research project, we can point out:

- Review of bibliographic including PWA, EUD, meta-design and mobile adaptation;
- Analysis of a study in which users interact on web, native and PWA platforms;
- Evaluation of PWA-EU proposal on end-users and developers viewpoint.

We can also point out the limitations of this work, which includes points we did not set as goals:

- Evaluate the traditional PWA architecture;
- Create an approach that simplifies PWA development;
- Propose a new application platform for mobile development.

## 1.2 Methodology

The methodology proposed to achieve this project's goal consisted of a six-step cycle, illustrated by Figure 1 and detailed as follows.

**S1 - Literature Review:** review of relevant studies based on this project's fundamentals, with the aims of finding trends, limitations and new findings on this field;

**S2 - Experimental Study I:** conduction of an experimental study with end-users in order to compare user interactions on three distinct mobile platforms, web, native and PWA;

**S3 - Outline Approach:** elaboration of the PWA-EU proposal based on a literature review and lessons learned from Experimental Study I;

**S4 - Experimental Study II:** conduction of an experimental study to validate our approach by an end-user perspective;

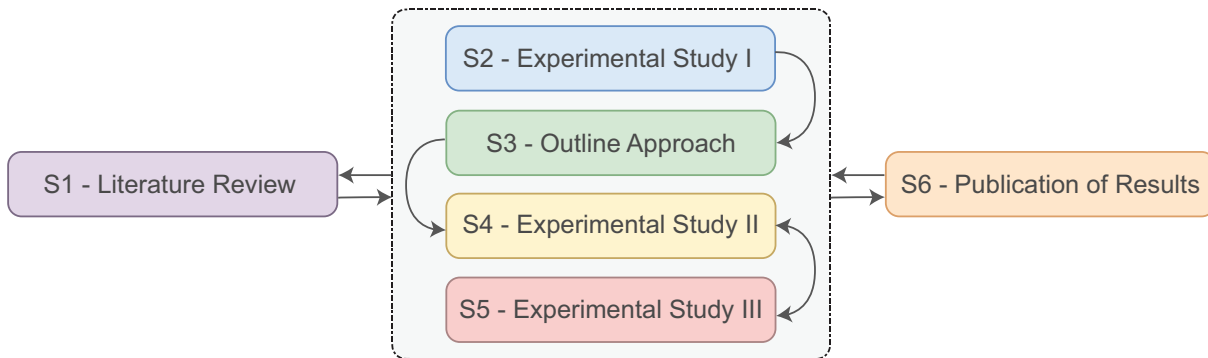
**S5 - Experimental Study III:** conduction of an experimental study to validate our approach on a developers perspective;

**S6 - Publication of Results:** writing of scientific papers and technical reports.

## 1.3 Contributions

The main contributions of this master's project were:

Figure 1 – Research cycle overview



- To provide a discussion comparing PWA, web and native mobile apps on user and specialist perspectives;
- To propose the PWA-EU approach, a PWA architecture extension that considers users' preferences by including meta-design aspects into the traditional PWA architecture;
- Validate the PWA-EU approach by the means of a study in which end-users with distinct technical profiles acted as co-designers on a PWA. This validation showed that users who act as co-designers may achieve a more satisfying experience. Still, playing this role is easier to end-users who are familiar with technology;
- To validate the PWA-EU approach by the means of a study on developers perspective, which pointed out that developers found the approach useful. Yet, novice developers might have difficulties with architecture and implementation.

## 1.4 Organization

This work consists of seven chapters that are described as follows. In Chapter 1, an introduction regarding this study's domain are presented, as well as the research goals, methodology, and contributions. In Chapter 2, we discuss fundamental concepts and related works to this research. Chapter 3 presents all steps of the experimental study about UX on three mobile platforms, besides discussing its results. In Chapter 4, a proposal to a PWA architecture extension with the aims of including meta-design concepts is featured. In Chapter 5, we analyze Chapter's 4 proposal on an experimental study that focused on analyzing end-users acting as co-designers on PWAs. Chapter 6 presents an experimental study evaluating Chapter's 4 proposal on a developers' perspective. At last, Chapter 7 concludes this work by discussing the findings of this project.



## 2 Fundamentals and Related Work

The search and gathering process of the related studies reported in this chapter were based on two main strategies: keywords-based search and snowballing strategy (WOHLIN et al., 2012). The keywords-based search was performed on bases such as Scopus<sup>1</sup>, Google Scholar<sup>2</sup>, ACM Digital Library<sup>3</sup> IEEE Xplore<sup>4</sup> and Web of Knowledge<sup>5</sup>. The following keywords were used in the process of searching for related works: Interface Adaptation, Interface Elements Adaptation, UI Elements, Interface Elements, End-User Development, Meta-design, Progressive Web Applications, Mobile Applications and Native Applications. Additionally, we searched for studies that combined more than one keyword. We refined our searching by examining their references and looking for new ones by applying the snowballing strategy.

### 2.1 UI Elements and Interface Adaptation

UI elements are interface object collections tied to a feature, which allow user interactions with an application (TIDWELL, 2011). All elements must be meaningful to users and have a purpose in performing tasks or fulfilling needs, otherwise, a given element is only noise and should not be included on the UI (GALITZ, 2007). These elements are reusable, optimized and organized solutions to recurring UI problems. Designers must apply and adapt these patterns according to each project's context (Interaction Design Foundation, 2017).

Some examples of UI elements in a system are windows (primary, secondary, or dialog boxes), menus (menu bar, tabs, pop-up, cascading), icons that represent objects, assorted screen-based controls (text boxes, list boxes, combination boxes, settings, scroll bars, and buttons), a mouse or other pointing device, and the cursor. The main goal of these elements is to represent the real world on the most realistic, meaningful, simple and clear way as possible to the user (GALITZ, 2007).

The concept of user interface adaptation on a system is not new. The literature describes many approaches that can be used to create a design able to adapt itself depending on requirements and/or user needs. These approaches can be classified into two distinct categories, adaptable and adaptive (GULLÀ et al., 2015).

---

<sup>1</sup> Link: [www.scopus.com/home.uri](http://www.scopus.com/home.uri)

<sup>2</sup> Link: [scholar.google.com](http://scholar.google.com)

<sup>3</sup> Link: [dl.acm.org/](http://dl.acm.org/)

<sup>4</sup> Link: [ieeexplore.ieee.org/](http://ieeexplore.ieee.org/)

<sup>5</sup> Link: [www.webofknowledge.com/](http://www.webofknowledge.com/)

An adaptive UI is based on the principle that a system should be capable of identifying the circumstances that require an adaptation and apply the appropriate actions according to the adopted strategy. Therefore, the adaptive technique requires that both user and system status be constantly monitored. On the other hand, an adaptable UI offers to the end-users a chance to choose between alternative presentation and interaction characteristics already built into the system, similar to selecting a user profile from a predefined list. Nonetheless, the line that separates these two categories is very thin. Even though adaptable systems allow users to have total control of the individual appearance and UI, adaptive systems can facilitate user interactions, minimizing the need to request help and helping users to deal with complex systems. At last, an adaptive UI with user control is a good arrangement between adaptive and adaptable interfaces approaches. In doing so, the system is responsible for the adaptation process under user's supervision (GULLÀ et al., 2015; BUENO; ZAINA, 2017).

## 2.2 Responsive Web Design, Native and Hybrid Applications

One of the most known proposals in interface adaptation area is Responsive Web Design (RWD) (MARCOTTE, 2011). This approach assumes website adaptation according to the characteristics of a mobile device, instead of developing multiple versions of a website to support a large number of devices (BRYANT; JONES, 2012). RWD is based on three technical bases: flexible grid layouts with relative dimensions, flexible images adjusted according to screen size, and CSS3 media queries that automatically adapts layout behavior to screen sizes and orientations (MARCOTTE, 2011; BRYANT; JONES, 2012). Ghiani et al. (2014) state that even though RWD has been recently adopted by many designers and developers, it has limitations that can affect the user experience, such as lack of multimodality support on certain situations.

Native applications are developed with a programming language from a certain platform, such as Java on Android. The code is already compiled when the application is installed on a device, which makes it faster when compared to web apps. Hybrid applications require tools as the framework PhoneGap<sup>6</sup>. Its purpose is to create native apps converting code from web technologies such as HTML, CSS, and JS to a compiled code to a certain platform. These applications have access to features that native applications have. However, an API accessed via JS is still required (CHARLAND; LEROUX, 2011).

Usually, web mobile apps are known for having performance issues when compared to native apps. However, according to Charland & Leroux (2011) there are two problems on this line of reasoning: (i) high cost to develop applications written in platform-specific programming languages and (ii) lack of striking differences on web mobile apps performance

---

<sup>6</sup> Link: <https://phonegap.com/>

Figure 2 – PWA features. Source: created by the author



when compared to their native counterparts, except for applications on 3D games or image processing fields (CHARLAND; LEROUX, 2011).

## 2.3 Progressive Web Apps

PWA is a new approach to develop applications by joining resources from web and native applications. This approach mixes a set of strategies, techniques, and APIs in order to deliver a native mobile-like experience (SHEPPARD, 2017). This term was defined in 2015 by Alex Russell, a software engineer at Google, and Frances Berriman, a designer (RUSSELL, 2015). These applications are initially presented as web mobile apps using RWD, however, as the user progressively builds a relationship with the application over time, they become more complete and similar to native apps. Besides progressive, a PWA is (i) *responsive*, adapting its interface with RWD, (ii) *connectivity independent*, working offline or in low-quality networks, (iii) *similar to native applications*, regarding interactions, navigation and appearance, (iv) *re-engageable*, allowing features as push notifications to recover user involvement with the app, (v) *installable*, allowing users to add apps to their home screen, and (vi) *fresh*, checking for new updates when online (PETELE, 2016). Figure 2 illustrates these features.

Since PWAs are a recent technology, not many empirical investigations have been carried out focusing on the perspective of user experience (UX). Still, Google has conducted

case studies with companies such as Twitter, OLX, and Jumia with the aims of analyzing the impact PWA brought on apps that migrated from the web approach. Jumia is the leading e-commerce in Africa, in which many users have intermittent connectivity and devices with data limitations. In this scenario, it is difficult to induce users to download a native app. By replacing their web app to a PWA, the company perceived that their traffic migrated from their native app by more than 12 times and is still growing (Google Developers, 2017a).

Besides the characteristics described above, authors claim that PWA's main characteristics are being reliable, fast and engaging. They are considered reliable since they use Service Workers, loading instantaneously detached from network connection state. These apps are fast because they have a fast response with smooth animation to user interactions. Engaging since they have an application manifest, that enables similarities to native applications, providing an immersive user experience with full screen and push notifications (Google Developers, 2018c; SHEPPARD, 2017).

Service workers (SW) are JS scripts run by the browser in the background. This technology uses resources that are independent of a web page or user interactions, such as push notifications and offline mode. These scripts have a life cycle detached from the web page and should follow three steps (see Fig (SHEPPARD, 2017)), (i) registration: the JS code registers the SW and requires the browser to start installation as a background process, (ii) installation: when static files are stored on cache, and (iii) activation: where old cache files are handled. However, an extra step is required when the Service Worker needs to be updated with new data from the app (SHEPPARD, 2017).

In order to implement a PWA that is reliable and fast, its architecture must be arranged using an application shell architecture. This architecture consists of loading only the shell containing the application's UI, usually formed by HTML, CSS and JS files, besides static content such as images and fonts (see Fig 4). The specific dynamic content from each web page is not initially loaded. Because of that, in a shell architecture, it is important to separate page layout content from navigation, and to load as little data as possible and to load static resources from the local cache. Furthermore, there is no need to download new versions of the application, as the most recent release will be automatically displayed when the online server sends the most recent files (SHEPPARD, 2017).

Google Developers (2017b) claim that PWAs bring radically better web experiences. In order to make web experiences pleasant to users, three key principles can be followed during PWA application development. These principles are known as Feel Good Principles and are based on the fundamentals that applications should be (i) reactive, (ii) predictable and (iii) in control.

A reactive app is based on perceived performance as the app reacts to each user interaction. By assuring users had the impression of short loading time, it avoided

Figure 3 – Service Worker Life cycle. Source: adapted from (SHEPPARD, 2017)

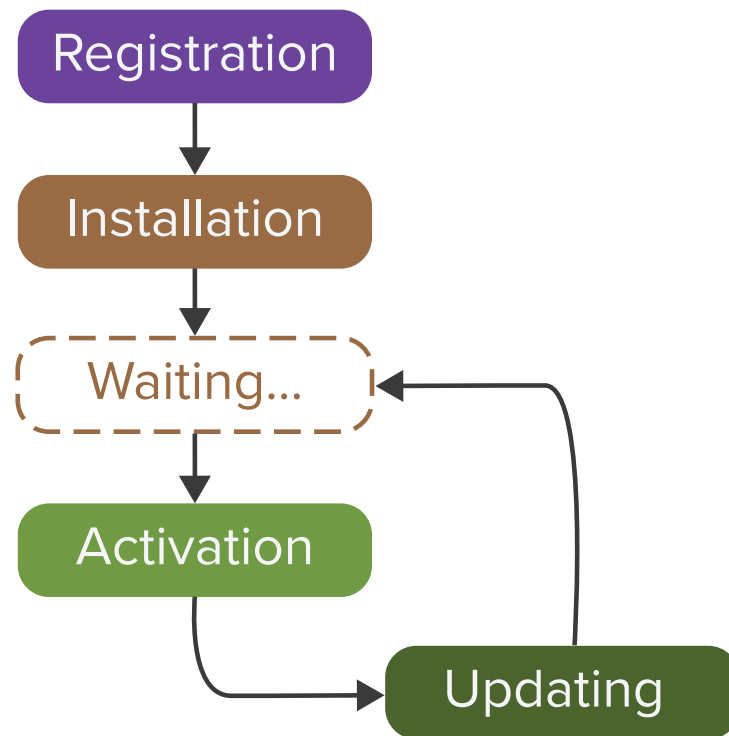
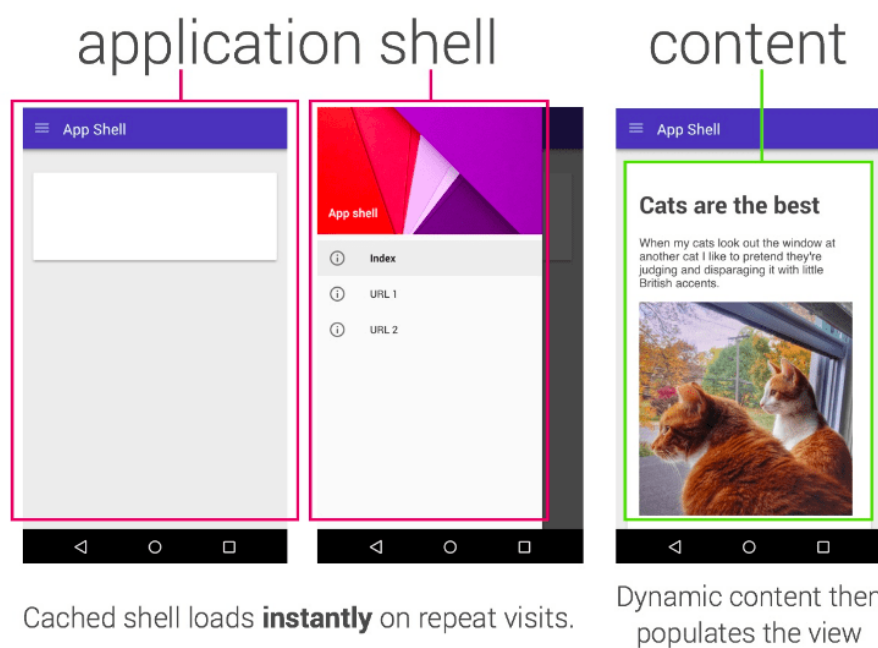


Figure 4 – PWA Application Shell. Source: (Google Developers, 2018b)



frustrations. Yet, this perception is achieved by relying on an app shell architecture in which part of the UI loads instantly and another waits for the server's response time. Moreover, it is important that every user action (i.e. touch or swipe movements) have an observable response.

A predictable app offers a UX in which users know what to expect to lead to a better comprehension of their experience. For instance, the side menu UI element, popularly known as the hamburger menu usually allows swipe and touch gestures. However, if one of these movements is missing on a menu, users might be frustrated as their expected an answer by performing this action.

When an application allows users to always be in control, it is developed focusing on the display of information, notifications or permissions in a subtle and stable manner, not by surprising the user. In this case, users can interpret the meaning of these actions, which avoids mis-clicks or the performance of an unwanted action.

## 2.4 End-User Development and Meta-design

End-user development (EUD) can be defined as a set of methods, techniques, and tools that allow users of software systems to create, modify, or extend a software artifact according to their needs and preferences. These users act as non-professional software developers and are not primarily interested in software development (LIEBERMAN et al., 2006; FISCHER, 2009). EUD's main goal is empowering end-users to develop and adapt the systems by themselves. Nonetheless, recent studies point out the previous definition is too restrictive, as there are currently many possibilities provided by technology, brought by web 2.0 and 3.0, Internet of Things and smart devices. A new definition of EUD can include methods, situations, and socio-technical environments that allow and empower end-users to express themselves. Moreover, users can be independent of developers while performing EUD activities (FISCHER; FOGLI; PICCINNO, 2017). On meta-design, users are in charge of solving their own problems, instead of relying on technology experts (FISCHER; NAKAKOJI; YE, 2009).

EUD appeals to distinct audiences by considering them when creating a design and building artifacts, besides developing tools to democratize knowledge creation (FISCHER; FOGLI; PICCINNO, 2017). Some examples of widely used applications in commercial software systems that have EUD concepts are recording macros in word processors, setting up spreadsheets for calculations, and defining e-mail filters. Yet, even though these applications have flaws and are only a fraction of EUD's potential, they can still set good examples of why empowering end-users to modify the systems they use is a relevant contribution to the society (LIEBERMAN et al., 2006).

An influential EUD approach is Meta-design, in which the end-user acts actively in

a system's design process. Its main goal is to create a user empowered environment, where all stakeholders can actively participate in application development and are not restricted to its use (FISCHER, 2009; FISCHER; FOGLI; PICCINNO, 2017). In this scenario, user preferences and needs are addressed by users' intervention who are able to modify and evolve the system (PROENÇA; NERIS, 2017). This approach contributes to the creation of cultures in which humans can express themselves and engage in personally meaningful activities. During these activities, the partial control is shifted from designers to users, who can contribute with their own visions and objectives. A meta-design approach is not an option, it is a necessity in order to match user's needs with a system (FISCHER, 2007).

Meta-design is a "*design for after design*", which indicates that this approach allows users to modify and evolve the system at use time, creating an open system. While user-centered design and participatory design promotes "*design for use before use*" with the aims of forcing all design intelligence on the beginning of the design process when there is little information about what is really needed (FISCHER; FOGLI; PICCINNO, 2017; FISCHER et al., 2007).

The Meta-design framework proposes a set of guidelines to support domain expert to adopt the EUD approach, described as follows (FISCHER; NAKAKOJI; YE, 2009; FISCHER; FOGLI; PICCINNO, 2017):

1. **Support Human-problem Interaction.** Domain experts want to focus on solving their own problems, not on interacting with computers or software development. Therefore, the design must support human-problem interaction and not human-computer interaction. In order to achieve this, the software domain and their problems must be defined by users.
2. **Underdesign for Emergent Behavior.** Meta-design systems should be underdesigned because users should not treat them as finished products, but view them as a continuous beta version that can incorporate emerging design behaviors during use. However, underdesign does not mean that design responsibilities should be transferred to end-users.
3. **Enable Legitimate Peripheral Participation.** Newcomers to a community must be able to participate with contributions. Transparent procedures are required to incorporate user contributions into the systems, which should be visible and have a recognizable influence.
4. **Share Control.** The original metadesigners must find a way to transfer partial control to the end-users. These users can play different roles, with their own responsibility and authority, depending on their involvement. This assists in sustaining user participation and system evolution, since the users that become stakeholders and acquire ownership in the system will probably continue to contribute.

5. **Promote Mutual Learning and Support.** Users do not have the same skill and knowledge levels as their peers. In order to get involved and to contribute to the system, they need to learn many things. A metadesigned system should have knowledge-sharing mechanisms associated with it to encourage users to learn from each other.
6. **Reward and Recognize Contributions.** Humans are not motivated only for material gain, but also for several reasons such as psychological well-being, social integration, recognition and improving reputation. Because of that, paying attention to users' motivation is essential for the continuous user participation and consequently evolution of the metadesigned system. Users' motivation is derived from their satisfaction from getting involved in shaping the software system to solve their problems.
7. **Foster Reflective Communities.** The knowledge required to solve a complex design issue is likely distributed among many domain experts. Due to that, it is important to create a mechanism for collaboration between these experts, in which they can share often controversial viewpoints which can lead to new insights, ideas, and artifacts.

## 2.5 Related Work

This section aims to present related work regarding the topics of the master's project. Table 1 shows the most relevant studies categorizing them into (i) topics the studies are related to (i.e. interface adaption (IA), and EUD), and (ii) platform they focus on (desktop, web, mobile web, PWA, and native). Each study is described in the following paragraphs.

UIFlex (PROENÇA; NERIS, 2017) is a web-browser extension that performs adaptations the UI from existing websites. It adapts considering users' profiles using meta-design principles. There are two steps on the adaptation: the first step takes information from a questionnaire answered by users and suggestions from authorities such as the W3C and Mozilla Developer Network to create an interaction profile. The second step adapts UIs from the visited web pages considering each user interaction profile and using JS, CSS and HTML code injections.

HyMobWeb (BUENO; ZAINA, 2017) is a hybrid approach that adapts UIs from web applications on mobile devices. It is based on context sensibility and multimodality. The proposal extends front-end frameworks by which developers are able to include adaptation points on the code. A hybrid strategy was adopted to combine both static when modifications occur during implementation (i.e. during coding), and dynamic adaptations



Table 1 – Related work comparison

<b>Work</b>	<b>Topic</b>	<b>Platform</b>
UIFlex (PROENÇA; NERIS, 2017)	IAM, EUD	Web
HyMobWeb (BUENO; ZAINA, 2017)	IAM	Mobile web
Bueno (BUENO; ZAINA, 2016)	IAM	Mobile web
Adaptive Management Interface (GULLÀ et al., 2015)	IAM	Undefined
Tukuchiy (BARRERA-LEON et al., 2016)	IAM	Desktop
Ghiani (GHIANI et al., 2014)	IAM	Mobile web
Manca (MANCA et al., 2013)	IAM	Mobile web
W3Touch (NEBELING; SPEICHER; NORRIE, 2013)	IAM	Mobile web
Biørn-Hansen (BIØRN-HANSEN; MAJCHRZAK; GRØNLI, 2017)	None	PWA
Fortunato & Bernardino (FORTUNATO; BERNARDINO, 2018)	None	PWA
Puzzle (DANADO; PATERNÒ, 2014)	EUD	Native
Costabile (COSTABILE et al., 2008)	EUD	Desktop
Bosetti (BOSETTI et al., 2017)	EUD	Desktop, Mobile Web
Namoun (NAMOUN et al., 2016)	EUD	Native
<b>PWA-EU</b>	<b>IAM, EUD</b>	<b>PWA</b>

when the web app changes its behavior according to the adaptation points during run time.

Bueno & Zaina (2016) conducted an exploratory study comparing two web mobile applications using RWD focusing on UI elements, modalities, and usage contexts. The first application had automatic adaptations from Bootstrap framework<sup>7</sup>, and the second used adaptations defined by the authors with the aim of solving possible interaction issues. The results of an evaluation with end-users demonstrated that the addition of multimodality and new adaptations brought positive changes in users' interactions efficiency and satisfaction. Besides, the outcomes pointed out gaps in adaptation resources in the front-end frameworks.

Gullà et al. (2015) proposed the Adaptive Management Interface, a method to develop adaptive UIs. This method follows a "design for all" paradigm, in which the UI attempts to provide an easy and friendly interaction for different users, including elderly users. Besides that, the UIs should be able to define and represent user behavioral models according to an identified scenario. This method is based on three information models: the user model (i), the environment model (ii) and the interaction model (iii). The user

<sup>7</sup> Link: <https://getbootstrap.com/>

model (i) provides a description of the user's profile pattern including his/her cognitive and physical structure, status and preferences. The environment model (ii) supplies the information pattern and includes the functionalities, physical and local characteristics of the HCI environment. The interaction model (iii) is the core of the adaptive process and controls both user and environment models. Moreover, this model must recognize the user and store his/her preferences, extract HCI interaction information, provide logical and task interpretation, allow suitable environmental usability and define the event activation schedule.

Tukuchiy ([BARRERA-LEON et al., 2016](#)) is a framework that generates adaptive UIs based on the Runa-Kamachiy model. This model combines a set of HCI standards (i.e. Usability components), information gathered from the user and use contexts to develop UIs for user-centered desktop applications. Adaptations are dynamic since they occur during the application's execution time. Tukuchiy was validated through two prototype evaluations by specialists. The results indicate that this framework can improve applications' usability. However, there are still features that can be enhanced.

The studies from Manca ([2013](#)) and Ghiani ([2014](#)) on mobile web present solutions in which the interfaces are self-adapted automatically considering distinct modalities and context of use, and are based on rules and HTML, CSS and JS code injections. Positive results indicating that users can benefit from adaptations were obtained on both solutions. The proposal of Ghiani ([2014](#)) also demonstrated that is more flexible and provides wider support when compared to RWD.

W3touch ([NEBELING; SPEICHER; NORRIE, 2013](#)) presented an automatic adaptation solution of mobile devices web pages. This study introduced a toolkit that produces analytics from user interactions aiming to assist designers to find issues related to the touch movement. The adopted metrics are mis-clicked links ratio and zoom levels necessary to visualize page components. From the metrics and rules, the toolkit can detect the need to make modifications using CSS and JS code injection.

Biørn-Hansen et al. ([BIØRN-HANSEN; MAJCHRZAK; GRØNLI, 2017](#)) proposed a technical comparison between PWA, Native, Hybrid, and Interpreted applications, besides describing concepts from PWAs and their technologies. They concluded there is potential for PWAs to unify web and native development without the need for cross-platform frameworks. Fortunato & Bernardino ([2018](#)) presented PWA's characteristics and a comparison between development frameworks. Their conclusion, the authors pointed out that it was possible to achieve a high-quality software using PWA approach.

Danado & Paternó ([2014](#)) proposed Puzzle, a framework in which end-users with no previous experience on programming can develop or customize mobile apps. Puzzle has an architecture that enables apps to connect to web services, native functions from the mobile device and smart devices. Additionally, Puzzle provides an environment UI

that allows users to create complex apps through a simple process. It is developed with JS, HTML, CSS.

Costabile et al. (2008) proposed a EUD desktop application based on the software shaping workshop (SSW) method. SSW adopts a meta-design participatory approach that continues throughout a whole software life cycle. Their application considers four types of end-users and defines that only the power users can modify and delete content. The other three types of end-users are limited to visualizing content.

Bosetti et al. (2017) presented a tool based on mobile web augmentation (MoWA) in which end-users can develop apps from desktop and mobile devices even when they have no programming skills. Web augmentation is a technique to manipulate and enhance web apps with new features. MoWA is a framework to extend web desktop applications to the mobile field taking the app's environment context into account. To evaluate their approach, an experiment with 21 participants was conducted. In this study, users had to implement an app on the tourism domain using the MoWA authoring tool developed by the authors. In the end, most participants were able to create their own mobile web experience without having programming knowledge.

Namoun et al. (2016) proposed a model linking the features of performing EUD in mobile devices to end-users' attitudes towards and intent of doing this. The proposal arose from different studies carried out with participants about the motivations and activities they played on EUD field. The studies considered participants with experience or not on EUD in mobile. The results show that rather than creating apps the end-users have more interest in customizing apps to improve their experience.

Even though the studies presented in this section have multiple ways to solve interface adaptation issues, they have limitations. Most of the authors' proposed techniques, frameworks and intelligent algorithms that automatically generate adapted interfaces (BARRERA-LEON et al., 2016; BUENO; ZAINA, 2016; BUENO; ZAINA, 2017; GHIANI et al., 2014; MANCA et al., 2013; NEBELING; SPEICHER; NORRIE, 2013). Others consider only technological aspects and contexts of use on their adaptations (BUENO; ZAINA, 2016; BUENO; ZAINA, 2017; GHIANI et al., 2014; NEBELING; SPEICHER; NORRIE, 2013; MANCA et al., 2013; GULLÀ et al., 2015). Furthermore, these ones leave native mobile apps dimension uncovered. In some cases user preferences are taken into account as a foundation of the adaptation process (BARRERA-LEON et al., 2016; GULLÀ et al., 2015). However, these studies did not target at providing freedom to users to take the role of the active agent of their adaptation choices. UIFlex (PROENÇA; NERIS, 2017) proposal contemplates user preferences and user active participation to change adaptations automatically made by the system. Nonetheless, it is limited to a web environment. Furthermore, UIFlex does not enable users to change UI elements appearance, content display format, and interaction methods.

Even though Puzzle (DANADO; PATERNÒ, 2014), Bosetti's (BOSETTI et al., 2017) and Namoun's (NAMOUN et al., 2016) studies take the user empowerment in setting the mobile apps, PWA-EU has a different focus. While PWA-EU aims to empower users by combining their preferences into interface adaptation on a PWA UI, Puzzle focuses on allowing end-users to create complex mobile applications that connect to smart devices from scratch. Furthermore, Bosetti's (BOSETTI et al., 2017) tool extends an existing web desktop app. The application presented by Costabile (COSTABILE et al., 2008) do not take into account mobile environments and/or interface adaptation on their proposal. At last, Namoun et al. (NAMOUN et al., 2016) do not consider PWAs into their research. Still, the works related to PWA do not take interface adaptation and/or EUD into account.

An unexploited gap is joining the adaptive nature from the EUD approach, such as meta-design, and PWAs concepts. On the proposal presented in this master's dissertation, end-users would actively take part in all UI adaptations, by progressively modifying their preferences such as UI elements appearance, content display format, and interaction methods.

## 3 Analyzing user experience on mobile web, native and PWA applications

We conducted an experimental study with 8 participants aiming to identify aspects of user experience on three different platforms: PWA, web mobile (with RWD) and native Android. We considered only the Android operating system since both PWA and Android are technologies proposed by Google. This experimental study was motivated by two aspects. First, there had not been reports of academic studies that investigated the use of PWA in the perspective of user experience. In addition, although Bueno & Zaina's study (BUENO; ZAINA, 2016) had explored the end-user perspective, the present study conducted a quantitative analysis regarding some UI elements on web mobile application. Our current study differs from others by conducting a qualitative analysis comparing the user experience during users' interactions with distinct application platforms, considering both user and HCI specialist perspectives. Guidelines proposed by Lazar et al. (2017) were addressed in this experiment organization, following planning, conduction and analysis phases. In the next sections, we present all of the steps.

### 3.1 Planning

Before starting the planning of the experimental study, we carried out an investigation with the aims of setting which UI elements will be explored in this study. A literature review was conducted looking for studies that reported UI elements issues in web and native applications. The review was based on two strategies. First, we conducted a keywords-based search on the following bases: Scopus<sup>1</sup>, Google Scholar<sup>2</sup>, ACM Digital Library<sup>3</sup> IEEE Xplore<sup>4</sup> and Web of Knowledge<sup>5</sup>. Keywords such as *interface elements*, *interaction elements*, *interface elements pattern* and *user interface patterns* were searched. Taking into account the set of works found out, we refined our searching by examining their references and looking for new ones by applying the snowballing strategy (WOHLIN et al., 2012). Table 2 shows the selected UI elements.

UI elements are interface object collections tied to a feature, allowing user interactions with the web page (TIDWELL, 2011). These patterns are reusable, optimized and organized solutions to recurring UI problems. Designers must apply and adapt these

---

<sup>1</sup> Link: [www.scopus.com/home.uri](http://www.scopus.com/home.uri)

<sup>2</sup> Link: [scholar.google.com](http://scholar.google.com)

<sup>3</sup> Link: [dl.acm.org/](http://dl.acm.org/)

<sup>4</sup> Link: [ieeexplore.ieee.org/](http://ieeexplore.ieee.org/)

<sup>5</sup> Link: [www.webofknowledge.com/](http://www.webofknowledge.com/)

Table 2 – Selected UI elements and general issues regarding each

UI element	Description	Issues
Search input	Most common search interface. A search term is entered in an input. (MEIER; HEIDMANN; THOM, 2014)	Lack of efficiency when typing search terms, however, voice modality can be added in order to increase efficiency (BUENO; ZAINA, 2016).
Calendar	One of the most efficient ways to avoid format errors on date insertions. Commonly used on travel websites to book hotels and flights due to its easiness observing the exact day of the week and month (BARGAS-AVILA et al., 2011).	Possibility to enter the date manually in the field, which may lead to an incorrect date (BARGAS-AVILA et al., 2011).
Select	Used when only one option must be chosen (BARGAS-AVILA et al., 2011). Easy to recognize as something that can be selected (RIBEIRO, 2012).	An extensive menu can fill the entire page, being ineffective on mobile devices. Difficult to achieve a consistent look across platforms (RIBEIRO, 2012).
Map	Commonly used in mobile applications to find specific places and get directions (HARLEY, 2014). Can also be used to select results from a search query (MEIER; HEIDMANN; THOM, 2014).	Icons are too close for selection through touch movements on mobile devices. Furthermore, the map may appear unresponsive to user interactions on slow network connections (HARLEY, 2014).
Tab menu	Content is separated into sections with clear indications of user current location (UI PATTERNS, 2017).	Section names must be short. Menu option number must be between 2 to 9 (UI PATTERNS, 2017).
Table	Allow users to analyze, observe and manipulate data. Distinct design structures, techniques and patterns, such as horizontal scroll and pagination, should be combined on mobile devices (COYLE, 2017).	Data visualization may be damaged if a table design is poorly structured (COYLE, 2017).

patterns according to each project’s context (Interaction Design Foundation, 2017). We selected these elements since they are commonly used to provide a communication channel between a user and a system (GALITZ, 2007).

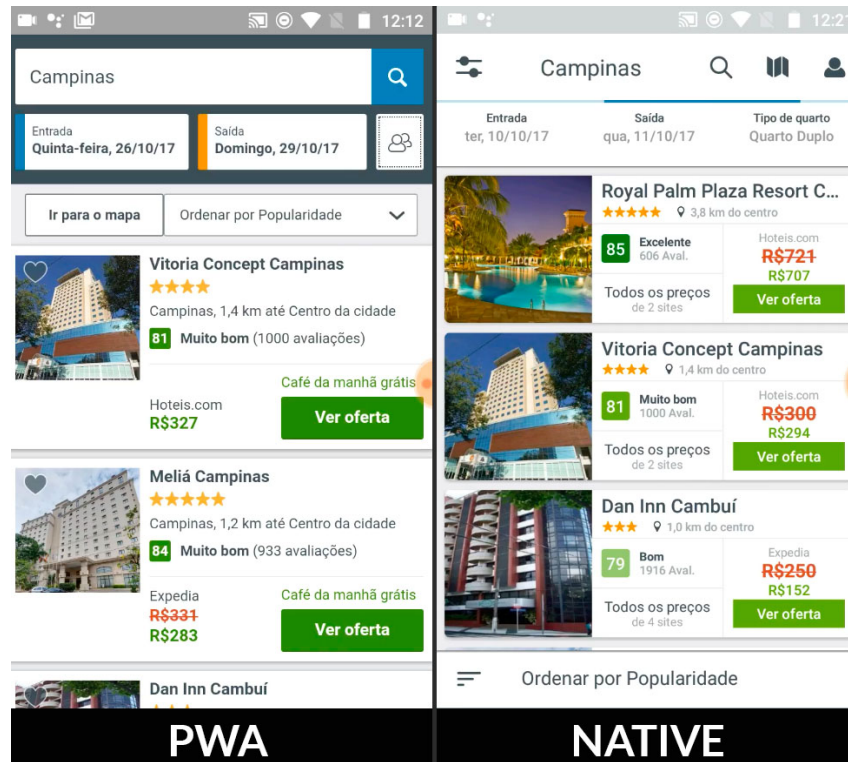
In the next step, we browsed the web and Google Play to choose which PWA, web and Android native applications would be used. We considered two requirements for this search. First, the applications should belong to the same category (business, news, game, shopping and so on) and preferably to the same organization either on Google Play or the web. The search for PWA applications was performed on websites that report these applications, such as PWA Rocks <sup>6</sup>, awesome-pwa GitHub repository <sup>7</sup> and Google Developers case studies showcase <sup>8</sup>. Second, the presence of a UI element (see Table 2) in at least two of three approaches (PWA, web and Android native). We checked a group of 17 applications of 6 categories. However, we could not find applications in the same category for all three platforms. We observed that most organizations that have a PWA application replaced its web application on mobile devices for this new technology.

<sup>6</sup> Link: <https://pwa.rocks/>

<sup>7</sup> Link: <https://github.com/hemanth/awesome-pwa>

<sup>8</sup> Link: <https://developers.google.com/web/showcase/>

Figure 5 – User interface from PWA and Native applications - Trivago



Therefore, we set our study by comparing the application approach in pairs. For ease of reference, *PWA-native* will be used to reference the comparison between PWA and native application. For this comparison, we chose PWA and Android native applications from Trivago<sup>9</sup>, a tourism category. By adopting applications from the same organization we ensure that a similar visual identity will be present in both UI elements. Figure 5 show examples of UIs from the chosen application. However, we could not find the same UI element matching all approaches and application category. In order to keep the same category, TripAdvisor web mobile and Android native applications were selected for *web-native* comparison. Figure 6 shows an example of TripAdvisor's interfaces.

A questionnaire to collect participants' profile was elaborated. Data such as gender, age, level of education, application download frequency, relevant factors to download an application, the frequency of internet access through mobile devices and the frequency of accessing some types of websites (see Appendix B).

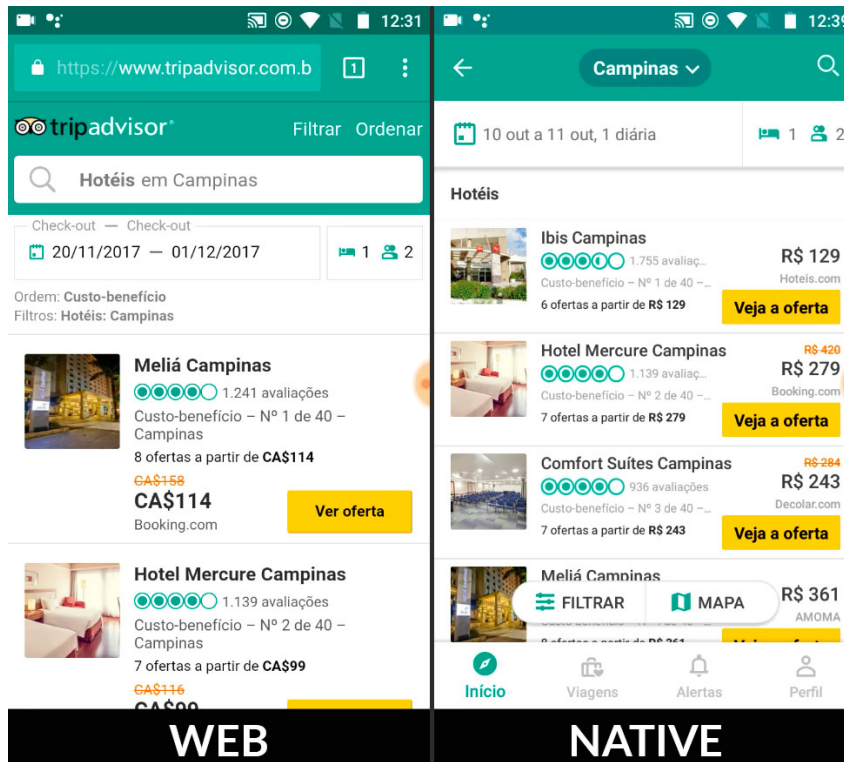
Based on the rationale for the use of the applications, we specified tasks related to the UI elements (see Table 2) considered in this study. The aim is that the tasks guide users' interactions with the application, and hence, we could explore their user experience. Participants' had access to the list of tasks using Google Forms<sup>10</sup>.

For *PWA-native* comparison, five tasks were elaborated. For some tasks, we proposed

<sup>9</sup> Link: <http://trivago.com/>

<sup>10</sup> Link: <https://www.google.com/forms/about/>

Figure 6 – User interface from Web and Native applications - TripAdvisor



more than one interaction modality such as voice and touch, and for others, we put together different UI elements. The tasks were specified as follow: ( $T1$  - *search input*) fill the input field with a pre-defined term; ( $T2$  - *calendar*) select a pre-defined date to his/her booking and try to insert an invalid date where the check-out date occurs before the check-in; ( $T3$  - *select*) choose one option on a drop-down element; ( $T4$  - *map*) select a hotel on the map using pinch movement; and ( $T5$  - *tab menu and table*) select one of the *tab menu* options and browse on the table content.

For *Web-native* comparison, the same five tasks were conducted following same definitions from *PWA-native* comparison. However, only *search input* and *datepicker* tasks were available on both approaches. Elements *map* and *select* were present on the web mobile application, although not available on native application. While *tab menu and table* was exclusively analyzed on the native application.

Aiming to analyze two different perspectives, user and HCI specialist, we selected two instruments for analyzing user experience aspects. First, focusing on obtaining users' feedback, we chose SAM - Self Assessment Manikin (BRADLEY; LANG, 1994) to collect and explore the participants' emotional feedback when interacting with the applications. Our choice was based on Hassenzahl's (2010) study that defined UX as a user's holistic perception of the software's functionality and quality characteristics. Moreover, the author affirms the perception of UX on an academic context concentrates on hedonic aspects and emotions. SAM is a pictographic evaluation method to measure emotional responses from



some sort of stimulus. Three dimensions are considered by this technique: pleasure (if the participant had a positive or negative reaction), arousal (body stimulation level from an event or object) and dominance (feeling in control of the situation or controlled by it). The user chooses a value on a scale of 1 to 9 on each dimension, using images. We opted to use a different set of images than the original approach, similar to Hayashi et al. (2016) (see Figure 7), to represent emotions in his/her perspective. We added questions about pleasure, arousal, and dominance to the participants' questionnaires, associated with an open question allowing participants to remark their impressions and possible issues on each task.

In addition, the ten emotion heuristics proposed by De Lera & Garreta-Domingo (2007) were adopted aiming to gather data regarding an HCI specialist viewpoint (see Table 3). The authors proposed to analyze user experience by evaluating facial emotional signs. The ten heuristics were created by analyzing and identifying the most common facial and body expressions during user evaluation. Its goal was providing a positive (P), neutral or negative (N) value about general user experience. Therefore, such heuristics allowed us to gather data regarding HCI specialist perspective and complement the user's viewpoint represented by SAM evaluation. In order to identify specific interaction issues and participants' facial expressions, both participants' face and interactions with the applications were recorded during the experiment conduction.

Emotion heuristics are commonly used in the HCI field in different conferences and journals. Xavier et al. (2012) presented a proposal that associated human emotional feedback to UI elements. Its fundamentals relied on identifying emotions using the ten emotion heuristics (LERA; GARRETA-DOMINGO, 2007) and Scherer's semantic space for emotions structure. Alegria et al. (2013) created a proposal for the analysis and identification of behavioral characteristics from users while evaluating the usability of a web application. The ten emotion heuristics (LERA; GARRETA-DOMINGO, 2007) supported the evaluation of emotions associated with facial expressions, which were identified using the Behavioral Coding System (BCS). Marques et al. (2014) evaluated different aspects of user experience in a serious game. The ten emotion heuristics (LERA; GARRETA-DOMINGO, 2007) and game experience questionnaire were selected as evaluation methods. Sánchez et al. (2012) stated that traditional techniques for measuring user experience are not sufficient to analyze a gaming system. The ten emotion heuristics (LERA; GARRETA-DOMINGO, 2007) was one of the selected methods to acquire information about the emotional interaction process, revealing emotional feedback during their interactions with the system.

A pilot test with five participants was conducted to verify if guide instructions were sufficiently clear and objective, and in addition, we observed possible improvements regarding tasks execution and data collection. From each participant's result, we could

Figure 7 – SAM indexes scale

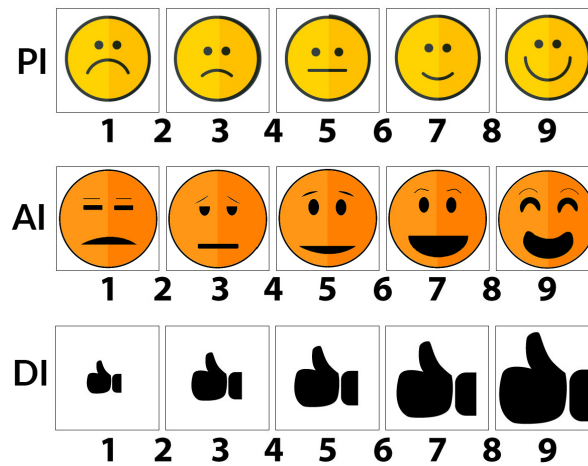


Table 3 – Emotion heuristics

Heuristic	Description
H1 - Frowning (N)	Possible need of concentration, displeasure or perceived lack of clarity.
H2 - Brow raising (N)	Sign of uncertainty, surprise, disbelief and exasperation.
H3 - Gazing away (N)	Might be a sign of deception.
H4 - Smiling (P)	Sign of satisfaction. The user might have found a moment of joy during the evaluation process.
H5 - Compressing the lip (N)	Sign of frustration or confusion.
H6 - Moving the mouth (N)	If the user is moving his/her mouth or speaking to himself/herself might be a sign of uncertainty or being lost.
H7 - Expressing vocally (N)	Vocal expressions such as sighs, gasps, coughs and the volume of these expressions might be signs of frustration or deception.
H8 - Hand touching the face (N)	Elevating one of his/her hands to the face might indicate the user is lost or tired.
H9 - Drawing back on the chair (N)	Indicate that user has negative emotions, showing a desire to get away from the current situation.
H10 - Forward leaning the trunk (N)	Might indicate frustration and depression, however, user approximation might indicate a rise in his/her attention.

simplify interactions for each task in order to decrease experiment conduction time if needed.

After the pilot test, we defined the approach to conducting the experiment. Initially, tasks would be done, in which each user should accomplish a pre-determined set of tasks (see Appendix C and D). At the end of each task, the user should fill a questionnaire with SAM and an open question to report possible difficulties or remarks about their interaction with the element. Users' face and interactions also will be recorded on video during the whole experiment process in order to conduct a detailed analysis and emotion heuristics evaluation afterward.

## 3.2 Conduction

The study was conducted over three days in different locations. Participants were invited to take part in the study voluntarily. They were undergraduate students in three different universities in Sorocaba-SP and Itapetininga-SP. All 10 participants accepted the term of consent about the use of data and images for academic ends. Participants were randomly sorted into two groups with the purpose of balancing the comparison order. One group started by using PWA and web mobile applications and the other used native applications before PWA and web.

The participants undertook the tasks individually. Initially, each participant received instructions about the applications and how to perform tasks. The differences among the three approaches were not introduced to participants since it could affect their interactions evaluation. This avoids that participants with previous knowledge about the limitations and advantages of each approach evaluated differently when compared to those who were unaware of those characteristics.

Tasks were exhibited on a computer screen in text format, where only instructions from the current task were available. All participants did not have a time limit to accomplish all tasks and used the same mobile device, a Motorola Moto G4 Play, connected to the local wi-fi network. It avoided that differences in Android operating system versions could introduce bias on interaction data. The participants' interactions with the applications were recorded by using DU Recorder <sup>11</sup>, installed on the mobile device. In addition, the participants' facial expressions were captured by a laptop camera placed in front of the participant. After finishing each task, the participant answered SAM questionnaire and an open question related to the accomplished task.

## 3.3 Analysis

Data were collected through forms, recording of user interactions on mobile device and recording of user's facial expressions. In order to discard invalid samples, a pre-analysis of 6 hours approximately of the users' facial expressions recordings was conducted. Two of the ten participants were removed from the analysis due to their faces not being completely visible on the recordings. In the end, eight participants were considered in our analysis.

First, we conducted an analysis of facial expressions. The original proposal of ten emotion heuristics was made in 2007 when mobile devices interactions were not popular. Hence, the final results of the evaluation were considered negative when five negative heuristics occurred in an event. Recently, interactions on a device might take only a few seconds and a few numbers of facial expression can be caught. Thus, we focused on the

---

<sup>11</sup> Link: <http://www.duapps.com/product/du-recorder.html>

qualitative analysis of each facial expression rather than considering the absolute number of negative and positive heuristics found in each interaction. For each facial expression identified, we assigned the correspondent heuristics.

We mapped each user facial expression to an emotion heuristic. To do this, an HCI specialist examined in parallel, side-by-side, the videos that show the user face and his/her interaction with the apps. This examination was repeated three times by the same specialist to assure a precise identification of the heuristics was conducted since a previous training was not executed. Each examination took about 45 minutes for each participant since all interactions with tasks and UI elements were analyzed. Besides pointing out the occurrence of a heuristic, the expert also made notes of when (the moment) and where (in the apps and related to which UI element) it happened. After that, the specialist associated his/her remarks with users' notes about their interactions with UI elements. This association allowed the researcher to identify the emotion or set of emotions linked to each heuristic occurrence. The goal of this mapping was to define the reason a certain heuristic occurred if it was a participants' interaction issue, an external factor and so on.

We used the open coding technique to analyze the qualitative data obtained with heuristics' mapping. Open coding consists of examining data and labeling the chunks with codes that give meaning to them. It does not use a pre-defined set of codes, on the contrary, the codes emerge from the data (STRAUSS; CORBIN, 1998).

We followed four steps to define the codes from this technique. First, we grouped the heuristics, SAM values and participants' notes from the open question in order to analyze the impact both perspectives had on each participant's interaction. Second, we gathered data obtained from the first step and grouped them by the same application platform (PWA, web or native) UI element. In doing so, we could compare if a UI element issue occurred multiple times on a platform. After that, we grouped issues by the participant, with the aims of identifying if the same issue occurred on multiple platforms for the same participant. Finally, we gathered data from step one and added codes that represent situations, reported by users and/or observed by the specialist, that triggered a set of emotions. As a result of these steps, we defined 13 different codes, as Table 4 presents. Description represents details of why such emotion happens.

Tables 5 and 6 presents summaries regarding which emotions and codes were identified, and medians obtained using SAM questionnaire in both comparisons for each UI element. The trigger of the emotions column refers to the open coding technique, as Table 4 presents.

Table 4 – Results obtained with open coding technique

Trigger of emotions	Description
C1 - Instructions misinterpretation	The participant misinterpreted task instructions and interacted with another element, not connected to the task.
C2 - Slow feedback after interaction	The response after interaction takes more time than expected.
C3 - Elements' slow loading	The UI element takes time to load while other page elements are already visible.
C4 - Satisfaction after interaction	The participant exhibits satisfaction after his/her interaction with the UI element.
C5 - User not sensing control of an element	The participant sensed the application was not being controlled exclusively by his/her actions.
C6 - User's inattention regarding element feedback	The participant got confused observing element's feedback, and mis-concluded an error occurred.
C7 - User's surprise with interaction response	The participant is positively or negatively surprised with an interaction's feedback.
C8 - Searching element on list or page	Difficulties searching for a specific element in an infinite or paginated list.
C9 - Movement/modality repetition	Repeating pinch movement or voice modality usage until the task is fulfilled.
C10 - Previous task influence	The participant was influenced by a negative interaction with the same UI element on a previous task or another UI element before the current task.
C11 - Participants' input error	The participant had typing errors.
C12 - Design flaw	The UI element's design caused an issue during participant's interactions.
C13 - Other	External factors such as participants' sickness and environment noises.

## 3.4 Results and discussion

A total of 8 users were considered in the results. Their ages ranged from 18 to 27 years old, 70% were undergraduate students, 80% used the Android operating system and 90% used the Google Chrome browser. All participants accessed the internet from a mobile device on their homes on a daily basis; from those, 90% browse social networks and search engines every day.

The presentation of results was split into subsections focusing on the *PWA-native* comparison and *Web-native* comparisons. We provided a presentation-oriented by the tasks which were linked to UI elements.

### 3.4.1 PWA-native comparison

Regarding *search input* on PWA application, three of eight participants faced from slow feedback after interaction (C2) that consequently caused frustration and uncertainty. One of these participants was frustrated with typing errors (C11). However, another one showed positive surprise and stated: "*Despite taking more time to open the app, it has*

Table 5 – PWA-native evaluation - positive emotions in bold; PI (Pleasure index); AI (Arousal index); DI (Dominance index) median results

UI element	Platform	Modality	Emotions	Trigger of emotions	PI	AI	DI
Search input	PWA	-	Frustration, <b>positive surprise</b> , uncertainty	C2, C4, C11	8	8	8.5
Search input	Native	Voice	Confusion, deception, negative surprise, <b>satisfaction</b> , uncertainty	C2, C4, C5	7.5	8	8
Datepicker	PWA	-	Confusion, frustration, negative surprise, <b>satisfaction</b> , uncertainty	C4, C5, C7	8	7	7
Datepicker	Native	-	Confusion, frustration, negative, <b>satisfaction</b> , uncertainty	C2, C4, C5, C6, C7	8	8.5	7.5
Select	PWA	-	<b>Positive surprise, satisfaction</b>	C7, C10	8	8	8
Select	Native	-	Attentiveness, confusion, deception, frustration, negative, <b>satisfaction</b> , tired, uncertainty	C2, C4, C5, C7, C10, C12	7.5	7.5	9
Map	PWA	Pinch	Confusion, negative surprise, tired, uncertainty	C3, C4, C7, C9, C12	7	8	8
Map	Native	Pinch	Attentiveness, confusion, frustration, <b>positive surprise</b> , uncertainty	C3, C4, C5, C9, C12	7	7	7
Tab menu and table	PWA	-	Confusion, negative, negative surprise, tired, uncertainty	C3, C4, C5, C8, C12	7	7	8
Tab menu and table	Native	-	Confusion, disbelief, negative, <b>satisfaction</b> , tired, uncertainty	C1, C3, C4, C10	8.5	8	9

a better, more fluid, sequence for each step". On the other hand, the third participant showed low satisfaction and motivation after the interaction. S/he selected values 4, 5 and 4 for SAM's pleasure, arousal, and dominance indexes (see Figure 7), respectively. The other five participants did not exhibit emotions related to their interactions with this UI element.

The use of the voice modality on native application's *search input* was controversial. Only two participants successfully used voice on the *search input* on their first try, and both demonstrated satisfaction after interaction (C4). A third participant also faced C4, after s/he accomplished the task on his/her the second try and stated that voice modality allows faster searching than typing. However, success using voice modality was achieved only on the second or third try for the other five participants. Uncertainty and confusion associated with the slow feedback of the voice modality (C2) were reported by one participant: *"The only problem I had was speaking into the microphone because it had a lag and I had to click again over it"*. Participants did not sense they were controlling the element after an unsuccessfully try of using voice modality, which caused uncertainty,

deception, and negative surprise. One of the five participants points that s/he would not use voice modality again since it did not work at first and selected value 4 for SAM's dominance index (see Figure 7).

Regarding PWA's *datepicker*, three participants had rich experiences, and showed satisfaction after interaction (C4) and provided values over 7 for all SAM dimensions (see Figure 7). A participant reported that the check-in date limitation fulfilled his/her expectations and another one stated that s/he was happy because the invalid check-out date could not be inserted. Still, another two participants reported they had a poor experience. The first stated *"What a confusing datepicker!"* and the other reported this element seemed slower and less intuitive when compared with the native *datepicker*. Observing these participants videos, we concluded both were confused regarding how the *datepicker* distinguishes check-in from check-out date, also, because the element was hidden by the application when a check-out date was manually included. Both sensed negative emotions like confusion, negative surprise, and frustration, and provided values lower than 6 for all SAM dimensions. This indicates they were not sensing control during their interactions (C5). Automatically hiding the *datepicker* produced a positive surprise on a participant (C7) after his/her interactions. S/he selected values over 7 for pleasure and arousal indexes on SAM, however, s/he attributed 6 to the dominance index on SAM, indicating s/he also was not sensing control during his/her interactions (C5). This was reinforced by the comment: *"Number selection is relatively small"*. Two participants did not express emotions during their interactions with the PWA *datepicker*.

The native *datepicker* had only one positive emotion occurrence, which was satisfaction after interaction (C4). The participant was satisfied because an invalid check-out date could not be inserted since the element automatically modified the check-in date to a day before the newly selected check-out date. The same automatic feedback was a negative experience to four participants, they showed confusion (C5, C7) and uncertainty (C7). Participants stated *"An inattentive user might not notice the change on check-in date"*, *"This element assumes that I want a check-in date one day before the check-out"* and *"System changes check-in date when I use it for the first time, which may cause difficulties when choosing dates"*. Another two participants misunderstood the element's response (C6), sensing uncertainty and frustration. Both commented that the *datepicker* allows users to choose a check-out date that occurs before the check-in, which did not happen according to the videos. A participant sensed confusion (C7) after finishing his/her interactions. However, this did not affect his/her overall experience since 9 was selected for all of SAM's dimensions (see Figure 7).

The PWA's *select* did not cause difficulties among participants. One participant was satisfied and positive surprised when the *select* response was fast regarding updating the hotels' list (C7). Another participant was satisfied after his/her interactions with the

element (C10). All the other six participants did not show emotions while interacting with the *select*.

The only positive emotion identified during participants' interactions with the native *select* was satisfaction (C4). The participant stated, *"I found interesting how the layout gets cleaner and straightforward when the select is on the inferior part"*. However, another participant reported a different opinion (C12) on the same issue and stated: *"Select is not grouped together with other search controls, which may not bring users' attention"*. While waiting for the hotels' list update with the new values, four participants felt slow feedback after interaction (C2), which caused attentiveness, uncertainty, confusion, deception and tired. One of these three also felt surprised by interaction response (C7) and sensed frustration. One participant accidentally chose the wrong option on the *select*. The *select* option did not have a border, which might be considered a design flaw (C12). This incident made the participant choose a lower value (5) on both pleasure index and dominance SAM dimensions (see Figure 7). Another participant did not sense emotions related to his/her interactions with this UI element during this task's execution.

The main issue reported during participants' interaction with the *map* on PWA application was the slow loading of elements (C3). Overlapping elements were instantly displayed, however, the *map* took seconds to load. This extra loading time caused six participants' confusion, negative surprise, and uncertainty. One of these participants reported that *"Pinch's response time might have some lag"*. Repetition of pinch movement (C9) was observed on two participants, including one of the six that also sensed C3, both felt uncertainty and negative surprise. One participant displayed uncertainty related to a large number of hotels on the *map*, a design flaw (C12). S/he reported, *"Map use gets complicated due to a large number of results"*. In the opposite direction, a participant stated that the PWA *map* is cleaner when compared to the native application *map*. Even with this positive comment and showing satisfaction after interaction (C4), the participant felt tired during his/her interactions. Surprise with the interaction response (C7) was identified with one participant, who also experienced C3. S/he expected the selected hotel on the *map* would be prioritized on the hotels' list, which caused uncertainty.

Elements' slow loading (C3) was also an issue on the native application *map*. Two participants sensed uncertainty when overlapping elements displayed while the *map* was still loading. Nonetheless, the major issue observed with this element was how overlapping elements, such as icons representing hotels, interfered on participants' interactions. Another four participants faced this design flaw (C12), sensed frustration and confusion, and reported this issue on the open question. Some statements are *"The information box for the first hotel on the list was unnecessary, it hampers how the map is displayed"*, *"Map has a square that hampers navigation"*, *"Information block hampers the task execution"* and *"The pop-up element on the map is too big"*. One participant had issues with repetition



of pinch movement (C9), s/he stated that the pinch movement was needed many times in order to find the searched hotel, which caused uncertainty. Differences in the hotel position are connected to the experiment's conduction being held on different days, that caused the hotel guests' evaluation numbers to change. Also regarding pinch movement, one participant who also experienced C3, felt frustrated and uncertain due to the feeling that s/he was not controlling the element (C5). S/he stated that the pinch movement was not very precise. Despite all the negative interactions, one participant sensed positive surprise and attentiveness during his/her interactions with the *map*. S/he reported that navigation was easier when compared to the PWA and, because of that, s/he found the searched hotel faster. S/he was satisfied (C4) with the interaction and selected 9 to all SAM dimensions (see Figure 7).

Searching elements on a list (C8) was the most common issue on *tab menu and table* task regarding PWA. Three participants sensed negative surprise and confusion while trying to find a hotel on the list when the hotel was located on the end of the list or on the second page. These differences in the hotel position might occur since the experiment was held on different days. One of the same three participants was unsatisfied due to *table* and list elements' slow loading (C3), s/he selected value 4 on SAM's pleasure index (see Figure 7), and felt confused and tired. S/he stated his/her preference to *tab menus* when compared to infinite lists or tables. One participant considered *table* content was located too close to each other and felt tired because of this design flaw (C12). Another participant sensed s/he was not controlling the element (C5) and pointed out the issue was connected to the *table* content format that has no filters. During interactions, s/he felt tired, uncertain, confused and wanting to get away from that situation. Still, the participant select values above 7 on all SAM dimensions, which indicates this issue did not impact on his/her overall experience. Satisfaction after interaction (C4) was experienced by one participant. However, two participants did not show emotions related to this task during their interactions.

On the native application *tab menu and table*, two participants misinterpreted the instructions (C1) and sensed confusion and uncertainty. The instructions clearly stated the hotel's name should be touched and both participants initially touched on the hotel's photo, the action that opened a photo gallery instead of the hotel's details. One of these two participants also sensed element's slow loading (C3). Images did not load on the hotel's details page for a few seconds, which caused disbelief and uncertainty. These two issues affected this participant's satisfaction and motivation. S/he selected values 4 and 5, respectively, on SAM's pleasure and arousal indexes (see Figure 7). The same issue related to C3 affected another participant's satisfaction. S/he also sensed uncertainty and selected the value 3 on SAM's pleasure index. On the other hand, one participant felt satisfaction after interaction (C4) and stated s/he was motivated because the information is more organized and complete when compared to the PWA application. Besides that,

one participant had a positive experience and sensed satisfaction and tiredness. S/he experienced previous task influence (C10) and stated that information was easier to understand when compared to the PWA application. Three participants did not express emotions during this task.

Comparing interactions on both platforms, we observed that either slow feedback after interaction (C2) or elements' slow loading (C3) provoked negative emotions in all native UI elements. This is a surprise since web applications are known for a slower performance when compared to native (CHARLAND; LEROUX, 2011). However, we also observed this finding did not impact the positive perspective regarding applications under users' viewpoint, since SAM results were mainly high values (see Table 5 and Figure 7). On the other hand, we identified a large number of positive occurrences of satisfaction after interaction (C4), which appeared in almost all UI elements excluding only PWA's *select*, that demonstrates the applications brought a pleasant experience to the user. This brings an important result to PWA applications since its premise is to bring the user experience as close as possible to the native app experience. Even though web applications are known for providing less pleasant user experience when compared to native apps (GAZDECKI, 2015).

### 3.4.2 Web-native comparison

Unlike applications from the PWA-Native comparison, web's *search input* had an auto-complete feature that displayed possible search terms after the user typed, at least, four characters. Typing errors (C11) occurred with one participant, who felt frustrated due to the auto-complete. This feature also caused a surprise with the interaction response (C7), which triggered confusion in one participant who selected the value 6 for SAM's dominance index and stated: "*The app is very suggestive regarding the auto-complete*". C7 and confusion were also identified with another participant. However, this occurrence is linked with the *datepicker* automatically opening after interactions with the *search input* were finished. Two participants sensed frustration, negative surprise, and tiredness related to a design flaw (C12) on the application's initial screen. As reported by one participant, the initial screen had too many actions, which caused this participant to select value 6 for all SAM's dimensions (see Figure 7). The second participant opened a secondary *search input* by touching a *search input* icon. S/he interacted with the wrong *search input* and navigated to a page with touristic information for about a minute, and then figured out how to return to the previous page and interact with the required *search input*. Despite this frustrating experience, the participant felt satisfied and selected values 7, 6 and 8 for SAM's pleasure, arousal and dominance indexes, respectively. One participant sensed elements' slow loading (C3) before starting the task. Nonetheless, two participants did not feel emotions related to their interactions with the *search input*.

Table 6 – Web-native coding summary - positive emotions in bold; PI (Pleasure index); AI (Arousal index); DI (Dominance index) median results

UI element	Platform	Modality	Emotion	Trigger of emotions	PI	AI	DI
Search input	Web	-	Confusion, frustration, negative, negative surprise, tired, uncertainty	C3, C7, C11, C12	8.5	8	9
Search input	Native	-	Confusion, deception, negative, negative surprise, <b>positive surprise, satisfaction</b> , tired	C1, C2, C3, C4, C7	8	8.5	7.5
Datepicker	Web	-	Confusion, frustration, negative surprise, <b>positive surprise, satisfaction</b> , uncertainty	C4, C5, C7, C10, C12	7	8	8
Datepicker	Native	-	Confusion, deception, disbelief, negative surprise, <b>positive surprise, satisfaction</b> , uncertainty	C1, C2, C4, C5, C6, C7	8.5	8.5	6.5
Select	Web	-	Frustration, tired, uncertainty	C2, C4, C7, C12	8	8	7
Map	Web	Pinch	Confusion, deception, frustration, negative, negative surprise, <b>satisfaction</b> , tired, uncertainty	C3, C4, C7, C8, C12	7	7	7
Tab menu and table	Native	-	Attentiveness, confusion, <b>positive surprise</b> , tired, uncertainty	C3, C4, C8	8.5	8.5	9

Native application *search input* also had an auto-complete function, which caused a surprise with the interaction response (C7) on three participants who sensed confusion, deception, and negative surprise. Still, all participants felt satisfied according to SAM (see Figure 7). Other two participants felt satisfaction and positive surprise after their interaction (C4). They stated *"I enjoyed more this app search experience when compared to the other ones"* and *"This application organizes better its search items, making easier to use them"*. Slow element's loading (C3) occurred with one participant, who felt tired and had to wait some seconds for the application initial load. Slow feedback after interaction (C2) was identified in one participant's interaction and caused a negative feeling on the participant while s/he waited for the hotel's list to load. However, both C3 and C2 occurrences did not affect SAM dimensions. One participant did not express emotions during his/her interactions with the native *search input*.

Uncertainty, frustration, negative surprise, and confusion were identified on the web's *datepicker* and relates to users not sensing control of an element (C5) and users' surprise with interaction response (C7). Two main aspects caused emotions among four participants, three who faced C5 and C7, the other two had C5 or C7. First, when a participant tried to insert an invalid check-out date, the application automatically changed

the check-in date to one day before the check-out. Second, one participant stated there is no easy way to close the *datepicker* since it closes automatically when a check-out date is selected. Still, only one of the four participants selected a low (5) value on SAM's pleasure index, and another one chose value 6 for the arousal index (see Figure 7). One participant stated that using other applications' *datepicker* made him have a poor experience (C10). Satisfaction after interaction (C4) occurred with one participant. Initially, the participant was confused about how the *datepicker* worked, however, after a few seconds s/he sensed positive surprise and satisfaction when accomplished the task. Emotions related to the *datepicker* were not identified in two participants.

Unlike other *datepickers*, the most common emotions were positive surprise and satisfaction. C4 was identified on four participants, who reported *"What a wonderful datepicker!"*, *"Regarding all tested datepickers, I'm more satisfied with this one"* and *"This app allows to visualize more dates with the scroll, which makes it easier to the user"*. One of the four participants was confused sense s/he was not in control of interactions with the UI element (C5). Scrolling movements caused confusion, negative surprise, and uncertainty on three participants' initial interactions. Participants' surprise with the interaction response (C7) affected two of the three participants dominance index, with values between 4-5 (see Figure 7). Another participant mistakenly stated that a check-out date that occurred earlier than the check-in date was allowed on the application. His/her inattention regarding element feedback (C6) is linked to value 5 for SAM's dominance index. One of the three participants that experienced C7 initially had the same wrong impression and sensed deception, disbelief, and uncertainty. After a few seconds, s/he realized that the check-in date was automatically changed to one day before the check-out. However, this created an impression the element was not being controlled by his/her actions (C5), which is connected to values 4, 5 and 4 for SAM's dimensions pleasure, arousal, and dominance indexes, respectively. Another participant complained about slow feedback after interaction (C2) and stated that response time was longer than other applications, and selected values 6 for all SAM dimensions. Observing SAM's medians in Table 6, we can conclude 6,5 is the lowest value on the table. This lower value may be associated with a different kind of navigation adopted on this *datepicker*. Even causing satisfaction and positive surprise in half of the participants, it provoked confusion and uncertainty in most participants.

Regarding web application's select feature, one participant felt tired with the large number (12) of options on the *select*, a design flaw (C12). Another was tired from slow feedback after interaction (C2) since the hotels' list took some seconds to reload. A participant was surprised with the interaction response (C7) and felt uncertain looking for altered values with the new *select* option. This *select* options did not have borders, similar to the PWA-native comparison's native *select*. A participant who had previous issues with this design flaw felt frustrated and uncertain thinking the same issue would occur (C10), which did not happen. The only positive experience was satisfaction after interaction (C4),

experienced by one participant. S/he reported, *"This app offers a cleaner version that makes easier to choose an item"*. Other three participants did not show emotions regarding their interactions with the *select*.

In order to open the *map* on the web application, users had to click on a button located on the hotel's details page. Four participants had issues trying to locate this button (C8, C12) and experienced deception, frustration, negative surprise, tiredness, and uncertainty. Five participants, including two that already experienced C8, sensed confusion, frustration, and uncertainty while facing slow elements' loading (C3). This occurred due to overlapping elements were displayed prior to the map's loading. Two of the five participants were also surprised with interaction response (C7) and sensed confusion, deception, and negative surprise. One participant had an issue with the web platform, s/he was confused when a new browser tab opened with the *map*. The second participant was surprised and not satisfied because s/he could not compare hotels on the map since it had only one hotel. S/he selected values 6 for both pleasure and dominance SAM indexes (see Figure 7). However, one participant felt satisfaction after interacting with the *map* (C4). S/he stated the map shows less content, which makes selecting an item easier.

Regarding native *tab menu and table*, three participants felt confused and tired due to slow loading of images and table contents (C3) on the hotel's details page. However, it did not affect the overall experience, since participants selected values above 7 in all SAM dimensions. One participant felt confused and uncertain searching for the hotel on a list (C8). Still, three participants were satisfied (C4) and reported attentiveness and positive surprise during their interactions. They stated *"Everything was easier and more practical"* and *"Unlike other apps, this shows users' evaluation in a direct way. On the other apps this information was not available in such a direct way (on the hotel's first page)"*. Emotions related to this UI element were not identified in one participant.

Observing results obtain on both platforms, elements' slow loading (C3) was a common trigger for negative emotions in native UI elements. Even being different platforms and applications, both occurrences reinforce the results obtained on the PWA-native comparison, disagreeing with the previous information stating web applications are usually slower than native (CHARLAND; LEROUX, 2011). Nonetheless, this finding did not affect users' perspective, since the results obtained with SAM have high medians (see Table 6 and Figure 7). However, satisfaction after interaction (C4) was identified on almost all UI elements, except the web's search input. This finding points participants had an overall rich user experience during their interactions.

### 3.5 Threats to validity

Four validity threats: internal, external, construction and conclusion, determined by Wohlin et al. (WOHLIN et al., 2012), were discussed.

For mitigating the internal threats, we had two actions. The identification of the platform where interaction has happened can affect participants' perception. In order to avoid this threat, participants were not informed which platform each application was developed with. All four applications were saved as home screen icons on the mobile device aiming to hamper the identification of the technology used to develop each application. Besides that, we took care of planning a reduced number of tasks in order to avoid participants fatigue.

We ensured that the selected participants were familiar with browsing the internet using mobile devices in order to avoid barriers to using this platform. Therefore, we could generalize the results to a portion of the population that commonly interact with native and web mobile applications.

Participants were randomly divided into two groups to shuffle the comparison order. Participants from the first group initially tested PWA and web applications in their respective comparisons. Group native first tested native applications and PWA and web afterward. This strategy was adopted aiming to avoid the influence of positive or negative interaction on the following application, and hence, affect the construction of the conclusions. Besides that, in order to obtain an accurate qualitative analysis, we crossed collected data from both the ten emotion heuristics and SAM instruments.

Three different data sources were used to mitigate the conclusion threats. An online application form and two recording applications were used for data gathering. One application was on the mobile device, recording the participant's interactions on the screen, and the other one was on an auxiliary laptop, recording the participant's face. With that, we could associate both recordings and gathered feedback from application forms, allowing data triangulation, adding reliability to the collected data and consequently to the analysis.

### 3.6 Conclusion

This section presented a qualitative analysis of user experience in PWA, web mobile and native applications investigating user and HCI specialist perspectives. User perspective pointed out that participants had an overall good experience, even with issues on different UI elements. HCI specialist angle allowed us to identify common issues on different platforms related to users' interaction and their causes. Associating both platforms, we concluded that even with interaction issues, participants can achieve a rich experience interacting with UI elements on different platforms. There is no bias indicating

that either a specific platform or UI element offers more enjoyable interactions.

We could not find specific studies evaluating user experience in PWA applications. Based on that, interaction comparison in a PWA with native and web applications is important to open discussions about user experience improvements this new approach can deliver. Analyzing the HCI specialist perspective was relevant, since PWA is a recent technology, and acknowledging the feedback obtained from users' allow this approach to become more mature. One of the main contributions of this study is a comparison between PWA and native applications and as far as we know this is the first analysis of its kind. Besides that, by identifying emotions and their triggers using both users' and specialist perspectives, we collected insights that bring improvements on the use of certain UI elements use.

This chapter was published as a full paper named "*Analyzing User Experience in Mobile Web, Native and Progressive Web Applications: A User and HCI Specialist Perspectives*" on the Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems (CARDIERI; ZAINA, 2018).





## 4 The PWA-EU Approach

The discussion of related works on Section 2.5 presented studies that proposed solutions for interface adaptation on mobile web (BUENO; ZAINA, 2016; BUENO; ZAINA, 2017; GHIANI et al., 2014; MANCA et al., 2013; NEBELING; SPEICHER; NORRIE, 2013; PROENÇA; NERIS, 2017). These studies are relevant due to their contribution to the mobile web app field. Still, none of the authors exploited PWA apps or EUD. Conversely, some authors investigated PWA apps but left out any regards to end-users (BIØRN-HANSEN; MAJCHRZAK; GRØNLI, 2017; FORTUNATO; BERNARDINO, 2018). Based on these studies, we proposed our initial experimental study to compare end-user interaction on native, web and PWA apps (see Chapter 3). This study pointed out that there are no differences between users interactions on these three mobile platforms.

Our related work section also brought discussions about works on EUD field. These works propose frameworks, tools, and models that allow user empowerment on mobile and desktop applications (DANADO; PATERNÒ, 2014; COSTABILE et al., 2008; BOSETTI et al., 2017; NAMOUN et al., 2016). After combining the findings of the first experimental study, related works, and concepts of EUD and Meta-Design (see Section 2.4), we noticed the potential in the association of PWAs and meta-design. This potential occurs as both approaches have an adaptive nature by becoming more complete when users interactions increase. Yet, the traditional PWA approach does not empower users and does not take their preferences into account.

Finally, by combining the findings of the first experimental study and the discussion of the related work, we propose PWA-EU. Our proposal is an extension of the PWA architecture in which users' preferences are considered to adapt the interface during run time. During their interactions, users can access and modify options such as appearance, interaction methods, and the content that is displayed. These options support users on fulfilling preferences that were not previously identified by designers and developers. The PWA-EU approach empowers users to take their own decisions about what they want in that interaction.

### 4.1 Architecture Proposal

Figure 8 shows the PWA-EU architectural scheme. The PWA-EU architecture is displayed on the right side, while on the left side the traditional PWA is shown. In our approach, the architecture was designed to enable end-user empowerment. We defined three architectural elements into The PWA-EU architecture, the *User Design Model (UDM)(A)*, *Management Layer(B)* and *Connection Layer(C)* (see Figure 8). Our architecture is an

adaptation of the event-driven and the layered architectures for software development (RICHARDS, 2015). The event-driven part is relevant since the PWA-EU architecture depends on events triggered by users, such as modifying the  $UDM(A)$  or loading a new UI. The layered part relates to the separation of components with distinct functionalities. On the PWA-EU scenario, these components are responsible for connecting the app with storage systems (*Connection Layer(C)*) and defining user preferences (*Management Layer(B)* and  $UDM(A)$ ). Moreover, the connection with the traditional PWA approach occurs by adapting the UI (*Adaptation Engine(F)*). These three elements and the connection with the *Adaptation Engine(F)* are discussed in the following subsections.

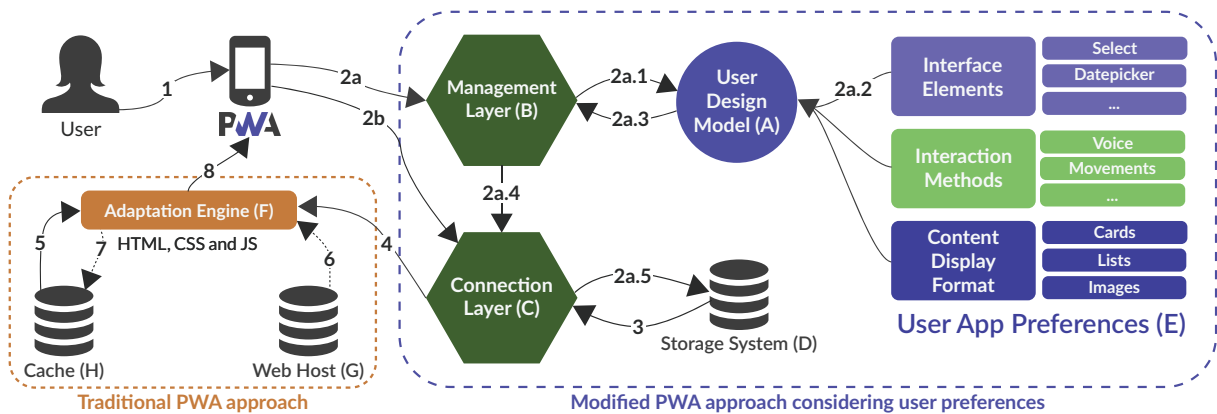
The interface adaptation on the PWA-EU proposal was outlined considering the definition of adaptable and adaptive interfaces proposed by Gullà et al. (2015). The authors of that study emphasize the line between these two categories is very thin and each one has advantages and disadvantages. Hence, we selected a combined approach between these two categories. PWA-EU's UI adaptation is based on adaptive UIs regarding the way the system performs the adaptation. Still, it is also adaptable as the adaptation process is monitored by end-users who select their user app preferences on the UDM.

With regards to EUD, PWA-EU is based on the meta-design framework as it follows most of the guidelines proposed by Fisher, Nakakoji & Ye (FISCHER; NAKAKOJI; YE, 2009) (see Section 2.4). Following, we describe how PWA-EU meets these guidelines:

1. **Support Human-problem Interaction** - Designers and developers should analyze the app domain and its end-users in order to specify their problems. After this analysis, the user app preferences are defined and included as part of PWA-EU's UDM.
2. **Underdesign for Emergent Behaviour** - A PWA developed with PWA-EU has default options that can be modified by end-users via the UDM. Moreover, a new design behavior is generated by architectural elements every time the user changes their user app preferences on the app. Yet, end-users are not in charge of design responsibilities as the preferences were previously defined by designers and developers.
3. **Enable Legitimate Peripheral Participation** - New users can contribute to the app by selecting their user app preferences. The Management and Connection Layers assure these preferences will be visible during users' interaction.
4. **Share Control** - Designers and developers are the original metadesigners who define multiple user app preferences during development. During run time, the control of these preferences, that affect the app's functionalities and appearance, is transferred to end-users who are acting as co-designers.

Three of the guidelines proposed by Fisher, Nakakoji & Ye (2009) were not followed on PWA-EU: 5. *Promote Mutual Learning and Support*, 6. *Reward and Recognize Contri-*

Figure 8 – PWA-EU architectural scheme



contributions and 7. Foster Reflective Communities. These guidelines are related to knowledge sharing features and the first version of PWA-EU do not contemplate these one. However, PWA-EU architecture is flexible and new elements can be added to implement such features.

#### 4.1.1 PWA-EU Architectural Scheme

The main goal of the *User Design Model (UDM)* shown in Figure 8(A) is to establish and combine different *User App Preferences* (see Fig 8(E)). This model was based on the user model proposed by Benyon & Murray (1993). On this proposal, end-users' preferences and characteristics are stored on this model and may be altered by the end-user. We adapted Benyon & Murray's (1993) model into the *UDM(A)*, as the user preferences are defined on runtime by end-users and stored on the app. Still, during design/development time, it is the designers' and developers' role to analyze the app's domain and the targeted end-users to define the set of preferences that will be available to end-users on runtime. Some examples of the *User App Preferences(E)* are UI elements appearance, interaction methods, and content display formats such as data and images. While planning the *UDM(A)*, the designers and developers should include, at least, a default and a secondary option for each *User App Preference(E)*, such as lists and cards to display content. Finally, the *User App Preferences(E)* must have default settings and follow meta-design principles, in which users can change their preferences.

Developers must select a technology to store these preferences and be the foundation of the *UDM(A)*, such as the JavaScript Object Notation (JSON) format. JSON is a widely adopted approach on web development for both sending and receiving requests that can be easily transformed into a JS object (FINK; FLATOW, 2014). An example of a *UDM(A)* implemented with JSON is detailed in Figure 9.

The *Management Layer* on Figure 8(B) is responsible for including, editing and deleting the *User App Preferences(E)* on the *UDM(A)*. It can receive two different requests

Figure 9 – UDM Example in JSON

```
{
  'id': attribute_id,
  'type': attribute_type,
  'name': attribute_name,
  'value': attribute_value
}
```

from the PWA: (i) receiving and sending modifications to the  $UDM(A)$ , and (ii) sending an updated  $UDM(A)$  to the *Connection Layer*( $C$ ). This layer is requested only when users modified their design preferences into the  $UDM(A)$ . Moreover, this layer is responsible for all algorithms needed to edit the  $UDM(A)$  structure to modify, include and delete new *User App Preferences*( $E$ ). The  $UDM's(A)$  structure should not be modified by other elements and no connections to *Storage Systems*( $D$ ) should be created.

The *Connection Layer* on Figure 8(C) has three functions: (i) sending the  $UDM(A)$  to the *Storage System*( $D$ ), such as `LocalStorage`<sup>1</sup>, (ii) retrieving the  $UDM(A)$  from the *Storage System*( $D$ ), and (iii) sending the latest  $UDM(A)$  to the *Adaptation Engine*( $F$ ). The first scenario only occurs when the  $UDM(A)$  was updated by the user on the *Management Layer*( $B$ ). The other scenarios happen when it is necessary to load a new UI on the PWA, which can occur when a new  $UDM(A)$  *User App Preference*( $E$ ) is set or when the user is browsing through the app. Further, the *Connection Layer's*( $C$ ) code should include only functions that connect to the defined *Storage System*( $D$ ). This element should not modify the  $UDM(A)$  structure, only transport it.

Both *Management*( $B$ ) and *Connection Layers*( $C$ ) were created by two reasons: (i) as a way to create a software architecture that can be replicated and more easily understood by software developers; and (ii) keep the traditional architecture of a PWA on the *Adaptation Engine*( $F$ ) to clearly separate it from the PWA-EU architecture.

The *Adaptation Engine* (see Fig 8 (F)) represents the combination of the traditional architecture of a PWA with the PWA-EU approach. On the traditional architecture, proposed by Google (2018b), the PWA must have an application shell that stores the UI (i.e. HTML, CSS and JS files) and is instantly loaded when an end-user opens the app. The combination with PWA-EU comes as this engine takes into account the *User App Preferences*( $E$ ) defined on the  $UDM(A)$  and combines these preferences with the application shell in order to alter the app's UI. With that in mind, all CSS and JS files that somehow modify the PWA UI can be considered part of this engine. Another task performed by this element is obtaining new files from the *Web Host*( $G$ ) if an internet

<sup>1</sup> [https://www.w3schools.com/html/html5\\_webstorage.asp](https://www.w3schools.com/html/html5_webstorage.asp)

connection is available. Further, this engine should store the app UI on *Cache(H)* memory and recover it when necessary.

We decided to define an *Adaptation Engine(F)* based on the studies from Manca et al. (2013) and Ghiani et al. (2014). Both studies define proposals in which an *Adaptation Engine(F)* is part of the UI adaptation process. More specifically, this engine is responsible for deciding how the adaptation will occur in these studies. As PWA-EU adaptation requires end-user participation and is based on PWAs, we adapted how this element works considering our proposal's fundamentals.

### 4.1.2 Privacy

As pointed out by Proença & Neris (2017) and Gullà et al. (2015), privacy is one of the main concerns of users. Individuals feel insecure about sharing their personal data in an application. Due to that, this approach recommends storing data from UDM locally on the browser using HTML5's `localStorage`<sup>2</sup> instead of a relational database located in a remote web server. Hence, users' personal information will not be sent to a web server, anticipating possible security issues.

### 4.1.3 Single-page Application Frameworks

One possible approach to developing an app shell architecture is using Single-page Applications (SPAs). This approach proposes that one HTML page is used as a shell for all other web pages and that all end-user interactions are implemented using HTML, CSS, and JS. A SPA enables apps to be less dependent on web server interactions, not needing to reload a page on each navigation. SPAs are similar to native apps regarding behavior, nonetheless, they run inside a browser (FINK; FLATOW, 2014).

SPA is an approach recommended by Google to develop PWA applications' architecture (Google Developers, 2018b). Recently, a high amount of SPA frameworks are available to the front-end development of a web, hybrid or PWA application. Some of the most popular SPA frameworks are Angular<sup>3</sup>, React.js<sup>4</sup>, and Vue.js<sup>5</sup>.

## 4.2 Proof of Concept

Calendar is a PWA app that allows users to manage events and their time (i.e. event name, date, start time, end time, location and a category). Calendar was chosen because it represents a common sense domain avoiding the need for users learning about

---

<sup>2</sup> [https://www.w3schools.com/html/html5\\_webstorage.asp](https://www.w3schools.com/html/html5_webstorage.asp)

<sup>3</sup> <https://angular.io/>

<sup>4</sup> <https://reactjs.org/>

<sup>5</sup> <https://vuejs.org/>

the application domain. Moreover, we were interested in exploring a set of UI elements in the same app to provide a diversity of interaction with users.

To build the Calendar, we considered the most popular frameworks in web mobile app design, such as Bootstrap<sup>6</sup>, Foundation<sup>7</sup> and Materialize<sup>8</sup>. As PWA is a Google proposal based on Material Design, we chose Materialize as the default design for Calendar's UI elements. Material Design consists of guidelines, components, and tools that support the best practices of UI design available as open-source code (CLIFTON, 2015).

At the start we set up the *user app preferences*. After conducting an investigation on Material Design guide and considering the nature of the app (i.e. Calendar), we decided to cover five *UI elements*: calendar, checkbox, input, select and timepicker. We also introduced a second option in which users acting as co-designers could select. For each element, users acting as co-designers could choose between a Material Design element and a browser-default appearance as part of the *user app preferences* of the UDM. A browser-default look changes according to the device and browser the user is accessing. For instance, the input from Safari Mobile for iOS shows a vertical scroll for the day, month and year, while on Chrome Mobile for Android it displays a pop-up calendar.

Gestures and voice (interaction methods - see Fig 8) were available. The swipe gesture was selected due to some Material Design components recommending its use, such as lists and cards. The voice synthesis and speech recognition were included considering they became a standard feature on smartphones (CORBETT; WEBER, 2016). Complementing the user app preferences, *content display formats* were selected to provide users with distinct ways to read the information on the app. Following Material Design guidelines, we defined that users can choose between list and cards formats to display data, besides showing or hiding images on the cards.

Recently, many JS frameworks became available to develop the front-end part of a web app (i.e. Angular<sup>9</sup>, React.js<sup>10</sup>, and Vue.js<sup>11</sup>). Still, we chose Vue.js due to three factors: (i) it has a fast learning curve (Tech Magic, 2018; NEUHAUS, 2017; TARNOWSKI, 2017), (ii) its applications are smaller to store when compared to the other frameworks (TARNOWSKI, 2017; VUE.JS, 2018), and (iii) availability of an official Vue.js PWA template<sup>12</sup> (SHEPPARD, 2017).

Each architectural element from PWA-EU and the *Adaptation Engine* are represented on the Calendar app. All three PWA-EU architectural elements (UDM, Management,

---

<sup>6</sup> <https://getbootstrap.com/>

<sup>7</sup> <https://foundation.zurb.com/>

<sup>8</sup> <http://materializecss.com>

<sup>9</sup> <https://angular.io/>

<sup>10</sup> <https://reactjs.org/>

<sup>11</sup> <https://vuejs.org/>

<sup>12</sup> <https://github.com/vuejs-templates/pwa>

Figure 10 – UDM in Calendar App

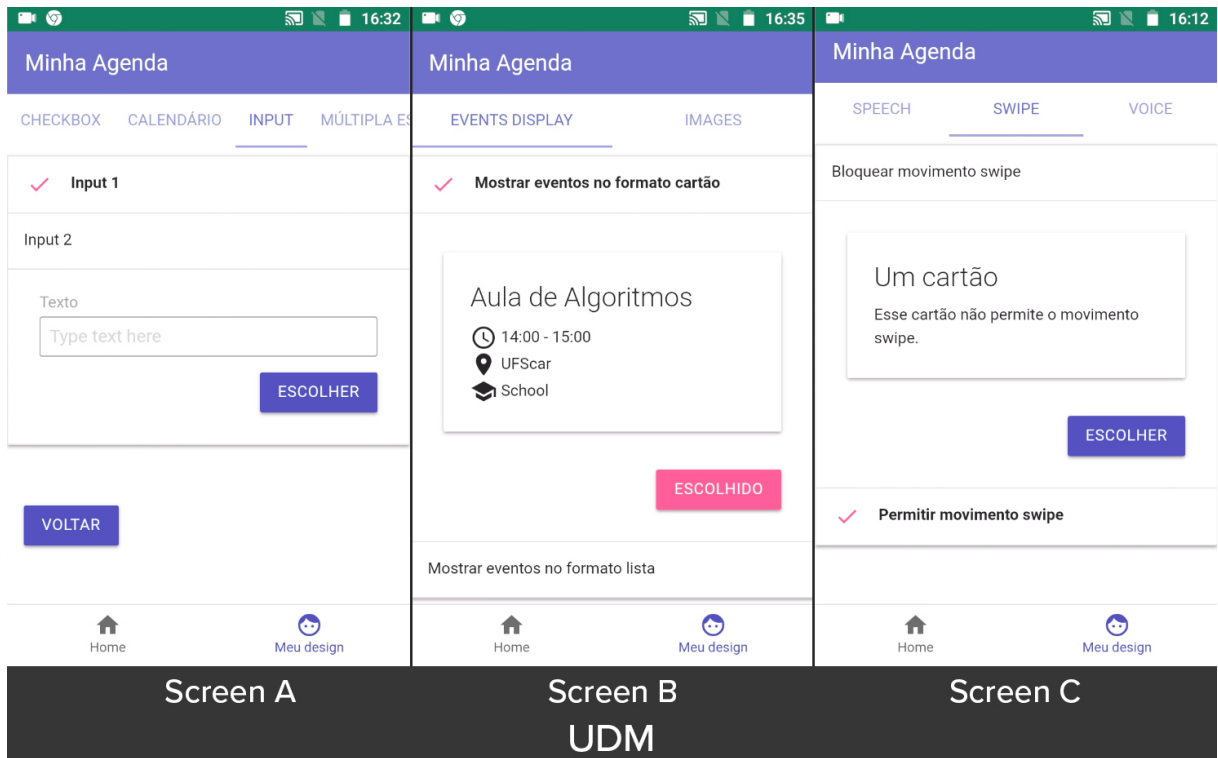


Figure 11 – Adaptation Engine in Calendar App as Content Display Formats

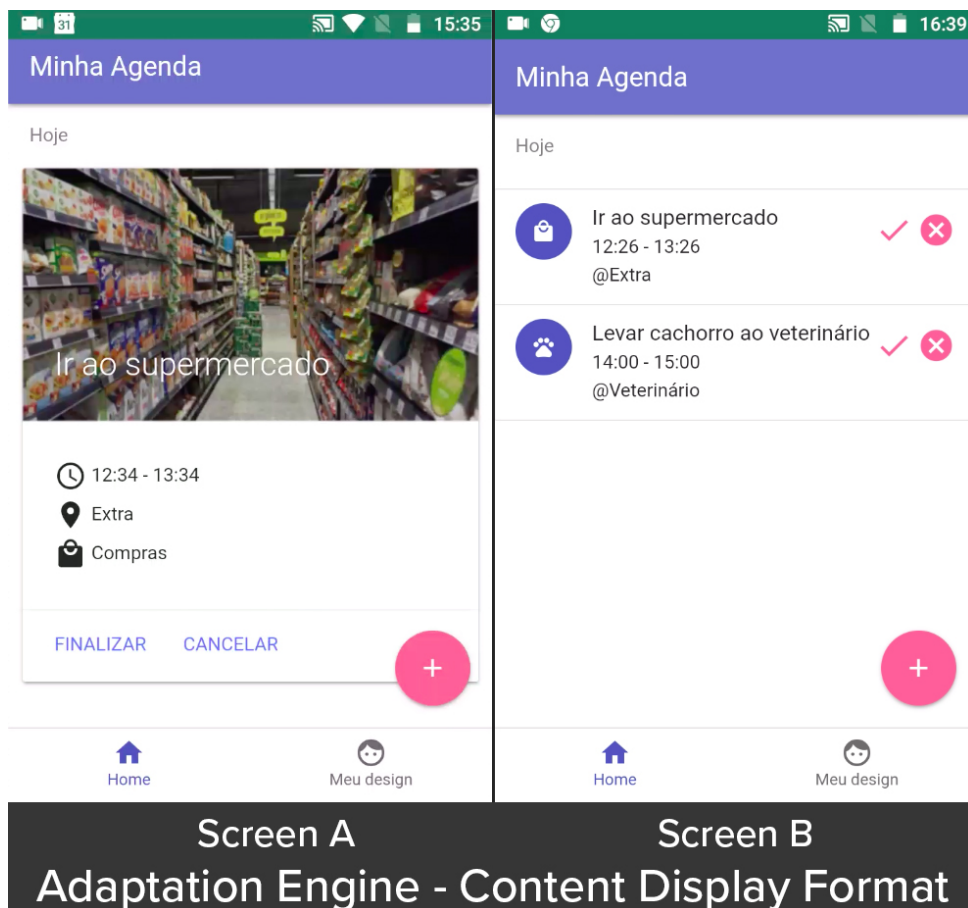
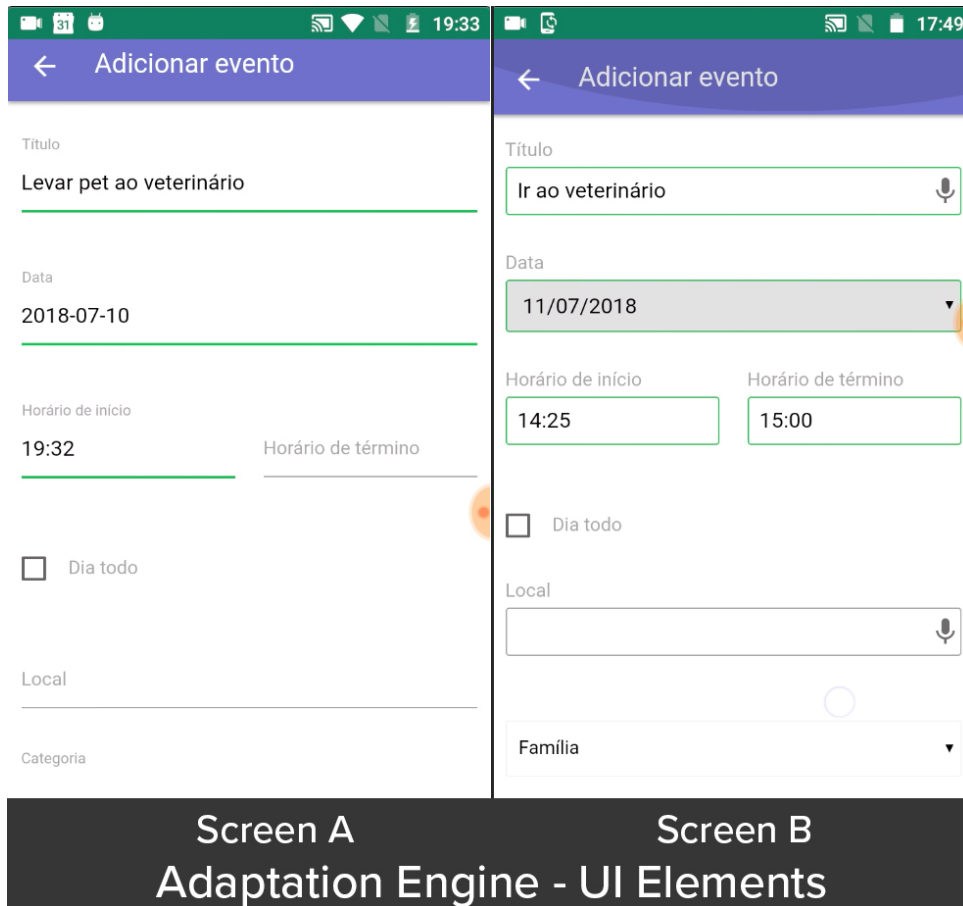


Figure 12 – Adaptation Engine in Calendar App as UI elements



and Connection Layers) are defined in the My Design section, which is shown in Figure 10. On this section, participants could select and edit their *user app preferences*. While the *UDM's* values were visually represented by My Design's UI, the *Connection* and *Management Layers* were built as sets of JS functions. *Connection Layer's* functions store and retrieve data from LocalStorage and the *Management Layer's* code changes the values of *user app preferences* on the *UDM*. On the other hand, the *Adaptation Engine* is formed by multiple CSS and JS files that modify *user app preferences* on Home and Include New Event sections. Figure 11 shows differences on the Home section regarding content display formats, i.e. lists (Screen B on Fig 11) and cards (Screen A on Fig 11). Figure 12 shows distinct UI elements preferences on the Include New Event section.

### 4.3 Final Considerations

There are no technologies tied to developing applications applying PWA-EU. This approach is extensible and may integrate technologies chosen by the developers. We opted for this extensible nature since new technologies rise frequently on mobile app field. Still, we suggest both Connection and Management layers be composed by JS functions, the use of LocalStorage as storing system and JSON as the basis for the UDM structure.



---

In the next chapters, we describe the evaluation of PWA-EU architecture, which includes the results from both end-user and developers perspectives. This proposal was presented with the title "*PWA-EU: uma abordagem para o desenvolvimento de aplicações PWA baseadas em EUD*" at the 3rd Thesis and Dissertations Workshop at the Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems.



# 5 Analyzing PWA-EU from an End-User perspective

This chapter presents an experimental study conducted in order to analyze end-users' experience when they act as co-designers using the PWA-EU proposal. We based our study on the guidelines proposed by Lazar (2017), the following sections present the planning, conduction, and analysis of the study.

## 5.1 Planning

PWA's adaptive nature brings great potential to be associated with meta-design since both approaches allow apps to become more complete while the user builds a relationship with it. Due to that, we wanted to investigate issues and the UX of end-users acting as co-designers on PWAs. To conduct our investigation, we defined two research questions (RQs): (i) *What communication breakdowns come up when users play the role of co-designers in a PWA?* and (ii) *How are users' perception of UX when interacting as co-designers?*

Participants were invited to take part in the study voluntarily via social networks like Twitter, Facebook, and Instagram. Nonetheless, in order to obtain a more diverse sample, half of the recruited participants were professionals or students who act in a technological area and the other half was not from that area and both groups were selected by convenience. We did not introduce any age limitation for participation in the study.

A questionnaire to collect participants' profile was elaborated (See Appendix F). We collected data such as gender, age, level of education, mobile operating system and browser they use, their professional area or course they study, the reason to download native apps, the frequency of use of voice synthesis on mobile devices, internet access from mobile devices and accessing specific websites.

Three tasks related to the Calendar were established with the aim of supporting users' interactions with the PWA. The initial task allowed *co-designers* to select their preferences. In the second and third tasks, all participants had to include, cancel and finish events. Participants' had access to the list of tasks via an offline-based web application developed by the authors (Available at <https://lgt6snm.dlvr.cloud>).

We selected the Communicability Evaluation Method (CEM) (LEITÃO; FARIA, 2009) to guide our analysis of communication breakdowns (RQ1). We used the method to separate into codes the recurring issues and interactions regarding users' communication

breakdown's situations. CEM is a semiotic engineering-based method that aims to explore communication breakdowns between the designed system and the user through the observation of how a group of users interacts with a particular system. Considering that in our study part of the users are also the designers (i.e. co-designers), these breakdowns can point out whether these individuals' roles as co-designers lead to more satisfying interactions, and understand if even acting as co-designers users still have communication breakdowns. Moreover, we took the Self Assessment Manikin (SAM) (BRADLEY; LANG, 1994) to collect the participants' experience after their interactions with the application (RQ2). SAM is a pictograph evaluation method to measure emotional responses from some sort of stimulus. Three dimensions are considered by this technique: pleasure (if the participant had a positive or negative reaction), arousal (body stimulation level from an event or object) and dominance (feeling in control of the situation or controlled by it). The user chooses a value on a scale of one to nine on each dimension, using images, to represent their emotions after interactions. Participants answered questions regarding the three dimensions after each task. Furthermore, when all tasks were accomplished we conducted an interview with four questions to have a broader comprehension of the end-users' perspective.

A pilot test was carried out with two participants with the aim of verifying whether instructions and tasks were sufficiently clear to the users or whether they needed improvements. We also checked if the mechanisms applied to gathered data (i.e. video collecting application, interface recording app, offline SAM questionnaire app) working acceptably. We concluded that no changes were necessary and the study could be run with the participants.

## 5.2 Conduction

The study was conducted over four days in October 2018 in different public locations in Sorocaba-SP. We decided to run the study in public locations such as shopping malls to simulate a more realistic environment of the use of smartphones. We had a total of 20 participants who were between 18 and 59 years old, median 23 years old. Most participants were undergraduate students or had a degree. All participants accepted the term of consent about the use of data and images for academic ends (See Appendix E).

The participants' observation took place individually and the researcher made notes regarding each communication breakdown. Initially, each participant received instructions about Calendar and the conduction of the study. Only the description of the current task was displayed in text format on a laptop screen. We did not introduce a time limit to the participants accomplished the tasks. All the participants used the same mobile device, a Motorola Moto G4 Play running Android 8.1. This avoided that differences in Android

operating system versions could introduce bias on interaction data. The participants' interactions with the PWA were recorded by using DU Recorder<sup>1</sup>, installed on the mobile device. In addition, the participants' facial expressions and voice were captured by a laptop camera placed in front of the participant. After finishing each task, the participant answered a SAM questionnaire related to the task they attempted to undertake. After finishing all the tasks each participant was interviewed by the researcher.

We obtained two distinct groups of participants who were separated based on their academic and professional experience. One group had participants from the technology field (developers and designers) and the other had participants from other professional and academic fields. For ease of reference, we will refer to the technology field participants as *tech* and *non-tech* for the group from other areas. This separation is similar to what was done in by Namoun et al. (2016) and it: (i) avoids that only individuals with previous knowledge on UI design or development acted as co-designers, (ii) provides a more diverse group of participants, and (iii) indicates whether or not a difference stands between end-users *co-designers* who have a technical profile and those who do not have.

Besides the technical profile separation, half of the participants were randomly selected and named *co-designers* due to having active participation in the setting of the UI design. The other ten participants interacted only with features that did not make changes on the UI and were named *non-designers*.

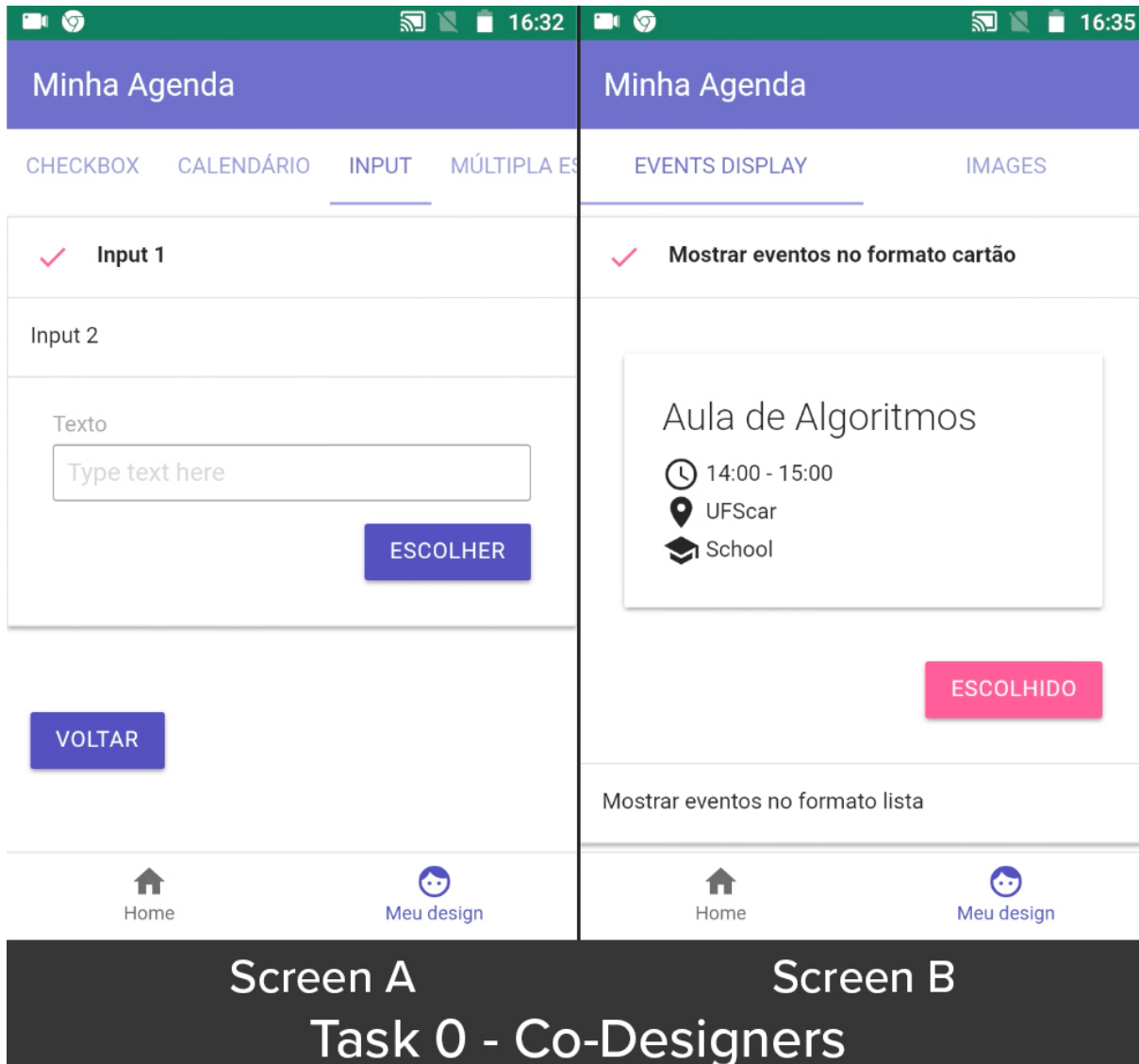
For the *co-designers*, three tasks were proposed. The tasks were specified as follow: **Task 0 - Co-designers** - participants could set the UDM by selecting their preferences regarding UI elements (See Screen A - Fig 13), interaction methods, and display format options (See Screen B - Fig 13); **Task 1 - Include events** - participants included two new events on the app (See Fig 14); and **Task 2 - Browse events** - Participants canceled (See Screen A Fig 15) and marked as finished (See Screen B Fig 15) the previous included events. *Non-designers* run only Task 1 and Task2.

The think-aloud protocol (PREECE; SHARP; ROGERS, 2015) was adopted to provide a better understanding of users interaction and to assist the application of CEM method. This protocol requires users to speak their thoughts during their interactions. Following the CEM method, we prepared a semi-structured interview that had four initial questions: (i) *Do you think that the choices you made regarding UI elements, interactions methods and content display format contributed for a better UX during your interactions with the app?* (ii) *Did you have difficulties while selecting design options?* (iii) *Did you have difficulties while adding events on the app?* (iv) *Did you have difficulties while browsing through events on the app?*

---

<sup>1</sup> <http://www.duapps.com/product/du-recorder.html>

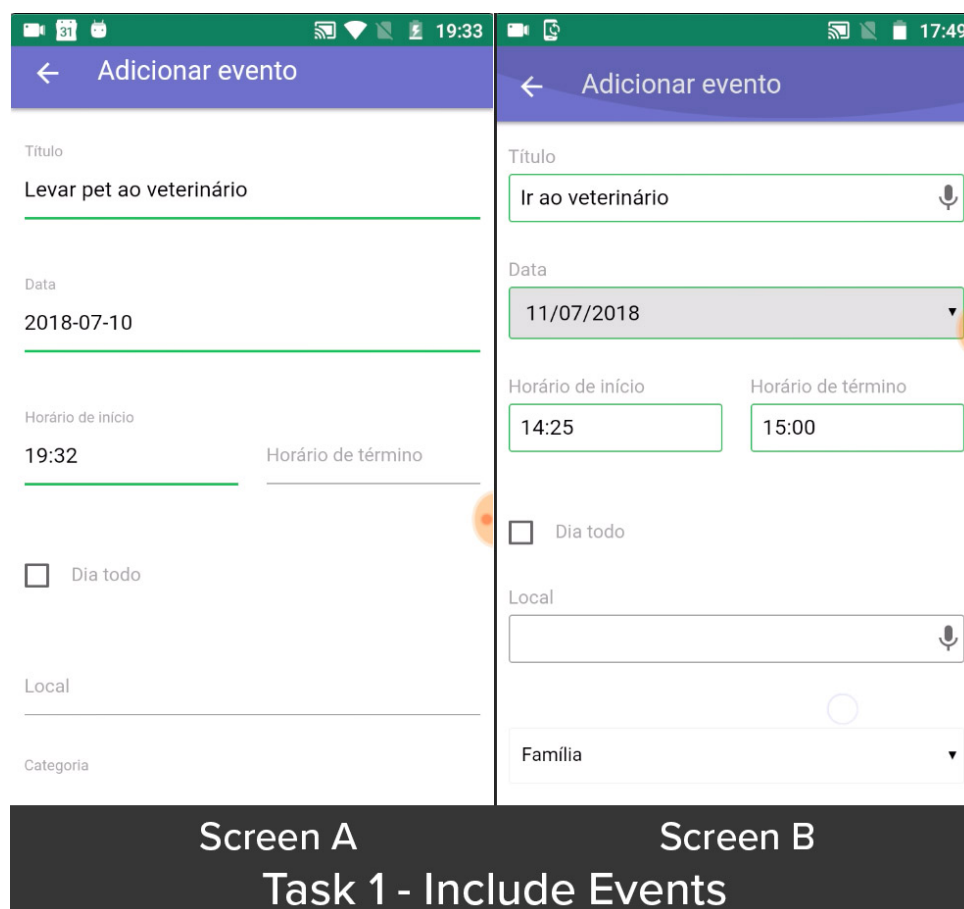
Figure 13 – Task 0 - Co-designers



### 5.3 Analysis

The analysis was conducted by crossing-over four data sources (i.e. web app questionnaire, video recordings, screen recordings, and the main researcher's notes). Approximately five hours of video recordings from user interactions on the mobile device and recordings of user's faces were collected. On top of that, around two hours of audio from the interviews were collected. To discard invalid samples, a pre-analysis of all video and audio recordings was conducted. During this step, two samples were discarded. These participants were *co-designers*, from both *tech* and *non-tech* groups, who did not complete Task 0, which resulted in an incomplete selection of user app preferences. Considering part of our analysis is based on the communication breakdowns, these participants were removed from our analysis. In the end, we had 18 participants, from which eight acted as *co-designers* and the others as *non-designers*.

Figure 14 – Task 1 - Include events



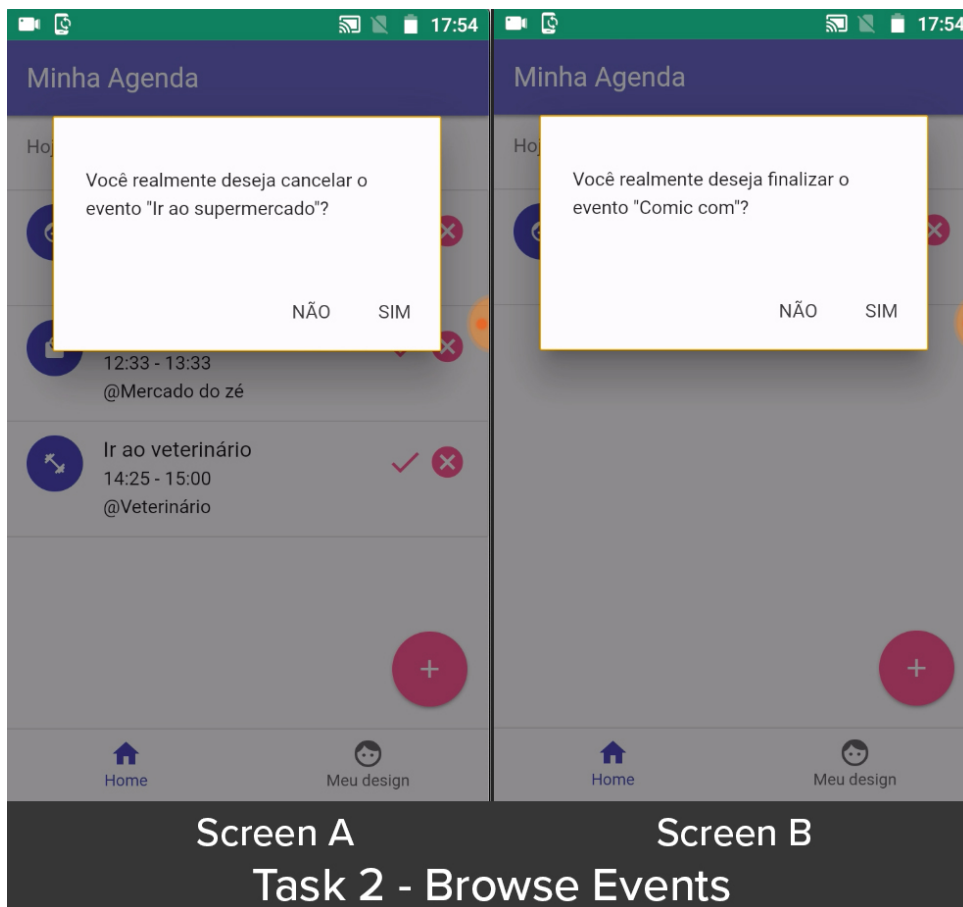
First, the videos containing the interaction on apps and the users' facial expressions were put together on the same video aiming to enable easier and more precise identification of each evidence of a communication breakdown. For this, we used the iMovie Software<sup>2</sup>. Regarding the interview, we transcribed to text all conversations recorded on audio between the users and the researcher.

After all the videos were combined, we started CEM's tagging step. The main researcher watched all video recordings and assigned a tag when a communication breakdown was identified. We used the 13 tags proposed by the CEM method (PRATES R. O.; BARBOSA, 2000). These tags are natural language expressions commonly found in human communications, which the participants might utter during the thinking aloud process. With the aim of identifying the breakdowns, the researcher watched all videos twice and took note of when, why and where it occurred. This information was saved into spreadsheets.

In the second round, the chunks that were tagged were revisited for the identification of the main communication breakdowns. Each tag was listed and analyzed in two perspectives: (i) the frequency and context of occurrence which was identified by

<sup>2</sup> <https://www.apple.com/imovie/>

Figure 15 – Task 2 - Browse events



listing when, where and the probable reason each tag occurred, and (ii) the existence of pattern in sequences of tag types, such as grouping tags connected to the similar communication breakdowns on multiple or single participants. This analysis was conducted creating spreadsheets and associating tags connected to the same breakdown. In each step of tagging, the main researcher conducted a double-checking of the results.

In the third round, we cross-over the issues that have caused the observed breakdowns, with the participants' profile, SAM answers, recordings and interview data.

## 5.4 Findings

We organized our findings in three subsections. In the first one, we present the participants' profile and in the others, we answer the RQs.

### 5.4.1 Profile

Table 7 shows participants' profile. Overall, most participants were under 30 years old and had, at least, two years of professional or academic experience. Moreover, all participants from the *tech* group had at least one year of experience.



Table 7 – Participants profile - group, technical profile (T. Profile), age, education and professional/academic (P/A) experience.

Participant	Group	T.Profile	Age	Education Level	P/A Experience
P1	Co-designer	Non-tech	23	Undergraduate Student	2-4 years
P2	Co-designer	Non-tech	18	High School	N/A
P3	Co-designer	Tech	22	Graduated	5 years or more
P4	Co-designer	Non-tech	25	Graduated	1-2 years
P5	Co-designer	Non-tech	25	Graduated	Less than a year
P6	Co-designer	Tech	21	Undergraduate Student	2-4 years
P7	Co-designer	Tech	25	Graduated	2-4 years
P8	Co-designer	Tech	22	Graduated	5 years or more
P9	Non-designer	Non-tech	59	Graduated	5 years or more
P10	Non-designer	Non-tech	23	High School	2-4 years
P11	Non-designer	Non-tech	25	Undergraduate Student	5 years or more
P12	Non-designer	Tech	32	Specialization	5 years or more
P13	Non-designer	Non-tech	22	Undergraduate Student	1-2 years
P14	Non-designer	Tech	23	Undergraduate Student	5 years or more
P15	Non-designer	Tech	25	Graduated	1-2 years
P16	Non-designer	Non-tech	21	High School	1-2 years
P17	Non-designer	Tech	23	Undergraduate Student	5 years or more
P18	Non-designer	Tech	23	Undergraduate Student	5 years or more

### 5.4.2 Communication Breakdowns

We analyzed the communication breakdowns that were caused by the *co-designers* choices because only these participants interfered on the app's design. These are problems influenced by the user preferences selected when participants were acting as co-designers. Moreover, we counted seven communication breakdowns which came up from five participants on a total of eight *co-designers*. The breakdowns were categorized into four of the 13 tags provided by CEM. Figure 16 shows the tags that represent the breakdowns which arose when the users were interacting with the Calendar app. The rightmost diagram indicates which breakdowns also occurred as a consequence of co-designer actions. Following we concentrated on discussing the tags from the third diagram. The other breakdowns are out of the scope of this work due to they are related exclusively to UI elements and technological expertise issues.

**T6 - What happened?** is identified when users repeat an operation because they could not see or understand the effects of their actions. We saw evidence of T6 from [P4] interaction on Task 2. Playing the role of co-designer (i.e. performing Task 0), [P4] changed the default design of *content display format* (see Fig 8) from card to list. Explaining the preference after the execution of Task 0, [P4] said: *"I prefer the list rather than card viewing...it shows the same data but allows me to see more data at the same time"*. Additionally, the list format changed how the cancel and finish buttons were displayed in a way the buttons contained only symbols (i.e. an X and a checkmark,

Figure 16 – Participants communication breakdown tags divided by co-designers/non-designers, tech/non-tech and co-designers from both tech/non-tech.

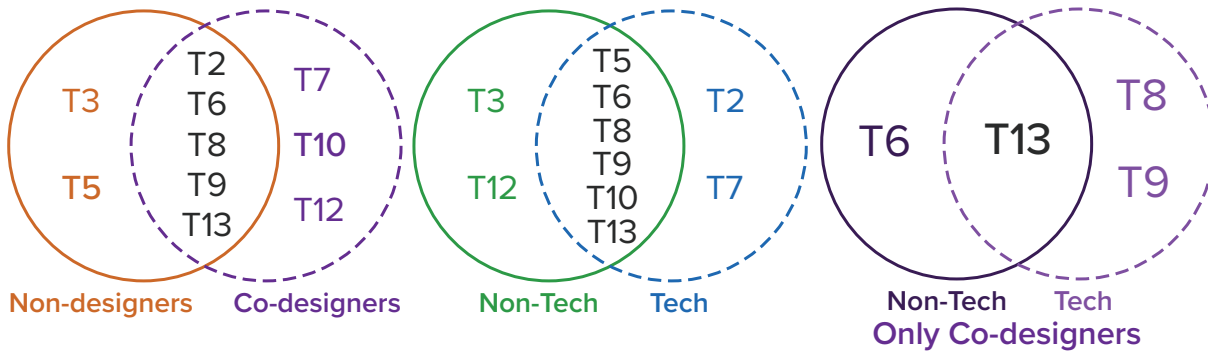
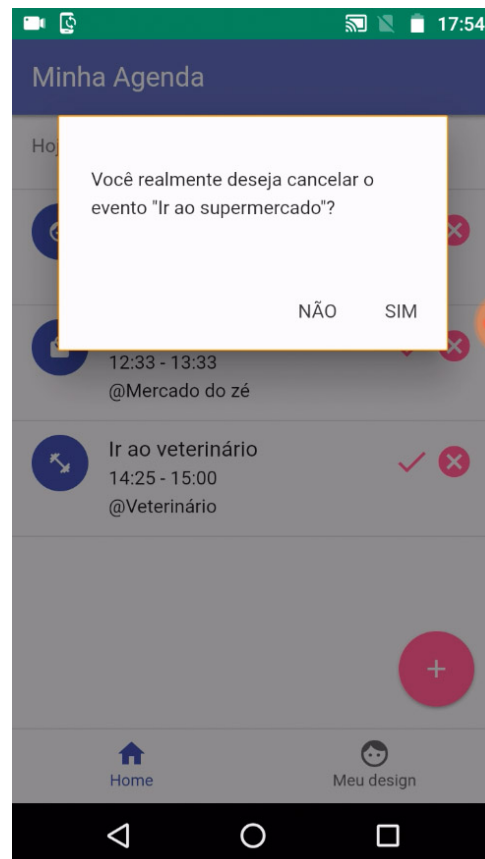


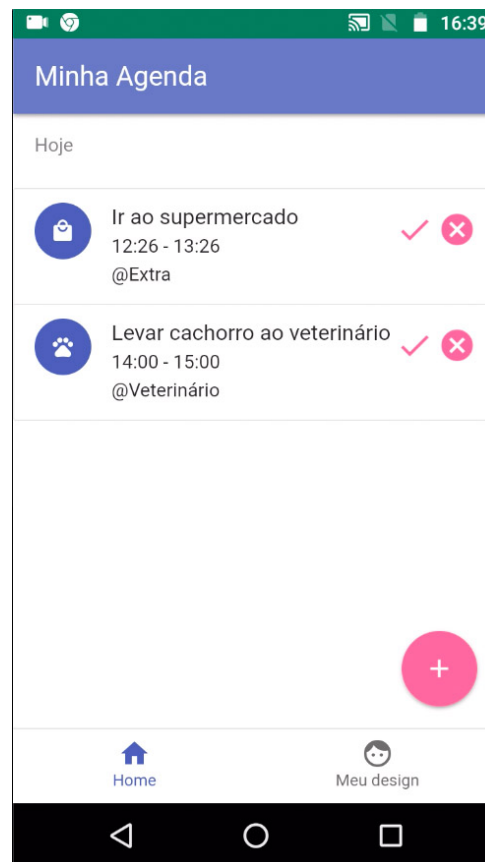
Figure 17 – The cancel event modal window



respectively) representing the actions instead of the text format from when the card format was selected (see Fig 18). While performing Task 2 [P4] was not sure which button canceled an event. After a time of hesitation, [P4] pressed the cancel button and the app displayed a modal window requesting the participant to confirm the action (see Fig 17). Rather than confirming the cancellation, [P4] closed the modal and reopened it right away, this time confirming the action. On the interview [P4] stated *"That was very practical, I just tested it once to be sure I was clicking on the correct button"*.

**T8 - Where am I** tag represents the communication breakdown when a user did not find a particular feature by pausing and searching it. T8 was identified on [P8]'s

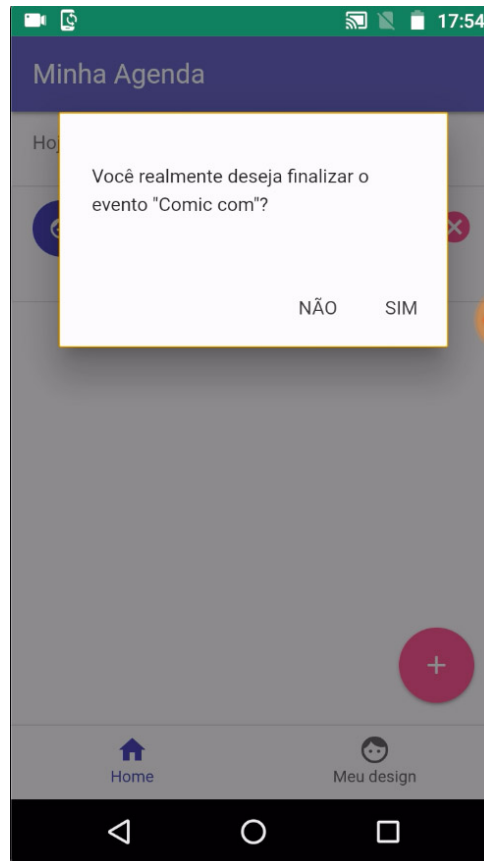
Figure 18 – The main screen when the list option was selected



interactions on Task 2, who likewise [P4] selected the list as default *content display format*. Commenting on the selection, [P8] said: *"It's an advantage as is faster to check events using the list format"*. Nonetheless, [P8] was insecure about which button would perform the action to mark an event as finished, as the buttons had only symbols and not text (see Fig 18). S/he stated: *"Mark the event as finished? ... Let me click here ... oh, ok! .. Yay!"*. By analyzing this snippet, we noticed that initially s/he was not sure which UI element would perform this action. Yet, [P8] figured it out in a few seconds and explained this breakdown over interview *"I didn't realize the finalize button was there, I initially tried to click over the list"*. However, a couple of minutes after this statement [P8] recalled this issue affirming *"Finalizing an event was easy."*

The **T9 - Oops** tag occurs when a user momentarily makes a mistake and immediately corrects it. S/he sees that s/he has made a wrong step and usually activates the “undo” function immediately. This breakdown was identified on [P7]’s interactions on Task 2, a *tech co-designer*. Resembling [P4] and [P8], [P7] selected the list as the preferred *content display format* on Task 0. S/he stated: *"I prefer the list because it fits more data into the screen, it's very clear and not cramped."*. Yet, s/he had issues to understand the difference between cancel and mark as finished buttons (see Fig 18). While trying to finish an event, [P7] spoke *"The button is asking me if I want to finish an event... I don't want to finish .. wait? Is it the same thing?... finish an event, ok."*. By [P7]’s speech, we observe

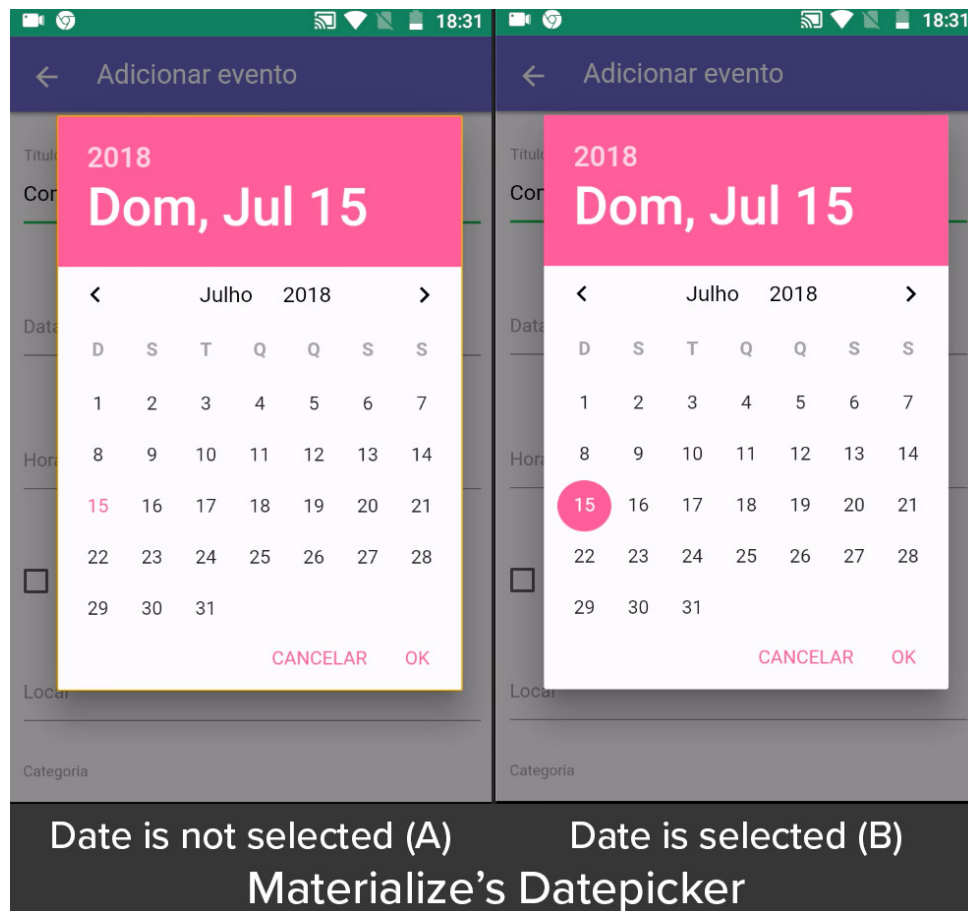
Figure 19 – The mark as finalized modal window



a misunderstanding regarding the verbs cancel and finish (see Fig 19). Over the interview, s/he explained: *"The action buttons were not clear, I understand one of them is positive and the other is negative. Still, the message was very similar in both actions"*.

On the **T13 - Why doesn't it?** tag, the user is trying to make sense of the designer message by repeating the steps of previous unsuccessful communication in order to find out what went wrong. The issue experienced by [P7] in the previous paragraph is also connected to T13. On Task 2, [P7] repeated the 'mark as finished task' until s/he understood the difference between this and the cancel action. Additionally, T13 also occurred with two other participants on Task 1, in which participants had to include an event. [P1] and [P2], *co-designers* from the *non-tech* group, selected the Materialize's *datepicker* instead of the browser-default. The default behavior of Materialize's *datepicker* displays the current day with a distinct font color that led participants to believe a date was selected (see Fig 20(A)). Still, in order to choose a date, users had to touch over a day and it would acquire a different background color (see Fig 20(B)). This behavior is a feature from Materialize, which is not considered a design error by the library. Both participants did not comment on why they selected this *datepicker* instead of the second option, which did not present this behavior. Yet, [P1] explained *"I did not enjoy the datepicker because instead of performing two commands I wish I could complete it in one."*. Conversely, [P2] did not state difficulties. Other three participants selected this *datepicker*,

Figure 20 – Materialize’s Datepicker - the default option



[P3] from the *tech* and [P4], [P5] from the *non-tech* group. However, [P4] had the same breakdown with the Materialize UI element on Task 0, which lead him/her to select the other *datepicker* option.

As Materialize’s *datepicker* was the default on the Calendar app, we can make a comparison of the T13 breakdown with the *non-designers*. For *non-designers*, four (P11, P13, P15, and P17) of the ten participants experienced a similar issue. Two (P15 and P17) of these four were from the *tech* group. This comparison points out that even people who are acquainted with technology may have issues understanding Materialize’s *datepicker* behavior. Most *co-designers* from the *tech* group selected the second option, which displays a UI element associated to the browser.

Even though three participants (P4, P8, and P7) had distinct issues related to the list and action buttons on Task 2, two other *tech* participants, [P3] and [P6], also selected the list as their *content display format* preference. Both [P3] and [P6] did not experience breakdowns, instead, both stated in the interview that the action buttons were very clear. As the list was not available as *content display format* for participants who did not act as *co-designers*, we could not make a relation with the *non-designers* participants.

Further, we looked at the results of the interview. One question asked whether the

*co-designers* felt they had better interactions due to their design choices. Six from eight participants felt positive about this question, [P8] felt neutral and [P2] disagreed. [P2] stated *"The choices don't change a lot of things"*. Yet, s/he was a *non-tech* participant who initially had difficulties understanding how the design options would be later applied. The *tech* participant [P8] said the choices did not interfere with posterior interactions. The other six participants explained the preferences selection made the navigation easier and more pleasant to interact. *"The app was more pleasant as it identifies with your choices."*[P3]. *"The app gets easier to interact"*[P4].

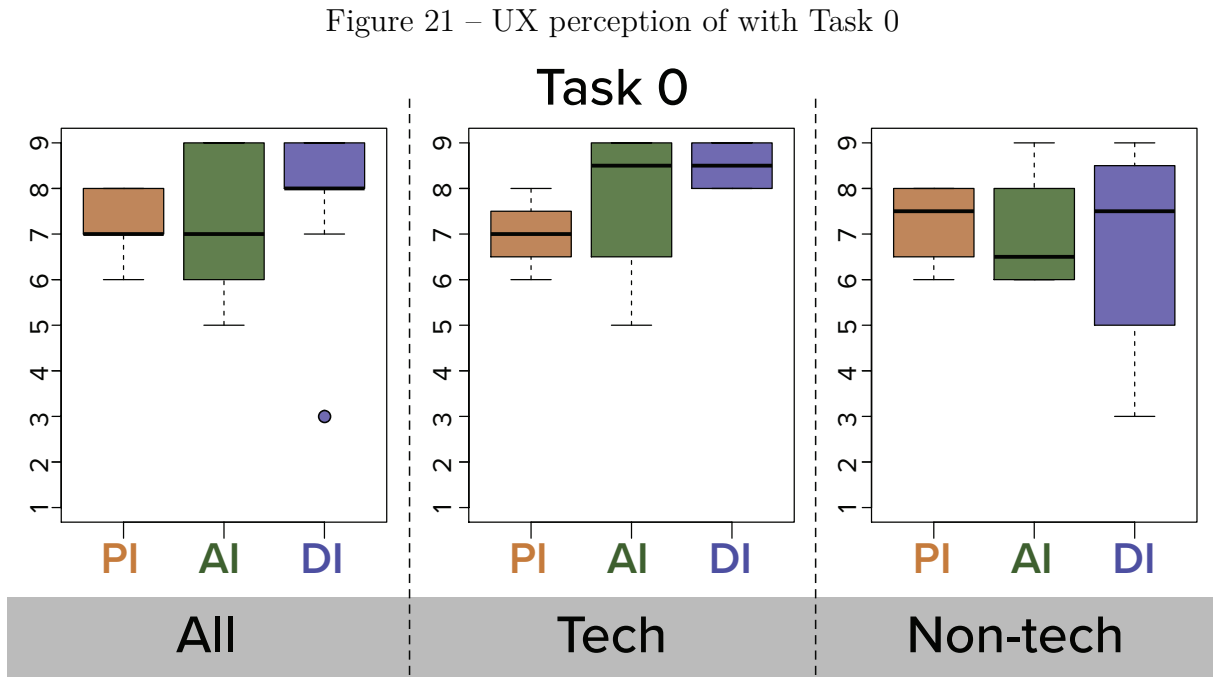
### 5.4.3 UX Perception

Our discussion on UX perception is performed on a holistic overview throughout the interpretation of the users' feedback from SAM dimensions. Furthermore, we answer the RQ2 *"How are users' perception of UX when interacting as co-designers?"*. Table 8 summarizes the SAM dimensions and the responses of the participants per task. When we look at participants' results divided by their technical profile, we notice a contrast between them. Hence, Figure 21 presents a comparison of both groups on the *co-designers'* feedback on Task 0. Further, all the SAM medians for *co-designers* are higher on Task 1 in which interactions with UI elements are linked to user app preferences. Yet, there were no differences in the SAM values for Task 2, as users' choices were not as visible as Task 1.

Table 8 – SAM results. PI - Pleasure Index, AI - Arousal Index, DI - Dominance Index

Participant	Group	Technical Profile	Task 0			Task 1			Task 2		
			PI	AI	DI	PI	AI	DI	PI	AI	DI
P1	Co-designer	Non-tech	8	7	9	9	9	9	9	9	9
P2	Co-designer	Non-tech	7	6	8	9	8	8,5	9	9	9
P3	Co-designer	Tech	8	9	9	7,5	7,5	8,5	9	9	9
P4	Co-designer	Non-tech	6	6	3	8	8,5	8,5	8	8	7,5
P5	Co-designer	Non-tech	8	9	7	9	9	9	9	9	9
P6	Co-designer	Tech	7	9	8	8	9	8	8	9	7
P7	Co-designer	Tech	6	8	8	5,5	5,5	6,5	5,5	4	5,5
P8	Co-designer	Tech	7	5	9	9	9	7,5	9	9	9
P9	Non-designer	Non-tech	-	-	-	9	9	9	9	9	9
P10	Non-designer	Non-tech	-	-	-	8,5	9	8,5	9	9	9
P11	Non-designer	Non-tech	-	-	-	7,5	7,5	7,5	9	8	9
P12	Non-designer	Tech	-	-	-	7,5	8,5	9	9	9	9
P13	Non-designer	Non-tech	-	-	-	7	7,5	6,5	7	9	6
P14	Non-designer	Tech	-	-	-	7,5	9	9	8	9	9
P15	Non-designer	Tech	-	-	-	7	8,5	7,5	7,5	8,5	7,5
P16	Non-designer	Non-tech	-	-	-	7,5	7,5	8	9	9	9
P17	Non-designer	Tech	-	-	-	7	8	5	5	7,5	3,5
P18	Non-designer	Tech	-	-	-	8	9	8	8,5	8	9

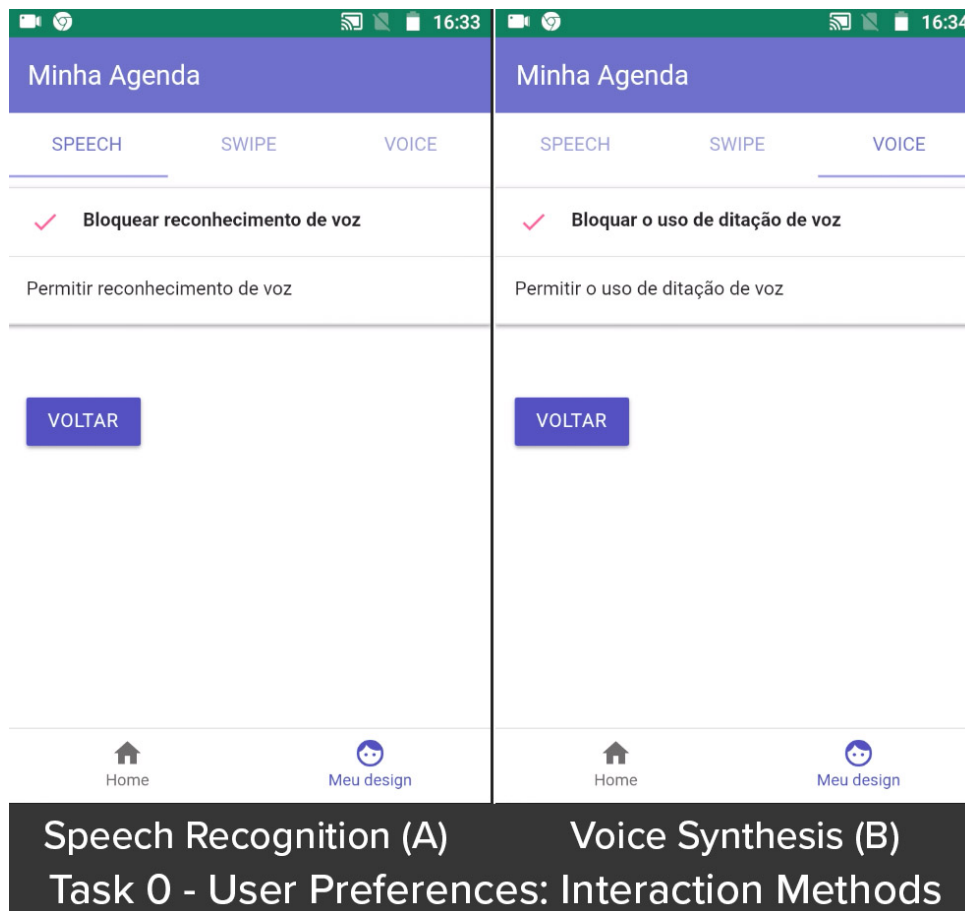
By associating the results of Table 8, Figure 21 and taking into account all the



*co-designers* (i.e. tech and non-tech), we see an outlier regarding dominance index. This is a consequence of [P4] having selected value 3 for the index. Considering the videos of the participants' interactions, we noticed that three *non-tech* users, [P2], [P4] and [P5], had difficulties in understanding how to initiate Task 0 and were confused about their role as co-designers. Yet, they were having doubts about why and how their preferences would affect their interactions. [P4], for instance, was browsing through the design options for a minute and asked the researcher for help: "I can't understand what I should do here". On top of that, [P2] had questions about the consequences of the changes: "Where can I see this element if I choose it?". Moreover, [P5] said: "But how do I know what is the difference between this or that checkbox?". Briefly, the researcher answered all the questions and the participants had no further issues regarding their initial difficulties. Even though [P4]'s SAM values appear as outliers when we take all *co-designers* into account, his/her dominance index value is not an outlier when we focus only on *non-tech* participants. As pointed out on Figure 21, *non-tech* individuals who took the role as *co-designers* selected lower values on arousal and dominance SAM dimensions, which might be the cause of this remark.

[P2], [P5] and [P7] had difficulties as co-designers. However they struggle more on choices of *interaction methods* (see 8). Both from the *non-tech* group, [P2] and [P5] asked the researcher for help, and [P2] stated: "This speech recognition, what does it specifically do?". Still, [P2] was confused about other design choices such as *swipe movement* and the *datepicker*. In both situations, the researcher gave a brief explanation about what the design choices meant and participants continued their tasks. Yet, the *tech* group participant [P7] stated that "I don't like to use voice, I don't like voice synthesis or speech recognition

Figure 22 – The selection of User Preferences on Task 0 - Interaction Methods

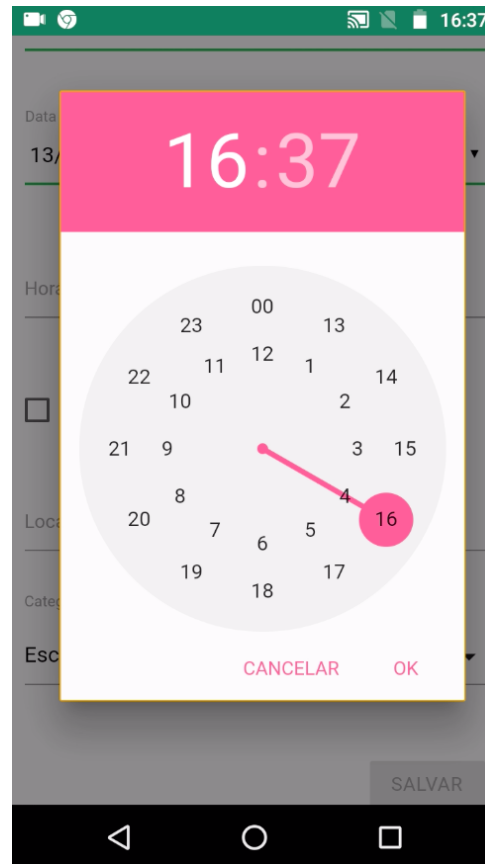


... I don't understand why both methods are separated, but it's fine" (see Fig 22). This issue could also be one of the causes of [P2] having attributed lower values for pleasure and arousal indexes. On the other hand, [P5] selected high values for both pleasure and arousal indexes. As the only representative of the *tech* group, [P7] selected 6 for pleasure index. However, [P7] chose high values for the other SAM indexes pointing out that the issue did not affect the experience.

By observing Figures 25 and 24, we see the homogeneous feedback of the participants independently of the task and the group they took part. Task 1 boxplots show outliers linked to [P7] and [P16]. Both participants are from the *tech* group, yet, [P7] was a co-designer and [P16] was not. Comparing the data from both CEM and SAM methods, we found out that [P7] and [P16] had a common issue that was also experienced by [P1], [P2], and [P3]. The issue came up during the use of *timepicker* element, in which they had to include an event with a specific time slot (see Fig 23). Actually, all the participants had some kind of difficulties while selecting a precise time slot. We noticed this issue on both *timepicker* options, materialize and web pattern. During the interview, [P16] related issues regarding the *timepicker*'s accuracy: "It's nice to have a *timepicker* that spins to a precise time, but I had many difficulties trying to select both zeros from 15:00. I'm not sure if it is too accurate or not accurate at all". No other participants commented on this issue.



Figure 23 – Materialize’s Timepicker - the default option



Taking a look at SAM indexes of [P1], [P2] and [P3], we notice that this issue did not affect their UX. Nonetheless, [P3] had lower values when compared to [P1] and [P2]. The results showed that this barrier has affected only the SAM indexes of the *tech* participants.

The issue related to the *datepicker* described in the previous subsection affected only part of the participants. When we observe [P1] and [P2] SAM results, we notice they did not select lower indexes on Task 1. The *non-designers* (P11, P13, P15, and P17) who had this issue selected lower values on SAM indexes for Task 1. This points out that the *co-designers'* experience was not affected by this breakdown, yet, the *non-designers* had a less satisfactory experience.

Also connected to the communication breakdowns, the issue regarding the list UI element did affect *co-designers* UX. In Figure 24 and 25, we see the outliers in the boxplots of *co-designers* (pleasure and arousal indexes) and of *techs* (arousal index) group from Task 2. Besides that, all of SAM indexes of [P4] are pointed out as outliers on the *non-tech* boxplot. Yet, the values from [P4] did not come as outliers on the *co-designers* boxplot. The experience reported by [P8] showed the communication breakdown s/he has faced did not bring consequences to his/her experience due to SAM values are high.

Other two outliers on Task 2 for pleasure and dominance indexes were found from [P12] feedback when seeing as the *non-tech* participant. His/her issues are related

Figure 24 – Boxplots with SAM results categorized by participants role

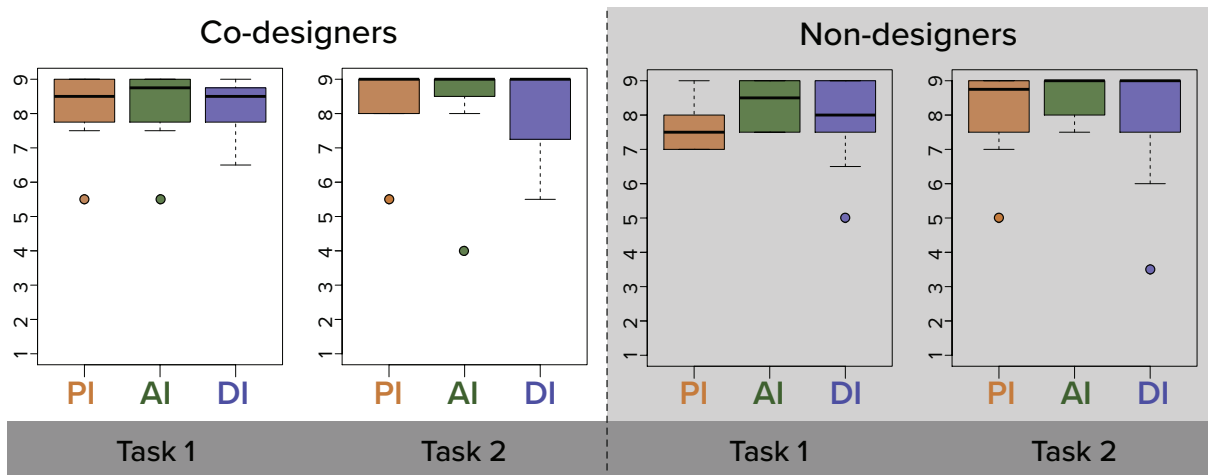
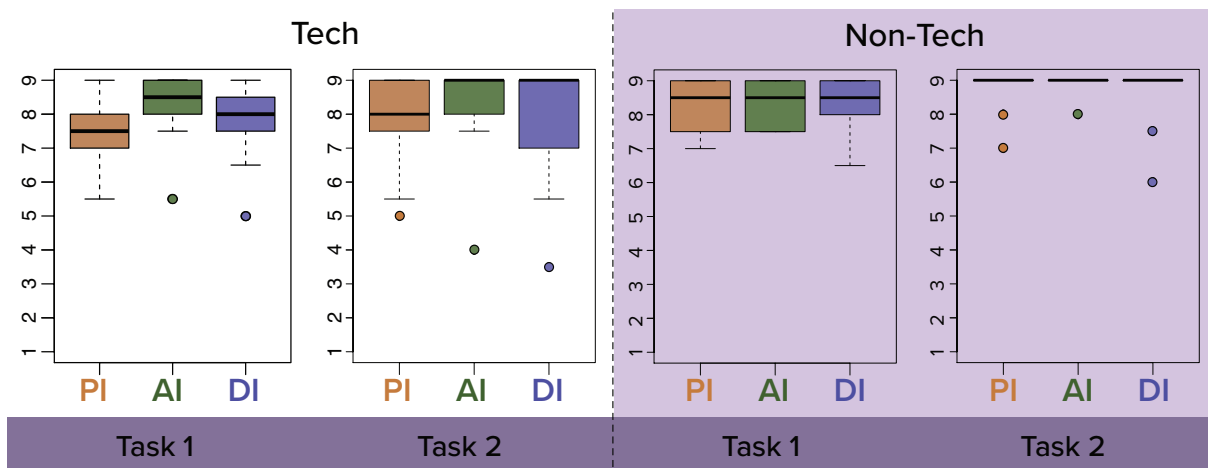


Figure 25 – Boxplots with SAM results categorized by participants experience



to his/her own expectations regarding the app (i.e. s/he wished a specific functionality existed), not their UI elements or the PWA-EU approach. Instead, the lower values arose when the participant wanted extra functionality on the app in which the finished and canceled events could be restored. [P12] stated: *"Wow, the event I marked as finished just vanishes? I'm sad"*. Likewise, [P12] and [P16] outliers presented in Task 2 for pleasure and dominance indexes on both non-designers and *tech* boxplots are linked to the same expectation.

## 5.5 Discussion

Taking into account the communication breakdowns results, we answer the RQ1 *"What communication breakdowns come up when users play the role of co-designers in a PWA?"*. Even when users acted as *co-designers* they experienced communication breakdowns. This remark is affected by users' technical background, such as programmers and designers who are used to building and designing applications. Tech participants had

fewer breakdowns as they had more expertise in the behavior and functionality of different UI elements than *non-tech* ones. Although EUD brings interest and motivation to *non-tech* end-users, their lack of technical skills makes them not engage as *co-designers* (NAMOUN et al., 2016).

Supporting by SAM results we can answer RQ2 "*How are users perception of UX when interacting as co-designers?*". By playing the role of *co-designers* the participants had a more positive perception as their active participation in the app design could improve their experience. This remark could be noticed especially in pleasure index. When considering the technical profile, the *tech* group demonstrate less satisfaction than the *non-tech*. However, while playing the role of *co-designers* the *non-tech* individuals felt less dominant and aroused comparing to the *tech* ones. Technical issues can introduce situational limitations to *non-tech* end-users (CASADEI; GRANOLLERS; ZAINA, 2017). These limitations are any kind of problem that comes up from environmental characteristics and may jeopardize the UX (HENRY; ABOU-ZAHRA; BREWER, 2014). One important motivation for end-users engaging in EUD actions is the improvements that it can bring to their experience (FISCHER; FOGLI; PICCINNO, 2017). Yet, one of the biggest challenges to overcome is to simultaneously give end-users the empowerment on designing apps and provide ways to support them with how to do EUD (NAMOUN et al., 2016). In order to provide a more guided interaction, we could establish guidelines for the development of the PWA-EU. For instance, only after the user interacts with a specific user app preference (i.e. a UI element) the different preference options are displayed. Before the user's initial interaction, an element or functionality is not visible as a user app preference. Further, the app section in which the user can select their preferences is progressively built according to user interactions.

We notice that most issues pointed out in the results section are related to three UI elements: datepicker, timepicker, and list. With that in mind, we can discuss if the identified breakdowns and issues that affected the user's experience is connected to the interaction with UI elements. First, despite bringing satisfaction to users, datepickers can cause confusion and frustration on PWA users. As we pointed out on Chapter 3. Moreover, many users have issues while including data using forms on the mobile app as these elements have not changed much since their first design. Finally, lists seem like a practical design for users, yet it raises serious accessibility concerns (CASADEI; GRANOLLERS; ZAINA, 2017). Still, likewise to our initial experimental study (See Chapter 3) participants achieved a satisfying UX even with issues when interacting with the UI elements.

## 5.6 Limitations of the study

As the sample of our work can be considered a limitation (i.e. 18 participants) we took different actions to mitigate this as discussed below. First, we adopted four distinct data sources, an online questionnaire application, researcher' notes, and two recording applications. A recording application was on the mobile device and recorded the participants' interactions with the PWA. The second was a desktop app on an auxiliary laptop that recorded the participants' faces. Therefore, we associated both recordings and gathered data from the questionnaire, which allowed us to triangulate data that provides reliability to the analysis. Yet, about the data gathered we assured all participants used the same mobile device, a Motorola Moto G4, avoiding bias which can be caused by different Android versions. By adopting different methods to conduct the data analysis (i.e. SAM and CEM), we could explore the data on different lenses and perspectives and consequently getting rich discussions of the results.

Besides, we divided participants into two different conduction groups. The first completed all three tasks and acted as *co-designers* on the app. The second group was the non-designers and they did not actively participate in the application's design choices. All selected participants were familiar with mobile devices and apps. We ensured they were frequent users in order to avoid problems related to the interaction with the mobile platform. Still, our work is limited to the use of certain UI elements. These elements were part of the user app preferences and are connected to the user's experience, as pointed out in the discussion.

## 5.7 Conclusion and future work

This chapter presented an investigation about end-users acting as co-designers on PWAs. The findings reveal that people with technology background were less satisfied, but felt more dominant when acting as co-designers. Conversely, grouping participants as those who acted as co-designers and the ones who did not, we noticed the co-designers group had more satisfying interactions on the posterior task when compared to the ones who did not include their preferences into the PWA app.

The discussion about PWA apps and EUD are an important contribution to our work. The findings of our study provide insights for new investigations in different perspectives. First, it opens the discussion of how end-users could be effectively empowered on mobile apps and whether meta-design is an easier approach to implement with technology-related people. The results show that there are some difficulties in playing the role of co-designers and these can have a direct impact on the overall UX. Working on more guided mobile apps could be a start point to achieving an inclusive environment to end-users. Second, our work presents an important discussion regarding the potential of

combining PWAs and EUD. PWA-EU is an approach that can be extended to explore other perspectives of EUD. It could be seen a contribution for developers of apps in the sense it arranges the responsibilities of the application.

As future work, we intend to continue our analysis of the PWA-EU approach. we plan to analyze the end-user perspective from a more diverse group of users taking into account different end-users profile and increase our sample.



## 6 Analyzing PWA-EU from the perspective of developers

After evaluating the proposal from end-users' point of view, we conducted an experimental study with 23 participants focusing on the evaluation from the perspective of developers. The study aimed to analyze the acceptance and difficulties that the PWA-EU elements can bring to the developers. We selected participants from undergraduate and graduate computer science courses. We made this choice considering that Salman et al. (SALMAN; MISIRLI; JURISTO, 2015) provided pieces of evidence of a small difference between students and professionals when they perform new activities. Even though professional developers have more experience, they are not familiar with new knowledge such as PWAs, which enables them to be compared to students. The motivations for this study were to understand if developers can understand the PWA-EU architecture and whether they find it useful. In order to conduct our investigation, we defined two research questions: *RQ1 - What do the developers acceptance when using PWA-EU approach?* and *RQ2 - How is developers' feedback regarding the usefulness of the PWA-EU approach?*. This study was organized into planning, conduction and analysis steps.

### 6.1 Planning

The first step in planning was the elaboration of a pre-study questionnaire to understand the participants' profile (see Appendix G). This questionnaire was necessary in order to conduct an analysis to prepare the supporting materials to our study (i.e. the technologies needed to build a PWA). We asked about their knowledge in front-end technologies and which programming technologies they were familiar with. For the front-end technologies, we collected their experience using a five-item Likert scale. Regarding programming technologies knowledge, the questionnaire inquired if their experience was restricted to academic works or if they had acted as developers in companies. Additionally, the participants' experience with web, mobile, hybrid, and PWA development was collected. This questionnaire was filled by all participants 17 days prior to the study conduction via Google Forms. Moreover, participants were selected by convenience and with no restriction of age.

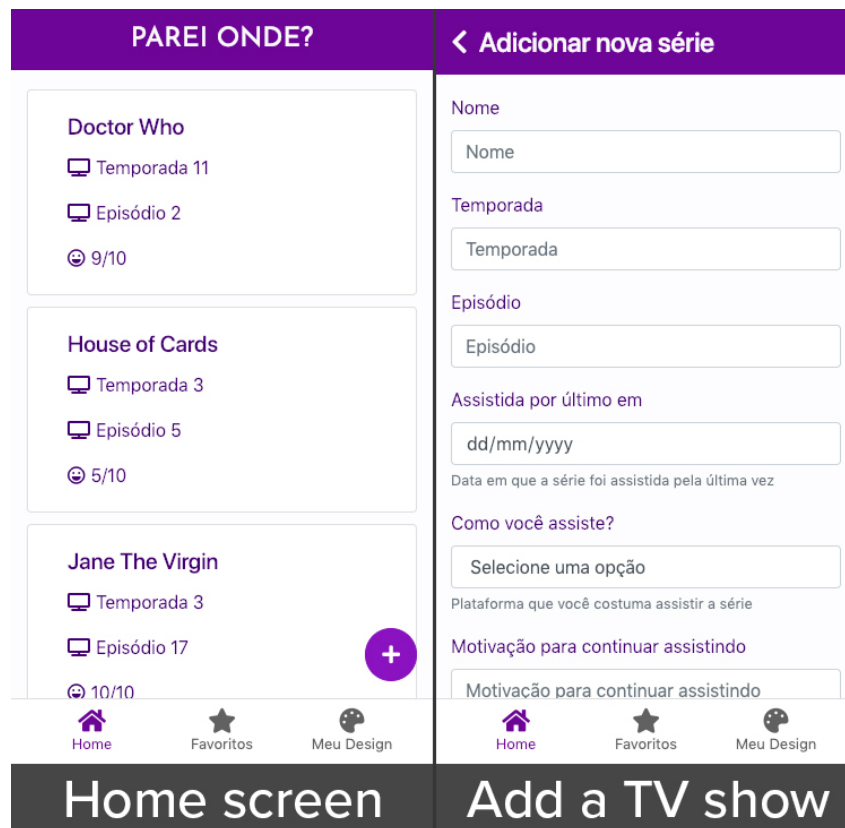
After analyzing the participants' profile, we prepared a training session to be conducted a week before the study with the participation of all individuals. The session encompassed all the technologies that were required to use PWA-EU. Aside from the training, we conducted a warm-up session on the same day as the experiment's conduction.

Table 9 – Participants profile - Likert scale: I have never heard of it(1), I have heard of it(2), I have some theoretical knowledge, but no practical(3), I have good theoretical and practical knowledge(4), I have deep theoretical and practical knowledge(5)

Part.	Group	Age	Education Level	HTML5	CSS	JS	jQuery	LS	Bootstrap	Hjs	JSON	SPA	RWD	PWA
P1	Student	21	Undergraduate student	4	4	4	2	4	4	1	5	4	3	2
P2	Student	21	Master's degree	2	2	3	2	2	2	2	2	2	2	1
P3	Student	25	Undergraduate student	4	4	2	1	2	4	1	2	1	2	1
P4	Student	24	Undergraduate student	4	4	4	2	3	4	2	4	2	3	2
P5	Student	22	Undergraduate student	5	5	4	2	4	5	1	4	2	4	1
P6	Student	23	Undergraduate student	3	3	3	1	3	3	1	4	3	3	2
P7	Student	20	Undergraduate student	4	4	4	1	3	4	1	3	2	2	2
P8	Student	23	Undergraduate student	4	4	4	1	3	4	1	2	2	3	2
P9	Student	23	Undergraduate student	4	4	3	2	4	4	2	4	2	2	1
P10	Student	22	Undergraduate student	2	3	3	2	2	2	1	2	1	2	0
P11	Student	20	Undergraduate student	4	4	4	1	4	2	1	2	2	2	0
P12	Student	21	Undergraduate student	4	4	4	2	2	2	1	2	1	2	1
P13	Student	23	Undergraduate student	4	4	4	1	2	2	1	2	1	3	0
P14	Professional	22	Master's degree	2	3	4	2	4	4	1	4	2	2	2
P15	Professional	21	Undergraduate student	5	5	5	3	5	5	1	5	2	5	2
P16	Professional	26	Master's degree	5	4	4	5	5	4	3	4	1	4	3
P17	Professional	36	Undergraduate's degree	2	3	3	2	3	2	2	3	2	3	2
P18	Professional	24	Undergraduate student	4	5	4	2	4	4	1	4	4	4	2
P19	Professional	26	Undergraduate student	3	3	3	3	2	2	3	4	1	3	1
P20	Professional	24	Undergraduate student	4	4	4	3	4	4	1	4	3	3	2
P21	Professional	23	Undergraduate student	2	2	3	1	2	2	1	3	1	2	1
P22	Professional	33	Undergraduate's degree	2	3	2	2	2	2	2	3	1	2	1
P23	Professional	27	Undergraduate's degree	4	4	4	4	4	4	2	4	3	4	3



Figure 26 – Parei Onde? main UIs



Based on the results from the analysis of the profile questionnaire (see Table 6.1), we developed an app called Parei Onde?<sup>1</sup>. This app follows the traditional PWA architecture so participants could complement it with PWA-EU elements. The initial structure of Parei Onde? allows end-users to include and mark as favorite their currently watched TV shows. The main and add shows UIs are displayed in Figure 26. We developed this app considering the profile analysis by selecting technologies most participants were familiar with. Due to that, we opted to use HTML, CSS, and JS with no JS frameworks. Instead, we adopted the front-end component library Bootstrap<sup>2</sup> to assist on mobile-first UI development and the jQuery library, required by some Bootstrap elements. Still, in order to allow it to be a SPA, the foundation of PWA applications, we included the dynamic template tool Handlebars.js<sup>3</sup> that assured pages could show new content without the need to make new HTTP requests.

After the Parei Onde? implementation was ready, we aimed at defining which user preferences should be included on the task we established. We defined user app preferences would be restricted to editing and deleting favorite platforms to watch TV Shows, such as Netflix, Hulu, Amazon Prime, Cable, so on. By including this feature, all three PWA-EU elements (UDM, Management Layer, Connection Layer) should be added

<sup>1</sup> <https://github.com/giuliacardieri/pareionde>

<sup>2</sup> <https://getbootstrap.com/>

<sup>3</sup> <https://handlebarsjs.com>

to the PWA. Hence, there were a few pre-requisites to accomplish this task, as participants were requested to use the same technologies as the app was implemented and the UDM was requested to be in JSON format and managed via LocalStorage.

We established a task named *Implementing PWA-EU into Parei Onde?* that was available in PDF format (see Appendix H). During this task, participants should include the PWA-EU architectural elements into a traditional PWA architecture on the code from the Parei Onde? app. Like the warm-up session task, participants should base their implementation on an app section called "*My Design*". This section should have functionality in which users could include and remove the user app preferences (see Fig 8), represented by TV Shows platforms on this app. Still, other elements should be added to the PWA code to represent other PWA-EU elements (i.e. Management and Connection Layers).

With the aims of analyzing PWA-EU acceptance and usefulness, we based our analysis on an adaptation of the Technology Acceptance Model (TAM) (DAVIS, 1989). This model analyzes both participants' acceptance and behavior on the use of technology. TAM is divided into two dimensions, the perceived ease of use and the perceived usefulness. The ease of use is related to the perception that technology can be adopted with no efforts. The perceived usefulness dimension represents how a person believes the use of a specific technology may improve their performance. In our analysis, we adapted the TAM and considered only the usefulness dimension.

To complement TAM, we also analyzed the codes from the PWAs developed by participants. Our goal was to analyze PWA code taking into account the completeness of each PWA-EU architectural element and the Adaptation Engine. Additionally, all participants edited their code using Brackets<sup>4</sup> and ran it on Google Chrome Browser.

## 6.2 Conduction

The training took place one week prior to the study, at a laboratory in UFSCar Sorocaba and took about one and a half hour. Technologies such as Bootstrap<sup>5</sup>, jQuery<sup>6</sup>, PWA, SPA and Handlebars.js<sup>7</sup> were presented on both theoretical and practical ways. The participant's task was to build a PWA that worked as a to-do list, called *My Tasks*. On this list app, participants should implement the functionality to include a new task storing data via LocalStorage, retrieving it with jQuery, and displaying using Bootstrap UI elements.

---

<sup>4</sup> <http://brackets.io/>

<sup>5</sup> <https://getbootstrap.com>

<sup>6</sup> <https://jquery.com>

<sup>7</sup> <https://handlebarsjs.com>

On the warm-up session participants had to include PWA-EU’s architectural elements (described in Chapter 4) into *My Tasks* PWA. Moreover, they had to use the technologies learned in the training session to implement a section called "My Design". In this section, users could select if they preferred to see their task in card or list formats. With this warm-up session, we expected participants to acquire a solid understanding of PWA-EU architectural elements.

The experimental study was conducted at a laboratory at UFSCar Sorocaba and participants had two hours to complete one task. All 23 participants were voluntary students from undergraduate and graduate computer science programs at UFSCar Sorocaba. Their ages ranged from 20 to 36 years old, and most individuals (70%) were undergraduate students. Participants were daily users of the internet on mobile devices and attended both the training and warm-up sessions.

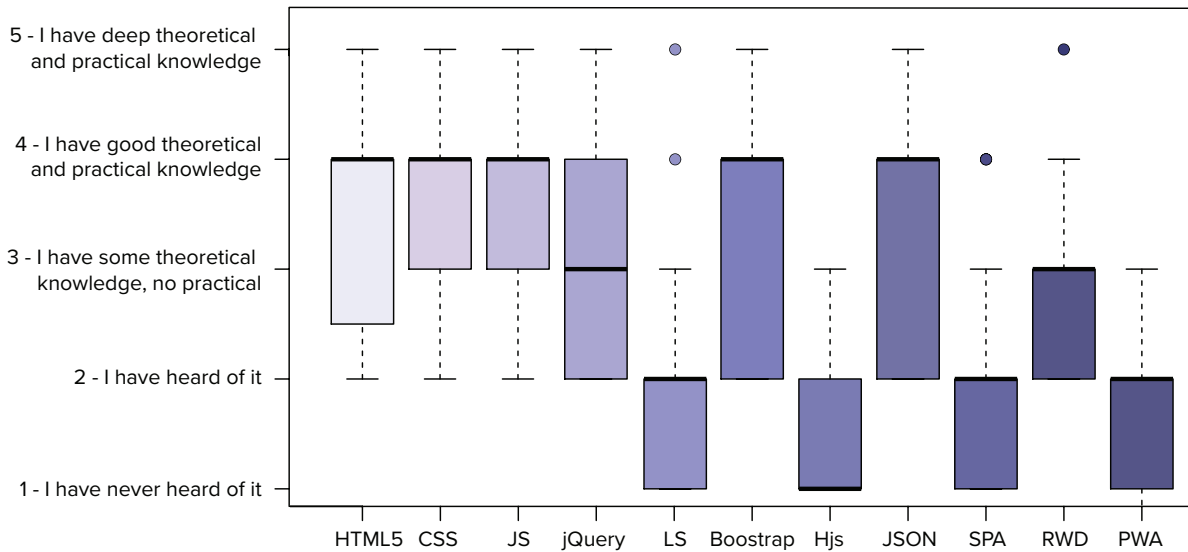
We set up participants environment prior to the study’s conduction by setting a unique text editor (Brackets) and a browser (Google Chrome). Yet, participants were free to use their personal laptops or laboratory computers to complete the task as long as they followed our text editor and browser specifications. Furthermore, all participants should develop and debug the app simulating an iPhone 6/7/8 device on Google Chrome’s device toolbar. We set these variables to avoid validity threats.

Initially, all participants received instructions about the task they should accomplish. A guide with instructions was provided to all participants in a PDF file (see Appendix H). We collected data using two sources, Google forms and participants’ final codes using Google Classroom. After finishing the task, participants should submit their code via Google Classroom and the post-study questionnaire (see Appendix I) that included TAM’s perceived usefulness dimension and two open questions, as Table 10 points out. Even though TAM has two dimensions, we opted to select only the perceived usefulness category. Hence, we want to focus on whether developers could implement the PWA-EU approach and their remarks about this architecture usefulness. Further, we adopted a six-point Likert scale going from *Strongly Disagree* to *Strongly Agree*.

Table 10 – Questions from the post-study questionnaire

Question	Format	Optional
U1 - Learning the PWA-EU architecture was easy	Likert	No
U2 - I can use the PWA-EU architecture the way I want	Likert	No
U3 - I find PWA-EU architecture useful for PWA development	Likert	No
U4 - I can make an app closer to end-user characteristics and/or environment through PWA-EU architecture	Likert	No
U5 - I clearly understand each architectural element from the PWA-EU architecture	Likert	No
Q1 - Do you understand all PWA-EU architectural parts? If not, which one?	Open	Yes
Q2 - Do you have any notes about PWA-EU’s usage or implementation?	Open	Yes

Figure 27 – Medians from participants previous knowledge, HTML5, CSS, JS, jQuery, LocalStorage(LS), Bootstrap, Handlebars.js(Hjs), JSON, SPA, RWD, PWA



### 6.3 Analysis

Prior to the experiment conduction, we analyzed the answers of the participants from the profile questionnaire. Figure 27 presents the medians calculated from Likert scale values displayed on Table 6.1. We noticed most participants were experienced with the fundamentals, such as HTML, CSS, and JS. However, there was a lack of practical knowledge of specific tools such as jQuery, LocalStorage, Handlebars.js, RWD, SPA, and PWAs. Regarding Handlebars.js, most participants never heard of it. These results guided us on the training, warm-up session and also to develop Parei Onde? app.

A pre-analysis was conducted in order to assure all participants filled the profile and post-study questionnaires. Later, we joined data from three data sources: participants' profile, the post-study questionnaire and the codes produced by the participants that were implemented applying the PWA-EU architecture. With regards to the code, we observed the completeness of each PWA-EU architectural element and the Adaption Engine, as it is an important part of the connection of the traditional PWA architecture with PWA-EU. We created a three-point scale to define the completeness of each element *1 - Non-existent*, *2 - Incomplete or with errors*, and *3 - Complete with no errors*.

Based on the participants' profile, the analysis was conducted by separating the participants into two groups. For ease of reference, we named the first group of *professionals* (i.e. 10 individuals) which grouped the participants who had had professional experience in working on programming area, and *students* (i.e. 13 individuals) the ones whose work experience on programming was limited to academic activities. With this categorization, we can examine if PWA-EU is more acceptable and/or useful for developers with certain previous experience.

Table 11 – Results from analysis divided by participant. TAM caption available at Figure 44. Code caption - Non-existent(1), Incomplete or with errors(2) and Complete with no errors(3)

Participant	Group	TAM					Code				
		U1	U2	U3	U4	U5	Management Layer	Connection Layer	UDM	Adaptation Engine	
P1	Students	1	3	5	5	3	2	2	3	3	3
P2	Students	2	2	5	5	4	2	2	3	3	2
P3	Students	1	1	1	1	1	2	2	2	3	1
P4	Students	3	2	5	6	3	1	1	3	3	2
P5	Students	2	2	4	4	5	2	2	3	3	2
P6	Students	3	3	4	4	3	2	2	3	3	2
P7	Students	3	4	5	6	3	2	2	2	3	3
P8	Students	4	3	4	4	5	2	2	3	3	2
P9	Students	1	1	1	1	1	2	2	3	1	1
P10	Students	2	2	2	2	2	2	2	3	2	1
P11	Students	4	2	4	4	3	1	1	2	1	1
P12	Students	6	3	5	5	6	1	1	3	3	1
P13	Students	1	1	6	6	1	1	1	2	1	2
P14	Professionals	4	5	6	6	6	3	3	3	3	3
P15	Professionals	3	4	5	6	5	2	2	3	3	3
P16	Professionals	5	5	6	6	5	2	2	3	3	3
P17	Professionals	3	2	5	5	5	2	2	3	3	1
P18	Professionals	6	6	6	6	6	2	2	3	3	3
P19	Professionals	1	1	1	1	1	2	2	3	3	1
P20	Professionals	1	1	3	3	2	2	2	3	3	2
P21	Professionals	1	1	4	3	3	2	2	2	1	1
P22	Professionals	4	3	5	6	5	1	1	3	3	1
P23	Professionals	6	5	5	5	5	2	2	3	3	2

Figure 28 – Complete UDM example

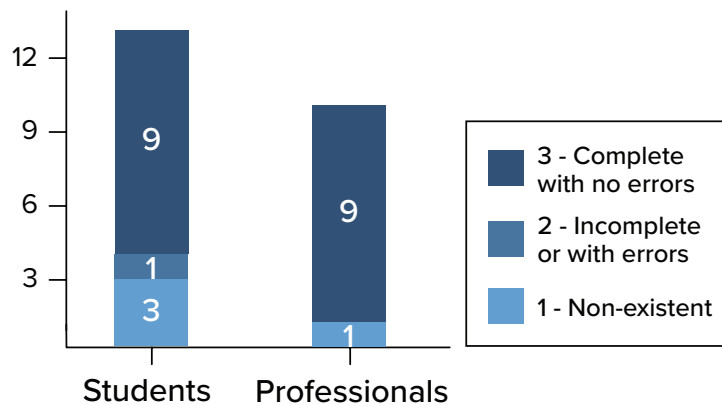
```

let pareionde_udm = [
  { id: 'Amazon Prime', name: 'Amazon Prime'},
  { id: 'HBOgo', name: 'HBOgo'},
  { id: 'Hulu', name: 'Hulu'},
  { id: 'Netflix', name: 'Netflix'},
  { id: 'Now', name: 'Now'},
  { id: 'TV', name: 'TV'},
  { id: 'YouTube', name: 'YouTube'}
]

```

Figure 29 – User Design Model (UDM) code analysis

## User Design Model



## 6.4 Results

We divided the discussion of the results by the baseline defined for our analysis. Due to that, we created subsections for each PWA-EU architectural element and the Adaptation Engine. Moreover, we also analyzed the data obtained using the post-study questionnaire. A bar graph was elaborated for each of the four architectural elements that compose the combined PWA-EU and traditional PWA architecture. The caption for each numeric labels on the code analysis is described at the corresponding figures.

### 6.4.1 User Design Model (UDM)

The UDM stores the user app preferences. Participants should create a JSON object containing different platform options to watch TV shows. We considered a task was complete when the JSON was correctly formatted considering the user app preferences. Figure 29 shows the distribution of the completeness of participants' from both groups code. We show a code snippet that illustrates a fully-functioning UDM in JSON format in Figure 28

Comparing participants from the *professionals* group, we noticed most of them

implemented a complete UDM. An exception was found for P21 that did not create a UDM. Even though the participant stated s/he had good theoretical and practical knowledge with JSON, s/he could not implement it.

When we look at the *students* group, we conclude most of them had a good performance. Nonetheless, three participants could not implement the UDM (P9, P11, P13) and one (P10) implemented an incomplete version. The UDM implemented by P10 was in the JSON format but the user preferences were not related to Parei Onde? app as s/he created a model similar to the JSON example from the warm-up session. This model contained two attributes, one that represents the name of the preference and a second, named *active*, that indicated if the preference was visible on the main screen. Again, this attribute was linked to the warm-up session in which the user preferences could be selected between a list or a card to display tasks. Furthermore, P10, P11, and P13 issues might be related to the fact they did not have experience with JSON technology. By contrast, P9 had deep theoretical and practical knowledge of JSON.

On top of that, we can also discuss the distinct ways participants correctly implemented this model. Even though we will not discuss the codes from all participants, the implementations are available at a GitHub repository <sup>8</sup>. In all the cases the PWA-EU architecture was not compromised, as the UDM fulfilled its role. P6 (students group) and P14 and P23 (professionals group) created a similar model to the example showed at the beginning of this subsection. The UDM implemented by P16 and P18 differs from the example as it does not contain an id attribute. Likewise, P15 created a plain array with no id or name attributes. Conversely, P3, P4, P17, P19, P20, and P22 created a UDM with a foundation similar to the one from the warm-up session. Unlike P10 (professionals group), these participants adapted the model to Parei Onde?'s domain but kept the *active* attribute, which was not necessary but did not hamper the development. In Figure 30, we show an example of P20's code, which is similar to P3, P4, P17, P19 and P22 model's. Moreover, P1, P2, P5, P19, P8, and P12, all from the *students* group, created a list with one attribute, called types or list\_types. An example from P12 is described on Figure 31.

By taking note of participants' implementation, we learned that the UDM could be built with no errors with different approaches. Regarding participants' issues, we learned they were related to failures in the understanding of PWA-EU and lack of knowledge of JSON. Many participants implemented their UDM structure based on the warm-up session app. This could be avoided by a more detailed definition of this element. For instance, the researcher could emphasize that the structure of the UDM should be rebuilt for each PWA, as the user app preferences change. Additionally, the developers who implement PWA-EU should be familiar with JSON.

---

<sup>8</sup> <https://github.com/giuliacardieri/pwaeu-devs-codes>

Figure 30 – UDM code from P20

```

let model = [
  {id: 'Amazon Prime', active: 'true'},
  {id: 'HBOgo', active: 'true'},
  {id: 'Hulu', active: 'true'},
  {id: 'Netflix', active: 'true'},
  {id: 'Now', active: 'true'},
  {id: 'TV', active: 'true'},
  {id: 'YouTube', active: 'true'}
]

```

Figure 31 – UDM code from P12

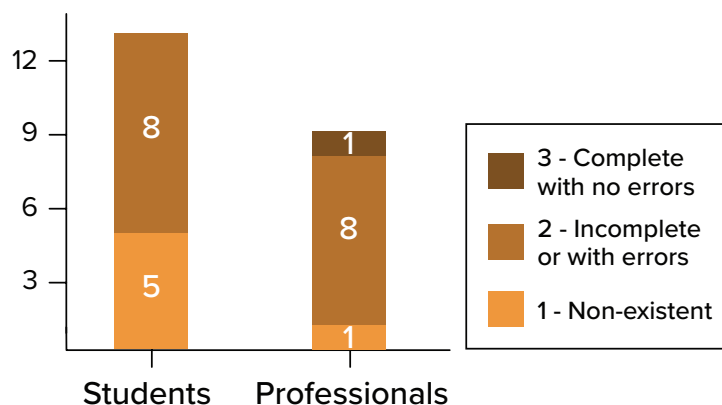
```

let model = [
  {lista_tipos: 'Amazon Prime'},
  {lista_tipos: 'HBOgo'},
  {lista_tipos: 'Hulu'},
  {lista_tipos: 'Netflix'},
  {lista_tipos: 'Now'},
  {lista_tipos: 'TV'},
  {lista_tipos: 'YouTube'},
]

```

Figure 32 – Management Layer code analysis

## Management Layer



### 6.4.2 Management Layer

The Management Layer is a PWA-EU architectural element in which programmers should include, edit and delete user app preferences on the UDM. On Parei Onde? domain, this layer should be represented by a JS file including functions to include and remove TV shows platforms. In order to implement this element, participants should be able to use JS basic resources.



When we take note of the *professionals* bar graph on Figure 32, we observe that almost all participants had not achieved the completeness of this layer. Only P14 implemented a complete layer. Nine participants (P15, P16, P17, P18, P19, P20, P21 and P23) built incomplete layers. P21's code was composed by a skeleton of a function to include new preferences. Moreover, s/he had no practical JS knowledge. On the other hand, P15, P16, P18, P20, and P23 were familiar with JS they pointed out at least good theoretical and practical JS knowledge. These skills are observable on all participants' code, even with errors and missing functions. P15 and P16 did not have a complete understanding of the difference between both layers from the PWA-EU architecture since both included Management Layers functions (remove and edit platform, respectively) on the Connection Layer file. In another way, P18's remove function contains the delete call to the data storage, a functionality that should belong to the Connection Layer. Conversely, P19, P20, and P23 created the include function, yet, they did not implement a remove platform functionality. Meanwhile, P19 and P17 indicated they had some theoretical JS skills. However, P17 could not implement both insert and remove functions correctly, which provoked multiple errors on the browser's developer console. P22 was the only participant from the *professionals* group who did not create a Management Layer and did not have any JS knowledge.

Similarly, none of the 13 *students* participants implemented a complete Management Layer, as Figure 32 points out. Still, on the profile questionnaire, only P3 pointed out a lack of JS knowledge. P1, P2, P3, P5, P6, P7, P8, and P9 developed a layer incomplete or with errors. While P4, P11, P12, P13 did not create a Management Layer. P3 created a file for this layer, yet, s/he left it empty. P9 and P10, who also had issues with the UDM, created a file but just copied the Connection Layer code from the warm-up session. P2, P5, P6, and P8 experienced a common issue, an incomplete layer in which the include and delete functions were not working. Although P1's and P7's layers were incomplete, their code was working. Likewise, P16 from the *professionals* group, P1, and P7 did not understand the limits between the Management and Connection layers. This issue made them leave functions from both architectural elements on the Connection Layer file.

With regards to code, we have developed an example from a complete Management Layer taking Parei Onde? app's domain. The code is shown in Figure 33. Figure 34 presents the implementation from P14, the only participant who built a complete Management Layer. Despite having fewer lines when compared to the example, P14's code also takes all functionalities from this layer into account. Codes from all participants are available at Github<sup>9</sup>

With these results in mind, we noticed that participants' issues can be categorized into two categories: lack of expertise on JS and misunderstanding about Management

---

<sup>9</sup> <https://github.com/giuliacardieri/pwaeu-devs-codes>

Figure 33 – Management Layer code example

```

const deletePlatform = function deletePlatform(id) {
  let old_model = getModel(),
      new_model_list = [],
      new_model = new Object(),
      j = 0

  for (let i = 0; i < Object.keys(old_model).length; i++) {
    if (old_model[i].id !== id) {
      new_model_list[j] = old_model[i]
      j++
    }
  }
  new_model = new_model_list
  updateModel(new_model)
}

const addPlatform = function addPlatform(data) {
  let old_model = getModel(),
      new_model_list = [],
      new_model = new Object(),
      new_platform

  if (data[0].value.trim().length === 0 || data[0].value.trim() === '')
    return false

  for (let i = 0; i < Object.keys(old_model).length; i++) {
    new_model_list[i] = old_model[i]
  }
  new_platform = {
    id: old_model.length,
    name: data[0].value
  }
  new_model_list[old_model.length] = new_platform
  new_model = new_model_list
  updateModel(new_model)
}

```

Figure 34 – Management Layer code from P14

```

function add(Name){
  let model = getModel()
  let nid = Object.keys(model).length+1
  model.push({id: nid, name: Name})
}

function remove(index){
  let model = getModel()
  model = model.filter(function(jsonObject) {
    return jsonObject['id'] !== index;
  });
}

```

Figure 35 – Connection Layer code example

```
const getModel = function getModel() {
  if (!localStorage.getItem('pareionde_udm'))
    setModel()
  return JSON.parse(localStorage.getItem('pareionde_udm'))
}

const setModel = function setModel() {
  let model = new Object(), list
  list = [
    { id: 'Amazon Prime', nome: 'Amazon Prime'},
    { id: 'Baixada', nome: 'Baixada'},
    { id: 'HBOgo', nome: 'HBOgo'},
    { id: 'Hulu', nome: 'Hulu'},
    { id: 'Netflix', nome: 'Netflix'},
    { id: 'Now', nome: 'Now'},
    { id: 'Online', nome: 'Online'},
    { id: 'TV', nome: 'TV'},
    { id: 'YouTube', nome: 'YouTube'}
  ]
  model = list
  localStorage.setItem('pareionde_udm', JSON.stringify(model))
}

const updateModel = function updateModel(new_model) {
  localStorage.setItem('pareionde_udm', JSON.stringify(new_model))
}
```

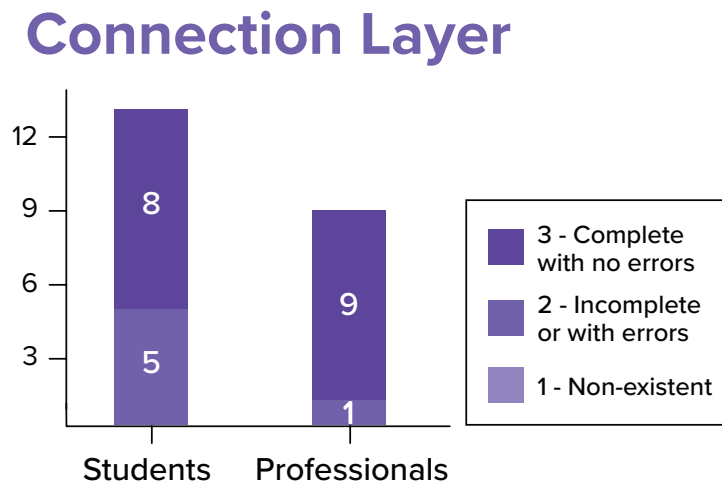
and Connection Layer roles. From these issues, we learned that in order to implement PWA-EU developers should have experience developing with JS. Further, our proposal should put more emphasis on the differences between the Management and Connection Layers.

### 6.4.3 Connection Layer

The Connection Layer is the part of the PWA-EU architecture that is responsible for storing and retrieving the UDM from a storage technology. Code that edits the UDM and is not related to storing data should not be present on this layer's code. On Parei Onde? the Connection Layer should be a JS file containing functions to set, update and get the JSON from LocalStorage. The update receives a modified UDM from the Management Layer. The set function is called only when users first access the app. The get function is requested when the Adaptation Engine is building the UI. We created a code example for all set, get and update functions, which is displayed in Figure 35.

When we look at the *professional* group performance on Figure 36, we note P17 delivered a complete implementation of this layer. Interestingly, most participants had no practical knowledge of LocalStorage. However, P21's implementation was incomplete since his/her functions attempt to obtain a nonexistent entry on the LocalStorage. This nonexistent model should be the UDM, another element that P21 could not implement, as pointed out in Section 4.1.1.

Figure 36 – Connection Layer code analysis



Unlike the *professionals* group, five participants (P1, P3, P7, P11 and P13) had incomplete Connection Layers on the *students* group, as Figure 36 indicates. None of the participants from this group were familiar with LocalStorage on either practical and theoretical perspectives. P3 did not have any JS experience, which might have influence under his/her error as s/he did not create an update UDM function. Still, P1, P7, P11, and P13 had practical skills with JS, as we observe their JS knowledge as functions were created. Yet, P11's functions were never called by the app. While P13 implemented the get function but missed out the set and update parts. On a different way, P1 and P7 were confused regarding layers limitations and included the algorithms to remove a platform from the UDM JSON on this layer, instead of the Management one. The delete function from the Connection Layer should only deal with connections to the LocalStorage.

Since most participants implemented a complete Connection Layer, we can compare the codes with our example from this Subsection's first paragraph. Most participants (P2, P5, P6, P8, P9, P10, P12, P14, P16, P17, P19, P20, and P22) built their code similar to our example. This remark is linked to the warm-up session, in which a similar logic was implemented on the group activity.

Unlike our example, P4's code had only get and set functions (see Fig 37). Rather than creating a separate function to update the UDM, s/he joined both set and change functions into a unique method. Similarly, P7's implementation (see Fig 38) also had only a get and set functions. Nonetheless, s/he set default platforms on the get model. On the other hand, P18's and P23's code did not include default values into LocalStorage. Instead, value is included later, when the user includes his/her first preference. Yet, both participants implemented this layer using distinct methods. On P18's code, an empty model returns null, while for P23, it returns an empty array. Below, we show P18's (see Fig 39) and P23's (see Fig 40) implementations, respectively.

Finally, the issues experienced by participants may be divided into two categories:

Figure 37 – Connection Layer code from P4

```

const setModel = function setModel(platform) {
  let model = [
    { id: 'Netflix', active: true },
    { id: 'Hulu', active: true },
    { id: 'Amazon Prime', active: true }
  ]

  if (plataforma !== undefined)
    model.push(plataforma)

  localStorage.setItem('list', JSON.stringify(model));
}

const getModel = function getModel() {
  let model;

  if (!localStorage.getItem('list')) {
    setModel();
  }
  return JSON.parse(localStorage.getItem('list'));
}

```

Figure 38 – Connection Layer code from P7

```

const setModel = function setModel(platform) {
  let list = getModel()

  let type = {type: platform}
  list.push(type)
  localStorage.setItem('list', JSON.stringify(list))
}

const getModel = function getModel() {
  let default_types = [
    {type: 'Amazon Prime'},
    {type: 'HBOgo'},
    {type: 'Hulu'},
    {type: 'Netflix'},
    {type: 'Now'},
    {type: 'TV'},
    {type: 'YouTube'}
  ]

  if(!localStorage.getItem('list'))
    localStorage.setItem('list', JSON.stringify(default_types))
  return JSON.parse(localStorage.getItem('list'))
}

```

Figure 39 – Connection Layer code from P18

```

const setModel = function setModel(model) {
  localStorage.setItem('list', JSON.stringify(model));
}

const getModel = function getModel() {
  let model
  if (localStorage.getItem('list')) {
    return JSON.parse(localStorage.getItem('list'));
  }
  return null;
}

```

Figure 40 – Connection Layer code from P23

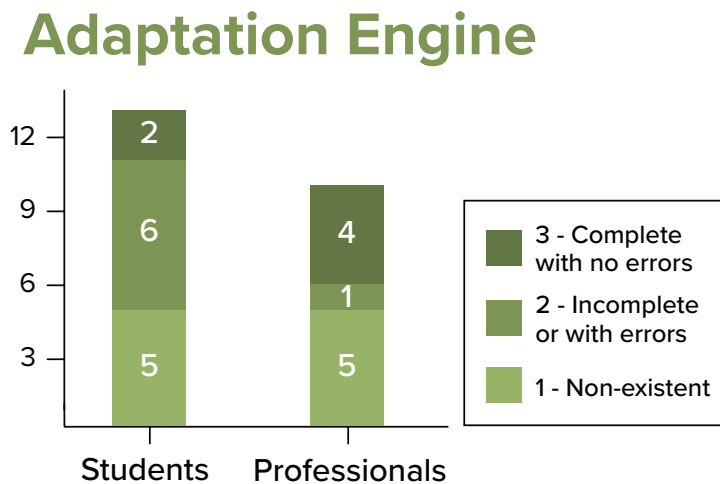
```

const setPlataformDB = function setPlataformDB(db) {
  localStorage.setItem('plataform', JSON.stringify(db))
}

const getPlataforaDB = function getPlataformDB() {
  if (localStorage.getItem('plataform'))
    return JSON.parse(localStorage.getItem('plataform'));
  return [];
}

```

Figure 41 – Adaptation Engine code analysis



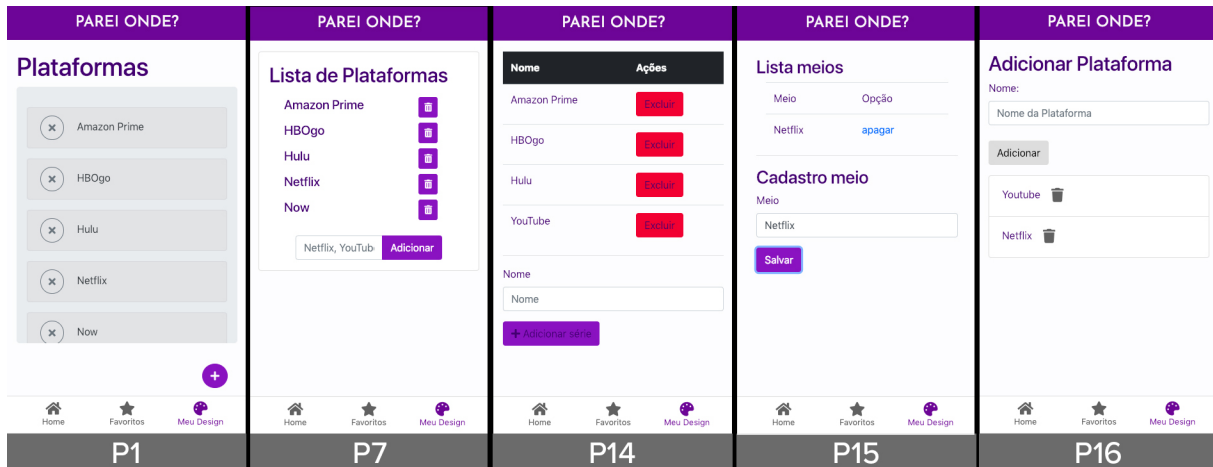
lack of LocalStorage and/or JS knowledge and lack of understanding of the Management and Connection Layers roles. We noticed the incomplete implementations come from participants who were novice developers. The second issue was previously discussed in the Management Layer subsection, in which we learned improvements to the description of each layer could be included on the PWA-EU approach.

#### 6.4.4 Adaptation Engine

The Adaptation Engine is part of the traditional PWA approach, as it changes the app UI by obtaining the app shell from the cache, as described in Section 4.1.1. On this study, participants should implement the engine in two steps: (i) design a UI on the "My Design" section using Bootstrap elements, and (ii) modify the include new TV shows from considering user app preferences defined on the UDM. On the UI, all the preferences and a form to include new ones should be visible. Additionally, all the user preferences must have remove buttons.

Figure 41 shows the participants who implemented a complete Adaptation Engine. We note that even on the *professionals* group, more than half of the developers could not create a complete engine. Moreover, four participants (P14, P15, P16, P18) delivered a complete Adaptation Engine. Regarding the *students* group, only two participants (P1,

Figure 42 – P1, P7, P14, P15 and P16 UIs from "My Design" section



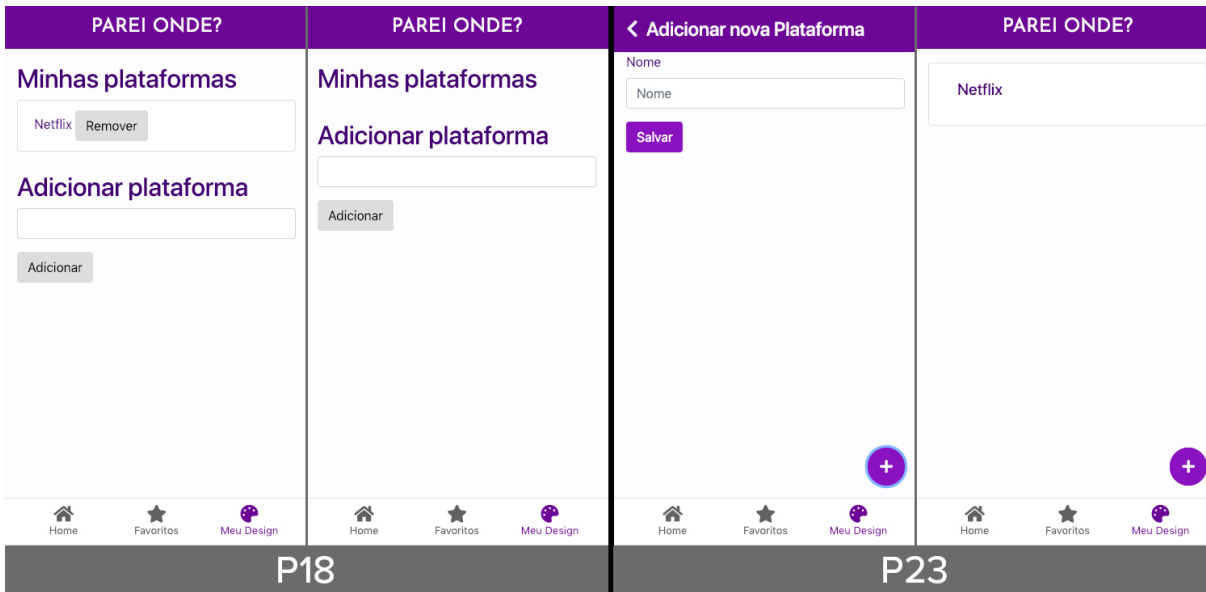
P11) successfully accomplished this task. This lower number of completeness is linked to the non-existent and incomplete code from the previously described architectural elements. We observe this relation as all three PWA-EU elements were needed in order to implement a complete Adaptation Engine. Nonetheless, there is a case scenario in which participants had an incomplete Management Layer and yet could implement a complete Adaptation Engine. On the case, participants (P1, P15, P16, and P7) misplaced some of the Management Layer on the Connection Layer file, but still implemented all the necessary functionalities and could create a complete Adaptation Engine.

As the Adaptation Engine combines many JS and CSS files, we will not compare the codes between participants. Instead, we will compare UIs developed on the "My Design" section. On this UI, end-users can select their preferred platforms to watch TV Shows. Figure 42 presents the UIs developed by P1, P7, P14, P15 and P16. P1's did not implement a form to include user preferences, in exchange, s/he implemented a window dialog when the button with the plus icon was touched. The other four developers implemented both include and remove features on the UI.

As pointed out on Section 4.1.1, P18's and P23's (see Figure 43) UIs are distinguished from the others as an initial empty state was developed. No platform is displayed while the user has not included a preference yet. P23's code was incomplete as remove buttons from each platform were missing together with the remove function on the Management Layer. Still, we decided to include his/her UI on this comparison due to the Adaptation Engine was almost complete and also to the relevant differences on the other layers' implementation.

In light of these results, we learned the Adaptation Engine is complete if all PWA-EU architectural elements were implemented. Yet, errors related to misunderstanding architectural layers, such as misplacing functions on wrong files, did not hamper the engine role. We noticed participants did not have issues on the steps related to implementing

Figure 43 – P18's and P23's UIs from "My Design" section



this element (i.e. build a UI using Bootstrap). Instead, participants issues with other architectural elements led to an incomplete engine. When we put together the elements described in the previous subsections, we obtain the functionality needed to combine both PWA-EU and the traditional PWA architectures.

### 6.4.5 Participants' Feedback

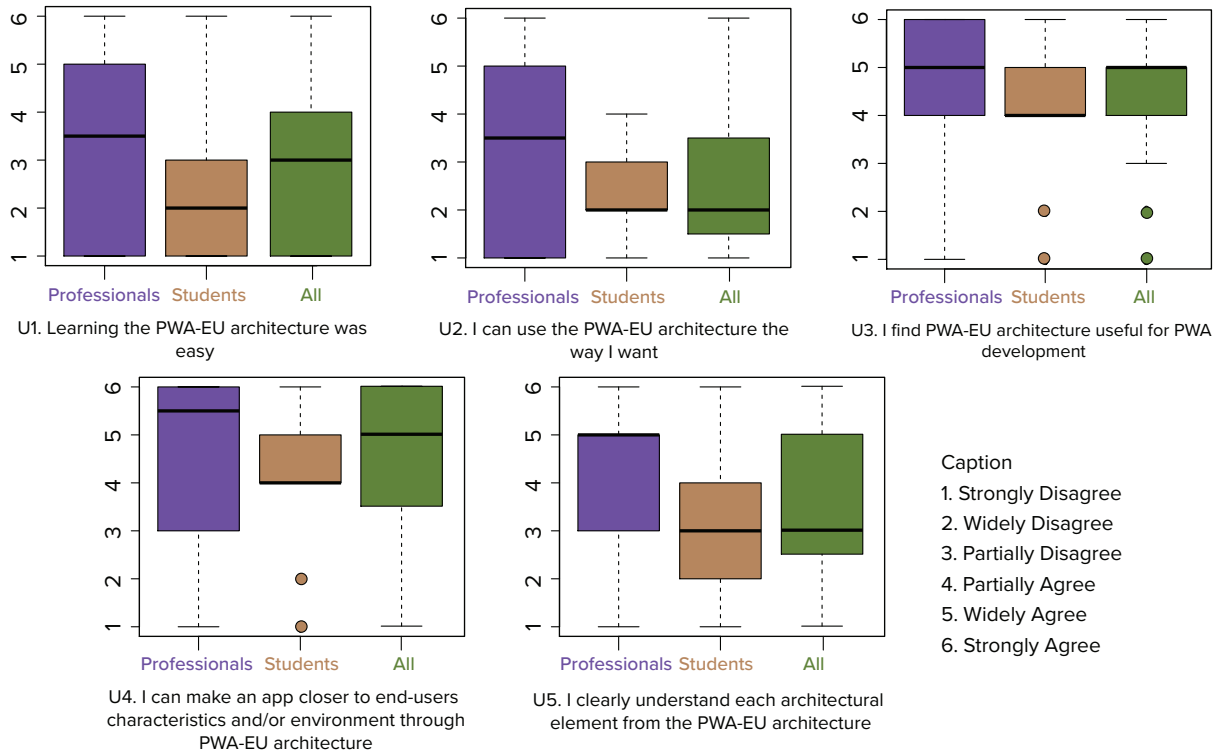
In this subsection, we will associate the code analysis results. Besides Table 11 that shows individual results from the questions of acceptance of the proposal, Figure 44 displays the group results in box plots.

On the post-study questionnaire participants could answer two open questions, *Q1 - Do you understand all PWA-EU architectural parts? If not, which one?* and *Q2 - Do you have any notes about PWA-EU's usage or implementation?*. Most participants answered Q2, while only a few (7) responded to Q1. Their remarks assisted us to comprehend more about their questions of acceptance of the proposal results.

Regarding questions of acceptance of the proposal, three affirmatives are directly connected to the elements from the PWA-EU and traditional PWA joint architectures: *U1 - Learning the PWA-EU architecture was easy*, *U2 - I can use the PWA-EU architecture the way I want*, and *U5 - I clearly understand each architectural element from the PWA-EU architecture*. Observing the boxplots 44 from these three affirmatives notice a difference between 1.5 to 2 points on the median between *professionals* and *students*. Moreover, we observe that while *professionals* felt neutral regarding U1 and U2, *students* widely disagreed on both affirmatives. Yet, on U5, *students* partially disagreed about the understanding of the PWA-EU elements and *professionals* widely agree with the statement. Hence, we can



Figure 44 – Questions of acceptance of the proposal



associate this contrast with the code analysis from the previous section. Counting all three PWA-EU elements and the Adaptation Engine, more developers from the *professionals* group delivered complete implementations.

We adopted Fisher’s exact test (FISHER, 1922) to analyze the existence of a statistical significance between the professionals’ and students’ groups. This test is recommended for comparing categorical data of small samples because it calculates the exact significance of the deviation from a null hypothesis using the p-value, instead of an approximation as other methods. The statistical analysis was conducted with the TAM adaptation data from Table 11. As we explored a small sample, we adopted a confidence interval of 95% (0.05) to mitigate the errors in the results. We defined a generic null hypotheses and corresponding alternative hypotheses:

- **H0**: There is no influence of the type of individual on the acceptance of  $\langle U \rangle$
- **HA1**: There is influence of the type of individual on the acceptance of  $\langle U \rangle$

We tested both hypotheses replacing  $\langle U \rangle$  for each TAM question (i.e U1,...,U5). Table 12 shows the p-values obtained. All p-values were above the defined confidence interval (0.05) and due to that we cannot reject the null hypothesis. This finding means no statistical significance was identified between *students’* and *professionals’* groups in all questions of acceptance of the proposal. Even though no significance was identified,

by observing the p-values we learned that a bigger difference between those groups was identified on U2 and U4 when compared to U1, U3 and U5. By associating this findings with other data sources (i.e. the code analysis) we learned *professionals* may have less issues to use the PWA-EU architecture (question U2) and less difficulties to make an app that empowers end-users (question U4) when compared to *students'* group.

Table 12 – Results from Fisher's exact test statistical analysis

Question	p-value
U1 - Learning the PWA-EU architecture was easy	0,63
U2 - I can use the PWA-EU architecture the way I want	0,14
U3 - I find PWA-EU architecture useful for PWA development	0,54
U4 - I can make an app closer to end-user characteristics and/or environment through PWA-EU architecture	0,19
U5 - I clearly understand each architectural element from the PWA-EU architecture	0,29

Associating remarks of the participants from all three data sources, we can learn more about their experience developing PWA-EU. The lower medians for U1, U2 and U5 are connected to three main issues: (i) difficulties implementing the elements; (ii) issues regarding inexperience with specific technologies and (iii) struggle to understand the differences between Management and Connection Layer. We will discuss all the issues in the following paragraphs.

The implementation issue was observed in 12 participants (P1, P2, P3, P5, P17, P20, P8, P9, P21, P10, and P13). Yet, only P17, P20, and P21 were from the *professionals* group. From these participants, P17 and P21 stated they clearly understood the PWA-EU architecture and their issues are connected only to the technologies required to implement. P20 complained about the large number of technologies needed to implement PWA-EU. P3 mentioned *"It was hard to understand how the architecture integrates on the code"*. However, P3 was the only developer with no JS practical or theoretical knowledge on the *students* group.

The second problem, related to inexperience was identified in 8 participants (P14, P15, P6, P19, P7, P20, P10, and P11), even experienced ones commented about it. P14 from the *professionals* group was the only participant who completed the entire PWA-EU architecture, s/he selected values 4 and over for all questions of acceptance of the proposal. Even so, s/he acknowledged *"It wasn't hard to understand the architecture, still, the activity required much knowledge about PWA development and specific libraries."* In the same way, P20 *"It was really hard to implement something completely new that encompassed so many technologies, such as the HTML/CSS and JS requirements."* P15 mentioned s/he needed more time as PWA was something new to him/her. Looking at *students* viewpoint, P6 also attributed issues to his/her inexperience with PWAs. Similar to P15, P7 stated an extra time to study the technologies would have helped him/her. On the other hand, P10 stated s/he did not know how to implement the code in JS. Also with a different opinion,

P11 attributed his/her difficulties to a steep learning curve of the selected technologies. However, we did not use any JS framework with the aims of reducing the learning curve.

The third issue, related to the misunderstanding of Management and Connection Layers was experienced by 8 participants (P1, P20, P22, and P13). From the *professionals* group, P20 affirmed s/he could not understand the architectural parts and needed more time to learn it. Conversely, P22 did not understand each layer's specificity. Focusing on the *students* group, P1 expressed issues with Management Layer on Q1 *"I could not comprehend the communication between different architectural parts, especially, the Management Layer"*. Looking at his/her code, we observe incomplete Management and Connection Layers. Moreover, P13 shared his/her remarks *"I found the architecture very cool, but the time to learn it and apply it was too short"*.

Despite lower U1, U2 and U5 affirmatives values, U3 - *I find PWA-EU architecture useful for PWA development* and U4 - *I can make an app closer to end-users characteristics and/or environment through PWA-EU architecture* had higher acceptance rates. Nonetheless, the contrast between *professionals* and *students* was still noticeable on U4. Most *students* only partially agreed the PWA-EU assists an app to get closer to end-user characteristics. On the other hand, *professionals* tended to strongly agree with this statement. Our statistical analysis pointed out this bigger contrast between both groups on U4, which we can observe in Table 12. Additionally, we can discuss if this difference relates to *professionals* group background, as these developers are accustomed to implementing new software architectures. Further, the outliers on U3 and U4 are related to the three participants from the *students* group. P3 and P9 selected 1 and P10 2 on all five acceptance of the proposal statements. P9 did not explain his/her issues. All three had issues on most architectural elements.

By observing participants' individual opinion, we notice that even with issues many agree about PWA-EU usefulness. Some examples are *"I found the PWA-EU architecture useful, but I had difficulties to implement it"*[P2], *"Learning is interesting and useful, but the implementation was extremely complicated to me"*[P3], *"The idea to let user on the center of design choices, participating and modifying the UI as s/he likes is extremely interesting..."*[P15]. Unlike other developers, P18 selected 6 to all acceptance of the proposal statements and did not disclosure issues *"The architecture is pretty simple and easy to learn and use"*.

## 6.5 Discussion

Based on the findings presented in each PWA-EU architectural element results section, we can answer RQ1 - *what do the developers acceptance when using PWA-EU approach?*. By observing participants profile we notice all of them were not familiar with

some of the technologies such as LocalStorage, Handlebars.js, SPA, and PWA. Moreover, even though Salman et al. (2015) point out only a small difference between students and professionals when they perform new activities, we noticed a significant contrast between these groups regarding the PWA-EU proposal acceptance. Most participants from the *professionals* group implemented a more complete PWA-EU architecture when compared to the *students* group. Additionally, only one participant (professionals group) implemented a complete Management Layer. Hence, we learned PWA-EU has limitations regarding its acceptance, as novice developers have more challenges on implementation, architectural and technological perspectives than experienced ones. Still, other factors might influence participants while writing code. Namoun et al. (2016) in the meta-design field affirms that writing code is a powerful yet boring activity when compared to programming by visual approaches (i.e. drag and drop), which could have contributed to participants acceptance on the study.

The questions of acceptance of the proposal support our answer to *RQ2 - How is developers' feedback regarding the usefulness of the PWA-EU approach?*. By gathering participants remarks we learned web-experienced developers may accept and understand the PWA-EU approach more easily when compared to those who are beginners. This limitation is connected to the need for a broad set of technologies and also the complete comprehension of PWA-EU layers specificity. Given that, participants from all technical levels identified the usefulness of the PWA-EU architecture on both PWA development and end-users experience viewpoints. With that in mind, we can affirm that developers found the PWA-EU approach useful, but not easy to implement. This lack of easiness could have its foundation on multiple constructs, as pointed out by Namoun et al. (2016). Furthermore, we could improve PWA-EU easiness by providing a more detailed comparison between each PWA-EU architectural element and a group discussion with multiple developers in order to define which user app preferences are adequate for a specific application domain. This discussion is related to one of the meta-design guidelines proposed by Fisher, Nakakoji & Ye (2009). Which is linked to PWA-EU as the user app preferences should be previously identified by designers and developers (see Chapter 4).

## 6.6 Threats to validity

Four validity threats: conclusion, construction, internal and external, determined by Wohlin et al. (2012), were discussed.

With the aims of mitigating conclusion threats, we adopted two distinct data sources, an online questionnaire application via Google Forms, and participants' code obtained via Google Classroom. Thus, we triangulate data from both sources which provides reliability to the analysis. Construction threats were avoided by assuring all participants

had some experience with web technologies by having taken a web development course, using the same code editor (Brackets) and browser (Google Chrome). Furthermore, we conducted training and warm-up sessions that followed a similar scenario to the study. These sessions benefited participants as they could clarify any doubts which led to more quality implementations that could be analyzed in a more safe way.

Internal threats are related to participants' fatigue during the study, we created one task and established a time limit (one and a half hours). Even though the external threats are related to the choice of selecting students as developers, Salman et al. (2015) point out these individuals are acceptable on an experimental study. The authors gathered evidence of a small difference between the performance of students and professionals when carrying out new tasks. As both groups did not have previous knowledge of PWA, we can compare *professionals* to *students*.

## 6.7 Final Considerations

This chapter presented a validation of the PWA-EU approach with developers with distinct technical skills, divided into *professionals* and *students* groups by their background programming experience. In our study, 23 developers had to implement the three architectural parts from the PWA-EU proposal and one from the traditional PWA approach into an existing PWA called Parei Onde?. We conducted an analysis based on users' profile, questions of acceptance of the proposal, open questions and a code analysis pinpointed to the four architectural elements. By analyzing developers' performance and remarks, we concluded novice developers found the PWA-EU challenging on implementation, architectural and technological perspectives. Conversely, experienced developers could implement and understand it in a simpler way. Nevertheless, all developers agreed about PWA-EU usefulness on PWA development and end-user experience viewpoints. With that in mind, we can discuss whether the association of PWA and the meta-design technique brings benefits not only to end-users but also to the web and mobile developers.

As contributions, we can point out the discussion on a technical level about the combination of PWA approach and EUD methods. Further, PWA-EU can assist developers interested in empowering end-users while developing mobile apps, as our architecture may guide how to arrange the responsibilities of an app.



## 7 Conclusion

This project proposed PWA-EU, an approach that allows end-user empowerment on Progressive Web Apps (PWAs). Initially, we compared users' interactions on mobile apps (ESI). Based on our early findings, we outlined the PWA-EU approach, an extension of the traditional PWA architecture in a way that end-users are empowered by including their preferences. Furthermore, we conducted two experimental studies (ESII and ESIII) to validate this proposal on end-users and developers perspectives.

During the first experimental study (Chapter 3), we analyzed UX on the mobile web, native Android and PWA apps on both user and HCI specialist viewpoints. As a result of this study, we found out that participants had good experiences even when interaction issues arose. Moreover, we identified common issues related to users' interaction with certain UI elements. With this in mind, there was no bias indicating whether a specific platform or UI element provided more satisfying interactions.

After our initial investigation, we noted similarities between the PWAs and meta-design concepts as both have an adaptive nature. PWA progressively changes its UI according to user interactions, as it becomes more complete regarding interactions and functionalities. This is similar to meta-design, in which the UI can change through time as end-users take the role of active participants. Still, users had no input on PWA's changes, in order to unify these apps and this approach, we outlined the PWA-EU.

The PWA-EU approach extends the traditional PWA architecture with the aims of providing end-users a role as active co-designers of the app. In order to accomplish that, we defined a three-element based architecture that connects to the PWA via its own architectural elements. If a PWA is developed following the PWA-EU proposal, the three architectural parts allow end-users to select their preferences and adapt the app's UI on runtime.

After PWA-EU was outlined, we conducted an experimental study (Chapter 5) to validate this approach with end-users. At first, we developed Calendar, a PWA based on the PWA-EU approach. Later, we selected participants that had experience in technology fields, such as developers and designers and individuals from other professional and academic areas. Moreover, half of the participants played the role of an application's co-designer and selected their preferences on the PWA's UI. Our analysis led us to find out that co-designers had more satisfying interactions than those end-users who did not actively participate in the design choices. On top of that, the results indicated that co-designers familiarized with technology felt less satisfied but more dominant when compared to non-tech users.

Complementing the end-users' perspective, we conducted an experimental study that evaluated this approach on developers viewpoint. Additionally, we grouped developers by their technical expertise, which resulted in a group of professional developers and one for those who experienced programming only on academic courses. In our study, 23 developers implemented the three architectural parts from the PWA-EU proposal and one from the traditional PWA approach into an existing PWA, called *Parei Onde?*. Further, we conducted an analysis that indicated novice developers had challenges on PWA-EU's implementation, architectural and technological perspectives. Alternatively, experienced developers could implement and understand our proposal in an easier way. However, all developers agreed about PWA-EU usefulness on PWA development and end-user experience viewpoints.

Lastly, we can answer the RQ we proposed on the Section 1.1 **How can we join PWA and meta-design in a way that users become co-designers of a PWA app?** by joining our analysis from all three experimental studies and the PWA-EU proposal. Furthermore, we can this project investigated end-user interactions when they play the role of co-designers on PWAs. Initially, we investigated differences between mobile app platforms. Later, we outlined a proposal, PWA-EU, with the aims of combining meta-design concepts into PWA's architecture. After PWA-EU was defined, we analyzed if this approach brought benefits to technical and non-technical end-users. Still, we considered developers' acceptance integrating EUD into PWA apps.

## 7.1 Contributions

Throughout this research project, we can enumerate relevant contributions achieved by the three experimental studies and the proposal of PWA-EU.

Our literature review indicates a lack of research studies on PWA's user perspective. Through searching and reading these works, we could note research in the PWA field is relevant to enable this approach to become more mature. Furthermore, we contribute by researching the potential of combining PWA apps and EUD, an area that has a gap in academic researches.

The initial study (Chapter 3) comparing interactions in a PWA, native and web apps is relevant as it open discussions about UX improvements on PWAs. Not to mention we conducted a comparison between PWA and native applications, which as far as we know, is the first analysis of its kind. Additionally, when we acknowledge the feedback from users and analysis on an HCI specialist perspective, we allow this approach to become more mature and collect insights that bring improvements on the use of certain UI elements.

The PWA-EU contributes to a proposal that includes meta-design aspects into the traditional PWA architecture. With this approach, we opened the discussion of how



end-users could be effectively empowered on mobile apps. The experimental study that evaluates PWA-EU with end-users acting as co-designers (Chapter 5) discusses whether meta-design is an easier approach to implement with technology-related people. On the other hand, our study with developers implementing the approach (Chapter 6) discusses the link between PWA and EUD on a technical level and points PWA-EU as a useful architecture to assist developers interested in developing mobile apps that consider end-user empowerment.

Further, this research had contributed to:

- Participation on the 3rd Thesis and Dissertations Workshop on the 17th Brazilian Symposium on Human Factors in Computing Systems submitting and presenting the work: *"PWA-EU: uma abordagem para o desenvolvimento de aplicações PWA baseadas em EUD"*;
- Publication: *"Analyzing User Experience in Mobile Web, Native and Progressive Web Applications: A User and HCI Specialist Perspectives"* on the Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems. (CARDIERI; ZAINA, 2018)

Moreover, we can list technical contributions related to talks and presentations about web development technologies:

- *"Usando CSS para fazer ilustrações"* - Presentation about CSS drawings in October 2017 at IFSP Itapetininga
- *"Progressive Web Apps"* - Presentation about PWA in January 2018 at GDG Meetup Bauru
- *"Front-end: O que eu posso fazer com HTML e CSS?"* - Training session about HTML and CSS in June 2018 at Rails Girls Sorocaba
- *"O que eu posso aprender com desenhos e jogos em CSS?"* - Presentation about learning CSS with games in July 2018 at Sorocaba CSS Meetup
- *"O que eu posso aprender com desenhos e jogos em CSS?"* - Presentation about learning CSS with games in October 2018 at Women Dev Summit
- *"PWA vs Nativo vs Web: o que a UX pode nos dizer sobre essa batalha?"* - Presentation about the study presented at Chapter 3 in October 2018 at 7Masters, video available <sup>1</sup>

---

<sup>1</sup> link: <https://imasters.com.br/design-ux/7masters-pwa-pwa-vs-nativo-vs-web-o-que-ux-pode-nos-dizer-sobre-essa-batalha>

- "*O que eu posso aprender com desenhos e jogos em CSS?*" - Presentation about learning CSS with drawings and games in December 2018 at Front in Floripa

## 7.2 Limitations and future work

Even though PWA-EU achieved its proposed goals, we can point out limitations that arise by contemplating data from our three experimental studies:

- There is a vast amount of user preferences that could be included as user app preferences into PWA-EU architecture. Still, we selected only a set of preferences related to UI elements, content display format and interaction methods on our study. Due to that, our evaluations are linked to these preferences;
- Our study did not evaluate the impact meta-design brings to apps. In order to achieve that, a comparison between an app with meta-design and an app that has no meta-design concepts should be carried out;
- PWA-EU adoption is not simple for novice developers. These developers have difficulties to either understand the architectural elements and implement the approach;
- Although our evaluation suggested perceived usefulness, we must consider that our scope is limited to software developers in an academic environment as the study was conducted on a university.

Considering the little exploitation of the combination of EUD and PWAs by the academic community, we can propose as future work:

- Explore other EUD topics on PWAs, such as web mashups and mobile web augmentation;
- Conduct studies comparing end-user interactions when they take the role of co-designers on native, web and PWA mobile apps;
- Extend PWA-EU architecture in a way that end-users can build an entire PWA when playing the role of co-designers;
- Implement apps based on PWA-EU on different domains than our research explored;
- Refine the PWA-EU with the aims of enabling an easier adoption by novice developers by setting up guidelines and/or environments that assist PWA-EU comprehension and implementation;
- Extend the PWA-EU to include the Meta-design's guidelines related to knowledge sharing features;

- Extend the PWA-EU in order to explore more PWA's features, such as offline availability.



# Bibliography

ALEGRÍA, Y. M. et al. Behavioral Assessment Proposal to Supplement the Emotional Evaluation of Users as they Interact with Websites. *Publicaciones e Investigación 8 (1)*, p. 185–201, 2013. Cited on page 39.

BARGAS-AVILA, J. A. et al. Working towards usable forms on the world wide web: Optimizing date entry input fields. *Advances in Human-Computer Interaction*, v. 2011, 2011. ISSN 16875893. Cited on page 36.

BARRERA-LEON, L. et al. Tukuchiy: A dynamic user interface generator to improve usability. *International Journal of Web Information Systems*, v. 12, n. 2, p. 150–176, 2016. Cited 3 time on pages 31, 32, and 33.

BENYON, D.; MURRAY, D. Applying user modelling to human-computer interaction design. *ArtifIntell Rev 7*, v. 3-4, p. 199–225, 1993. Cited on page 57.

BIØRN-HANSEN, A.; MAJCHRZAK, T. A.; GRØNLI, T.-M. Progressive Web Apps: The Possible Web-native Unifier for Mobile Development. *Proceedings of the 13th International Conference on Web Information Systems and Technologies*, n. Webist, p. 344–351, 2017. Cited 3 time on pages 31, 32, and 55.

BOSETTI, G. et al. An End User Development Approach for Mobile Web Augmentation. *Mobile Information Systems*, v. 2017, n. February, 2017. ISSN 1875905X. Cited 5 time on pages 20, 31, 33, 34, and 55.

BRADLEY, M.; LANG, P. J. Measuring Emotion: The Self-Assessment Manikin and the Semantic Differential. *Journal of Behavior Therapy and Experimental Psychiatry*, v. 25, n. 1, p. 49–59, 1994. ISSN 00057916. Cited 2 time on pages 38 and 66.

BRYANT, J.; JONES, M. Responsive Web Design. In: *Pro HTML5 Performance*. Berkeley, CA: Apress, 2012. v. 67. ISBN 9780984442577. Cited 2 time on pages 19 and 24.

BUENO, D. C.; ZAINA, L. M. A comparative exploratory study on mobile Web adaptation in front-end frameworks from the perspective of end-users. In: *15th Brazilian Symposium on Human Factors in Computing Systems - IHC'16*. São Paulo, Brazil: [s.n.], 2016. ISBN 9788576693468. Cited 6 time on pages 19, 31, 33, 35, 36, and 55.

BUENO, D. C.; ZAINA, L. M. HyMobWeb : A hybrid adaptation of context-sensitive Web interfaces with multimodality support in mobile devices. *SBC Journal on Interactive Systems*, 2017. Cited 6 time on pages 19, 24, 30, 31, 33, and 55.

CARDIERI, G. d. A.; ZAINA, L. M. Analyzing user experience in mobile web, native and progressive web applications: A user and hci specialist perspectives. In: *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2018. (IHC 2018), p. 9:1–9:11. ISBN 978-1-4503-6601-4. Availble from Internet: <<http://doi.acm.org/10.1145/3274192.3274201>>. Cited 2 time on pages 53 and 111.

CASADEI, V.; GRANOLLERS, T.; ZAINA, L. Investigating accessibility issues of ui mobile design patterns in online communities: A virtual ethnographic study. In: *Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2017. (IHC 2017), p. 33:1–33:10. ISBN 978-1-4503-6377-8. Available from Internet: <<http://doi.acm.org/10.1145/3160504.3160521>>. Cited on page 81.

CHARLAND, A.; LEROUX, B. Mobile Application Development : Web vs. native. *Communications of the ACM*, v. 54, n. 5, p. 49–53, 2011. ISSN 00010782. Cited 5 time on pages 19, 24, 25, 48, and 51.

CLIFTON, I. G. *Android User Interface Design: Implementing Material Design for Developers*. [S.l.]: Addison-Wesley Professional, 2015. Cited on page 60.

COMSCORE. *The Global Mobile Report*. 2017. Available from Internet: <<https://www.comscore.com/Insights/Presentations-and-Whitepapers/2017/The-Global-Mobile-Report>>. Accessed on: November 12th 2018. Cited on page 19.

CORBETT, E.; WEBER, A. What can i say?: Addressing user experience challenges of a mobile voice user interface for accessibility. In: *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY, USA: ACM, 2016. (MobileHCI '16), p. 72–82. ISBN 978-1-4503-4408-1. Available from Internet: <<http://doi.acm.org/10.1145/2935334.2935386>>. Cited on page 60.

COSTABILE, M. F. et al. End users as unwitting software developers. In: *Proceedings of the 4th international workshop on End-user software engineering - WEUSE '08*. [s.n.], 2008. p. 6–10. ISBN 9781605580340. ISSN 02705257. Available from Internet: <<http://portal.acm.org/citation.cfm?doid=1370847.1370849>>. Cited 5 time on pages 20, 31, 33, 34, and 55.

COYLE, A. *Design better data tables: The ingredients of a successful data table UI*. 2017. Available from Internet: <<https://uxdesign.cc/design-better-data-tables-4ecc99d23356>>. Accessed on: November 19th 2017. Cited on page 36.

DANADO, J.; PATERNÒ, F. Puzzle: A mobile application development environment using a jigsaw metaphor. *Journal of Visual Languages and Computing*, Elsevier, v. 25, n. 4, p. 297–315, 2014. ISSN 1045926X. Available from Internet: <<http://dx.doi.org/10.1016/j.jvlc.2014.03.005>>. Cited 5 time on pages 20, 31, 32, 34, and 55.

DAVIS, F. D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, JSTOR, p. 319–340, 1989. Cited on page 88.

FINK, G.; FLATOW, I. *Pro Single Page Application Development: Using Backbone.Js and ASP.NET*. 1st. ed. Berkely, CA, USA: Apress, 2014. ISBN 1430266732, 9781430266730. Cited 2 time on pages 57 and 59.

FISCHER, G. Meta-design: Expanding boundaries and redistributing control in design. In: *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction*. Berlin, Heidelberg: Springer-Verlag, 2007. (INTERACT'07), p. 193–206. ISBN 3-540-74794-X, 978-3-540-74794-9. Available from Internet: <<http://dl.acm.org/citation.cfm?id=1776994.1777019>>. Cited on page 29.

FISCHER, G. End-user development and meta-design: Foundations for cultures of participation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 5435 LNCS, p. 3–14, 2009. ISSN 03029743. Cited 3 time on pages 20, 28, and 29.

FISCHER, G.; FOGLI, D.; PICCINNO, A. Revisiting and broadening the meta-design framework for end-user development. In: *New Perspectives in End-User Development*. [S.l.]: Springer, Cham, 2017. p. 61–98. Cited 3 time on pages 28, 29, and 81.

FISCHER, G. et al. Converging on a science of design through the synthesis of design methodologies. In: *CHI '07 Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2007. (CHI EA '07), p. 2837–2840. ISBN 978-1-59593-642-4. Available from Internet: <<http://doi.acm.org/10.1145/1240866.1241090>>. Cited on page 29.

FISCHER, G.; NAKAKOJI, K.; YE, Y. Metadesign: Guidelines for supporting domain experts in software development. *IEEE Software*, v. 26, n. 5, p. 37–44, 2009. ISSN 07407459. Cited 4 time on pages 28, 29, 56, and 106.

FISHER, R. A. On the interpretation of  $\chi^2$  from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, [Wiley, Royal Statistical Society], v. 85, n. 1, p. 87–94, 1922. ISSN 09528385. Available from Internet: <<http://www.jstor.org/stable/2340521>>. Cited on page 103.

FORTUNATO, D.; BERNARDINO, J. Progressive web apps: An alternative to the native mobile Apps | Progressive Web Apps: uma alternativa às Apps móveis nativas. *Iberian Conference on Information Systems and Technologies, CISTI, AISTI*, v. 2018-June, p. 1–6, 2018. Cited 3 time on pages 31, 32, and 55.

GALITZ, W. O. *The Essential Guide to User Interface Design: An Introduction to GUI Design*. 3. ed. [S.l.: s.n.], 2007. 857 p. Cited 3 time on pages 19, 23, and 36.

GAZDECKI, A. *Why Progressive Web Apps Will Replace Native Mobile Apps*. 2015. Available from Internet: <<https://www.forbes.com/sites/forbestechcouncil/2018/03/09/why-progressive-web-apps-will-replace-native-mobile-apps>>. Accessed on: May 31st 2018. Cited on page 48.

GHIANI, G. et al. Beyond responsive design: Context-dependent multimodal augmentation of web applications. In: *11th International Conference on Mobile Web and Intelligent Information Systems - MobiWis'14*. [S.l.: s.n.], 2014. v. 8640 LNCS, p. 71–85. Cited 7 time on pages 19, 24, 31, 32, 33, 55, and 59.

Google Developers. *Jumia sees 33% increase in conversion rate, 12X more users on PWA*. 2017. Available from Internet: <<https://developers.google.com/web/showcase/2017/jumia>>. Accessed on: November 12th 2018. Cited on page 26.

Google Developers. *reating UX that “Just Feels Right” with Progressive Web Apps (Google I/O '17)*. 2017. Available from Internet: <<https://www.youtube.com/watch?v=mmq-KVeO-uU>>. Accessed on: June 21st 2018. Cited on page 26.

Google Developers. *Case Studies | Web | Google*. 2018. Available from Internet: <<https://developers.google.com/web/showcase/>>. Accessed on: January 23rd 2019. Cited on page 20.

- Google Developers. *Introduction to Progressive Web App Architectures*. 2018. Available from Internet: <<https://developers.google.com/web/ilt/pwa/introduction-to-progressive-web-app-architectures>>. Accessed on: July 10th 2018. Cited 4 time on pages 9, 27, 58, and 59.
- Google Developers. *Progressive Web Apps | Web | Google Developers*. 2018. Available from Internet: <<https://developers.google.com/web/progressive-web-apps/>>. Accessed on: May 10th 2018. Cited on page 26.
- GULLÀ, F. et al. Design Adaptable and Adaptive User Interfaces: A Method to Manage the Information. In: *Biosystems & Biorobotics*. [S.l.]: Springer International Publishing, 2015. v. 11, p. 47–58. ISBN 978-3-319-18373-2. Cited 7 time on pages 19, 23, 24, 31, 33, 56, and 59.
- HARLEY, A. *Maps and Location Finders on Mobile Devices*. 2014. Available from Internet: <<https://www.nngroup.com/articles/mobile-maps-locations/>>. Accessed on: November 19th 2017. Cited on page 36.
- HASSENZAHN, M. *Experience Design: Technology for All the Right Reasons*. [S.l.: s.n.], 2010. Cited on page 38.
- HAYASHI, E. C. S. et al. Exploring new formats of the self-assessment manikin in the design with children. In: *Proceedings of the 15th Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2016. (IHC '16), p. 27:1–27:10. ISBN 978-1-4503-5235-2. Available from Internet: <<http://doi.acm.org/10.1145/3033701.3033728>>. Cited on page 39.
- HENRY, S. L.; ABOU-ZAHRA, S.; BREWER, J. The role of accessibility in a universal web. In: *Proceedings of the 11th Web for All Conference*. New York, NY, USA: ACM, 2014. (W4A '14), p. 17:1–17:4. ISBN 978-1-4503-2651-3. Available from Internet: <<http://doi.acm.org/10.1145/2596695.2596719>>. Cited on page 81.
- Interaction Design Foundation. *User Interface (UI) Design Patterns*. 2017. Available from Internet: <<https://www.interaction-design.org/literature/topics/ui-design-patterns>>. Accessed on: October 28th 2017. Cited 2 time on pages 23 and 36.
- LAZAR, J.; FENG, J. H.; HOCHHEISER, H. *Research Methods in Human-Computer Interaction*. 2. ed. [S.l.: s.n.], 2017. 560 p. ISBN 9780128053904. Cited 2 time on pages 35 and 65.
- LEITÃO, C. S. d. S.; FARIA, C. *Semiotic Engineering Methods for Scientific Research in HCI*. [S.l.]: Morgan & Claypool, 2009. ISBN 9781598299441. Cited on page 65.
- LEIRA, E. de; GARRETA-DOMINGO, M. Ten Emotion Heuristics: Guidelines for assessing the user's affective dimension easily and cost-effectively. In: *21st BCS HCI Group Conference*. [S.l.: s.n.], 2007. v. 2, p. 7–10. ISBN 978-1-902505-95-4. Cited on page 39.
- LIEBERMAN, H. et al. End-user development: An emerging paradigm. *End User Development*, v. 9, p. 1–8, 2006. ISSN 09376429. Cited on page 28.



- MANCA, M. et al. Generation of multi-device adaptive multimodal web applications. In: *10th International Conference on Mobile Web and Intelligent Information Systems - MobiWis'13*. [S.l.: s.n.], 2013. v. 8093 LNCS, p. 218–232. Cited 6 time on pages 19, 31, 32, 33, 55, and 59.
- MARCOTTE, E. *Responsive web design, a book apart*. 2011. Available from Internet: <<https://alistapart.com/article/responsive-web-design>>. Accessed on: October 30th 2017. Cited on page 24.
- MARQUES, A. B.; LOPES, A.; CONTE, T. Aplicação de métodos de avaliação da experiência do usuário na utilização de serious game em sala de aula. In: *Congresso Brasileiro de Software: Teoria e Prática*. [S.l.: s.n.], 2014. p. 22–31. Cited on page 39.
- MEIER, S.; HEIDMANN, F.; THOM, A. A Comparison of Location Search UI Patterns on Mobile Devices. In: *16th International Conference on Human-computer Interaction with Mobile Devices and Services - MobileHCI '14*. [S.l.: s.n.], 2014. p. 465–470. ISBN 9781450330046. Cited on page 36.
- NAMOUN, A. et al. Exploring mobile end user development: Existing use and design factors. *IEEE Transactions on Software Engineering*, IEEE, v. 42, n. 10, p. 960–976, 2016. ISSN 00985589. Cited 8 time on pages 20, 31, 33, 34, 55, 67, 81, and 106.
- NEBELING, M.; SPEICHER, M.; NORRIE, M. W3touch: Metrics-based Web Page Adaptation for Touch. In: *SIGCHI Conference on Human Factors in Computing Systems - CHI'2013*. Paris, France: [s.n.], 2013. p. 2311. Cited 5 time on pages 19, 31, 32, 33, and 55.
- NEUHAUS, J. *Angular vs. React vs. Vue: A 2017 comparison*. 2017. Available from Internet: <<https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176>>. Accessed on: July 29th 2018. Cited on page 60.
- PATERNÒ, F. End User Development: Survey of an Emerging Field for Empowering People. *ISRN Software Engineering*, v. 2013, p. 1–11, 2013. ISSN 2090-7680. Available from Internet: <<http://www.hindawi.com/journals/isrn/2013/532659/>>. Cited on page 20.
- PETELE. *Your First Progressive Web App*. 2016. Available from Internet: <<https://codelabs.developers.google.com/codelabs/your-first-pwapp/index.html>>. Accessed on: September 26th 2017. Cited 2 time on pages 19 and 25.
- PRATES R. O., S. C. S.; BARBOSA, S. D. J. A Method for Evaluating the Communicability of User Interfaces. *methods&tools*, n. february, p. 31–38, 2000. Cited on page 69.
- PREECE, J.; SHARP, H.; ROGERS, Y. *Interaction Design-beyond human-computer interaction*. [S.l.: s.n.], 2015. 523–1 p. Cited on page 67.
- PROENÇA, M. Q.; NERIS, V. P. A. UIFlex : a meta-design solution for the web. In: *IHC '17*. [S.l.: s.n.], 2017. v. 2, p. 461–470. ISBN 9788576694052. Cited 8 time on pages 19, 20, 29, 30, 31, 33, 55, and 59.
- RIBEIRO, J. *Web Design Patterns for Mobile Devices*. 2012. Master's thesis. Cited on page 36.

RICHARDS, M. *Software Architecture Patterns*. [S.l.: s.n.], 2015. ISBN 9781491971437. Cited on page 56.

RUSSELL, A. *Progressive Web Apps: Escaping Tabs Without Losing Our Soul*. 2015. Available from Internet: <<https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/>>. Accessed on: November 15th 2017. Cited on page 25.

SALMAN, I.; MISIRLI, A. T.; JURISTO, N. Are students representatives of professionals in software engineering experiments? In: IEEE PRESS. *Proceedings of the 37th International Conference on Software Engineering-Volume 1*. [S.l.], 2015. p. 666–676. Cited 3 time on pages 85, 106, and 107.

SÁNCHEZ, J. L. G. et al. Playability: Analysing user experience in video games. *Behaviour and Information Technology*, v. 31, n. 10, p. 1033–1054, 2012. ISSN 0144929X. Cited on page 39.

SHEPPARD, D. *Beginning progressive web app development creating a native app experience on the web*. [S.l.: s.n.], 2017. 978–1 p. ISBN 978-1-4842-3089-3. Cited 5 time on pages 9, 25, 26, 27, and 60.

STRAUSS, A.; CORBIN, J. M. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. 2. ed. CA: Sage: Thousand Oaks, 1998. Cited on page 42.

TARNOWSKI, D. *Angular vs React—the DEAL BREAKER: A REAL, EXTREME framework comparison*. 2017. Available from Internet: <<https://hackernoon.com/angular-vs-react-the-deal-breaker-7d76c04496bc>>. Accessed on: July 29th 2018. Cited on page 60.

Tech Magic. *ReactJS vs Angular5 vs Vue.js—What to choose in 2018?* 2018. Available from Internet: <<https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d>>. Accessed on: July 29th 2018. Cited on page 60.

TIDWELL, J. *Designing Interfaces*. 2. ed. [S.l.: s.n.], 2011. 278 p. ISBN 1449379702. Cited 2 time on pages 23 and 35.

UIPATTERNS. *Navigation Tabs Design Pattern*. 2017. Available from Internet: <<http://ui-patterns.com/patterns/NavigationTabs>>. Accessed on: November 19th 2017. Cited on page 36.

VUE.JS. *Comparison with Other Frameworks*. 2018. Available from Internet: <<https://vuejs.org/v2/guide/comparison.html>>. Accessed on: July 29th 2018. Cited on page 60.

WOHLIN, C. et al. *Experimentation in software engineering: an introduction*. [S.l.: s.n.], 2012. 228 p. ISBN 0792386825. Cited 4 time on pages 23, 35, 52, and 106.

XAVIER, R. A. C.; GARCIA, F. E.; NERIS, V. P. d. A. Decisões de design de interfaces ruins e o impacto delas na interação: um estudo preliminar considerando o estado emocional de idosos. In: *11th Brazilian Symposium on Human Factors in Computing Systems, IHC'12*. [S.l.: s.n.], 2012. v. 5138, p. 127–136. ISBN 978-85-7669-262-1. Cited on page 39.



# APPENDIX A – TCLE 1

## Termo de Consentimento Livre e Esclarecido

1. Você está sendo convidado para participar da pesquisa "Comparação entre elementos de interação de diferentes tipos de aplicações mobile".
2. Você foi selecionado para ser voluntário e sua participação não é obrigatória.
3. A qualquer momento você pode desistir de participar e retirar seu consentimento.
4. Sua recusa não trará nenhum prejuízo em sua relação com o pesquisador, com a instituição.
5. Essa pesquisa tem por objetivo comparar diferentes aplicações, através de elementos de interação e recursos, em relação a aspectos de satisfação do usuário, eficiência e dificuldades encontradas.
6. Sua participação nesta pesquisa consistirá em seguir as tarefas definidas que serão entregues a você e responder a um questionário em relação ao elementos de interação de diferentes aplicativos.
7. A sua participação na pesquisa pode envolver algum desconforto relacionado ao tempo despendido com a realização da sessão e do preenchimento de questionários, sendo que faremos o possível para minimizar possíveis desconfortos.
8. As informações obtidas através dessa pesquisa serão confidenciais e asseguramos o sigilo sobre sua participação.
9. Os dados não serão divulgados de forma a possibilitar sua identificação.
10. Sua participação estará sendo gravada com o intuito de coletar dados sobre a interação com o *smartphone* e com os elementos de tela.
11. Você receberá uma cópia deste termo onde consta informações do pesquisador, podendo tirar suas dúvidas sobre o projeto e sua participação, agora ou a qualquer momento.

---

Giulia de Andrade Cardieri

Universidade Federal de São Carlos (UFSCar) -Departamento de Computação (DC)  
Rodovia João Leme dos Santos, (SP-264), Km 110, s/n - Itinga, Sorocaba - SP, 18052-780.  
Tel.: +55 16 3351-6000

**Declaro que entendi os objetivos, riscos e benefícios de minha participação na pesquisa e concordo em participar.**

**Itapetininga, 03/10/2017**

---

Assinatura do Sujeito da pesquisa



# APPENDIX B – Profile Questionnaire 1

## Questionário de perfil do participante

Obrigada pela sua inscrição para participar do experimento!

\* Required

1. Qual seu número do RG? \*

\_\_\_\_\_

2. Qual seu sexo? \*

Mark only one oval.

- Feminino  
 Masculino  
 Prefiro não informar

3. Qual sua idade? \*

\_\_\_\_\_

4. Qual seu nível de escolaridade? \*

Mark only one oval.

- Ensino fundamental incompleto  
 Ensino fundamental  
 Ensino médio incompleto  
 Ensino médio  
 Ensino superior incompleto  
 Ensino superior  
 Pós-graduação  
 Especialização

5. Qual sistema operacional você utiliza em seus dispositivos móveis com maior frequência? \*

Mark only one oval.

- iOS  
 Android  
 Windows Phone  
 Other: \_\_\_\_\_

6. Qual navegador você costuma utilizar em seus dispositivos móveis com maior frequência? \*

Mark only one oval.

- Navegador padrão iOS (Safari)
- Google Chrome
- Firefox
- Opera
- Internet Explorer
- Other: \_\_\_\_\_

7. Qual tipo de rede para acesso a internet você utiliza com maior frequência? \*

Mark only one oval.

- 3G/4G
- Wifi

8. Quantas vezes por mês você costuma baixar aplicativos? \*

Mark only one oval.

- Não baixo aplicativos todo mês
- Entre 1 e 2 vezes
- Entre 3 e 4 vezes
- Mais de 5 vezes

**9. Selecione a opção que mais representa sua opinião sobre diversos fatores ao baixar um aplicativo \***

*Mark only one oval per row.*

	Discordo totalmente	Discordo amplamente	Discordo parcialmente	Concordo parcialmente	Concordo amplamente	Concordo totalmente
Não vou baixar um aplicativo se já existe um site com conteúdo similar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Prefiro usar aplicativos em relação a sites no navegador	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Não vou baixar um aplicativo se estiver sem acesso a Wifi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Costumo ter pouco espaço de armazenamento no meu dispositivo móvel	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Não gosto da necessidade de pesquisar o aplicativo na Google Play/App Store	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O tempo de espera para baixar um aplicativo não me agrada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**10. Indique a sua frequência de acesso à internet (através de dispositivos móveis) nos ambientes listados abaixo \***

*Mark only one oval per row.*

	Sempre (todos os dias)	Frequentemente (em média, 3 vezes por semana)	Com frequência razoável (em média, 1 vez por semana)	Raramente (em média, 1 vez por mês)	Não se aplica
Em casa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
No trabalho	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Na casa de amigos/parentes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Na escola/faculdade	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Em locais públicos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



**11. Informe quantas vezes você costuma acessar (através de dispositivos móveis) os tipos de sites listados \***

*Mark only one oval per row.*

	Sempre (todos os dias)	Frequentemente (em média, 3 vezes por semana)	Com frequência razoável (em média, 1 vez por semana)	Raramente (em média, 1 vez por mês)	Não se aplica
E-commerce (sites de compras)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Redes Sociais	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Portais (G1, UOL, Terra)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Midiático (blogs, sites de jornais e revistas)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fóruns de discussão	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sites de busca (Google, Yahoo, Bing)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Jogos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sites de vídeos (YouTube, Vimeo)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ferramentas em geral (email, mapas)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Viagens (reserva de hotéis, passagens aéreas)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>







**3. Motivação \***

Você deverá escolher um valor entre 1 (totalmente desmotivado) e 9 (totalmente motivado), onde 5 é o valor neutro, para representar sua motivação ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

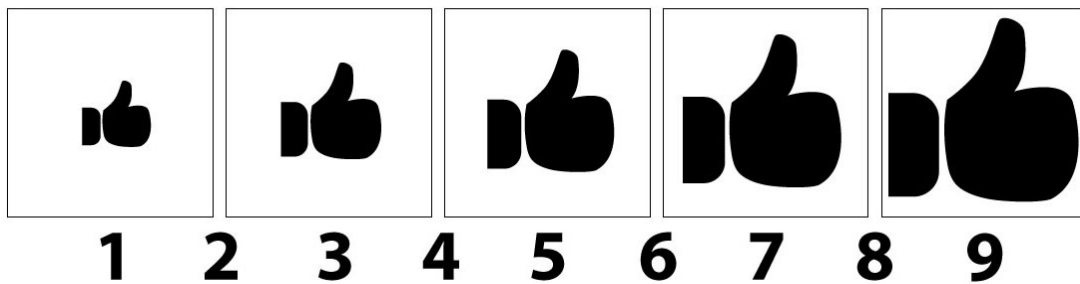
---

---

**4. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

---

---

**5. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**Trivago 1 - Tarefa 2 - Elemento datepicker/calendário**

**Passo 1: Escolher a data 20/11/2017 como entrada**

---

**Passo 2: Escolher a data 15/11/2017 como saída**

---

**Passo 3: Caso não consiga usar a data 15/11/2017, escolher a data 1/12/17 como saída**

---





**11. Motivação \***

Você deverá escolher um valor entre 1 (totalmente desmotivado) e 9 (totalmente motivado), onde 5 é o valor neutro, para representar sua motivação ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

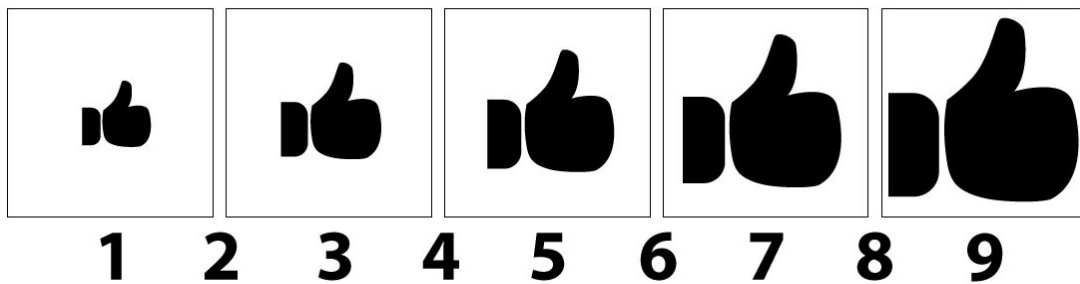
1    2    3    4    5    6    7    8    9

---

**12. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

---

**13. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**Trivago 1 - Tarefa 4 - Elemento mapas**

**Passo 1: Selecionar a opção "Ir para o mapa"**

---

**Passo 2: Usar o movimento de Pinch/Pinça (exibido na imagem abaixo) para aumentar o zoom deixando o centro do mapa mais próximo.**

---

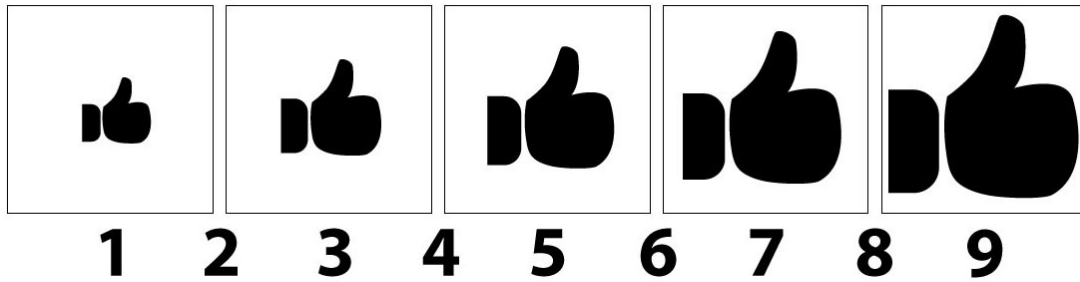
**Movimento de Pinch/Pinça**





**16. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1 2 3 4 5 6 7 8 9

---

---

**17. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**Trivago 1 - Tarefa 5 - Elemento tabela e menu de abas**

**Passo 1: Clicar no nome do resultado da busca, "Vitoria Concept Campinas"**

---

**Passo 2: Clicar na opção "Ofertas" no menu de abas superior**

---

**Passo 3: Navegar até o final da tabela de preços que apareceu e clicar na opção "Ver mais", e navegar novamente até o final da tabela.**

---

**Questionário sobre Trivago 1 - Tarefa 5**





**23. Motivação \***

Você deverá escolher um valor entre 1 (totalmente desmotivado) e 9 (totalmente motivado), onde 5 é o valor neutro, para representar sua motivação ao realizar a Tarefa. Utilize as figuras para ajudar.

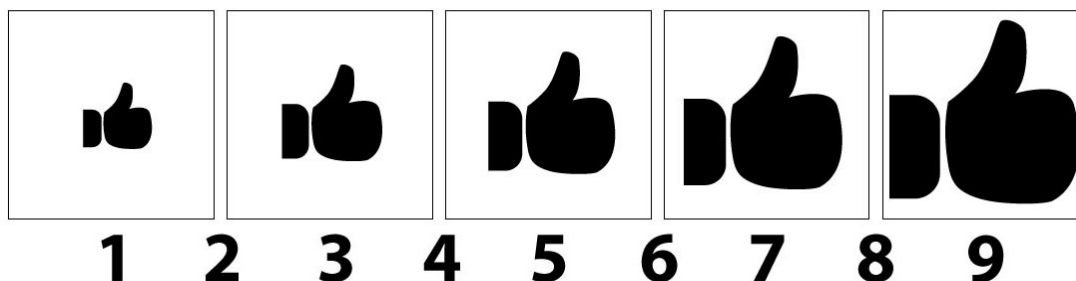


Mark only one oval.

1 2 3 4 5 6 7 8 9

**24. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1 2 3 4 5 6 7 8 9

**25. Você tem sugestões/reclamações/observações dessa tarefa? \***

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Trivago 2 - Tarefa 2 - Elemento datepicker/calendário**

**Passo 1: Clicar no campo "Entrada"**

---

**Passo 2: Escolher a data 20/11/2017 como entrada**

---

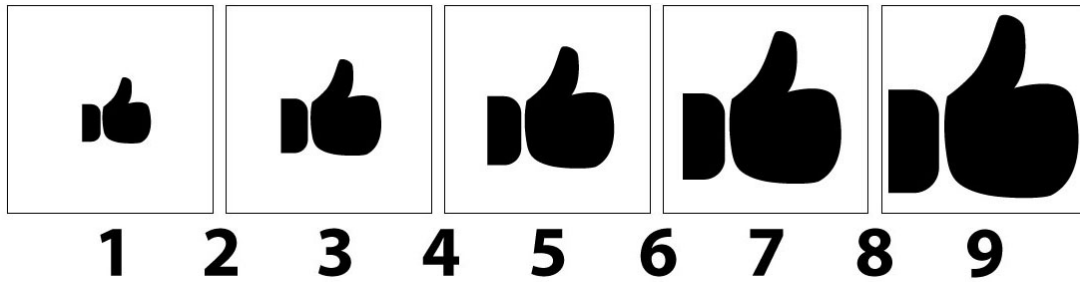
**Passo 3: Tentar escolher a data 15/11/2017 como saída**

---



**28. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



*Mark only one oval.*

1 2 3 4 5 6 7 8 9

---

---

**29. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**Trivago 2 - Tarefa 3 - Elemento select/múltipla escolha**

**Passo 1: Clicar no select "Ordenar por popularidade" localizado na parte inferior da tela**

---

**Passo 2: Selecionar a opção "Priorizar por avaliação"**

---

**Questionário sobre Trivago 2 - Tarefa 3**





33. Você tem sugestões/reclamações/observações dessa tarefa? \*

---

---

---

---

---

## **Trivago 2 - Tarefa 4 - Elemento mapas**

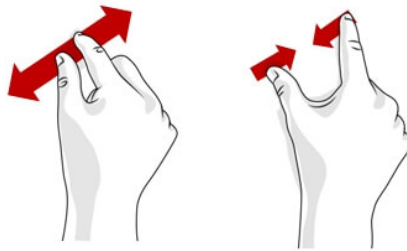
**Passo 1: Clicar no ícone que representa um mapa/marcador de mapa, localizado no topo direito da tela**

---

**Passo 2: Usar o movimento de Pinch/Pinça (exibido na imagem abaixo) para aumentar o zoom deixando o centro do mapa mais próximo.**

---

### **Movimento de Pinch/Pinça**



**Passo 3: Clicar no quadrado verde com o número 77**

---

**Passo 4: Clicar no ícone que representa uma lista, localizado no topo direito da tela**

---

## **Questionário sobre Trivago 2 - Tarefa 4**





**39. Motivação \***

Você deverá escolher um valor entre 1 (totalmente desmotivado) e 9 (totalmente motivado), onde 5 é o valor neutro, para representar sua motivação ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

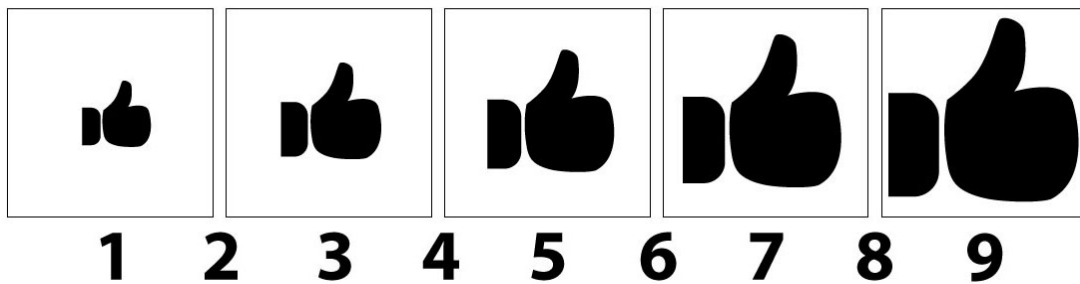
---

---

**40. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

---

---

**41. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**Aplicativo 3 - TripAdvisor 1**

Clicar no ícone do aplicativo TripAdvisor 1 (o ícone de fundo branco com o logo de uma cabeça de coruja)

**TripAdvisor 1 - Tarefa 1 - Elemento input/caixa de busca**

**Passo 1: Pesquisar por "Campinas" no campo de buscas de hotel/cidade**

---

**Questionário sobre TripAdvisor 1 - Tarefa 1**





**47. Motivação \***

Você deverá escolher um valor entre 1 (totalmente desmotivado) e 9 (totalmente motivado), onde 5 é o valor neutro, para representar sua motivação ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

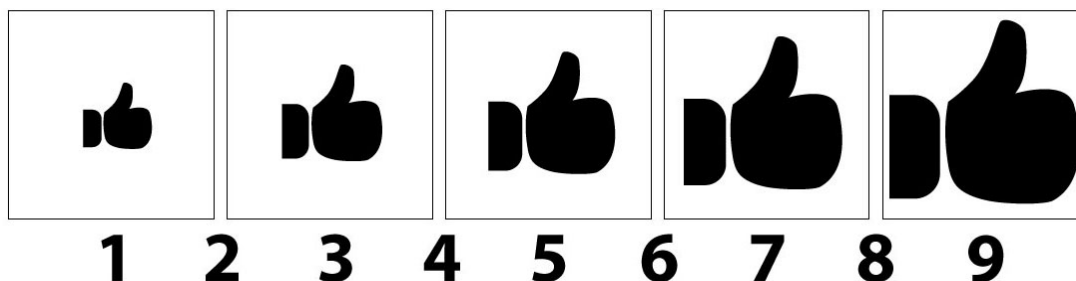
---

---

**48. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

---

---

**49. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**TripAdvisor 1 - Tarefa 3 - Elemento mapas**

**Passo 1: Clicar no nome do resultado da busca, "Vitoria Concept Campinas"**

---

**Passo 2: Navegar na página, no sentido para baixo, até encontrar a opção "Mapa" no canto esquerdo, clicar nesse link**

---





**51. Motivação \***

Você deverá escolher um valor entre 1 (totalmente desmotivado) e 9 (totalmente motivado), onde 5 é o valor neutro, para representar sua motivação ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

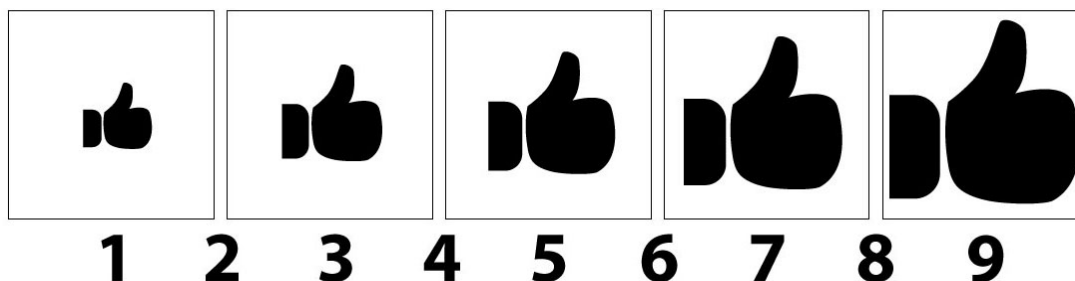
1    2    3    4    5    6    7    8    9

---

**52. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

---

**53. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**TripAdvisor 1 - Tarefa 4 - Elemento select/múltipla escolha**

**Passo 1: Clicar no select/múltipla escolha R\$ BRL**

---

**Passo 2: Selecionar a opção "CA\$ CAD Dólar Canadense"**

---

**Questionário sobre TripAdvisor 1 - Tarefa 4**





**59. Motivação \***

Você deverá escolher um valor entre 1 (totalmente desmotivado) e 9 (totalmente motivado), onde 5 é o valor neutro, para representar sua motivação ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

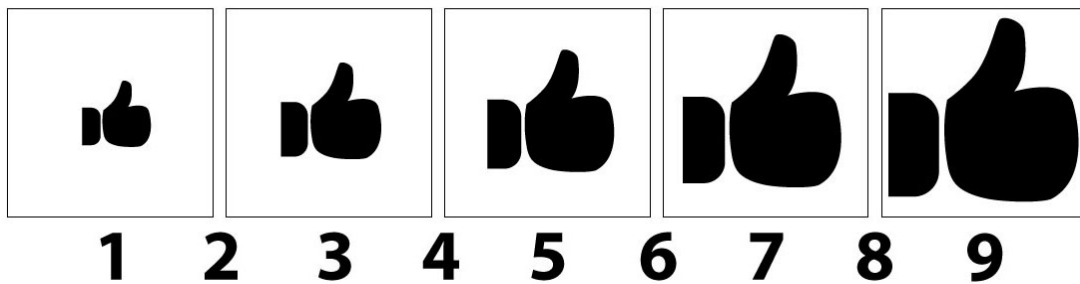
1    2    3    4    5    6    7    8    9

---

**60. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

---

**61. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**TripAdvisor 2 - Tarefa 2 - Elemento datepicker/calendário**

**Passo 1: Clicar no campo de data/Check-in, localizado no canto superior esquerdo embaixo da barra verde. Estará preenchido com a data de hoje e o texto "1 diária".**

---

**Passo 2: Escolher a data 20/11/2017 como Check-in**

---

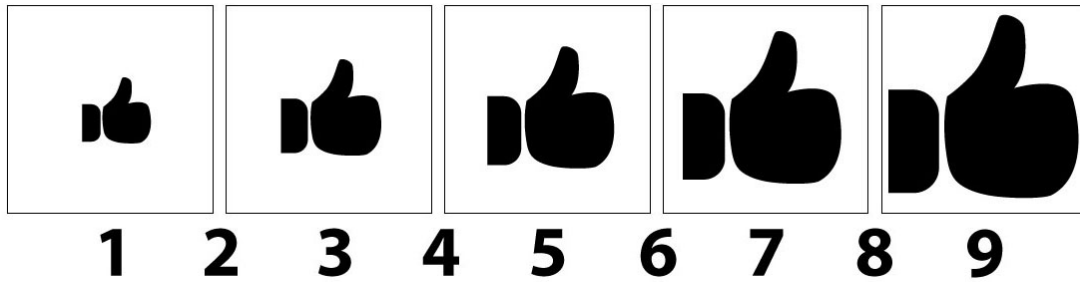
**Passo 3: Tentar escolher a data 15/11/2017 como Check-out**

---



**64. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



*Mark only one oval.*

1 2 3 4 5 6 7 8 9

---

---

**65. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**TripAdvisor 2 - Tarefa 3 - Elemento tabela e menu de abas**

**Passo 1: Clicar no nome do resultado da busca, "Vitoria Concept Campinas"**

---

**Passo 2: Clicar no botão "Ver X ofertas a partir de R\$..."**

---

**Passo 3: Navegar para baixo até aparecer um menu de abas no topo da tela. Clicar na opção "Avaliações" no menu superior**

---

**Questionário sobre TripAdvisor 2 - Tarefa 3**



69. Você tem sugestões/reclamações/observações dessa tarefa? \*

---

---

---

---

---





# APPENDIX D – Tasks/SAM Questionnaire - Type B

## Parte 1 - Experimento sobre Aplicativos

Nesse experimento serão testados elementos de interação de quatro aplicações de dispositivos móveis.

Na parte 1 serão testados elementos de interação num ambiente com acesso a internet. Para isso serão realizadas tarefas em quatro aplicativos diferentes, com o objetivo de avaliar o uso de diferentes elementos de interação. Após cada tarefa um questionário será exibido.

Em cada tarefa execute as instruções exatamente como pedidas e ao terminar continue para a próxima seção.

Obrigada pela sua participação! Qualquer dúvida favor perguntar para algum dos instrutores.

\* Required

1. Qual seu número de RG? \*

---

### Aplicativo 1 - Trivago 2

Clicar no ícone do aplicativo Trivago 2 (o ícone de fundo branco com o texto trivago escrito em três cores diferentes) e esperar ele abrir.

### Trivago 2 - Tarefa 1 - Elemento input/caixa de busca

**Passo 1: Apertar o ícone de microfone dentro da caixa de busca**

---

**Passo 2: Quando o microfone em um círculo verde/azul aparecer falar a palavra "Campinas". Caso essa tentativa dê errado após a terceira vez, digitar "Campinas" na caixa de busca.**

---

**Passo 3: Escolher a opção "Campinas - São Paulo, Brazil"**

---

**Questionário sobre Trivago 2 - Tarefa 1**





**7. Motivação \***

Você deverá escolher um valor entre 1 (totalmente desmotivado) e 9 (totalmente motivado), onde 5 é o valor neutro, para representar sua motivação ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

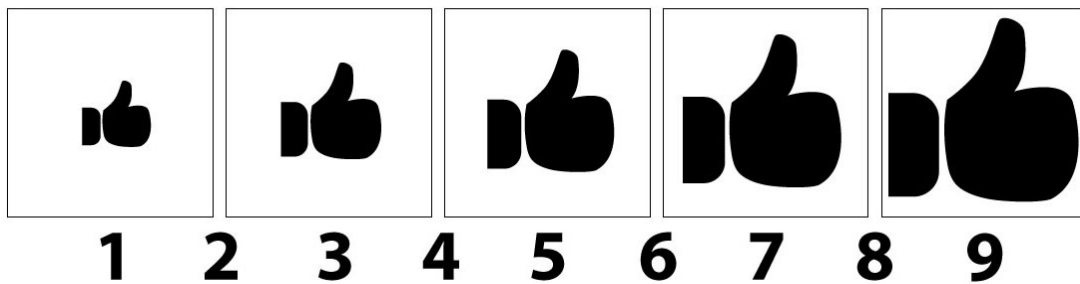
---

---

**8. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

---

---

**9. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**Trivago 2 - Tarefa 3 - Elemento select/múltipla escolha**

**Passo 1: Clicar no select "Ordenar por popularidade" localizado na parte inferior da tela**

---

**Passo 2: Selecionar a opção "Priorizar por avaliação"**

---

**Questionário sobre Trivago 2 - Tarefa 3**



13. Você tem sugestões/reclamações/observações dessa tarefa? \*

---

---

---

---

---

## **Trivago 2 - Tarefa 4 - Elemento mapas**

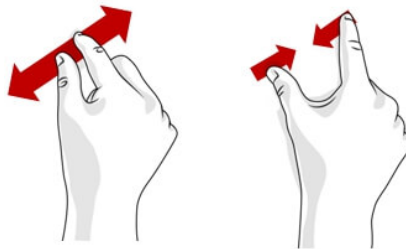
**Passo 1: Clicar no ícone que representa um mapa/marcador de mapa, localizado no topo direito da tela**

---

**Passo 2: Usar o movimento de Pinch/Pinça (exibido na imagem abaixo) para aumentar o zoom deixando o centro do mapa mais próximo.**

---

### **Movimento de Pinch/Pinça**



**Passo 3: Clicar no quadrado verde com o número 77**

---

**Passo 4: Clicar no ícone que representa uma lista, localizado no topo direito da tela**

---

## **Questionário sobre Trivago 2 - Tarefa 4**







**19. Motivação \***

Você deverá escolher um valor entre 1 (totalmente desmotivado) e 9 (totalmente motivado), onde 5 é o valor neutro, para representar sua motivação ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

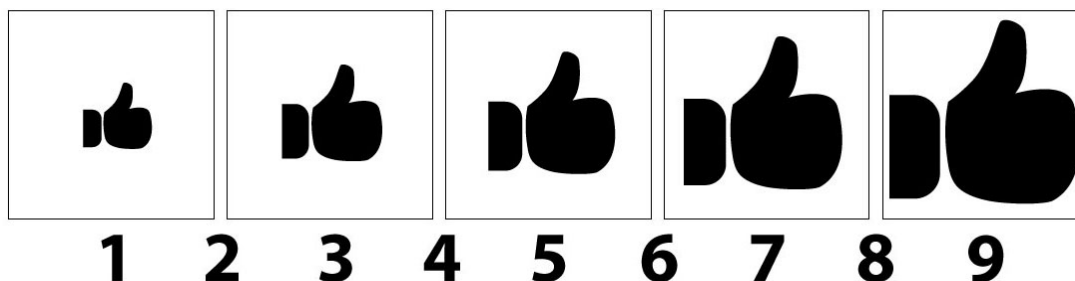
1    2    3    4    5    6    7    8    9

---

**20. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

---

**21. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**Aplicativo 2 - Trivago 1**

Clicar no ícone do aplicativo Trivago 1 (o ícone de fundo branco com três retângulos de cores diferentes) e esperar ele abrir.

**Trivago 1 - Tarefa 1 - Elemento input/caixa de busca**

**Passo 1: Pesquise pelo termo "Campinas" na caixa de busca**

---

**Questionário sobre Trivago 1 - Tarefa 1**





**27. Motivação \***

Você deverá escolher um valor entre 1 (totalmente desmotivado) e 9 (totalmente motivado), onde 5 é o valor neutro, para representar sua motivação ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

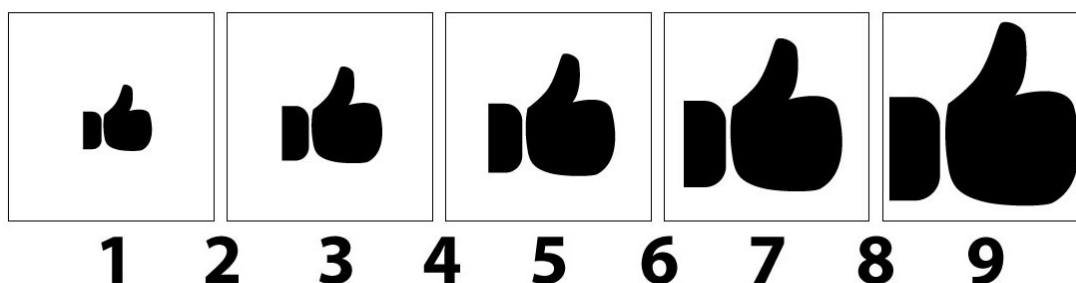
1    2    3    4    5    6    7    8    9

---

**28. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

---

**29. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**Trivago 1 - Tarefa 3 - Elemento select/múltipla escolha**

**Passo 1: Escolher a opção "Quarto Duplo" do select/múltipla escolha**

---

**Passo 2: Clicar no select "Ordenar por popularidade" localizado abaixo do datepicker no canto direito da tela**

---

**Passo 3: Selecionar a opção "Priorizar por avaliação"**

---





**35. Motivação \***

Você deverá escolher um valor entre 1 (totalmente desmotivado) e 9 (totalmente motivado), onde 5 é o valor neutro, para representar sua motivação ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

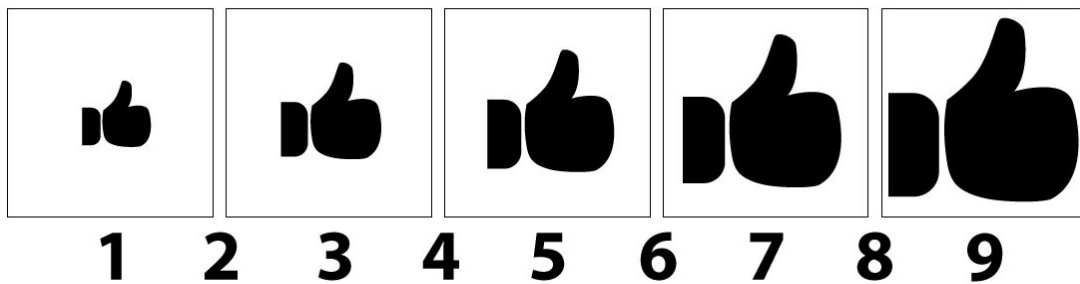
1    2    3    4    5    6    7    8    9

---

**36. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

---

**37. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**Trivago 1 - Tarefa 5 - Elemento tabela e menu de abas**

**Passo 1: Clicar no nome do resultado da busca, "Vitoria Concept Campinas"**

---

**Passo 2: Clicar na opção "Ofertas" no menu de abas superior**

---

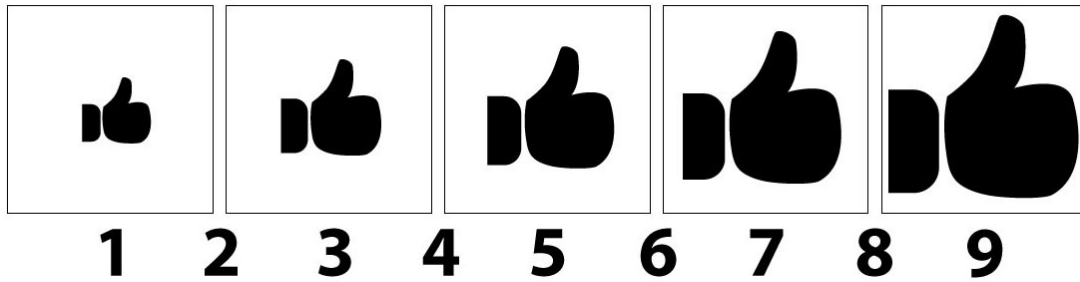
**Passo 3: Navegar até o final da tabela de preços que apareceu e clicar na opção "Ver mais", e navegar novamente até o final**





**40. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1 2 3 4 5 6 7 8 9

**41. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**Aplicativo 3 - TripAdvisor 2**

Clicar no ícone do aplicativo TripAdvisor 2 (o ícone de fundo verde com o logo de uma cabeça de coruja)

**TripAdvisor 2 - Tarefa 1 - Elemento input/caixa de busca**

**Passo 1: Digitar "Campinas" no campo de buscas**

---

**Passo 2: Clicar no botão "Hotéis"**

---

**Questionário sobre TripAdvisor 2 - Tarefa 1**





**47. Motivação \***

Você deverá escolher um valor entre 1 (totalmente desmotivado) e 9 (totalmente motivado), onde 5 é o valor neutro, para representar sua motivação ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

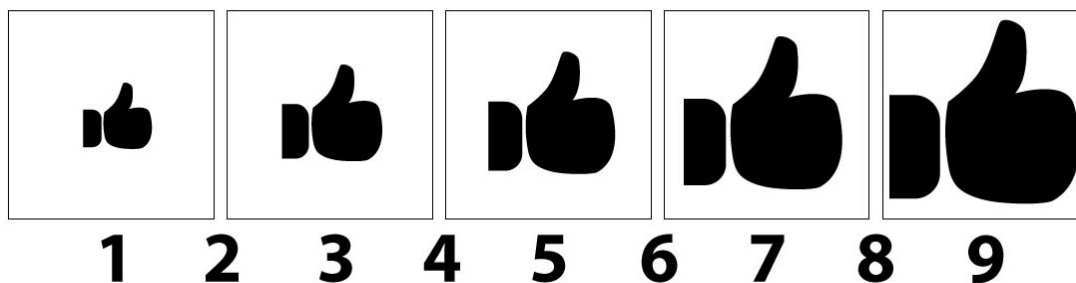
1    2    3    4    5    6    7    8    9

---

**48. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1    2    3    4    5    6    7    8    9

---

**49. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**TripAdvisor 2 - Tarefa 3 - Elemento tabela e menu de abas**

**Passo 1: Clicar no nome do resultado da busca, "Vitoria Concept Campinas"**

---

**Passo 2: Clicar no botão "Ver X ofertas a partir de R\$..."**

---

**Passo 3: Navegar para baixo até aparecer um menu de abas no topo da tela. Clicar na opção "Avaliações" no menu superior**

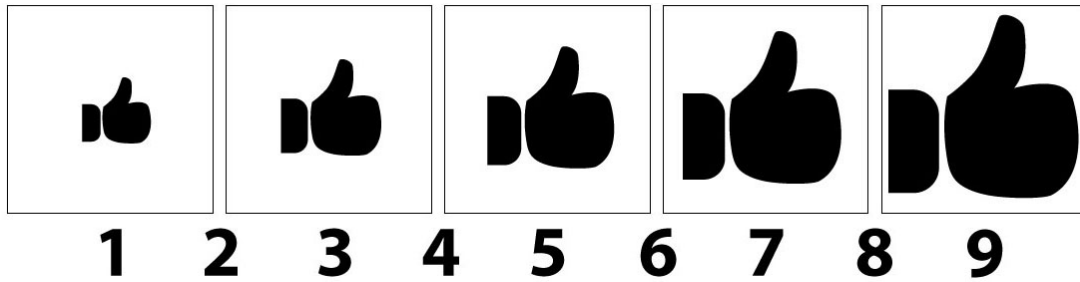
---





**56. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1	2	3	4	5	6	7	8	9
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**57. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---

**TripAdvisor 1 - Tarefa 2 - Elemento datepicker/calendário**

**Passo 1: Clicar no campo "Check-in"**

---

**Passo 2: Escolher a data 20/11/2017 como Check-in**

---

**Passo 3: Tentar escolher a data 15/11/2017 como Check-out**

---

**Passo 4: Caso não consiga usar a data 15/11/2017, escolher a data 1/12/17 como Check-out**

---

**Passo 5: Fechar o datepicker caso ele ainda esteja aberto**

---

**Passo 6: Clicar no botão "Encontrar hotéis"**

---

**Questionário sobre TripAdvisor 1 - Tarefa 2**





61. Você tem sugestões/reclamações/observações dessa tarefa? \*

---

---

---

---

---

### **TripAdvisor 1 - Tarefa 3 - Elemento mapas**

**Passo 1: Clicar no nome do resultado da busca, "Vitoria Concept Campinas"**

---

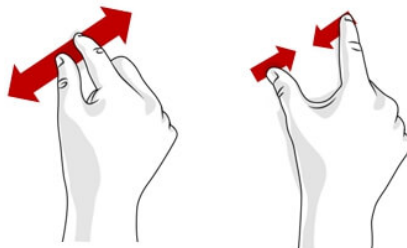
**Passo 2: Navegar na página, no sentido para baixo, até encontrar a opção "Mapa" no canto esquerdo, clicar nesse link**

---

**Passo 3: Usar o movimento de Pinch/Pinça (exibido na imagem abaixo) para aumentar o zoom deixando o centro do mapa mais próximo.**

---

**Movimento de Pinch/Pinça**



**Passo 4: Clicar no botão "Exibir preços"**

---

**Passo 5: Navegar ao fim da página até encontrar a seção Moeda/País**

---

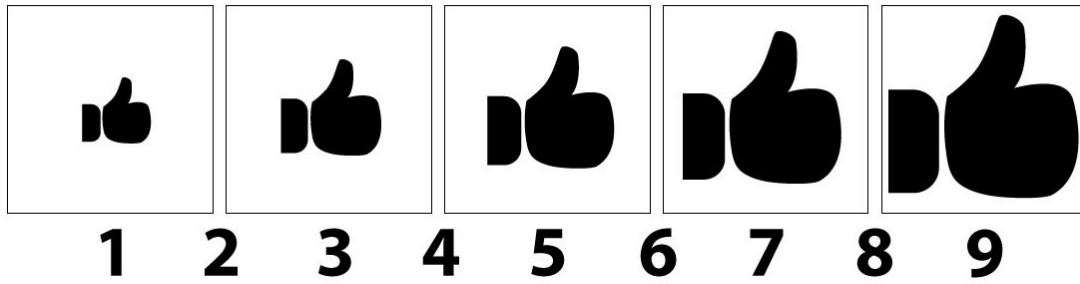
**Questionário sobre TripAdvisor 1 - Tarefa 3**





**68. Sentimento de controle \***

Você deverá escolher um valor entre 1 (controlado pela situação) e 9 (no controle da situação), onde 5 é o valor neutro, para representar o sentimento de controle ao realizar a Tarefa. Utilize as figuras para ajudar.



Mark only one oval.

1 2 3 4 5 6 7 8 9

---

---

**69. Você tem sugestões/reclamações/observações dessa tarefa? \***

---

---

---

---

---





# APPENDIX E – TCLE 2

## Termo de Consentimento Livre e Esclarecido

1. Você está sendo convidado para participar da pesquisa "Análise da experiência de usuário no app Minha Agenda".
2. Você foi selecionado para ser voluntário e sua participação não é obrigatória.
3. A qualquer momento você pode desistir de participar e retirar seu consentimento.
4. Sua recusa não trará nenhum prejuízo em sua relação com o pesquisador, com a instituição.
5. Essa pesquisa tem por objetivo comparar interações em uma aplicação PWA que permite aos usuários atuarem como co-designers na aplicação, através do meta-design.
6. Sua participação nesta pesquisa consistirá em seguir as tarefas definidas que serão entregues a você, responder a um questionário sobre sua satisfação, motivação e sentimento de controle após cada tarefa. E participar de uma entrevista com a pesquisadora após a condução do teste.
7. A sua participação na pesquisa pode envolver algum desconforto relacionado ao tempo despendido com a realização da sessão e do preenchimento de questionários, sendo que faremos o possível para minimizar possíveis desconfortos.
8. As informações obtidas através dessa pesquisa serão confidenciais e asseguramos o sigilo sobre sua participação.
9. Os dados não serão divulgados de forma a possibilitar sua identificação.
10. Sua participação estará sendo gravada com o intuito de coletar dados sobre a interação com o *smartphone* e com os elementos de tela.
11. Você receberá uma cópia deste termo onde consta informações do pesquisador, podendo tirar suas dúvidas sobre o projeto e sua participação, agora ou a qualquer momento.

---

Giulia de Andrade Cardieri

Universidade Federal de São Carlos (UFSCar) -Departamento de Computação (DC)  
Rodovia João Leme dos Santos, (SP-264), Km 110, s/n - Itinga, Sorocaba - SP, 18052-780.  
Tel.: +55 16 3351-6000

**Declaro que entendi os objetivos, riscos e benefícios de minha participação na pesquisa e concordo em participar.**

**Sorocaba, / /2018**

---

Assinatura do Sujeito da pesquisa





# APPENDIX F – Profile Questionnaire 2

## Questionário de perfil do participante

Olá,  
Por favor, responda todas as perguntas antes de iniciar o teste com a aplicação.

\* Required

1. Qual seu número do RG? \*

\_\_\_\_\_

2. Qual seu sexo? \*

*Mark only one oval.*

- Feminino  
 Masculino  
 Other: \_\_\_\_\_

3. Qual sua idade? \*

\_\_\_\_\_

4. Qual seu nível de escolaridade? \*

*Mark only one oval.*

- Ensino fundamental incompleto  
 Ensino fundamental  
 Ensino médio incompleto  
 Ensino médio  
 Ensino superior incompleto  
 Ensino superior  
 Pós-graduação  
 Especialização

5. Qual sistema operacional você utiliza em seus dispositivos móveis com maior frequência? \*

\*

*Mark only one oval.*

- iOS  
 Android  
 Windows Phone  
 Other: \_\_\_\_\_

6. **Qual navegador você costuma utilizar em seus dispositivos móveis com maior frequência? \***

*Mark only one oval.*

- Navegador padrão iOS (Safari)
- Google Chrome
- Firefox
- Opera
- Internet Explorer
- Other: \_\_\_\_\_

7. **Qual tipo de rede para acesso a internet você utiliza com maior frequência? \***

*Mark only one oval.*

- 3G/4G
- Wifi

8. **Quantas vezes por mês você costuma baixar aplicativos? \***

*Mark only one oval.*

- Não baixo aplicativos todo mês
- Entre 1 e 2 vezes
- Entre 3 e 4 vezes
- Mais de 5 vezes

9. **Qual sua principal área de estudo/trabalho?**

*Mark only one oval.*

- Tecnologia/Computação
- Design gráfico/digital
- Outra

10. **Há quanto tempo você está nessa área de trabalho/estudo?**

*Mark only one oval.*

- Menos de 1 ano
- Entre 1-2 anos
- Entre 2-4 anos
- Mais de 5 anos

**11. Selecione a opção que mais representa sua opinião sobre diversos fatores ao baixar um aplicativo \***

*Mark only one oval per row.*

	Discordo totalmente	Discordo amplamente	Discordo parcialmente	Concordo parcialmente	Concordo amplamente	Concordo totalmente
Não vou baixar um aplicativo se já existe um site com conteúdo similar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Prefiro usar aplicativos em relação a sites no navegador	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Não vou baixar um aplicativo se estiver sem acesso a Wifi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Costumo ter pouco espaço de armazenamento no meu dispositivo móvel	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Não gosto da necessidade de pesquisar o aplicativo na Google Play/App Store	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O tempo de espera para baixar um aplicativo não me agrada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**12. Indique a sua frequência de acesso de sites em navegadores em seu dispositivo móvel**

*Mark only one oval.*

- Sempre (todos os dias)
- Frequentemente (em média, 3 vezes por semana)
- Com frequência razoável (em média, 1 vez por semana)
- Raramente (em média, 1 vez por mês)
- Não se aplica

**13. Indique a sua frequência de acesso de aplicativos instalados (apps) em seu dispositivo móvel**

*Mark only one oval.*

- Sempre (todos os dias)
- Frequentemente (em média, 3 vezes por semana)
- Com frequência razoável (em média, 1 vez por semana)
- Raramente (em média, 1 vez por mês)
- Não se aplica

**14. Informe quantas vezes você costuma acessar (através de dispositivos móveis) os tipos de sites listados \***

*Mark only one oval per row.*

	Sempre (todos os dias)	Frequentemente (em média, 3 vezes por semana)	Com frequência razoável (em média, 1 vez por semana)	Raramente (em média, 1 vez por mês)	Não se aplica
E-commerce (sites de compras)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Redes Sociais	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sites de busca (Google, Yahoo, Bing)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sites de vídeos (YouTube, Vimeo)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ferramentas em geral (calendário, e-mail, mapas)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Viagens (reserva de hotéis, passagens aéreas)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**15. Indique a sua frequência de uso da modalidade de voz/áudio no lugar de digitar um texto em seu dispositivo móvel**

*Mark only one oval.*

- Sempre (todos os dias)
- Frequentemente (em média, 3 vezes por semana)
- Com frequência razoável (em média, 1 vez por semana)
- Raramente (em média, 1 vez por mês)
- Não se aplica





# APPENDIX G – Profile Questionnaire 3

## Questionário de perfil desenvolvedores

Olá,  
Por favor, responda todas as perguntas :)  
Obrigada

\* Required

1. Qual seu número do RG? \*

---

2. Qual seu sexo? \*

*Mark only one oval.*

- Feminino  
 Masculino  
 Other: \_\_\_\_\_

3. Qual sua idade? \*

---

4. Qual seu nível de escolaridade? \*

*Mark only one oval.*

- Ensino fundamental incompleto  
 Ensino fundamental  
 Ensino médio incompleto  
 Ensino médio  
 Ensino superior incompleto  
 Ensino superior  
 Pós-graduação  
 Especialização

5. Com qual frequência você usa dispositivos móveis? \*

*Mark only one oval.*

- Nunca  
 1 vez por semana  
 2-4 vezes por semana  
 5 vezes por semana  
 Todo dia



6. Com qual frequência você usa internet (web e/ou apps) em dispositivos móveis? \*

Mark only one oval.

- Nunca
- 1 vez por semana
- 2-4 vezes por semana
- 5 vezes por semana
- Todo dia

7. Selecione seu nível de conhecimento em cada uma dessas tecnologias \*

Mark only one oval per row.

	Eu nunca ouvi falar	Conheço um pouco/Já ouvi falar	Tenho um bom conhecimento teórico, mas nunca pratiquei	Tenho um bom conhecimento teórico e prático	Tenho profundo conhecimento teórico e prático
HTML5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
HTML5 - localStorage	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
CSS	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
JavaScript	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
jQuery	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Bootstrap	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Handlebars.js (template dinâmico)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
JSON - JavaScript Object Notation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Single Page Applications - SPAs	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Web Design Responsivo - RWD	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8. Selecione seu nível de conhecimento em cada um desses editores de código/texto \*

Mark only one oval per row.

	Eu nunca ouvi falar	Já ouvi falar, mas nunca usei	Já usei, mas não uso mais	Uso as vezes	É o editor que eu costumo usar sempre
Sublime	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
VSCode	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Atom	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eclipse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Notepad++	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Brackets	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**9. Qual sua experiência com programação? \***

*Check all that apply.*

- Já programei em disciplinas/projetos na faculdade/universidade
- Trabalho com programação em uma empresa/como freelancer
- Já trabalhei com programação numa empresa/como freelancer mas agora trabalho em outra área
- Não tenho nenhuma experiência com programação

**10. Qual seu nível de conhecimento sobre Progressive Web Apps (PWAs)? \***

*Mark only one oval.*

- Eu nunca ouvi falar *Skip to question 14.*
- Conheço um pouco/já ouvi falar *Skip to question 11.*
- Tenho um bom conhecimento teórico, mas nunca pratiquei *Skip to question 11.*
- Tenho um bom conhecimento teórico e prático *Skip to question 11.*
- Tenho profundo conhecimento teórico e prático *Skip to question 11.*

## Informações sobre PWA

**11. Você já desenvolveu um PWA? \***

*Mark only one oval.*

- Sim *After the last question in this section, skip to question 13.*
- Não *After the last question in this section, skip to question 14.*

**12. Você já usou um PWA? \***

*Mark only one oval.*

- Sim
- Não

## PWA na Prática

**13. Qual framework você usou para o desenvolvimento do PWA? \***

*Check all that apply.*

- Vue.js
- React
- Angular
- Nenhum, usei vanilla/puro js
- Other: \_\_\_\_\_

## Desenvolvimento web e mobile

14. **Você já desenvolveu algum app nativo para dispositivos móveis? (sem usar tecnologias híbridas) \***

*Mark only one oval.*

- Sim, iOS
- Sim, Android
- Sim, Android e iOS
- Sim, outro sistema
- Não

15. **Você já desenvolveu algum app híbrido? \***

*Check all that apply.*

- Sim, com Cordova.js
- Sim, com Electron
- Sim, com ionic
- Sim, com outra ferramenta
- Não

16. **Você já desenvolveu algum web app para dispositivos móveis? \***

*Mark only one oval.*

- Sim
- Não





# APPENDIX H – Developer Tasks Guide

## Estudo de Caso 2

### Implementar PWA-EU no PWA "Parei Onde?"

#### Instruções gerais

- O arquivo base "pareionde" está no Drive
- Utilizar o navegador **Google Chrome**
- Editar o texto usando o **Brackets**
- Não se esqueça de escrever um código legível, se possível deixe comentários explicando o que cada função faz
- Não se esqueça de olhar os comentários no código
- Se houver a necessidade de adicionar novos estilos, você pode criar novos arquivos CSS ou adicionar conteúdo aos que já existem, mas lembre-se de usar os componentes/elementos do Bootstrap
- Os usuários devem acessar as novas funcionalidades na aba "Meu Design"
- Utilizar LocalStorage para armazenar os dados
- Handlebars.js é a ferramenta de templates a ser utilizada
- Utilizar jQuery: <https://api.jquery.com/>
- Utilizar Bootstrap: <https://getbootstrap.com>
- Não se esqueça que as duas tarefas estão relacionadas à abordagem PWA-EU
- A tarefa 2/desafio é opcional e não vale nota, você faz se quiser
- **Não se esqueça de responder o questionário após finalizar a tarefa (e se você quiser, o desafio)**

#### Tarefa 1 - Modificação da lista de plataformas

Nessa tarefa você deve criar uma funcionalidade para os usuários finais criarem sua própria lista de plataformas (respostas do campo *Como você assiste?* ex: *Netflix, Baixada, TV, Hulu, Prime, Now...*). O usuário deve ter a liberdade de adicionar e remover plataformas. As mudanças devem ser exibidas assim que o usuário abrir o formulário.

Ao finalizar essa tarefa, o usuário final pode;

- Adicionar e remover plataformas através da aba Meu Design

Lembre-se

- Você está desenvolvendo uma SPA, pense quando os templates devem ser recarregados
- Comente no código quais partes da arquitetura você está usando, assim como foi feito no aquecimento
- Pense como os dados relacionados a essas plataformas podem ser armazenado, e como eles se relacionam com PWA-EU

- Você pode usar qualquer elemento de interface do Bootstrap que achar apropriado, e pode adicionar novo CSS a eles se quiser

### ***Dicas para possíveis problemas***

- Ao tentar adicionar o evento click a um elemento dinâmico adicionado no template, é necessário se preocupar com a delegação em jQuery (é só usar o `$(elemento pai não dinâmico).on('click', 'nome do elemento dinâmico')`)
  - <https://tableless.com.br/entendendo-o-event-delegation-jquery/>
- Se o template do handlebars não compilou, tente ver se todos os seguintes itens estão corretos:
  - Each/If/Else tem elementos de begin/end (é / e não \)
  - Você não está chamando errado o nome do elemento do JSON
  - Os dados estão corretos e compilados para o formato JSON
  - Ver as funções JS de compilação
- Pensar se vale a pena criar um template para o formulário (se sim, como criá-lo usando as ferramentas disponíveis no Handlebars.js)
  - O modelo de design do usuário pode ter opções para permitir que esse formulário seja exibido corretamente (se o campo é visível, qual tipo é o campo, se é text, number, select, checkbox...)
- Para dúvidas técnicas, pode usar o Google ou perguntar para gente

### ***Terminei a tarefa, e agora?***

- Se você decidiu fazer o Desafio - Tarefa 2 que está no fim desse guia, responda o questionário e envie os arquivos junto com o Desafio
- Responder o questionário
  - <https://goo.gl/CvJ8h4>
- Enviar o .zip de **todos os arquivos** via Classroom no seguinte formato:
  - **NomeSobrenome\_RG.zip**
  - É necessário enviar nesse formato para ser corretamente avaliada/o

### ***Desafio Opcional - Tarefa 2 - Alteração dos campos do formulário***

Essa tarefa é um desafio. Ela **não vale nota** e você pode fazer **se quiser**. Se você completar a tarefa de forma correta, pode ganhar até 2 pontos extras na nota do EC2.

Nessa tarefa você deve criar uma funcionalidade que permite que usuários do seu app possam escolher quais campos do formulário eles desejam ver. Não é necessário se preocupar com os dados exibidos ao usuário na página inicial e nos favoritos, **a funcionalidade edição é relacionada somente ao formulário de adicionar séries**.

Ao finalizar essa tarefa, o usuário final pode:

- Escolher quais campos são visíveis no formulário (**não deve ser possível editar os campos Nome e Favorito**) através da aba Meu Design

Lembre-se:

- Os campos Nome e Favorito não devem ser editáveis
- Você está desenvolvendo uma SPA, pense quando os templates devem ser recarregados
- No formulário alguns campos tem uma descrição, seria bom se essas descrições acompanhassem o campo também nessa funcionalidade
- Você pode usar qualquer elemento de interface do Bootstrap que achar apropriado, e pode adicionar novo CSS a eles se quiser
- Pensar qual a relação dos dados obtidos com essa funcionalidade com a arquitetura PWA-EU, e como eles podem ser armazenados

Você tem até o final da aula.

Bom Trabalho! :)







3. **Você não entendeu alguma parte da arquitetura? Qual?**

---

---

---

---

---

4. **Você tem alguma observação sobre a arquitetura PWA-EU, seu uso e/ou implementação? \***

---

---

---

---

---