



## Exploring Mobile Analytics

**BEATRIZ SAMPAIO BARBOSA**

Outubro de 2021

# **Exploring *Mobile Analytics***

**Beatriz Sampaio**

**Dissertação para obtenção do Grau de Mestre em  
Engenharia Informática, Área de Especialização em  
Engenharia de Software**

**Orientador: Dr. Paulo Baltarejo Sousa**

Porto, Outubro 2021



# Resumo

Neste documento é descrito o trabalho realizado no âmbito da disciplina TMDEI do Mestrado em Engenharia de Software (MEI) do Departamento de Engenharia Informática (DEI) do Instituto Superior de Engenharia do Porto (ISEP) do Politécnico do Porto (P.Porto).

O trabalho apresentado nesta dissertação foi realizado numa empresa, que por motivos de segurança e privacidade, irá ter o nome fictício 'X' no decorrer do documento. Este trabalho tem como intuito perceber a importância de ferramentas de *Analytics* na monitorização das aplicações móveis.

Atualmente, as aplicações móveis apresentam a maneira mais eficiente, direta e personalizável de fornecer informações sobre um produto e incentivar os clientes a permanecerem fiéis a uma determinada marca.

A experiência do utilizador está a tornar-se um recurso cada vez mais crucial no sucesso da aplicação, e desta forma, o uso de uma ferramenta de *Analytics* desempenha um papel importante na obtenção de informações sobre como o utilizador está a interagir com a aplicação, como a aplicação está a funcionar e o que se pode corrigir e melhorar. Ao longo desta dissertação, desenvolveu-se uma *app* onde se implementou uma ferramenta de *Analytics*, neste caso, o Countly. No final, foram obtidos resultados acerca da monitorização efetuada, e feita uma comparação em relação à *app* anteriormente existente, onde se pôde concluir que este desenvolvimento ajudou a melhorar o desempenho da *app* e a satisfação dos seus utilizadores.

**Palavras-chave:** *Analytics*, *Mobile*, aplicação móvel, monitorização



# Abstract

This document describes the work carried out in the scope of the subject TMDEI of the Masters in Software Engineering (MEI) of the Informatics Engineering Department (DEI) of the Porto School of Engineering (ISEP) of the Porto Polytechnic (P.Porto).

The work presented in this dissertation was developed in a company, which for security and privacy reasons, will have the fictitious name 'X' in the course of the document. This work aims to understand the importance of Analytics in monitoring mobile applications. Today, mobile apps represent the most efficient, direct and customizable way to provide information about a product and encourage customers to stay true to a particular brand.

The user experience is becoming an increasingly crucial resource in the success of the application, therefore, the use of an Analytics tool plays an important role in obtaining information about how the user is interacting with the application, how the application is working, and what can be fixed and improved.

Throughout this dissertation, an app was developed where an Analytics tool was implemented, in this case, County. In the end, results were obtained about the monitoring performed, and made a comparison with the previously existing app, where it was found that this development helped improve the performance of the app and the satisfaction of its users.

**Keywords:** Analytics, Mobile, mobile application, monitoring



# Índice

|  |               |
|--|---------------|
| <i>Lista de Figuras</i> .....                          | <i>xí</i>     |
| <i>Lista de Excerto de Código</i> .....                | <i>xííí</i>   |
| <i>Lista de Tabelas</i> .....                          | <i>xíííí</i>  |
| <i>Acrónimos e Símbolos</i> .....                      | <i>xííííí</i> |
| <b>1 Introdução</b> .....                              | <b>1</b>      |
| <b>1.1 Contexto</b> .....                              | <b>1</b>      |
| <b>1.2 Problema</b> .....                              | <b>2</b>      |
| <b>1.3 Objetivo</b> .....                              | <b>3</b>      |
| <b>1.4 Metodologia de Trabalho e Planeamento</b> ..... | <b>3</b>      |
| <b>1.5 Estrutura da Dissertação</b> .....              | <b>4</b>      |
| <b>2 Estado da Arte</b> .....                          | <b>7</b>      |
| <b>2.1 Contexto</b> .....                              | <b>7</b>      |
| <b>2.2 Métricas/Funcionalidades</b> .....              | <b>8</b>      |
| 2.2.1 Informação dos utilizadores .....                | 9             |
| 2.2.2 Eventos .....                                    | 9             |
| 2.2.3 Relatórios de falhas .....                       | 10            |
| 2.2.4 Monitorização de desempenho .....                | 10            |
| 2.2.5 Push Notifications.....                          | 10            |
| 2.2.6 Funnels .....                                    | 11            |
| 2.2.7 Cohorts .....                                    | 11            |
| 2.2.8 Testes A/B.....                                  | 11            |
| 2.2.9 Segmentos .....                                  | 12            |
| <b>2.3 Soluções Existentes</b> .....                   | <b>13</b>     |
| 2.3.1 Google Analytics – Firebase.....                 | 13            |
| 2.3.2 Mixpanel .....                                   | 15            |
| 2.3.3 Flurry.....                                      | 16            |
| 2.3.4 Countly.....                                     | 17            |
| 2.3.5 UXCam .....                                      | 18            |
| 2.3.6 Apple iOS Analytics .....                        | 20            |
| 2.3.7 Adjust.....                                      | 22            |
| 2.3.8 App Annie .....                                  | 23            |
| 2.3.9 Síntese .....                                    | 24            |
| <b>2.4 Tecnologia</b> .....                            | <b>25</b>     |
| 2.4.1 Flutter .....                                    | 25            |
| 2.4.2 Gestão de Estados .....                          | 27            |
| <b>2.5 Sumário</b> .....                               | <b>29</b>     |
| <b>3 Análise de Valor</b> .....                        | <b>31</b>     |

|            |  |                  |
|------------|--|------------------|
| <b>3.1</b> | <b>Processo de Inovação .....</b>                      | <b>31</b>        |
| 3.1.1      | Modelo de desenvolvimento de novo conceito (NCD) ..... | 33               |
| <b>3.2</b> | <b>Valor .....</b>                                     | <b>36</b>        |
| <b>3.3</b> | <b>Valor percecionado .....</b>                        | <b>37</b>        |
| <b>3.4</b> | <b>Valor para o cliente .....</b>                      | <b>37</b>        |
| <b>3.5</b> | <b>Proposta de Valor .....</b>                         | <b>38</b>        |
| <b>3.6</b> | <b>Modelo de negócio Canvas .....</b>                  | <b>38</b>        |
| <b>3.7</b> | <b>Sumário .....</b>                                   | <b>41</b>        |
| <b>4</b>   | <b><i>Análise e Design da Solução .....</i></b>        | <b><i>43</i></b> |
| <b>4.1</b> | <b>Análise .....</b>                                   | <b>43</b>        |
| 4.1.1      | Requisitos Funcionais .....                            | 44               |
| 4.1.2      | Requisitos Não Funcionais .....                        | 45               |
| <b>4.2</b> | <b>Desenho .....</b>                                   | <b>46</b>        |
| 4.2.1      | Condicionantes do Desenho .....                        | 46               |
| 4.2.2      | Arquitetura lógica do sistema .....                    | 47               |
| 4.2.3      | Mobile App .....                                       | 48               |
| 4.2.4      | Interface Gráfica .....                                | 49               |
| 4.2.5      | Desenho das Funcionalidades .....                      | 53               |
| <b>4.3</b> | <b>Sumário .....</b>                                   | <b>57</b>        |
| <b>5</b>   | <b><i>Implementação .....</i></b>                      | <b><i>59</i></b> |
| <b>5.1</b> | <b><i>Guidelines de Implementação .....</i></b>        | <b><i>59</i></b> |
| 5.1.1      | Customização de Eventos .....                          | 60               |
| <b>5.2</b> | <b><i>Implementação de Analytics .....</i></b>         | <b><i>62</i></b> |
| <b>5.3</b> | <b>Sumário .....</b>                                   | <b>68</b>        |
| <b>6</b>   | <b><i>Avaliação e Resultados .....</i></b>             | <b><i>69</i></b> |
| <b>6.1</b> | <b>Metodologia de Avaliação .....</b>                  | <b>69</b>        |
| 6.1.1      | Satisfação do Utilizador .....                         | 71               |
| 6.1.2      | Deteção de <i>bugs</i> .....                           | 73               |
| <b>6.2</b> | <b>Avaliação dos resultados .....</b>                  | <b>73</b>        |
| 6.2.1      | Resultados da Implementação .....                      | 74               |
| 6.2.2      | Satisfação do Utilizador .....                         | 82               |
| <b>6.3</b> | <b>Sumário .....</b>                                   | <b>94</b>        |
| <b>7</b>   | <b><i>Conclusões .....</i></b>                         | <b><i>95</i></b> |
| <b>7.1</b> | <b>Síntese .....</b>                                   | <b>95</b>        |
| <b>7.2</b> | <b>Objetivos realizados .....</b>                      | <b>96</b>        |
| <b>7.3</b> | <b>Limitações e Trabalho futuro .....</b>              | <b>97</b>        |
| <b>7.4</b> | <b>Apreciação final .....</b>                          | <b>97</b>        |
|            | <b>Referências .....</b>                               | <b>99</b>        |

**Anexos .....103**



# Lista de Figuras

|   |    |
|---|----|
| Figura 1 - Dashboard Firebase .....                       | 14 |
| Figura 2 - Dashboard Flurry .....                         | 16 |
| Figura 3 - Dashboard Countly .....                        | 18 |
| Figura 4 - Dashboard Apple Analytics .....                | 21 |
| Figura 5 - Dashboard Adjust .....                         | 22 |
| Figura 6 - Dashboard App Annie .....                      | 23 |
| Figura 7 - Flutter System Overview [21] .....             | 27 |
| Figura 8 - Arquitetura Gestão de Estados BLOC .....       | 28 |
| Figura 9 - Processo de inovação [21] .....                | 32 |
| Figura 10 - Modelo NCD [21] .....                         | 33 |
| Figura 11 - Modelo Canvas .....                           | 40 |
| Figura 12 - Diagrama de Casos de Uso .....                | 45 |
| Figura 13 – Divisão ficheiros .....                       | 47 |
| Figura 14 – <i>Flow</i> do BLOC .....                     | 47 |
| Figura 15 - Diagrama de Componentes .....                 | 48 |
| Figura 16 - Diagrama de Componentes Mobile App .....      | 49 |
| Figura 17 - Página principal (Parte 1) .....              | 51 |
| Figura 18 - Página principal (Parte 2) .....              | 51 |
| Figura 19 – Página Carteira do utilizador .....           | 51 |
| Figura 20 - Página de Atividade do Utilizador .....       | 53 |
| Figura 21 - Página de detalhe de uma atividade .....      | 53 |
| Figura 22 - Diagrama de Sequência UC1 .....               | 54 |
| Figura 23 - Diagrama de Sequência UC5 .....               | 55 |
| Figura 24 - Diagrama de Sequência UC6 .....               | 56 |
| Figura 25 - Evento de <i>Tap</i> no Countly .....         | 68 |
| Figura 26 - Países da <i>app</i> desenvolvida .....       | 74 |
| Figura 27 - Utilizadores da <i>app</i> .....              | 76 |
| Figura 28 - Países com maior número de utilizadores ..... | 77 |
| Figura 29 - Utilizadores ativos .....                     | 77 |
| Figura 30 - Frequência de sessão dos utilizadores .....   | 78 |
| Figura 31 - Sistemas operativos utilizados .....          | 79 |
| Figura 32 - Dispositivos mais utilizados .....            | 79 |
| Figura 33 - Pontuação <i>app</i> antiga: 1.9 .....        | 80 |
| Figura 34 - Pontuação <i>app</i> desenvolvida .....       | 81 |
| Figura 35 - Matriz Respostas Questão 1 .....              | 83 |
| Figura 36 - Matriz Respostas Questão 2 .....              | 83 |
| Figura 37 - Matriz Respostas Questão 3 .....              | 84 |
| Figura 38 - Matriz Respostas Questão 4 .....              | 85 |
| Figura 39 - Matriz Respostas Questão 5 .....              | 86 |
| Figura 40 - Matriz Respostas Questão 6 .....              | 86 |

|  |    |
|--|----|
| Figura 41 - Matriz Respostas Questão 7 ..... | 87 |
| Figura 42 - Matriz Respostas Questão 1 ..... | 88 |
| Figura 43 - Matriz Respostas Questão 2 ..... | 90 |
| Figura 44 - Matriz Respostas Questão 3 ..... | 92 |
| Figura 45 - Matriz Respostas Questão 4 ..... | 93 |

# Lista de Excerto de Código

|  |    |
|--|----|
| Excerto de Código 1 - Código para adicionar SDK Countly .....      | 62 |
| Excerto de Código 2 - Enumerável PointsProgramAnalyticsEvent ..... | 63 |
| Excerto de Código 3 - Classe PointsProgramAnalytics.....           | 64 |
| Excerto de Código 4 - Evento de Swipe .....                        | 65 |
| Excerto de Código 5 - Eventos com pedido ao repositório .....      | 67 |
| Excerto de Código 6 - Teste Shapiro-Wilk Questão 1 .....           | 89 |
| Excerto de Código 7 - Teste Wilcoxon Questão 1.....                | 90 |
| Excerto de Código 8 - Teste Shapiro-Wilk Questão 2 .....           | 91 |
| Excerto de Código 9 - Teste Wilcoxon Questão 2.....                | 91 |
| Excerto de Código 10 - Teste Shapiro-Wilk Questão 3 .....          | 92 |
| Excerto de Código 11 - Teste Wilcoxon Questão 3.....               | 93 |

# Lista de Tabelas

|   |    |
|---|----|
| Tabela 1 - Prós e Contras Firebase .....                  | 14 |
| Tabela 2 - Prós e Contras Mixpanel .....                  | 15 |
| Tabela 3 - Prós e Contras Flurry .....                    | 17 |
| Tabela 4 - Prós e Contras Countly .....                   | 18 |
| Tabela 5 - Prós e Contras UXCam.....                      | 20 |
| Tabela 6 - Prós e Contras Apple Analytics .....           | 21 |
| Tabela 7 - Comparação entre soluções .....                | 24 |
| Tabela 8 - Comparação entre BLOC e MobX.....              | 29 |
| Tabela 9 - Benefícios e Sacríficos .....                  | 37 |
| Tabela 10 - Definição Modelo de Negócios .....            | 39 |
| Tabela 11 - Segmentos <i>Standart</i> da <i>App</i> ..... | 60 |
| Tabela 12 - Segmentos <i>Standart</i> do Utilizador ..... | 61 |
| Tabela 13 - Exemplo de Evento.....                        | 62 |
| Tabela 14 - Países/Regiões da <i>app</i> antiga .....     | 75 |
| Tabela 15 - Tipo de dispositivos .....                    | 78 |
| Tabela 16 - Realização de Objetivos .....                 | 96 |



# Acrónimos e Símbolos

|             |  |
|-------------|--|
| <b>API</b>  | <i>Application Programming Interface</i> |
| <b>AV</b>   | Análise de valor                         |
| <b>BFF</b>  | BackEnd for FrontEnd                     |
| <b>BLOC</b> | <i>Business Logic Component</i>          |
| <b>CRM</b>  | <i>Customer Relationship Management</i>  |
| <b>DAU</b>  | <i>Daily Active Users</i>                |
| <b>FFE</b>  | <i>Fuzzy Front End</i>                   |
| <b>HTML</b> | <i>HyperText Markup Language</i>         |
| <b>KPI</b>  | <i>Key Performance Indicator</i>         |
| <b>LTV</b>  | <i>Lifetime value</i>                    |
| <b>NCD</b>  | <i>New Concept Development</i>           |
| <b>NPD</b>  | <i>New Product Development</i>           |
| <b>PC</b>   | <i>Personal Computer</i>                 |
| <b>SDK</b>  | <i>Software development kit</i>          |
| <b>UC</b>   | <i>Use Case</i>                          |
| <b>UI</b>   | <i>User Interface</i>                    |
| <b>UML</b>  | <i>Unified Modeling Language</i>         |





# 1 Introdução

Neste primeiro capítulo introdutório será apresentado o contexto da dissertação, do problema e o objetivo que se pretende atingir, a metodologia de trabalho utilizada e, no final deste capítulo, será explicada a estrutura do documento.

## 1.1 Contexto

As aplicações móveis, também designadas de *apps*, surgiram como resposta ao mercado emergente dos *smartphones* e ganharam popularidade no ano de 2008 através do lançamento da App Store<sup>1</sup>, da Apple.

A vulgarização e proliferação das *apps* alteraram o modelo convencional de comunicação social e despoletaram o desenvolvimento da tecnologia Android App, iOS App, Web App e PC App, um acontecimento que assinalou a Revolução Tecnológica [1].

O desenvolvimento de *apps* envolve processos complexos. Estes são reflexo da experiência do criador e proporcional à estrutura e configurações do software a ser produzido, ao número de dispositivos distintos em que estas vão operar, às especificações do hardware e às plataformas que as vão disponibilizar.

São inúmeras as etapas a considerar que devem anteceder a criação de uma *app*. Deve ter-se em conta o propósito da *app* a desenvolver, para que plataformas de distribuição as disponibilizar, em que tipo de dispositivos móveis vai ser possível operá-las. Assim como perceber se se vai criar uma *app* gratuita ou paga, como melhor rentabilizá-la e qual o nível de risco associado.

---

<sup>1</sup> Serviço de distribuição digital de aplicações móveis para dispositivos com a plataforma iOS

## 1.2 Problema

O uso de ferramentas *mobile analytics* desempenha um papel vital no sucesso de qualquer *app*. A comercialização da *app* não pára com o mero lançamento da *app* na loja (*App Store* ou *Google Play Store*<sup>2</sup>). Semelhante às aplicações *web*, as *apps* envolvem a monitorização contínua do desempenho da *app*.

Toda a experiência do utilizador é monitorizada, desde o *download* até à desinstalação. Através desses dados, geram-se relatórios sobre o desempenho da *app*.

Além disso fornece informações sobre como os utilizadores estão a interagir com a *app*, com quais partes da *app* eles interagem mais e que ações executam na *app* [2].

Os utilizadores da *app* podem levar ao sucesso, insucesso ou mesmo ao desaparecimento das *apps* com base no seu *feedback*, que é baseado nas suas perceções. Eles podem partilhar as suas experiências amplamente e grande parte da sua partilha é pública e, portanto, pode alcançar grandes públicos [3].

A falta de monitorização de *apps* é uma das mais importantes razões pelas quais estas falham e são eliminadas do mercado [4]. Antes de lançar a *app* é importante saber que elementos, critérios ou métricas se deverão avaliar e como avaliar. Então, quando a *app* já está no mercado, pode-se verificar o sucesso (ou não) destes elementos, permitindo sempre alterá-los ou melhorá-los.

Erros ou problemas técnicos são a razão número um pela qual as *apps* são desinstaladas pelos utilizadores, e 42% das *reviews* com zero estrelas na *Google Play Store* mencionam a instabilidade da *app* e erros da mesma. Por exemplo, a Knight Capital Group perdeu \$440 milhões de dólares (aproximadamente 362€ milhões de euros) num mero espaço de tempo de 30 minutos devido a um erro de *software* [5].

Com uma ferramenta de monitorização de erros, como as *mobile analytics*, podem-se descobrir defeitos rapidamente, poupando tempo aos *developers*, evitando erros críticos e permitindo entregar *apps* sem erros.

---

<sup>2</sup> Serviço de distribuição digital de aplicações móveis para Android

### 1.3 Objetivo

Pretende-se com o estudo da importância das ferramentas de *analytics*, verificar a eficácia das métricas existentes no ciclo de vida de uma *app*.

Para tal serão estudadas e avaliadas as métricas mais eficazes (por exemplo: o número de *active users*; o tempo de sessão; retenção) e será desenvolvido um protótipo para permitir o estudo dessas métricas e sua eficácia.

Por fim pretende-se sistematizar a forma mais eficaz de aplicar essas métricas no ciclo de vida de uma *app* e responder à pergunta: **Serão as *mobile analytics* a solução para o desempenho e sucesso de uma *app*?**

Sumariando, os objetivos deste trabalho são:

- Estudar e avaliar as métricas mais eficazes para monitorizar a *app* e melhorar o desempenho da mesma;
- Desenvolver uma *app* com a implementação de *mobile analytics*;
- Avaliar e comparar resultados com uma antiga solução pré-existente.

### 1.4 Metodologia de Trabalho e Planeamento

Durante o desenvolvimento deste projeto, foi adotada a metodologia Scrum e o desenvolvimento do projeto foi dividido em *sprints*, com a duração de duas semanas cada. A equipa é constituída pela autora desta dissertação, como *mobile developer*; mais dois *mobile developers*; dois *back-end developers*; uma *designer*; um *scrum master* e um *product owner*.

No início de cada *sprint* são feitas sessões de Sprint Planning, onde são definidas quais as tarefas a realizar no novo *sprint*, ordenadas de acordo com a sua prioridade, tendo em atenção as suas dependências, a complexidade e o esforço de implementação de cada uma, elaborando, assim, o Sprint Backlog.

São também realizadas diariamente sessões de Daily Meeting com a equipa que têm como objetivo pôr a equipa a par das tarefas realizadas no dia anterior, bem como quais as que iriam ser feitas no presente dia e potenciais impedimentos.

Ao longo da *sprint*, também são feitas sessões (uma ou duas, dependendo da necessidade) de Product Backlog Refinement, que têm como intuito rever o Product Backlog, e estimar as tarefas existentes.

No último dia da *sprint*, normalmente terça-feira, é realizada uma sessão de Sprint Review, onde é feita uma demonstração da *app* aos *stakeholders* e também uma sessão de Sprint Retrospective, onde é feita uma reflexão sobre o *sprint* decorrido, identificadas as principais falhas a serem corrigidas no próximo *sprint*, e, ainda, reconhecidas quais as tarefas que foram concluídas com sucesso.

## 1.5 Estrutura da Dissertação

Este documento divide-se em sete capítulos: Estado da Arte, Análise de Valor, Análise e Design, Implementação, Avaliação e por fim, Conclusões.

De acordo com a unidade curricular, no qual se insere esta dissertação, são esperados um conjunto de resultados definidos como *outcomes* e, ao longo dos capítulos, será dada a resposta a cada um dos *outcomes*, sendo estes:

- Outcome 1: Interpretar o Problema;
- Outcome 2: Contexto e Estado da Arte;
- Outcome 3: Avaliar Soluções e Abordagens Existentes;
- Outcome 4: Design da Solução;
- Outcome 5: Construção da Solução;
- Outcome 6: Avaliação da Solução;

No presente capítulo, Introdução, foi abordado o **Outcome 1: Interpretar o Problema**. No capítulo dois, intitulado de Estado da Arte, serão analisados e respondidos parte do **Outcome 2: Contexto e Estado da Arte** e o **Outcome 3: Avaliar Soluções e Abordagens Existentes**. No final será efetuada uma avaliação às soluções apresentadas.

De seguida, no terceiro capítulo, Análise de valor, será respondido parte do **Outcome 2: Contexto e Estado da Arte** e será realizada a análise de valor do projeto descrito neste relatório tendo em conta o processo respetivo.

No capítulo dedicado à Análise e Design será abordado o **Outcome 4: Design da Solução** e é apresentada a abordagem a seguir no desenvolvimento deste projeto bem como são especificados os respetivos requisitos funcionais e não funcionais. Neste capítulo, são também apresentados os diagramas que servem como base para a implementação do projeto.

No quinto capítulo, Implementação, será respondido o **Outcome 5: Construção da Solução**, onde será apresentado o que foi feito para o desenvolvimento da solução.

No capítulo da Avaliação, são apresentados os fatores que serão utilizados para avaliar a solução, incluindo metodologias de avaliação e testes estatísticos para avaliar cada fator. O objetivo deste capítulo é avaliar os resultados da solução e irá responder ao **Outcome 6: Avaliação da Solução**.

Por fim, no sétimo capítulo, Conclusões, irão ser apresentadas as conclusões finais de todo o trabalho desenvolvido ao longo deste projeto, os resultados obtidos, e por fim, o trabalho futuro.



## 2 Estado da Arte

Neste capítulo são apresentados os conceitos, assim como o estado da arte em tecnologia relevante.

No final, será apresentada uma comparação entre as soluções abordadas, e estas serão avaliadas tendo em conta alguns fatores tais como: funcionalidades, plataformas compatíveis, custo e prós e contras da sua utilização.

Neste capítulo será abordado o **Outcome 2: Contexto e Estado da Arte** e o **Outcome 3: Avaliar Soluções e Abordagens Existentes**.

Após a análise às soluções existentes atualmente será dada a resposta às seguintes questões:

**Quais as melhores soluções para monitorizar *apps*?**

**Quais são as métricas com mais relevância a serem monitorizadas?**

### 2.1 Contexto

Hoje em dia, as lojas de aplicações estão repletas de *apps* que desejam deixar uma marca. Embora esta seja uma grande vantagem para os utilizadores atuais, ainda assim, o enorme volume leva a uma imensa concorrência, o que se torna num desafio para as empresas que estão interessadas em criar *apps* para impulsionar os seus negócios.

Surpreendentemente, a realidade é que cerca de 80-90% das *apps* lançadas nas lojas são abandonadas logo após o primeiro uso. Tanto que a pesquisa diz que uma *app* tende a perder em torno de 77% dos DAUs (*Daily Active Users*) apenas três dias após a sua instalação. Esta é a razão pela qual muitas vezes as empresas questionam-se sobre o que realmente é preciso para fazer de uma *app* um sucesso [6].

Uma das razões para o insucesso destas, é ignorarem o *feedback* dos utilizadores.

Construir uma *app* é apenas metade da batalha. Uma vez feita, é hora de passar a otimizar a *app* e monitorizar o comportamento dos utilizadores desta. Com mais de

2,7 bilhões de utilizadores de *smartphones* e mais de 4 milhões de *apps* em todo o mundo, a concorrência está a ficar renhida. Para ter sucesso, as empresas têm que se manter atentas e oferecer uma boa experiência na *app* aos seus clientes. Mas como se pode saber se estão no caminho certo? Entrando no mundo das ferramentas de análise de *apps* [7].

Os utilizadores não devem ser vistos apenas como um meio de venda da sua ideia, mas também devem ser considerados importantes o suficiente para ouvir os seus problemas. O foco principal deve ser resolver os problemas que eles estão a enfrentar relacionados com a *app*. Se for dado um passo quase imediato em relação aos seus problemas, eles vão-se sentir importantes e isso fará aumentar o sucesso do produto [6].

A análise de *apps* é a melhor amiga de um *developer*. Ferramentas que analisam o desempenho da *app*, ajudam a entender bem como os seus utilizadores se estão a comportar e como a *app* pode ser otimizada para alcançar os seus objetivos.

De acordo com um estudo da *Harvard Business Review*, investigadores descobriram que "empresas que integram múltiplas fontes de dados de clientes e marketing superam significativamente outras empresas (...)", assim como retornos totais de *shareholders* drasticamente maiores [7].

Encontrar uma ferramenta de análise de *apps* adequada, no entanto, pode ser uma tarefa desafiadora e às vezes assustadora.

## **2.2 Métricas/Funcionalidades**

Nesta secção serão abordadas algumas métricas de monitorização de *apps*, assim como algumas funcionalidades, de forma a sustentar a seguinte secção de soluções existentes.

### 2.2.1 Informação dos utilizadores

Os utilizadores são visitantes que iniciaram uma sessão na *app* dentro de um período de tempo específico. Existem informações variadas acerca de quem está a utilizar a *app* como por exemplo [8]:

- *Active users* - Número de utilizadores ativos diários ou mensais da *app*;
- *New users* – Número de novos utilizadores da *app*;
- Localização – País ou local em que se encontra o utilizador;
- Dispositivo – Informações do tipo de dispositivo (*smartphone* ou *tablet*), sistema operacional (Android ou iOS), assim como versão do *software* instalado. Estas informações podem ajudar a ajustar o desempenho da *app*, e até melhorar a UI da *app* com o conhecimento do tamanho de ecrã mais utilizado;
- Tempo de sessão – O número de sessões, o seu comprimento e os intervalos de tempo que o utilizador passa na *app*.

### 2.2.2 Eventos

Os eventos são interações do utilizador com qualquer conteúdo, que podem ser medidas independentemente do carregamento de um ecrã. Transferências, cliques em *links*, envios de formulários, reproduções de vídeo, basicamente qualquer evento pode ser monitorizado e analisado [9].

#### **Heatmap**

Os *heatmaps*, ou mapa de calor, visualizam os elementos mais populares (quentes) e impopulares (frios) da *app* usando cores numa escala de vermelho para azul.

Ao agregar o comportamento do utilizador, os *heatmaps* facilitam a análise de dados e dão uma compreensão rápida de como as pessoas interagem com uma página individual da *app* — onde clicam, onde fazem *scroll* ou o que ignoram — o que ajuda a identificar tendências e otimizar para maior envolvimento.

### 2.2.3 Relatórios de falhas

Os relatórios de falhas permitem que se avalie o número e o tipo de falhas e exceções capturadas e não capturadas que ocorrem na *app*, com uma breve descrição para facilitar a correção [8].

É possível analisar quantos utilizadores são afetados pelas mesmas falhas na *app* e desta forma corrigir rapidamente o problema, de forma a não afetar mais público-alvo.

### 2.2.4 Monitorização de desempenho

Existem várias métricas a considerar na monitorização de desempenho da *app*, por exemplo:

- *Aquisição* – Número de *downloads* da *app*;
- *Revenue* – Valor da receita da *app*, seja receita anual ou média de receita por utilizador ganha;
- *Store views* – Quantidade de vezes que a *app* foi vista na *store*;
- *Análise de retenção* - Informações sobre quantos dispositivos ainda mantém a *app* instalada e o número de desinstalações. Fornece informação acerca de se os utilizadores ainda estão ativos, ou se a *app* é removida após o uso [8][10].

### 2.2.5 Push Notifications

*Push notification*, ou notificação *push*, é uma mensagem de alerta enviada automaticamente para o utilizador.

Ao contrário de outros serviços, estas mensagens são enviadas sem a necessidade de solicitação do utilizador, desde que ele as permita nas configurações [11].

Esta métrica permite analisar a notificação que teve mais taxa de leitura, a notificação que teve mais cliques e quais as notificações mais populares para os utilizadores. Com uma análise mais profunda, é também possível ver quais foram

os utilizadores específicos que clicaram ou não, numa certa notificação através da implementação de *mobile analytics* [12].

### **2.2.6 Funnels**

Um *funnel* é um caminho de navegação (conjunto de ecrãs da *app*) que se espera que os utilizadores sigam, para alcançar as metas da *app*. A análise do *funnel* permite visualizar os passos que os utilizadores tomam para concluir uma tarefa e ver se eles estão a ter sucesso ou a falhar em alguma etapa [9]. Por exemplo, se um utilizador estiver a efetuar uma compra pela *app*, se irá até ao fim do processo ou irá desistir na fase do pagamento.

### **2.2.7 Cohorts**

Um *cohort*, ou coorte, é um grupo de utilizadores que compartilham uma característica comum. Por exemplo, todos os utilizadores com a mesma data de aquisição pertencem à mesma coorte. O relatório de análise da coorte permite isolar e analisar o comportamento da coorte [9].

Uma análise de coorte, então, permite comparar o comportamento e as métricas de diferentes coortes ao longo do tempo. Pode-se encontrar as coortes de melhor desempenho (ou de menor desempenho) e quais os fatores que estão a impulsionar esse desempenho [13].

### **2.2.8 Testes A/B**

O teste A/B possibilita entender quais versões de uma ação de *marketing* podem ter melhores resultados. Assim, é possível trabalhar com hipóteses e, por meio das experiências, provar quais funcionam melhor em páginas, gerando mais conversões, cliques e outras ações desejadas.

Por vezes, uma simples mudança na disposição de um item numa página provoca mais cliques, por exemplo.

Desta forma, são criados dois modelos (A e B) de apresentação dessas práticas, para, então, avaliar qual deles tem um melhor desempenho.

Estes testes podem ser usados para experimentar diversos detalhes, como duas apresentações diferentes de *landing pages*, botões com cores ou mensagens diversas, títulos de uma página, posicionamento de *banners* e mais uma série de outras possibilidades.

No ambiente de *analytics*, os resultados são mostrados para gerar um comparativo preciso e fácil.

A ferramenta automatiza a criação destas duas versões do que precisa de ser testado e gera um processo de monitoramento delas durante o período determinado.

No fim, são mostradas as métricas separadas, gerando essa possibilidade de comparação [14].

### **2.2.9 Segmentos**

Um segmento é um subconjunto dos seus dados de *analytics*. Por exemplo, do conjunto total de utilizadores, um segmento pode representar utilizadores de um determinado país ou de uma determinada cidade. Outro segmento poderá representar utilizadores que compram uma linha específica de produtos ou que visitam uma parte específica da *app*.

Os segmentos permitem isolar e analisar esses subconjuntos de dados para que se possa examinar e responder às tendências dos componentes no negócio.

Pode-se utilizar segmentos como base para públicos-alvo. Por exemplo, pode-se criar um segmento de utilizadores que visitam as páginas de moda masculina e, em seguida, segmentar apenas esses utilizadores (o público-alvo) com uma campanha de *remarketing* centrada nos novos artigos que se está a adicionar a essas páginas [9].

## 2.3 Soluções Existentes

Nesta secção serão abordadas oito ferramentas existentes de *mobile analytics*. Este estudo tem como objetivo principal identificar as funcionalidades e vantagens de cada uma destas plataformas. As soluções selecionadas para este estudo foram escolhidas de acordo com uma pesquisa efetuada sobre as soluções mais populares no mundo das *apps*.

Dado que a solução já estava escolhida desde início por parte da empresa X, sendo esta, o Countly, no final desta secção irá ser feita uma breve comparação entre as soluções existentes e a escolhida.

### 2.3.1 Google Analytics – Firebase

O Firebase é uma plataforma criada pela Google para desenvolver aplicações móveis e *web*. Foi originalmente uma empresa independente fundada em 2011. Em 2014, a Google adquiriu a plataforma e agora é a sua principal oferta para o desenvolvimento de *apps* [15].

O Firebase não se concentra apenas em analisar a *app* em si, mas também fornece a infraestrutura para construí-la.

Uma grande vantagem desta ferramenta é que vários recursos como análise, relatórios de falhas, hospedagem e testes A/B vêm todos num mesmo produto.

O preço é baixo para *apps* com baixo tráfego, mas pode ficar caro para *apps* estabelecidas. O Firebase é uma ferramenta que se encontra presente no *Software development kit* (SDK) de muitas *apps* populares [16].

Na Figura 1 [17] é apresentado o *dashboard* do Firebase onde se é possível visualizar dados como ‘*Active users*’, ‘*Daily user engagement*’, ‘*Top conversation events*’ e ‘*Total revenue*’.

O Firebase tem compatibilidade com três plataformas: iOS, Android e Unity e disponibiliza uma versão grátis e outra paga, a partir de \$25 (aproximadamente 20€) /mês.

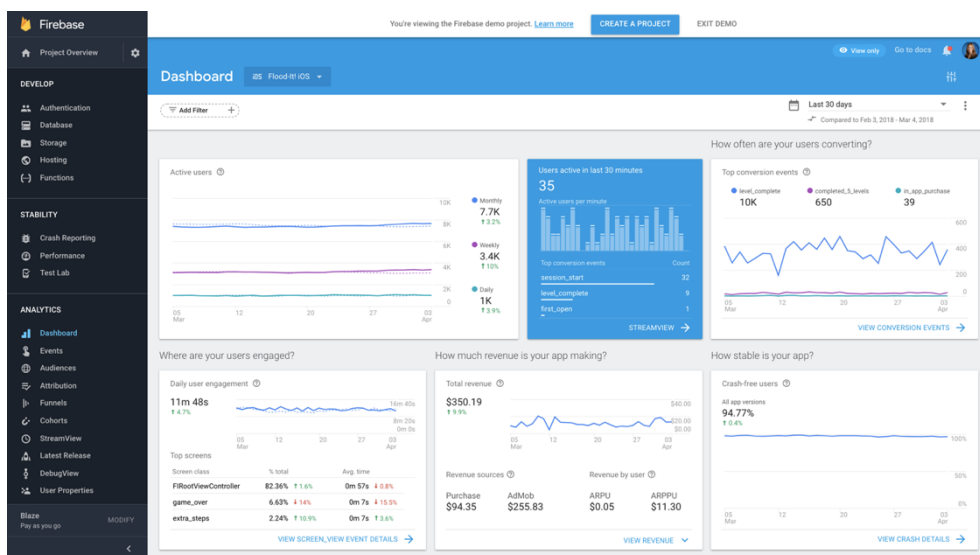


Figura 1 - Dashboard Firebase

O Firebase tem como funcionalidades as seguintes: *Hosting*, *Crash reporting*, *Test Lab* (para Android), Monitorização de desempenho de *apps*, *Push Notifications*, *Adwords* (Google Ads) e também a versão da *app* do Google Analytics está incluída no Firebase.

De seguida é apresentada a Tabela 1 com Prós e Contras da utilização do Firebase.

Tabela 1 - Prós e Contras Firebase

| PRÓS  | CONTRAS   |
|---|---|
| Relatórios de <i>crashes</i>  | Customização limitada                             |
| Tudo numa ferramenta para análise quantitativa, testes a/b e <i>hosting</i> | Pode ficar cara para <i>apps</i> com tráfego alto |

### 2.3.2 Mixpanel

O Mixpanel é uma solução de análise de produtos fundada em 2009. Ajuda as empresas a entender como os clientes interagem com as suas aplicações *web* e *móveis* para que possam melhorar os seus produtos.

O Mixpanel rastreia as interações do utilizador e permite que se construa relatórios personalizados.

Esta ferramenta de análise de *apps* permite avaliar as ações do utilizador segmentando-as e criando *funnels*.

Além disso, o Mixpanel é capaz de mostrar informações sobre dados, como localização, dispositivo, versões de *software* e muito mais. O Mixpanel também pode ser usado para comunicação direcionada através de notificações *push* [16].

O Mixpanel tem compatibilidade com três plataformas: iOS, Android e *web* e disponibiliza uma versão grátis e outra paga, a partir de \$999 (aproximadamente 824€) /ano.

O Mixpanel tem como funcionalidades as seguintes: *A/B Testing*, *Push Notifications*, *Funnels*, *Engagement Tracking* e compatibilidade com *mobile* e *web*.

De seguida é apresentada a Tabela 2 com Prós e Contras da utilização do MixPanel.

Tabela 2 - Prós e Contras Mixpanel

| PRÓS                              | CONTRAS   |
|-----------------------------------|---|
| Análise de <i>cohort</i>          | Requer muito tempo de desenvolvimento para a configuração |
| Rastreamento detalhado de eventos | Não é capaz de criar ciclos de CRM completos              |
| <i>Dashboards</i> personalizáveis |   |

### 2.3.3 Flurry

A Flurry é uma solução de análise de *apps*, criada pela Yahoo. A Flurry permite monitorizar o comportamento do utilizador em todas as plataformas sem nenhum custo.

Deve-se notar que o Flurry é gratuito porque a empresa pode compartilhar os seus dados, embora anonimizados, com terceiros [16].

Na Figura 2 [17] é apresentado o *dashboard* do Flurry onde se é possível visualizar dados como ‘*New Devices Daily*’, ‘*Active Devices Daily*’ e ‘*Medium Session Length Daily*’.

O Flurry tem compatibilidade com duas plataformas: iOS e Android e disponibiliza uma versão grátis.

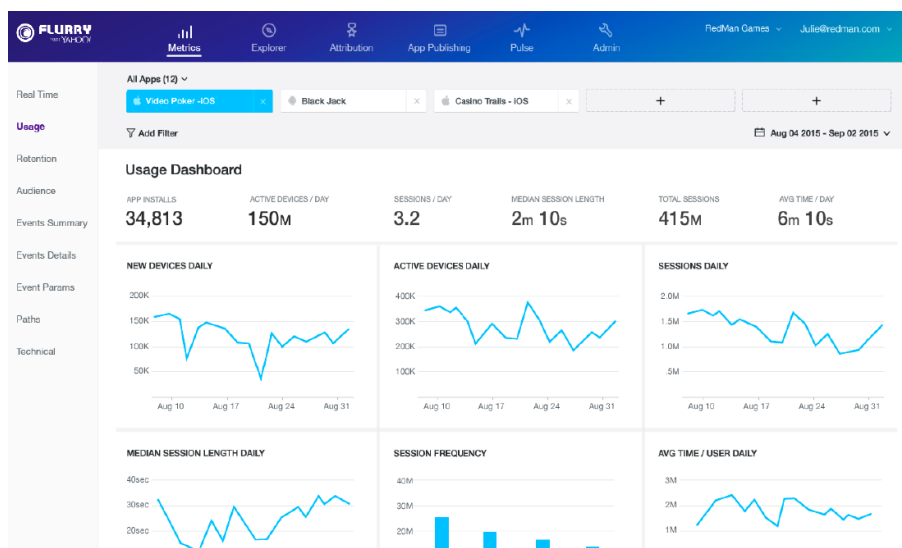


Figura 2 - Dashboard Flurry

O Flurry tem como funcionalidades as seguintes: Análise de *apps* grátis, Eventos, *Funnels*, Segmentos e *Cohorts*.

De seguida é apresentada a Tabela 3 com Prós e Contras da utilização do Flurry.

Tabela 3 - Prós e Contras Flurry

| PRÓS                                 | CONTRAS   |
|--------------------------------------|---|
| Grátis                               | Profundidade de análise não é tão robusta como outras ferramentas |
| Fluxos da utilização de cada cliente | Suporte lento   |

### 2.3.4 Countly

O Countly é uma plataforma de análise e marketing de nível corporativo para aplicações móveis, web e desktop. Mais de 2000 servidores countly em todo o mundo recolhem dados de mais de 16000 aplicações.

O Countly oferece uma *app* de análise de código aberto em tempo real que pode ser hospedado nos seus próprios servidores.

Na Figura 3 [16] é apresentado o *dashboard* do Countly onde se é possível visualizar dados como *'Online Users'*, *'Total Users'*, *'Time Spent'* e *'Average Time Spent'*.

O Countly tem compatibilidade com três plataformas: iOS, Android e Web e disponibiliza uma versão grátis e uma paga, a partir de \$350 (aproximadamente 288€) /mês [16].

O Countly tem como funcionalidades as seguintes: *app* de código aberto, extensível via *plugins*<sup>3</sup>, nuvem local ou privada, análise de retenção e análise de *funnel*.

<sup>3</sup> Programa de computador usado para adicionar funções a outros programas maiores

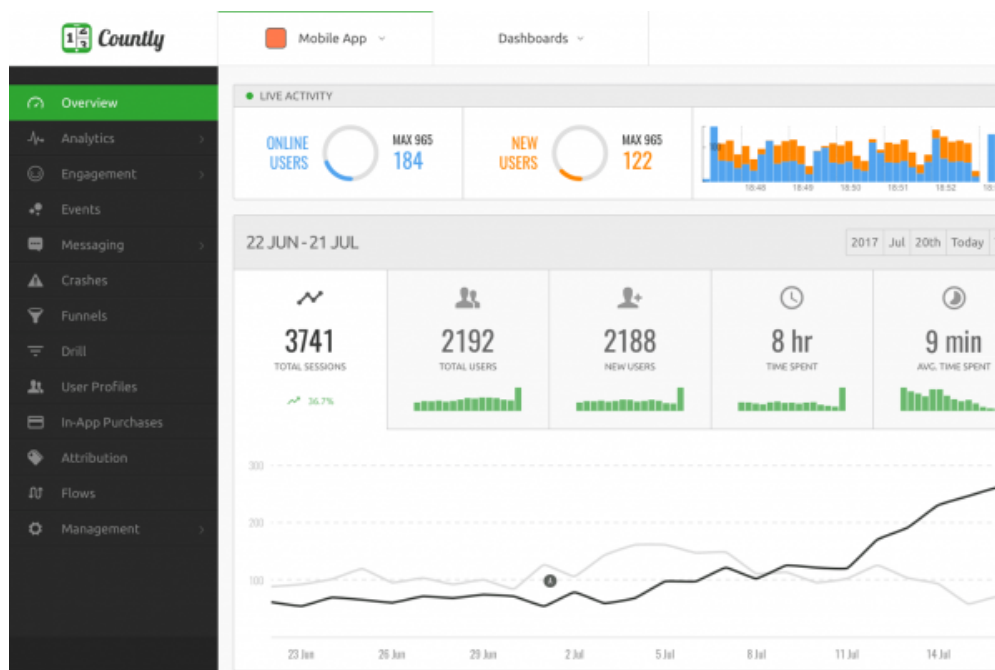


Figura 3 - Dashboard County

De seguida é apresentada a Tabela 4 com Prós e Contras da utilização do County.

Tabela 4 - Prós e Contras County

| PRÓS                                       | CONTRAS   |
|--|---|
| Código aberto, permite hospedagem no local | UI pouco intuitiva                                |
| Altamente customizável                     | Rastreamento limitado de <i>cohorts</i> e eventos |

### 2.3.5 UXCam

UXCam é uma solução de análise de experiência para *apps*. A UXCam permite que os profissionais de *apps* melhorem os KPIs (Indicador-chave de desempenho) das suas *apps* através de uma compreensão profunda do comportamento do utilizador.

A UXCam oferece benefícios que outras ferramentas de análise de *apps* não, como por exemplo:

1. Descobrir as verdadeiras causas de atrito e frustração. *Heatmaps* mostram onde os utilizadores tocam e quais os elementos que ignoram.
2. Obter mais contexto sobre onde e por que os utilizadores desistem. Descobrir como os utilizadores se comportam durante o seu *funnel*.
3. Replicar e reduzir os *crashes*. Economizar um valioso tempo do *developer* replicando cada *crash* em menos de 30 segundos.
4. Entender o caminho completo do utilizador com *replays* de vídeo de sessão.

Os clientes relataram taxas de conversão e retenção mais altas desde que começaram a usar o UXCam [16].

O UXCam tem compatibilidade com seis plataformas: iOS, Android, React Native, Flutter, Xamarin e Unity e disponibiliza uma versão temporária grátis e uma versão paga, consoante pedido.

O UXCAM tem como funcionalidades as seguintes: Entender facilmente a jornada completa do utilizador, descobrir as dificuldades do utilizador automaticamente, encontrar problemas e oportunidades em cada *screen*, analisar os *Heatmaps*, encontrar *bottlenecks* ocultos com *Screen Flow*, entender onde e por que os utilizadores deixam com o *Funnel Analytics*, encontrar facilmente sessões importantes com o *Session Replay* e marcação automática de elementos – sem necessidade de código adicional.

De seguida é apresentada a Tabela 5 com Prós e Contras da utilização do UXCam.

Tabela 5 - Prós e Contras UXCam

| PRÓS                                      | CONTRAS   |
|---|---|
| Análise qualitativas poderosa             | Plano gratuito é limitado                                       |
| Análise completa dos fluxos do utilizador | <i>Taps</i> não são preservados quando os vídeos são exportados |
| Replicar e resolver <i>crashes</i>        |   |
| Segurança elevada                         |   |
| SDK mais leve que das restantes           |   |

### 2.3.6 Apple iOS Analytics

O Apple iOS Analytics é a ferramenta de análise da Apple que está incluído na assinatura do Apple Developer Program e não requer qualquer implementação.

Com recursos limitados, é uma ferramenta de análise de *apps*, mas deve ser emparelhada com outra ferramenta de análise móvel mais avançada [16].

Na Figura 4 [17] é apresentado o *dashboard* do Apple Analytics onde se é possível visualizar dados como ‘*App Store Views*’, ‘*In-App Purchases*’, ‘*Sales*’, ‘*Sessions*’ e ‘*Active Devices*’.

O Apple iOS Analytics só tem compatibilidade com duas plataformas, ambas da Apple: iOS e tvOS e disponibiliza uma versão grátis.

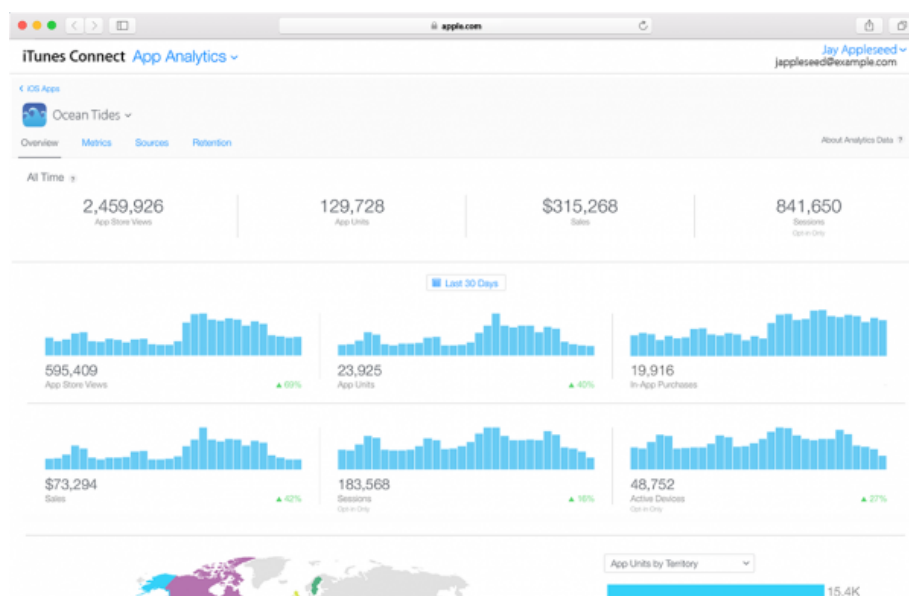


Figura 4 - Dashboard Apple Analytics

O Apple Analytics tem como funcionalidades as seguintes: dados de uso, dados de vendas e dados da *App Store*.

De seguida é apresentada a Tabela 6 com Prós e Contras da utilização do Apple Analytics.

Tabela 6 - Prós e Contras Apple Analytics

| PRÓS   | CONTRAS  |
|--|--|
| Não requer SDK                                     | Apenas para iOS (e com certeza que Android não será uma possibilidade) |
| Análise de aquisições ( <i>downloads</i> e vendas) | Sem <i>in-app analytics</i>  |
| Análise de atribuições                             | Análise básica   |

### 2.3.7 Adjust

Adjust é uma *app* de análise projetada para ajudar a identificar e construir a base de utilizadores. Esta *app* de análise rastreia tendências horárias, avalia LTV (*Lifetime value*) e analisa coortes.

A Adjust também permite agregar dados de utilizadores vitalícios num único painel. Também tem grande destaque o *Fraud Prevention Suite* da plataforma, que protege a *app* contra atividades fraudulentas [17].

Na Figura 5 [17] é apresentado o dashboard do Adjust onde se é possível visualizar dados como 'Deliverables', 'Fraud Prevention', e 'Cohorts'.

A Adjust tem compatibilidade com cinco plataformas: iOS, Android, Blackberry, Windows Phone e HTML5 e disponibiliza um plano básico a partir de \$129 (aproximadamente 106€) e *planos personalizados e de negócios com mais custos e mais recursos*.

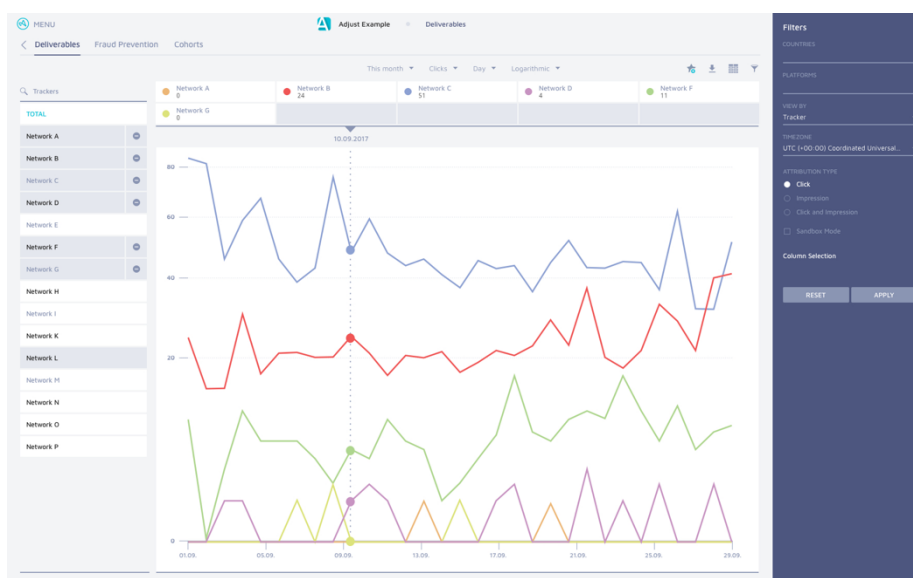


Figura 5 - Dashboard Adjust

## 2.3.8 App Annie

App Annie é uma plataforma de análise de dados que fornece visualizações sem precedentes dos dados da sua *app* e de outras *apps* no mercado.

Através da plataforma App Annie, ganha-se acesso imediato às inovações tecnológicas e conjuntos de dados da organização, permitindo acesso a dados compartilhados, identificando oportunidades e a capacidade de criar campanhas bem-sucedidas [17][18].

Na Figura 6 [17] é apresentado o dashboard da App Annie onde se é possível visualizar dados como 'Downloads', 'Revenue' e 'Usage'.

A App Annie tem compatibilidade com três plataformas: iOS, Android e Windows Phone e disponibiliza uma versão grátis com funcionalidades limitadas e planos personalizados com cálculo de preço consoante pedido.

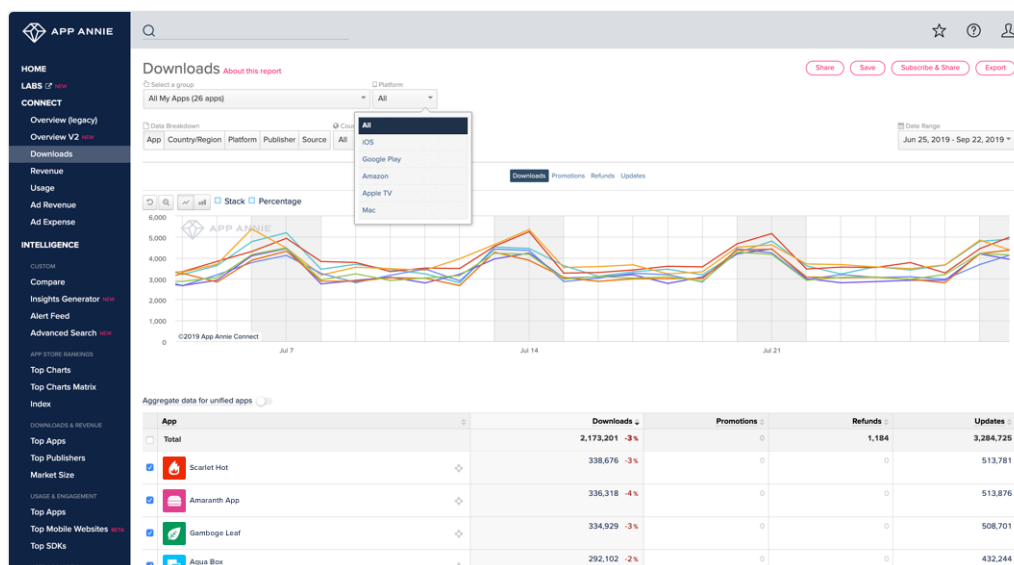


Figura 6 - Dashboard App Annie

### 2.3.9 Síntese

Nesta subsecção irá ser feita uma comparação de funcionalidades entre as soluções abordadas nas subsecções anteriores, com recurso à Tabela 7 e utilizando os seguintes critérios comparativos:

- Critério 1 – Informação de utilizadores
- Critério 2 - Eventos
- Critério 3 – Relatórios de falhas
- Critério 4 – Monitorização de desempenho
- Critério 5 – *Push Notifications*
- Critério 6 - Customização
- Critério 7 - *Funnels*
- Critério 8 - *Cohorts*
- Critério 9 – Tempo de sessão
- Critério 10 – Versão gratuita
- Critério 11 – Multiplataforma

Tabela 7 - Comparação entre soluções

|                        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8              | 9 | 10             | 11 |
|------------------------|---|---|---|---|---|---|---|----------------|---|----------------|----|
| <b>Firebase</b>        | X | X | X | X | X |   | X | X              | X | X              | X  |
| <b>Mixpanel</b>        | X | X |   | X | X | X | X | X              |   | X              | X  |
| <b>Flurry</b>          | X | X |   |   |   |   | X | X              | X | X              | X  |
| <b>Countly</b>         | X | X | X | X | X | X | X | X <sup>4</sup> | X | X              | X  |
| <b>UXCam</b>           | X | X | X |   |   |   | X |                | X | X <sup>5</sup> | X  |
| <b>Apple Analytics</b> | X |   |   |   |   |   |   |                | X | X              |    |
| <b>Adjust</b>          | X |   |   |   |   |   |   | X              |   |                | X  |
| <b>App Annie</b>       | X |   |   |   |   |   |   |                | X | X <sup>6</sup> | X  |

<sup>4</sup> Análise limitada

<sup>5</sup> Versão temporária

<sup>6</sup> Versão com funcionalidades limitadas

Como se pode observar pela análise à Tabela 7, existem três soluções que se destacam mais pelo número de funcionalidades que disponibilizam, sendo elas o Firebase, o Mixpanel e o Countly. O Countly, sendo esta a que disponibiliza mais funcionalidades, foi também a escolhida para a implementação desta solução.

Pelo contrário, as soluções com menos funcionalidades são o Apple Analytics, Adjust e App Annie. Destas últimas, o Apple Analytics é realmente a que mais peca em termos de oferta dado que só é compatível com dispositivos iOS e não encaixa nos requisitos que um *developer*/organização procura em termos de monitorizar as suas *apps* nas mais diferentes plataformas.

## 2.4 Tecnologia

Em relação à tecnologia a utilizar na solução, já foi pré-definido pela empresa X que se iria utilizar o Flutter. Esta tomada de decisão foi feita com base no Flutter ser uma *framework* relativamente recente e com bastante crescimento, e dado que alguns *developers* desta *framework* são desenvolvedores da empresa X nos Estados Unidos da América.

De seguida irá ser apresentado o Flutter e a sua gestão de estados.

### 2.4.1 Flutter

O Flutter é uma *framework* de desenvolvimento de *software* transversal que foi apresentado pela Google em 2015 e teve o seu primeiro lançamento em maio de 2017. O Flutter tem crescido de forma constante e oferece possibilidades não só para o desenvolvimento mobile de iOS e Android, mas também para aplicações web e desktop [19].

O Facebook já tinha lançado em 2015 a sua *framework* de desenvolvimento *mobile cross-platform* (para várias plataformas), o React Native.

Contudo, desde o lançamento do Flutter, muitos utilizadores preferem utilizar a tecnologia da Google, pela experiência de utilização, tempo de desenvolvimento e performance, tornando-se uma alternativa ao React [20].

A linguagem usada para construir o Flutter, foi o Dart. O Dart é uma linguagem de programação orientada a objetos que foi apresentada pela Google pela primeira vez em 2011. Desde então, o Dart evoluiu constantemente disponibilizando diferentes funcionalidades. A sua sintaxe é muito semelhante às línguas de Java, Swift ou Kotlin [19].

A *framework* do Flutter foi concebida utilizando alguns princípios que devem ser descritos separadamente. Estes princípios são: "Tudo é um *widget*", "Composição > Herança" e "*Widget tree*" [19].

Em Flutter quase tudo é um *widget* e isto é o básico de construção da aplicação. Em comparação com outras *frameworks*, o Flutter não tem *controllers*, *views* ou *layouts* separados. Quase todos os aspetos de desenvolvimento do Flutter são cobertos por um bloco de construção unificado - o *widget*. Um *widget* pode ser um botão, um elemento de estilo ou um ecrã *pop-up*, por exemplo [19].

A abordagem de composição é melhor do que a herança: Muitas vezes os *widgets* são compostos por outros *widgets* menores e essa é a abordagem baseada na composição. A utilização de uma Flutter API permite combinar vários *widgets* para provocar o comportamento que se necessita [19].

Um conceito de '*Widget tree*' é basicamente uma implementação de *nested widgets* que representam componentes da interface do utilizador. Estes *widgets* podem ser *stateless* ou *stateful* e a diferença entre eles está em manter o estado do *widget*, o que ajuda na gestão dos estados das aplicações [19].

A Google criou dois conjuntos de *widgets*, Cupertino (iOS) e Material (Android). Estes conjuntos de *widgets* são responsáveis pela interface do utilizador (UI) e incluem todos os componentes necessários para o desenvolvimento Android e iOS. Estes *widgets* não estão ligados à API nativa do iOS ou Android como em React Native, por exemplo, mas funcionam como componentes autónomos do Flutter com velocidade e animação de renderização adequadas. Esta é uma das principais

características do Flutter. Os utilizadores não serão afetados pela má experiência do utilizador, por causa disso [19].

A Figura 7 ilustra uma vista geral do sistema do Flutter.

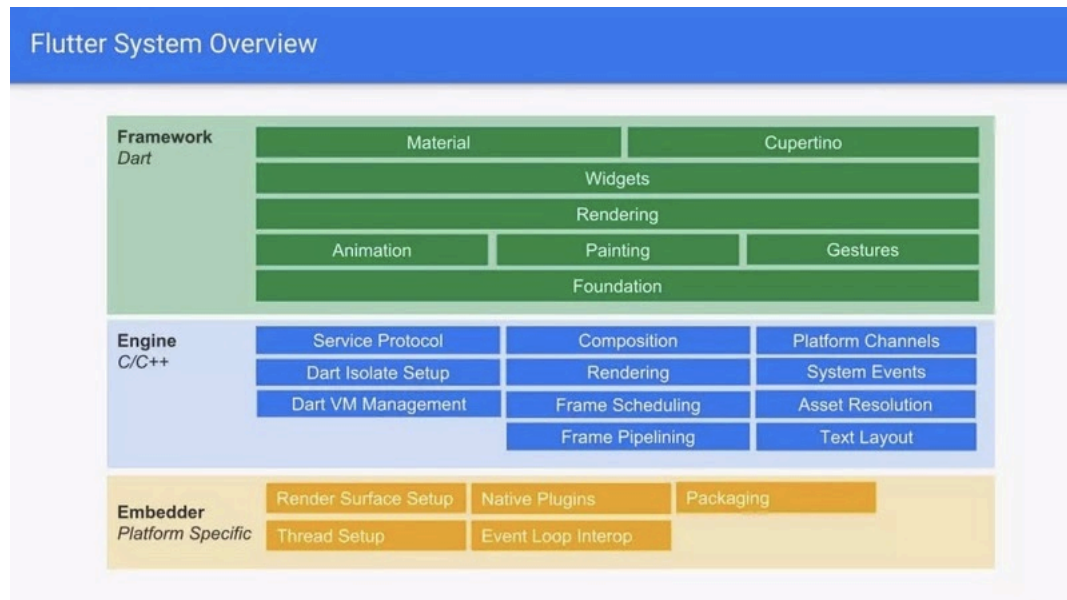


Figura 7 - Flutter System Overview [21]

### 2.4.2 Gestão de Estados

Além dos conceitos fundamentais apresentados anteriormente, o Flutter oferece a arquitetura básica que se pode aplicar à aplicação móvel e gerir facilmente o seu estado. A arquitetura que é usada em Flutter para a sua gestão de estados chama-se BLOC.

#### **BLOC**

O BLOC é basicamente uma técnica de gestão do estado da aplicação que fornece uma forma simples de o gerir durante o ciclo de vida da app. BLOC é acrónimo para Business Logic and Component, os componentes são divididos em três secções principais: a camada de dados, camada de bloco e a camada de UI/camada de apresentação.

As ações desencadeadas pela UI são chamadas de eventos, com base nestes eventos a chamada à camada de dados foi feita e de acordo com as respostas da chamada se o sucesso ou o fracasso do estado são devolvidos, o que por sua vez atualiza a UI [21].

Na Figura 8 pode-se compreender melhor a arquitetura da gestão de estados BLOC.

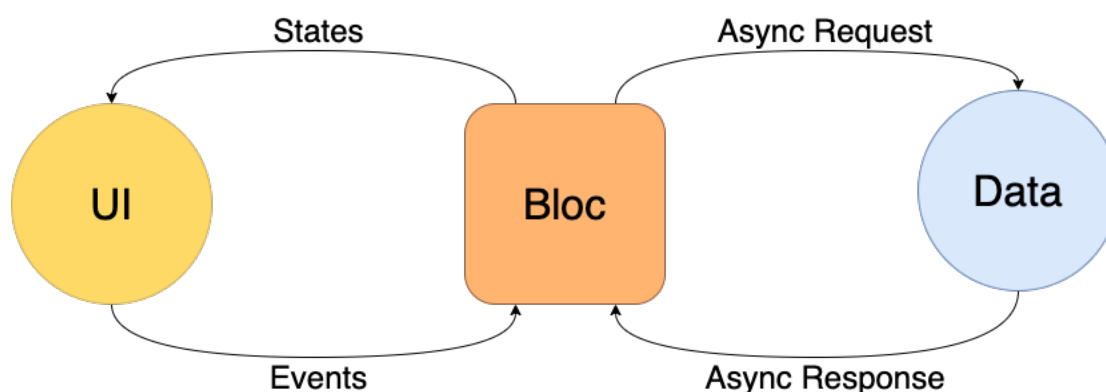


Figura 8 - Arquitetura Gestão de Estados BLOC

### **MobX**

Para além da arquitetura BLOC, existem outras alternativas conhecidas no mundo do Flutter. De seguida irá ser abordada uma alternativa, o MobX.

O MobX é uma biblioteca para gestão de estado, o que significa que se pode utilizar para estados locais (páginas) ou globais.

Este baseia-se em observáveis, ações e reação, um conceito que vem da programação reativa. No caso, toda ação executada gera uma reação, que procura um observável para atualizá-lo.

Ao contrário do BLOC que é um padrão de *design* e cuja implementação está aberta ao desenvolvedor, o MobX é uma biblioteca de gestão estatal de programação funcional transparente que implementa o padrão de observador com uma sintaxe amigável e uma API simples que facilita a aprendizagem quando se é um principiante. A MobX é escalável e tem sido usada em inúmeros grandes projetos como *Microsoft Office, Jenkins, Coinbase* [22].

## Comparação

Na Tabela 8 é apresentada uma comparação de Prós e Contras do BLOC e do MobX.

Tabela 8 - Comparação entre BLOC e MobX

| BLOC PROS  | BLOC CONS   |
|--|---|
| <ul style="list-style-type: none"><li>• Fácil de separar a UI da lógica de negócio;</li><li>• Flexível;</li><li>• Fácil de testar;</li><li>• Boa performance</li></ul> | <ul style="list-style-type: none"><li>• Difícil de entender;</li><li>• Não há implementação <i>standart</i></li></ul> |
| MobX PROS  | MobX CONS   |
| <ul style="list-style-type: none"><li>• Fácil de aprender e usar;</li><li>• Fácil de testar;</li><li>• Boa performance;</li><li>• Boa documentação</li></ul>           | <ul style="list-style-type: none"><li>• Geração de código lenta</li></ul>   |

## 2.5 Sumário

Neste capítulo foi efetuado o Estado de Arte, onde foram analisadas as métricas a analisar e outras soluções existentes para além da que será utilizada nesta dissertação.

Dado que a solução já se encontrava definida de início, no final foi efetuada uma comparação das funcionalidades existentes entre as oito soluções abordadas.

Ao longo deste capítulo foi possível responder às questões propostas:

### Quais as melhores soluções para monitorizar *apps*?

Não se pode dizer que existe uma melhor solução, mas baseado na comparação que foi efetuada, as soluções com mais funcionalidades/mais métricas disponíveis seriam o Firebase, o Mixpanel e o Countly, sendo esta última a escolhida para a implementação.

**Quais são as métricas com mais relevância a serem monitorizadas?**

As métricas com mais relevância foram as apresentadas neste capítulo: informação dos utilizadores, eventos, relatórios de falhas, monitorização de desempenho, *funnels*, *cohorts* e tempo de sessão.

## 3 Análise de Valor

Neste capítulo será apresentada a Análise de Valor (AV) desta dissertação, um breve contexto, valor, valor percebido, valor para o cliente, proposta de valor e modelo Canvas.

A AV neste ponto é referente ao problema apresentado no Capítulo 1 e visa criar valor na oportunidade existente.

A AV permite uma abordagem organizada para melhorar a rentabilidade de aplicações de produto [23].

A AV é um processo sistemático, formal e organizado de análise e avaliação. As técnicas que suportam as atividades de AV incluem técnicas comuns usadas para todos os exercícios de AV e outras apropriadas em determinadas condições.

Neste capítulo será abordado parte do **Outcome 2: Contexto e Estado da Arte** e no final irá ser dada a resposta à seguinte questão:

**Qual o valor do trabalho apresentado nesta dissertação para a melhoria do desempenho das *apps*?**

### 3.1 Processo de Inovação

Existem vários pontos de vista e conceitos referentes à inovação que afirmam, com base na análise de definições da literatura, que “a inovação é o processo de várias etapas através das quais as organizações transformam ideias em produtos novos/melhorados, serviços ou processos, a fim de avançar, competir e diferenciar-se com sucesso no seu mercado”.

A inovação pode ser vista como um processo ou como o resultado desse processo: inovação em produtos, em processos, em métodos de *marketing*, ou métodos organizacionais. Pode ser também vista na perspectiva da intensidade de resultado: inovação radical ou incremental [24].

Tal como indicado na Figura 9, o processo de inovação pode ser dividido em três subprocessos [25]:

- Modelo *Fuzzy Front End* (FFE);
- *New Product Development* (NPD);
- Comercialização.

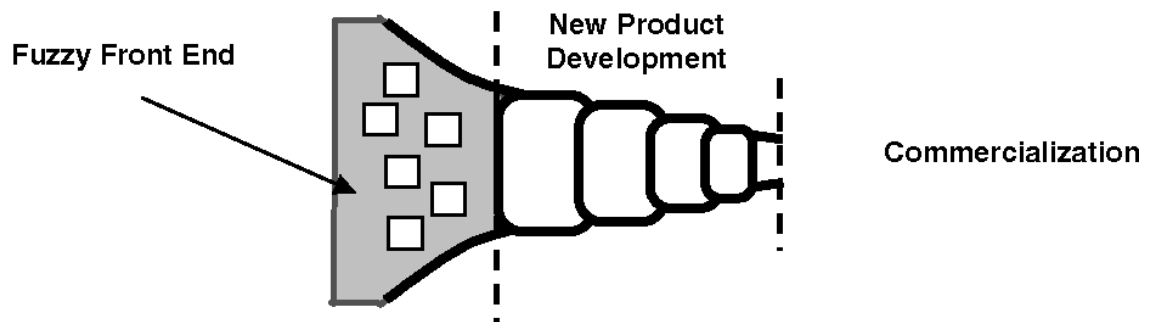


Figura 9 - Processo de inovação [21]

O primeiro processo – o FFE – é geralmente considerado como uma das maiores oportunidades de melhoria do processo global de inovação. Muitas empresas melhoraram drasticamente o tempo de ciclo e a eficiência, implementando uma abordagem formal de Stage-Gate<sup>7</sup> ou PACE<sup>8</sup> para gerir projetos na parte NPD do processo de inovação.

As atenções estão cada vez mais focadas nas atividades *front-end* que precedem esse processo formal e estruturado, a fim de aumentar o valor e a probabilidade de sucesso de conceitos de alto lucro entrando no desenvolvimento e comercialização de produtos [25].

Nesta dissertação será abordado o novo conceito de desenvolvimento de Peter Koen (NCD), uma vez que foi desenvolvido a partir da necessidade de identificar com maior clareza o *Fuzzy Front End*, o denominado *Front end of innovation* (FEI).

<sup>7</sup> Técnica de gestão de projetos em que uma iniciativa ou projeto é dividido em fases distintas, separadas por pontos de decisão (conhecidos por *gates*).

<sup>8</sup> Significa Excelência de Produto e Ciclo (*Product and Cycle-time Excellence*), e conta com sete elementos interrelacionados para garantir o sucesso do projeto e minimizar o tempo de mercado.

### 3.1.1 Modelo de desenvolvimento de novo conceito (NCD)

O modelo NCD mostrado na Figura 10 consiste em três partes-chave:

- A parte do motor ou *bull's-eye* é a liderança, cultura e estratégia de negócios da organização que impulsiona os cinco elementos-chave que são controláveis pela corporação.
- A área interna define os cinco elementos de atividade controláveis (identificação de oportunidades, análise de oportunidades, gênese de ideias e enriquecimento, seleção de ideias e definição de conceito) da FFE.
- Os fatores influenciadores consistem em capacidades organizacionais, o mundo externo (canais de distribuição, direito, política governamental, clientes, concorrentes e clima político e econômico), e as ciências (internas e externas) que podem estar envolvidas. Esses fatores afetam todo o processo de inovação até a comercialização. Esses fatores influenciadores são relativamente incontroláveis pela corporação [25].

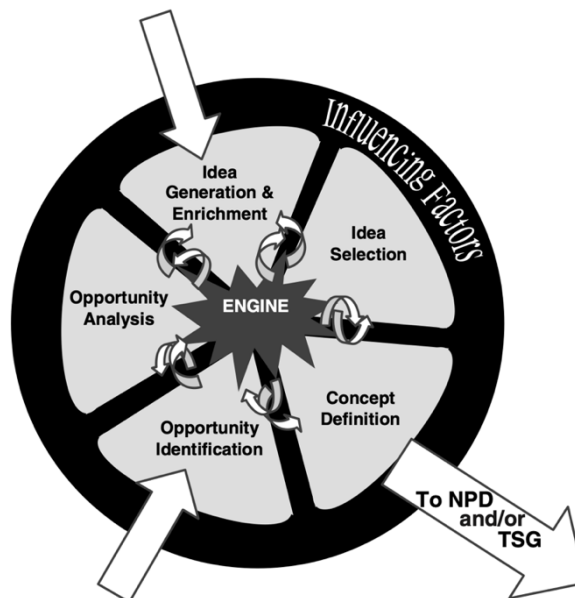


Figura 10 - Modelo NCD [21]

Várias características do modelo merecem ser notadas. As partes internas do NCD são chamadas de elementos, ao contrário dos processos.

Um processo implica uma estrutura que pode não ser aplicável e poderia forçar o uso de um conjunto de controlos mal projetados para gerenciar as atividades da FFE.

Além disso, o modelo tem uma forma circular, sugerindo que as ideias devem fluir, circular e iterar entre todos os cinco elementos. O fluxo pode abranger os elementos em qualquer ordem ou combinação e pode usar um ou mais elementos mais de uma vez. Isso contrasta com o processo sequencial de NPD ou Stage-Gate™, no qual as atividades de *looping* e redirecionamento (ou *redo*) estão associadas a atrasos significativos, custos adicionais e projetos mal geridos. Iteração e *loopbacks* fazem parte das atividades da FFE. Embora o *looping* inerente possa atrasar a FFE, ela normalmente encurta o tempo total do ciclo de desenvolvimento e comercialização do produto [25].

### ***Identificação da Oportunidade***

Neste elemento é onde a organização identifica as oportunidades tecnológicas e de negócio.

A oportunidade pode seguir uma direção totalmente diferente para o negócio ou um pequeno *upgrade* a um produto existente. Também pode ser uma nova plataforma de produtos, um novo processo, uma nova oferta de serviços ou uma nova abordagem de *marketing* ou vendas [26].

A solução apresentada nesta dissertação surge da oportunidade de melhorar a *app* da empresa X, assim como otimizar todo o processo de desenvolvimento da mesma, tornando-a mais fiável e menos suscetível a erros.

### ***Análise da Oportunidade***

Depois da identificação da oportunidade são necessárias informações adicionais para traduzir a identificação de oportunidades em oportunidades específicas de negócios e tecnologia.

Podem ser aplicados esforços extensivos para grupos focais, estudos de mercado e/ou experiências científicas [26].

Após serem pesados os prós e contras da sua implementação através do estudo de mercado e da pesquisa sobre abordagens a seguir, pode-se concluir que existe efetivamente uma oportunidade relevante para resolver os problemas a que este caso de estudo assume colmatar, fazendo-a corresponder às expectativas dos utilizadores e tornando-a uma presença ainda mais forte no mercado onde esta se insere.

### ***Génese da ideia***

Este elemento é o nascimento, desenvolvimento e amadurecimento da oportunidade numa ideia concreta. Isto representa um processo evolutivo no qual as ideias são construídas, combinadas, remodeladas, modificadas e atualizadas. A ideia pode passar por muitas iterações e mudanças à medida que é examinada, estudada, discutida e desenvolvida.

A génese da ideia pode ser um processo formal, incluindo sessões de *brainstorming* e bancos de ideias, de modo a fazer a organização gerar ideias novas ou modificadas para a oportunidade identificada [26].

Nesta fase, espera-se que existam sessões de *brainstorming* para a ideia estar em constante processo de aperfeiçoamento, onde é expectável que todos os intervenientes no processo de desenvolvimento desta solução contribuam com as suas opiniões (*developers, stakeholders, product owner, scrum master e designer*).

### ***Seleção da ideia***

Nesta fase, considerando que muitas ideias surgiram, é importante decidir as que escolher para alcançar o maior valor do negócio. É um processo difícil quando existe um grande volume de ideias, podendo haver a possibilidade de descartar ideias que poderiam ser muito vantajosas para a organização [26].

Aqui, foram pesados os gastos financeiros e temporais para o desenvolvimento da solução em questão, e após a reflexão sobre o valor que esta alteração poderá trazer para a organização, concluiu-se que a sua aplicabilidade tem grande probabilidade de trazer um valor comercial superior ao gasto financeiro e temporal que acarreta.

### ***Desenvolvimento do conceito***

O elemento final envolve o desenvolvimento de um caso de negócio com base em estimativas de potencial de mercado, necessidades do cliente, requisitos de investimento, avaliações de concorrentes, incógnitas tecnológicas e risco geral de projetos [26].

Após a seleção dos métodos e tecnologias corretos a seguir, é definido um caso de negócio, por meio de reuniões com as equipas e *stakeholders* da empresa, para definir os riscos da solução, as necessidades de mercado, de investimento e do esforço necessários.

## **3.2 Valor**

O valor de um produto é a visão (ou expectativa) do consumidor quanto aos seus benefícios em relação ao que oferece [27].

O valor de um produto está associado ao custo de fabrico e ao preço a que é vendido de forma a haver lucro. Corresponde ao valor sem expectativas exteriores, tanto do comprador como do vendedor. É ainda caracterizado pela razão entre os benefícios e os respetivos custos/sacrifícios.

Durante uma compra, o consumidor analisa e avalia o valor do produto ou serviço que a empresa oferece e o valor dessa transação. É de referir que este valor pode não ser só monetário, mas também temporal ou de esforço. Desta forma, para haver negócio terá de ser criado valor, uma vez que apenas com valor é que o consumidor compra repetidamente o produto/serviço [28].

Avaliando os possíveis resultados deste trabalho, encontram-se vários benefícios tanto para o cliente como para a empresa. Com a melhoria na performance das *apps*, a equipa de *developers* consegue testar e validar desenvolvimentos mais facilmente, o que naturalmente vai permitir a entrega dos projetos mais rapidamente, diminuindo os erros, permitindo tempos de resposta aceitáveis e uma maior cobertura de testes. O valor irá tentar ser alcançado através da comparação entre uma *app* com e sem *analytics*.

### 3.3 Valor percebido

O modelo de valor varia de cliente para cliente pois este baseia-se nas necessidades, desejos, características e capacidades financeiras de cada um.

Distintos clientes percebem diferente valor para os mesmos produtos/serviços. Para além disso, organizações que possam estar envolvidas no processo de compra podem possuir diferentes percepções do que seja valor para os clientes [29].

Podem-se observar os benefícios e sacrifícios do projeto descrito neste relatório na Tabela 9.

Neste caso, o valor percebido é alto para o cliente, pois a empresa/*developers* poderão ter grandes benefícios e sucesso, e por sua vez, alcançar mais clientes.

Tabela 9 - Benefícios e Sacrifícios

|                    | Serviço             | Relação   |
|--------------------|---------------------|---|
| <b>Benefícios</b>  | Melhoria no produto | Redução de erros;<br>Aumento da Satisfação;<br>Mais clientes. |
| <b>Sacrifícios</b> | Custos associados   | Tecnologia paga;<br>Formação para as novas tecnologias.       |

### 3.4 Valor para o cliente

O termo 'valor para o cliente' é usado dentro da literatura de *marketing* para retratar tanto o que é entregue ao cliente pelo fornecedor, quanto o que é entregue ao fornecedor pelo cliente.

O valor para o cliente é qualquer percepção pessoal de vantagem decorrente da associação de um cliente com a oferta de uma organização, podendo ocorrer como redução do sacrifício; presença de benefício (percebido como atributos ou desfechos); ou resultante de

qualquer combinação ponderada de sacrifício e benefício (determinado e expresso racional ou intuitivamente); ou uma agregação, ao longo do tempo, de qualquer ou tudo isso [30]. Neste caso, a solução apresenta valor para o utilizador dado que irá diminuir a quantidade de *bugs* e *crashes* da *app*, assim como tornar uma melhor experiência de utilização para este.

### 3.5 Proposta de Valor

Uma proposta de valor é compreendida como a afirmação dos benefícios oferecidos pela marca ao cliente e que lhe proporcionam valor. É entendida como uma estratégia eficaz ao conduzir a empresa a uma relação com o cliente, que o leva a decidir pela compra [31].

A proposta de valor de uma empresa é muito mais do que se posicionar num único atributo; é uma declaração sobre a experiência resultante que os clientes obterão com a oferta e seu relacionamento com o fornecedor. A marca deverá representar uma promessa relativa à experiência total resultante que os clientes podem esperar. Se a promessa será ou não cumprida, depende da capacidade da empresa em gerir seu sistema de entrega de valor. O sistema de entrega de valor inclui todas as experiências de comunicação e canais que o cliente terá a caminho da obtenção da oferta [32].

Sendo assim, a proposta de valor deste documento depreende-se no facto de *mobile analytics* permitirem aos *developers* melhorar as suas *apps* em termos de performance e menor tempo de produção, contribuindo para um melhor *feedback* por parte dos utilizadores, conseqüentemente trazendo um crescimento para a empresa.

### 3.6 Modelo de negócio Canvas

O *Business Model Canvas* ou "Quadro de modelo de negócios" é uma ferramenta de gestão estratégica, que permite desenvolver e esboçar modelos de negócio novos ou existentes numa única página. É um mapa visual pré-formatado que cobre os três pilares conceituais da definição de modelo de negócios, assim apresentado na Tabela 10 [33]:

Tabela 10 - Definição Modelo de Negócios

|                         |                                       |   |
|-------------------------|---------------------------------------|---|
| <b>Criação de Valor</b> | <b>Parcerias-chave</b>                | Rede de fornecedores e os parceiros que põem o modelo de negócio em execução                            |
|                         | <b>Atividades-chave</b>               | Ações mais importantes que uma empresa deve tomar para fazer funcionar seu modelo de negócio            |
|                         | <b>Recursos-chave</b>                 | Recursos mais importantes para fazer um modelo de negócio funcionar                                     |
| <b>Entrega de Valor</b> | <b>Canais</b>                         | Maneira como uma empresa se comunica e alcança os seus clientes a fim de realizar uma proposta de valor |
|                         | <b>Segmentos de clientes</b>          | Distintos grupos de pessoas ou organizações que uma empresa busca alcançar, servir e criar valor        |
|                         | <b>Relacionamento com os clientes</b> | Tipos de relacionamento que uma organização pode adotar com seus segmentos de clientes específicos      |
|                         | <b>Propostas de valor</b>             | Pacote de produtos e serviços bem como os valores que são entregues aos segmentos de clientes           |
| <b>Captura de Valor</b> | <b>Estrutura de custo</b>             | Custos envolvidos na operação de um modelo de negócio   |
|                         | <b>Fontes de receitas</b>             | Dinheiro que uma empresa gera a partir de cada segmento de cliente                                      |

# The Business Model Canvas

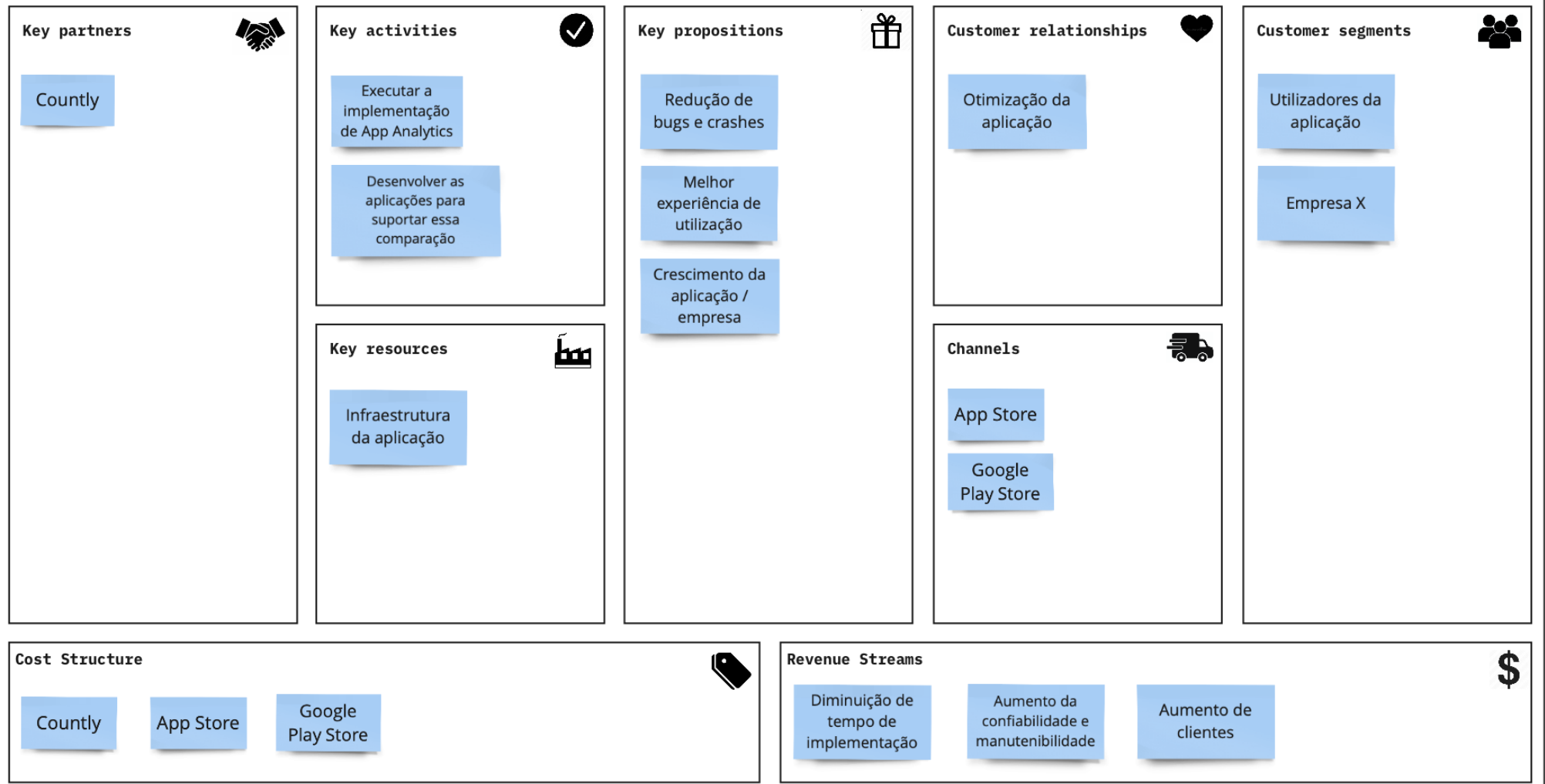


Figura 11 - Modelo Canvas

### 3.7 Sumário

Foi apresentada a AV do projeto de forma a perceber qual o real valor do projeto para o cliente. Verificou-se que a solução apresenta valor para o utilizador pois irá diminuir a quantidade de *bugs* e *crashes* da *app*, assim como tornar uma melhor experiência de utilização para este. Foi ainda realizado o modelo Canvas.

Neste capítulo, assim como proposto, foi respondida a seguinte questão:

**Qual o valor do trabalho apresentado nesta dissertação para a melhoria do desempenho das *apps*?**

O objectivo desta dissertação passa pelo estudo do problema e como este pode ser combatido utilizando *mobile analytics*. Este estudo ajudará a perceber quais as métricas a monitorizar e a importância destas no sucesso de uma *app* e no eventual crescimento de uma organização/empresa.



# 4 Análise e Design da Solução

Neste capítulo, irá ser feita uma análise mais profunda à solução, com vista em detalhar como esta estará estruturada tecnicamente, segundo as boas práticas, padrões e regras da programação aprendidos.

Serão utilizados diagramas Unified Modeling Language (UML) de forma a expor os artefactos em questão.

Neste capítulo irá ser abordado o **Outcome 4: Design da Solução**.

## 4.1 Análise

Neste documento será proposta uma solução para melhorar o desempenho das aplicações móveis.

A solução proposta passa pela criação de uma nova aplicação da empresa X dirigida aos utilizadores de uma marca de automóveis, com mais foco nas funcionalidades do utilizador de carros híbridos<sup>9</sup>.

A *feature* da aplicação a desenvolver chama-se **Programa de Pontos** e o seu propósito é incentivar os utilizadores que possuem carros híbridos, a conduzir de forma elétrica. Os utilizadores que o fizerem, irão ganhar pontos por cada quilómetro conduzido deste modo, assim como por cada sessão de carregamento do carro. Estes pontos posteriormente podem ser gastos para adquirir *vouchers* para por exemplo, efetuar carregamentos grátis. Após a análise e definição daquilo que se pretende para o desenvolvimento da solução é importante definir os casos de uso bem como os requisitos funcionais e não funcionais.

---

<sup>9</sup> Carro híbrido move-se com recurso a dois tipos de energia diferentes: motor a combustão e motor elétrico [42]

### 4.1.1 Requisitos Funcionais

Em engenharia de *software*, um requisito funcional define uma função de um sistema de *software* ou componente. O requisito funcional representa o que o *software* faz, em termos de tarefas e serviços [34].

Com o início do desenvolvimento do desenho da aplicação móvel, foram definidos os requisitos funcionais da mesma. Estes representam as funcionalidades chave e serão distribuídos pelos demais Casos de Uso ou *Use Case* (UC).

- **UC1: Adesão ao Programa de Pontos:** Apenas utilizadores com carros híbridos mapeados na *app*, podem aderir ao Programa de Pontos;
- **UC2: Consulta de Carteira:** O utilizador pode consultar a sua Carteira, com os pontos disponíveis, pontos pendentes, pontos a expirar e seus carros;
- **UC3: Consulta de Histórico de Atividade:** O utilizador pode consultar todo o seu histórico de atividade, desde carregamentos, viagens em modo elétrico, prémios resgatados, etc;
- **UC4: Consulta de Desafio para Ganhar Pontos:** O utilizador pode consultar o desafio mensal disponível para ganhar pontos com carregamentos efetuados;
- **UC5: Consulta de Prémios:** O utilizador pode consultar o catálogo de prémios disponíveis para adquirir;
- **UC6: Resgatar Prémios:** O utilizador resgata prémios disponíveis em troca de pontos que tenha na sua carteira;
- **UC7: Sair do Programa de Pontos:** O utilizador sai do Programa de Pontos.

Através da Figura 12, apresenta-se um diagrama dos UC descritos.

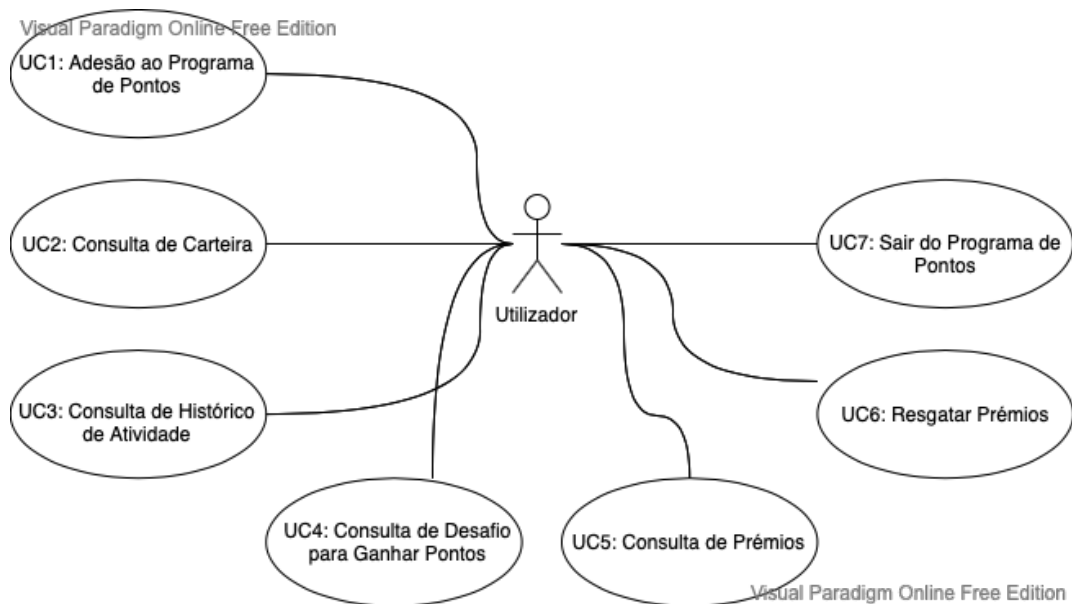


Figura 12 - Diagrama de Casos de Uso

#### 4.1.2 Requisitos Não Funcionais

Os requisitos não funcionais, são requisitos que especificam como o sistema se deve comportar. Os mesmos, podem ser considerados como atributos do sistema [35].

Para definir os requisitos não funcionais, é utilizado o modelo FURPS+ (*Functionality, Usability, Reliability, Performance e Supportability* e + para categorias adicionais), este modelo, permite a fácil classificação de requisitos não abrangidos pelos requisitos funcionais [36].

Serão apresentados os principais requisitos não funcionais para o sistema a desenvolver:

- **Usabilidade** – pode haver utilizadores que não estejam à vontade com aplicações móveis, como por exemplo, as pessoas mais idosas, sendo necessário garantir a melhor usabilidade possível para evitar que o utilizador passe demasiado tempo a tentar executar uma funcionalidade;
- **Confiabilidade** – a aplicação deve ter uma taxa de erros baixa (< 1 % por utilizador);

- **Desempenho** - O tempo de resposta da aplicação deve ser o mínimo possível de forma a melhorar a interação do utilizador;
- **Compatibilidade** – a aplicação deve ser suportada em dispositivos iOS e Android;
- **Restrições de interface** – a aplicação deverá ter interfaces intuitivas pretendendo-se que o sistema seja ágil, fácil de usar e completo em termos de informação.

## 4.2 Desenho

Nesta secção será apresentada a arquitetura que servirá como base para o desenvolvimento da aplicação.

### 4.2.1 Condicionantes do Desenho

O desenho arquitetural de uma solução é de extrema importância, uma vez que define a sua estrutura e como esta irá executar as funcionalidades pretendidas, detalhando os padrões utilizados.

A gestão de estados escolhida para a implementação da solução foi o BLOC, como analisado na Secção 2.4, e dito isto, a estrutura da aplicação vai estar diretamente relacionada com esta escolha.

Um exemplo dessa estrutura é apresentado na Figura 13.

Neste caso específico, para uma funcionalidade de '*Vouchers*', foi criado o ficheiro **vouchers\_bloc** que terá toda a lógica e chamadas necessárias ao repositório, **vouchers\_event**, que como o nome indica, terá os eventos necessários para a funcionalidade em questão, e por fim o ficheiro **vouchers\_state** que terá também os estados precisos para o seu desenvolvimento.

Na Figura 14 pode-se compreender melhor o funcionamento destes ficheiros na aplicação. O **vouchers\_bloc** irá comunicar com a UI através de estados presentes no **vouchers\_state**, e conseqüentemente a UI irá comunicar com o **vouchers\_bloc** através de eventos do **vouchers\_event**.

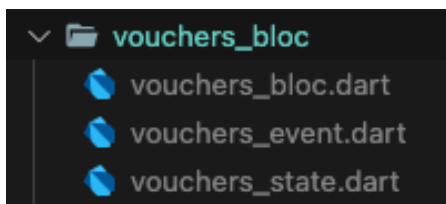


Figura 13 – Divisão ficheiros  
Arquitetura BLOC

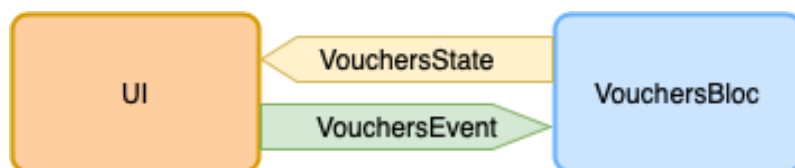


Figura 14 – Flow do BLOC

O BLOC é uma boa abordagem que separa a lógica de negócio da interface de utilizador e supervisiona os pontos-chave da lógica de negócio. As ideias centrais que foram usadas para a gestão de estados BLOC são a simplicidade, a escalabilidade e a testabilidade, e todos estes objetivos foram definitivamente alcançados dentro da arquitetura BLOC.

#### 4.2.2 Arquitetura lógica do sistema

Na Figura 15 está representado o diagrama de componentes da aplicação, que se divide essencialmente em quatro componentes:

- **Mobile App** – Componente referida ao longo de toda a dissertação e cuja implementação é um dos objetivos deste projeto. Este componente é responsável pela interface com o utilizador através da apresentação e manipulação de dados recebidos do servidor;
- **BFF API** – Componente responsável por gerir toda a informação apresentada e manipulada na camada da aplicação. Todos os dados que se pretendam obter ou enviar passam por esta camada devido à sua ligação com os serviços do componente Engine API;
- **Engine API** – Componente que fornece informação à BFF API de funcionalidades pertencentes a outras equipas externas;
- **Countly** – Componente que fornece ao componente da aplicação, os dados necessários para ser possível a implementação das *mobile analytics*.

A aplicação, **Mobile App**, comunica com o repositório através do componente **BFF** (BackEnd for FrontEnd) **API**, que por sua vez comunica com o componente **Engine API** que é desenvolvida por uma equipa externa à empresa.

Em relação ao componente do **Countly**, esta comunicação acontece através da instalação do SDK do Countly no projeto da aplicação, **Mobile App**.

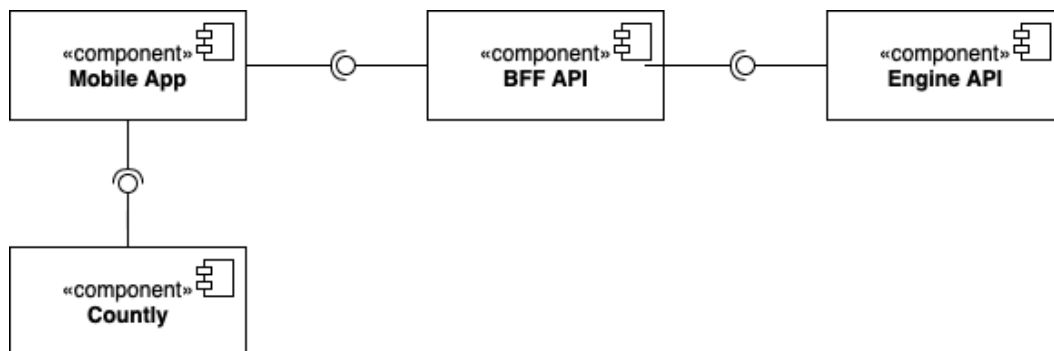


Figura 15 - Diagrama de Componentes

### 4.2.3 Mobile App

Dado que o componente mais importante no contexto deste trabalho é a **Mobile App**, nesta subsecção será apresentada a arquitetura lógica da mesma.

O diagrama de componentes da Mobile App, apresentado na Figura 16, divide-se essencialmente em:

- **UI** – Responsável por gerir a apresentação dos ecrãs da *app* e a interação com o utilizador;
- **State** – Responsável por gerir a comunicação entre o **Bloc** e a **UI**;
- **Bloc** – Responsável por gerir toda a informação que é apresentada na **UI**;
- **Event** - Responsável por gerir a comunicação entre a **UI** e o **Bloc**;

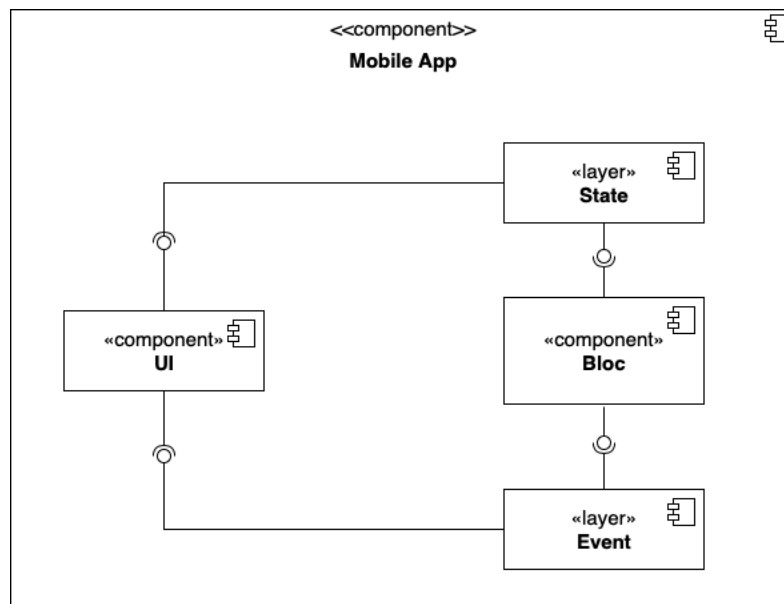


Figura 16 - Diagrama de Componentes Mobile App

Dada a existência destes componentes, todas as funcionalidades presentes na *app* irão ter um **Bloc** por cada uma delas. Assim sendo, foi adicionado o sufixo 'Bloc' a cada uma das classes necessárias para desenvolver estas funcionalidades. Por exemplo, para a funcionalidade do perfil do utilizador, chamada **ProfileCard**, o seu **Bloc** irá-se designar **ProfileCardBloc**.

#### 4.2.4 Interface Gráfica

A UI definida para a **Mobile App** foi desenhada pela *designer* pertencente à equipa de desenvolvimento, com apoio em *mock-ups* que eram depois passadas aos *developers* da aplicação.

Nesta subsecção serão apresentados *mock-ups* de algumas páginas da aplicação desenvolvida, como:

- Página principal;
- Carteira do utilizador;
- Atividade do utilizador;
- Detalhe de uma atividade.

Algumas partes das imagens irão estar ocultadas com retângulos laranja, por questões de confidencialidade.

### **Página principal**

Nesta página (Figura 17 e 18) estão representados os UC4: Consulta de Desafio para Ganhar Pontos e UC5: Consulta de Prêmios, e são mostradas todas as funcionalidades que estão disponíveis para o utilizador: no *header* está a carteira do utilizador, que pode depois ser acedida para mais detalhes; uma secção de atividade com um pequeno gráfico de atividade de pontos do utilizador na semana corrente, que pode ser expandida para uma página própria com toda a atividade do utilizador; uma secção de *challenge* com um cartão a indicar o progresso do utilizador no mês corrente; e uma secção com um catálogo de prémios que o utilizador pode trocar por pontos.

### **Carteira do utilizador**

Nesta página (Figura 19), está representado o UC2: Consulta de Carteira, onde o utilizador pode consultar os carros que tem mapeados na aplicação, os pontos que tem disponíveis na sua carteira, assim como pontos pendentes (ainda para serem creditados na carteira) e os pontos que irão expirar no futuro.

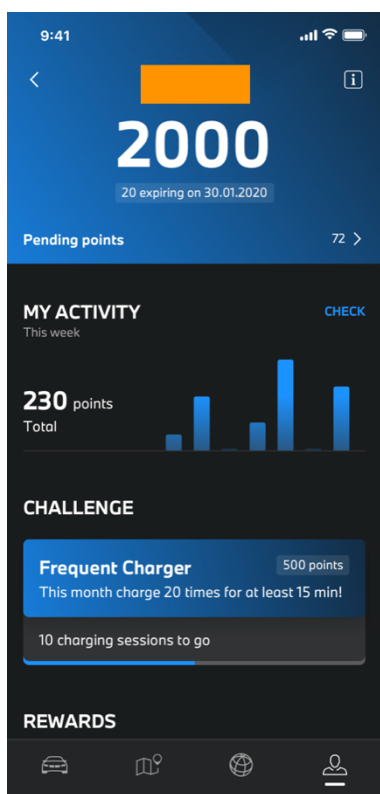


Figura 17 - Página principal (Parte 1)

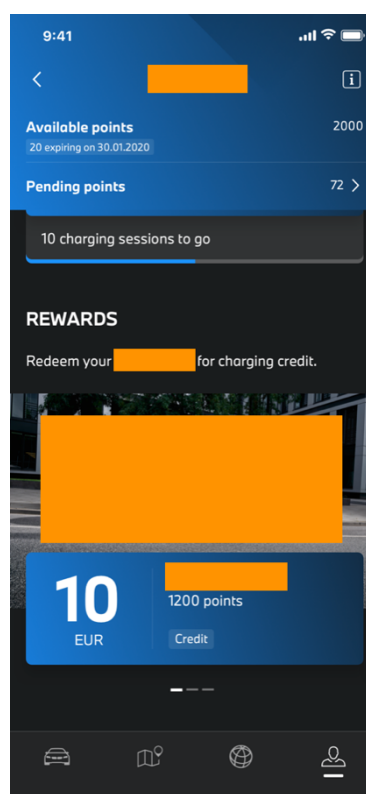


Figura 18 - Página principal (Parte 2)

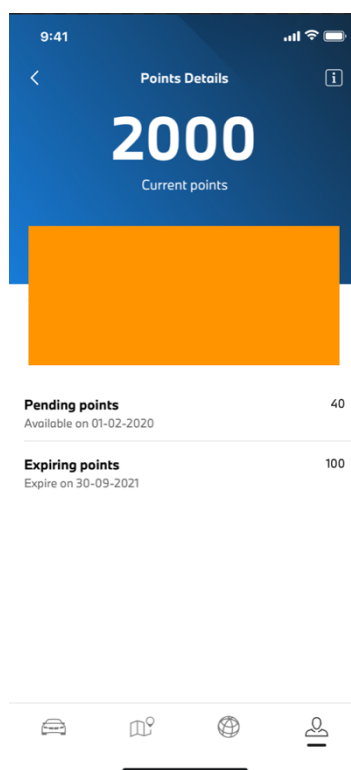


Figura 19 – Página Carteira do utilizador

### **Atividade do utilizador**

Esta página (Figura 20), para além da página principal, é a página com mais informação da *feature*, onde está representado o UC3: Consulta de Histórico de Atividade.

Aqui, é possível ver um gráfico *scrollable* sobre a atividade de pontos ganhos ao longo do tempo, num *range* de datas de uma semana.

Além deste gráfico, na secção seguinte do ecrã tem o histórico de utilizador, que apresenta toda a sua atividade dentro da *feature*, como: pontos ganhos por conduzir em modo elétrico, por carregamentos, por completar o *challenge* mensal ou por bónus oferecido; e também pontos perdidos pela troca por prémios.

### **Detalhe de uma atividade**

Neste último *mock-up* (Figura 21), é apresentado um exemplo de uma página de detalhe de atividade, neste caso, de uma viagem feita em modo elétrico.

Nesta página é possível ver a quantidade de pontos ganha com esta viagem, assim como a data em que foi feita, a distância conduzida (tanto em modo elétrico como em modo normal, se for o caso) e o veículo utilizado.

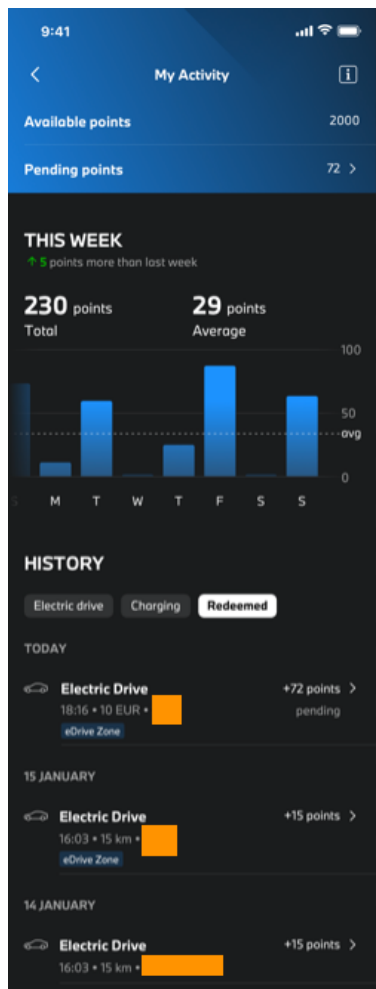


Figura 20 - Página de Atividade do Utilizador

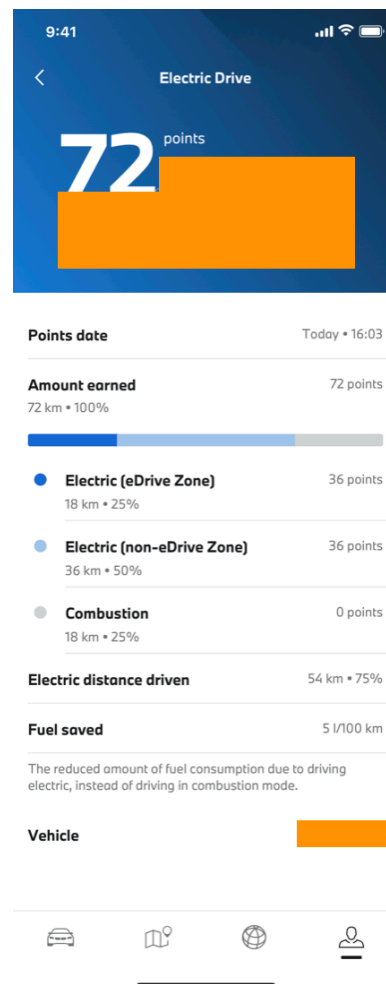


Figura 21 - Página de detalhe de uma atividade

#### 4.2.5 Desenho das Funcionalidades

Nesta subsecção irão ser descritas como serão implementadas as funcionalidades **UC1: Adesão ao Programa de Pontos**, **UC5: Consulta de Prémios** e **UC6: Resgatar Prémios**. Estas foram escolhidas por se entender que são as mais relevantes para a *feature* Programa de Pontos.

A descrição destas será feita com recurso a diagramas de sequência, que têm como objetivo ilustrar como os componentes necessários para realizar um UC interagem entre si.

### UC1: Adesão ao Programa de Pontos

Na Figura 22 está representado o diagrama de sequência do UC1. O processo de adesão ao programa começa pela interação do utilizador com a *navigation tab* da *app*, selecionando a *tab* do Perfil do utilizador.

Após isto, o cartão que permite a entrada no programa irá fazer *loading* enquanto a sua informação está a ser procurada através do método **getCardInformation()**.

Depois de se obter esta informação, o cartão irá comportar-se consoante. No caso do utilizador já ter aderido ao programa, irá mostrar informação sobre os pontos disponíveis e os seus carros. Se não tiver aderido, aparecerá um texto informando o utilizador para carregar no cartão para aderir. Quando o utilizador interagir com o cartão, irá ser apresentado um ecrã de *onboarding* para familiarizar o utilizador com o programa e um botão para aderir.

Depois do utilizador pressionar o botão, irá ser despoletado o método **optIn()**. Depois desta chamada terminal, o utilizador será redirecionado para o ecrã principal do programa, em caso de sucesso.

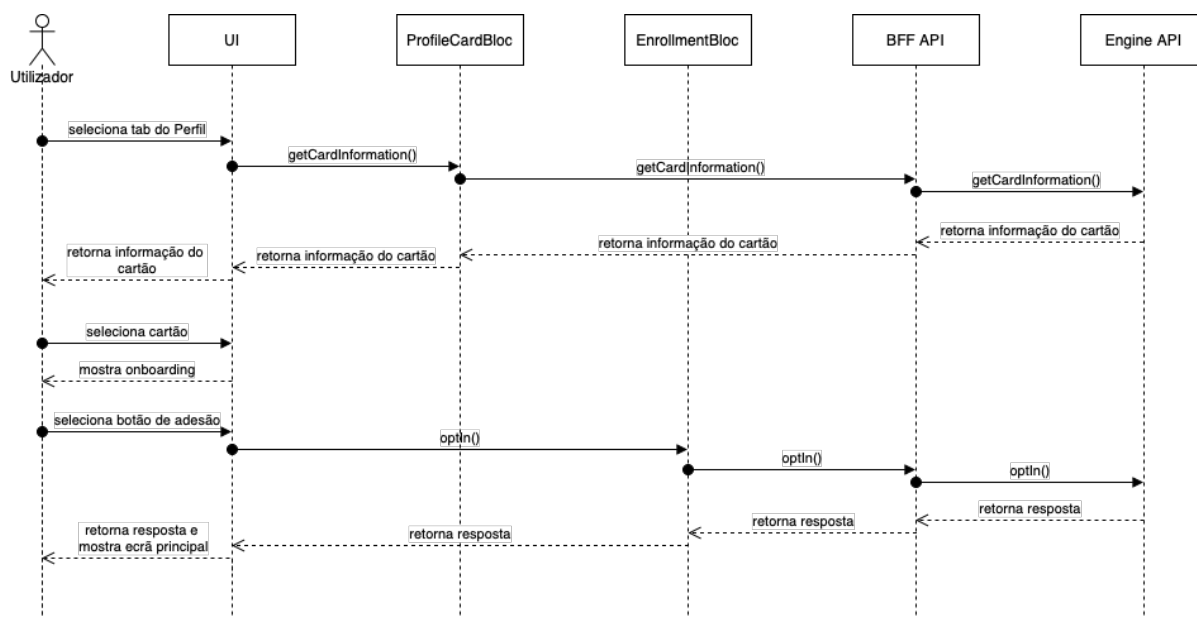


Figura 22 - Diagrama de Sequência UC1

## UC5: Consulta de Prêmios

Na Figura 23 está representado o diagrama de sequência do UC5. O processo começa pela interação do utilizador com a página principal. Quando este entra na página principal da *feature* é chamado o método **getRewardsCarousel()** que irá retornar um carrossel com os três primeiros prémios disponíveis para o utilizador resgatar. O utilizador consulta estes prémios, e carrega no botão “Mais” que desencadeia o método **getRewardsCatalog()** que irá retornar todos os prémios disponíveis existentes para o utilizador. Depois deste pedido terminar, irá ser apresentado um novo ecrã ao utilizador com um catálogo de todos os prémios, organizados por parceiro.

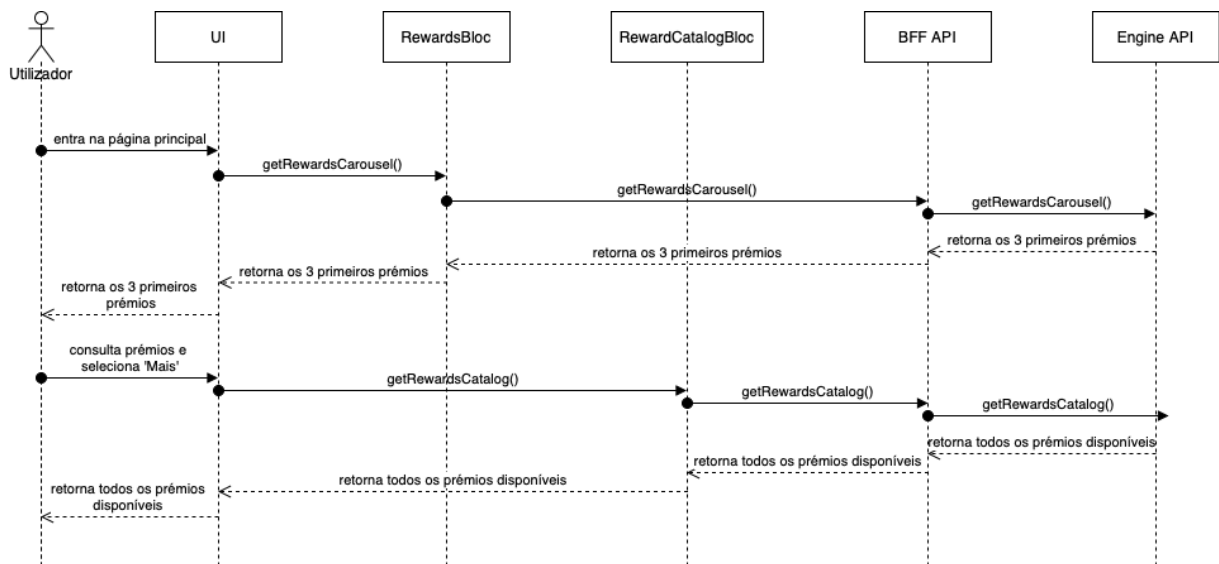


Figura 23 - Diagrama de Sequência UC5

## UC6: Resgatar Prêmios

Na Figura 24 está representado o diagrama de sequência do UC6. O processo começa pela interação do utilizador com o cartão do prémio que este pretende redimir. Quando este seleciona o prémio pretendido é mostrado um ecrã de detalhe com informação acerca deste, como: os pontos disponíveis na carteira; os pontos necessários para redimir o prémio; o parceiro pertencente a este prémio e o valor monetário associado ao prémio. Depois do utilizador consultar esta informação, pressiona o botão de “Resgatar prémio” que abre uma janela de *pop-up* questionando o utilizador se tem a certeza de que quer resgatar o prémio. O utilizador carrega de novo no botão “Resgatar prémio” que despoleta o serviço através do método **redeemReward(rewardID, partnerID)**. Após a resposta do serviço, é apresentado ao utilizador uma *snack-bar* a informar o sucesso da operação.

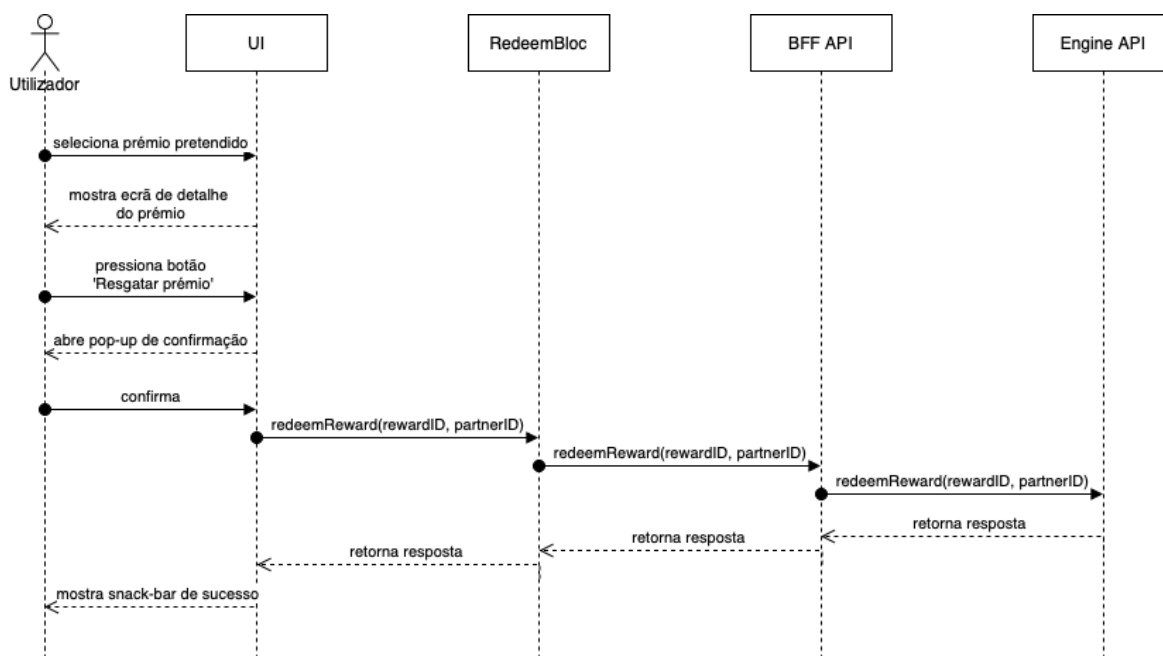


Figura 24 - Diagrama de Sequência UC6

### **4.3 Sumário**

Foi apresentada a Análise e Design da arquitetura do projeto com apoio a diagramas de UML, assim como os seus requisitos funcionais e não funcionais de forma a perceber melhor o intuito do projeto. Além disto, foram também apresentados alguns ecrãs *mock-up* da *app* a desenvolver.



# 5 Implementação

Tendo sido apresentadas as tecnologias a utilizar, assim como a análise e o desenho, é neste ponto onde será construída a solução que será submetida a testes de forma a validar, ou não, a análise efetuada ao problema em estudo.

Neste capítulo será abordado o **Outcome 5: Construção da Solução**.

## 5.1 *Guidelines* de Implementação

Para o desenvolvimento das *mobile analytics* foram elaboradas algumas *guidelines* de forma a uniformizar a implementação destas ao longo de toda a *app*.

Foram definidos alguns princípios para orientar a implementação:

1. Pensar nos ‘caminhos (sequências para realizar uma operação) críticos do utilizador’ da *feature* (ex: processo de registo ou de *onboarding*).
  - Cada atividade dentro de cada caminho deve ter o seu próprio evento;
  - Cada ecrã principal deve ter a sua própria vista.
2. Todas as ações distintas do utilizador devem ser novos eventos (ex: cliques).
  - Incluir segmentos personalizados para descrever atributos de um determinado evento.
3. Todos os ecrãs principais devem ser novas vistas.
  - Certificar que cada ecrã relevante (especialmente o primeiro de cada *feature*) obtém uma nova vista para fornecer uma visão completa do tempo correto gasto por ecrã.

### 5.1.1 Customização de Eventos

Para a criação de eventos personalizados para cada equipa e de forma a uniformizar todo o processo, foram definidas as seguintes regras para cada um dos atributos dos eventos.

#### Categoria do Evento

Utilizar grupos ou categorias para distinguir partes lógicas da aplicação. Os grupos de eventos ajudam a priorizar componentes chave da aplicação e podem ajudar na identificação de eventos.

#### Detalhes dos Eventos:

- Nome do evento / Chave: como o evento irá aparecer no Countly. O grupo dos eventos pode ser adicionado ao nome para ajudar na análise.
- Descrição do evento: Explicação de como o evento é desencadeado e gravado (fornecer contexto para membros da equipa)
- Prioridade: P1 (Alto), P2 (Médio), P3 (Baixo)
- Contagem: 1 como *default*
- Duração: Pode ser utilizada se necessário
- Segmentos personalizados: Pode-se introduzir segmentos personalizados se se precisar de segmentos específicos para os eventos. Usar apenas quando os segmentos *standard* (Tabelas 11 e 12) não forem suficientes, da seguinte forma:
  - Nome do segmento: como a propriedade vai aparecer no Countly;
  - Valor do segmento: valores potenciais que a propriedade pode ter.

Tabela 11 - Segmentos *Standart* da *App*

| SEGMENTOS DA APP |                                 |
|------------------|---------------------------------|
| Nome no Countly  | Segmento                        |
| Country          | País da localização por IP      |
| Language         | Linguagem definida no telemóvel |
| App_brand        | Nome da <i>app</i>              |

| SEGMENTOS DA APP |                                    |
|------------------|------------------------------------|
| App version      | Versão <i>client</i> da <i>app</i> |
| Device           | Dispositivo                        |
| OS               | Sistema operativo                  |

Tabela 12 - Segmentos *Standart* do Utilizador

| SEGMENTOS DO UTILIZADOR       |  |
|-------------------------------|--|
| Nome no Countly               | Segmento                               |
| ID                            | ID do utilizador                       |
| permissions_push_notification | Permissão para notificação <i>push</i> |
| permissions_location          | Permissão de localização               |
| permissions_calendar          | Permissão de calendário                |
| permissions_photos            | Permissão de fotos                     |
| permissions_camera            | Permissão de camera                    |

### Detalhes do Nome do Evento

O nome do evento é dividido em quatro partes, separadas por '-':

- Nome da *feature*;
- Nome da página (ecrã);
- Nome da etiqueta do objeto;
- Ação do utilizador ou resultado

Observações a ter em conta na criação do nome do evento:

- Apontar para no máximo 30 caracteres;
- Remover espaços em branco e caracteres especiais;
- Nome do ecrã / Etiqueta do objeto:
  - ◇ Usar um nome facilmente compreensível para descrever o ecrã/objeto em causa;
- Ação do utilizador ou resultado:

- ◇ As ações devem ser: *Tap, Swipe, Longpress*
- ◇ Os resultados devem ser: *Success* ou *Error*, e onde for aplicável, *Yes* ou *No*

Na Tabela 13, é apresentado um exemplo do que foi explicado anteriormente.

Tabela 13 - Exemplo de Evento

| <b>Categoria do evento</b> | <b>Nome do evento / chave</b> | <b>Descrição do evento</b> | <b>Contagem</b> | <b>Duração</b>  | <b>Prioridade</b> |
|----------------------------|-------------------------------|----------------------------|-----------------|---|-------------------|
| <i>Button pressed</i>      | Send2Car-TripCard-Send-Tap    | <i>Send Trip Card</i>      | 1               | Quantos segundos desde que abriu a página, demorou a carregar no botão? | <i>High</i>       |

## 5.2 Implementação de *Analytics*

Para a solução apresentada nesta dissertação foi utilizado o Countly como ferramenta de medição de *mobile analytics*.

Para tal, foi necessário em primeiro lugar importar o SDK do Countly como apresentado no Excerto de Código 1, no ficheiro de dependências do projeto:

```

1 dependencies:
2   countly_flutter:
3     git:
4       url:          "git@code.connected.XXX:mobile20/countly-
5         flutter.git"
6       ref: master

```

Excerto de Código 1 - Código para adicionar SDK Countly

Com o SDK integrado, é possível agora aceder a todos os seus métodos e propriedades. Por motivos de confidencialidade, vai-se supor que a *feature* a desenvolver se chama 'PointsProgram'.

Com isto, foi definido um enumerável **PointsProgramAnalyticsEvent** (Excerto de Código 2) que irá ter todos os eventos que irão ser criados.

```
1  enum PointsProgramAnalyticsEvent {
2      pointsProgramCardTappedToOnboarding,
3      pointsProgramCardTappedToMainPage,
4      pointsProgramTappedToWalletDetails,
5      pointsProgramTappedToMyActivity,
6      pointsProgramWeeklyChartSwiped,
7      pointsProgramCardLoadFailed,
8      pointsProgramOptedInFailure,
9      pointsProgramRedeemStarted,
10     pointsProgramRedeemSuccess,
11     pointsProgramRedeemFailure,
12     pointsProgramHistoryScrolledDown,
13     pointsProgramRewardsCatalogOpened,
14     pointsProgramVoucherShared,
15     ...
16 }
```

Excerto de Código 2 - Enumerável PointsProgramAnalyticsEvent

De seguida, foi definida uma classe **PointsProgramAnalytics** (Excerto de Código 3) com a responsabilidade de criar os eventos necessários com a ajuda do enumerável em cima, que irão ser recebidos e guardados pelo County.

```

1 class PointsProgramAnalytics {
2     static const Map events = {
3     PointsProgramAnalyticsEvent.PointsProgramCardTappedToOnboarding:
4         'PointsProgram-Onboarding-Card-Tapped',
5     PointsProgramAnalyticsEvent.PointsProgramCardTappedToMainPage:
6         'PointsProgram-MainScreen-Card-Tapped',
7     PointsProgramAnalyticsEvent.PointsProgramTappedToWalletDetails:
8         'PointsProgram-Wallet-Details-Tapped',
9     PointsProgramAnalyticsEvent.PointsProgramTappedToMyActivity:
10        'PointsProgram-MyActivity-Tapped',
11    PointsProgramAnalyticsEvent.PointsProgramWeeklyChartSwiped:
12        'PointsProgram-WeeklyChart-Swiped',
13    PointsProgramAnalyticsEvent.PointsProgramCardLoadFailed:
14        'PointsProgram-Card-Load-Failed',
15    PointsProgramAnalyticsEvent.PointsProgramOptedInFailure:
16        'PointsProgram-OptedIn-Failure',
17    PointsProgramAnalyticsEvent.PointsProgramRedeemStarted:
18        'PointsProgram-Redeem-Started',
19    PointsProgramAnalyticsEvent.PointsProgramRedeemSuccess:
20        'PointsProgram-Redeem-Success',
21    PointsProgramAnalyticsEvent.PointsProgramRedeemFailure:
22        'PointsProgram-Redeem-Failure',
23    PointsProgramAnalyticsEvent.PointsProgramHistoryScrolledDown:
24        'PointsProgram-History-Scrolled-Down',
25    PointsProgramAnalyticsEvent.PointsProgramRewardsCatalogOpened:
26        'PointsProgram-Rewards-Catalog-Opened',
27    PointsProgramAnalyticsEvent.PointsProgramVoucherShared:
28        'PointsProgram-Vouchers-Shared',
29        ...
30    };

```

### Excerto de Código 3 - Classe PointsProgramAnalytics

Com este enumerável e classe definidos, é possível então começar a implementar estes eventos nos ecrãs destinados.

De seguida, irão ser apresentados alguns exemplos desta implementação.

O primeiro exemplo (Excerto de Código 4) será de um evento de *Swipe* num gráfico existente no ecrã de detalhes da atividade do utilizador.

```
1  PageView.builder(  
2    scrollDirection: Axis.horizontal,  
3    reverse: true,  
4    onPageChanged: (index) {  
5      PlatformSdk.analytics?.recordEvent(  
6        eventName: PointsProgramAnalytics.events  
7        [PointsProgramAnalyticsEvent.pointsProgramWeeklyChartSwiped],  
8        eventDetails: {  
9          'numberOfWeeksBack': index.toString(),  
10       },  
11     );  
12  
13     BlocProvider.of<ActivityDetailsSummaryBloc>(context).add(  
14       ActivityDetailsNextWeekFetched(weekIndex: index),  
15     );  
16   },  
17 );
```

#### Excerto de Código 4 - Evento de Swipe

Neste caso, o widget que desenha o gráfico é um **PageView** que já contém um atributo **'onPageChanged'** que permite executar o código pretendido quando o utilizador muda de página com o movimento de *Swipe*.

Desta forma, na linha 5 pode-se comprovar que foi chamado o método **'recordEvent'** pertencente ao SDK das *analytics*, onde é enviado o nome do evento, **pointsProgramWeeklyChartSwiped** que foi definido no enumerável definido acima, e também os detalhes do evento, sendo que neste caso será um número indicando as semanas que o utilizador consultou para trás no gráfico.

Este evento servirá para perceber se o utilizador está a interagir com o gráfico, ou neste caso, para entender se o utilizador conseguiu alcançar o intuito do gráfico.

De seguida irá ser apresentado um exemplo de três eventos relacionados com um pedido feito ao repositório. Estes eventos ajudarão a perceber o tempo que o pedido demorou a ser efetuado. Neste caso desde que o utilizador clicou no ecrã para fazer uma redenção de pontos, até dar sucesso ou falha no serviço.

```
1  @override
2  Stream<PointsProgramRedeemState> mapEventToState(
3    PointsProgramRedeemEvent event,
4  ) async* {
5    if (event is PointsProgramRedeemStarted) {
6      yield PointsProgramRedeemInProgress(
7        reward: event.reward,
8      );
9
10     PlatformSdk.analytics?.recordEvent(
11       eventName: PointsProgramAnalytics
12 .events[PointsProgramAnalyticsEvent.pointsProgramRedeemStarted]
13       eventDetails: {
14         'partnerName': 'Charging',
15       },
16     );
17
18     try {
19       final redeem = await _repository.redeemReward(
20         rewardId: event.reward.id,
21         partnerId: event.reward.partnerId,
22       );
23
24       PlatformSdk.analytics?.recordEvent(
25         eventName: PointsProgramAnalytics
26 .events[PointsProgramAnalyticsEvent.pointsProgramRedeemSuccess]
27         eventDetails: {
28           'partnerName': 'Charging',
29         },
30       );
31
32       yield PointsProgramRedeemSuccess(reward: redeem);
33     } on Object catch (_) {
```

```

34     PlatformSdk.analytics?.recordEvent(
35         eventName: PointsProgramAnalytics
36     .events[PointsProgramAnalyticsEvent.pointsProgramRedeemFailure]
37         eventDetails: {
38             'partnerName': 'Charging',
39         },
40     );
41     yield PointsProgramRedeemFailure();
42 }
43 }

```

#### Excerto de Código 5 - Eventos com pedido ao repositório

Na linha 10 pode-se observar a chamada para guardar o evento **pointsProgramRedeemStarted** que informa que o começo do pedido ao repositório, mostrado na linha 19. Após isto, se o serviço retornar sucesso, é gravado o evento **pointsProgramRedeemSuccess**, se não, será gravado o evento **pointsProgramRedeemFailure**.

Este evento servirá para tirar conclusões acerca de quantos utilizadores estão a redimir pontos, mas também o tempo que este pedido está a demorar a ter uma resposta, e se esta é bem-sucedida ou não.

De seguida, será apresentado na Figura 25 um exemplo de um evento a ser monitorizado, já na plataforma Countyly.

Este evento **'PointsProgram-MyActivity-Tapped'** é um dos eventos apresentados no Excerto de Código 3 e monitoriza o número de cliques dados para consultar a página de atividade do utilizador.

Como se pode observar, de 4 de Agosto a 4 de Setembro existiu um total de 18.099 cliques para consultar a atividade. Este evento pode ajudar a equipa de desenvolvimento a entender se o utilizador está a consultar a sua atividade, e se esta funcionalidade está a cumprir o seu intuito, ou se o *design* deste ecrã não é intuitivo o suficiente para o utilizador perceber a sua funcionalidade.



Figura 25 - Evento de Tap no County

### 5.3 Sumário

Neste capítulo foi descrita, numa abordagem técnica, a implementação das *mobile analytics*. Foram apresentadas as *guidelines* de implementação das *analytics* de forma a uniformizar toda a sua implementação ao longo da *app*. De seguida foram apresentados excertos de código de forma a compreender como implementar as *analytics* na *app*, desde a instalação do Countly até à adição dos eventos a ser monitorizados.

## 6 Avaliação e Resultados

Este capítulo irá apresentar todo o processo de testes e avaliação durante a presente dissertação, apresentando as métricas a serem utilizadas, descrevendo as hipóteses, bem como o processo e os testes utilizados para testar essas hipóteses.

Irão também ser apresentados os resultados obtidos depois da implementação das *mobile analytics* e feita uma comparação com os dados existentes da antiga *app*.

Este capítulo tem como objetivo dar resposta ao **Outcome 6: Avaliação da Solução**.

No final deste capítulo será dada a resposta às seguintes questões:

**Que métodos e parâmetros irão ser usados para avaliar a solução?**

**Que procedimentos/ferramentas irão ser usadas para avaliar esses parâmetros?**

**A adoção de *mobile analytics* melhorou o funcionamento/monitorização da *app*?**

### 6.1 Metodologia de Avaliação

A Metodologia de Avaliação é um plano estruturado para ajudar outras pessoas a entender quais são as diversas fases necessárias para fazer uma avaliação válida e de qualidade. Esta metodologia consiste nos seguintes quatro passos principais [37]:

- Definir os parâmetros da avaliação - determinar a necessidade da avaliação; o uso para os resultados da avaliação; o que deve ser relatado e por quem; o que o avaliador precisa de relatar e escolher diretrizes a seguir para implementar a avaliação;
- Planear os métodos utilizados para a avaliação - escolher os critérios para usar na avaliação; determinar as evidências que serão recolhidas para cada critério escolhido; a amostra que será usada, se apropriado; maneiras de recolher as provas e estabelecer um plano para recolher as provas;

- Estabelecer padrões e recolher provas - determinadas as provas que serão recolhidas, deve ser definida uma escala para descrever como a qualidade é julgada. Esta escala pode ser uma escala numérica, uma rubrica, ou uma descrição. Após a escala e as decisões a serem tomadas, devem-se estabelecer padrões para as decisões. Essas normas definem "qualidade" com base no que será relatado pelo avaliador;
- Relatório e tomada de decisões – necessário fazer um relatório de avaliação, para que possam ser analisados os resultados em relação às normas que foram definidas.

Considerando a importância da solução e das pessoas interessadas na sua implementação, diversos métodos podem ser utilizados para avaliá-la.

Os fatores que serão utilizados para avaliar a solução são:

- Satisfação do Utilizador (pessoas envolvidas no processo) - o objetivo é perceber se o utilizador da *app* teve algum problema durante a utilização da mesma e se sentiu que teve uma boa experiência de utilização, de forma a ir de encontro aos objetivos definidos. Em relação à equipa, irá-se analisar se esta se sente confortável com a nova solução e se serve efetivamente para combater o problema identificado nesta dissertação.
- Detecção de *bugs* - indicador de qualidade, que servirá para perceber se a solução utilizada contribui ou não para a qualidade do código e para a diminuição de defeitos.

### **6.1.1 Satisfação do Utilizador**

Para medir a satisfação do utilizador, será realizado um questionário com diversas questões sobre usabilidade com o objetivo de entender se o utilizador sentiu algum tipo de impedimento/erro ao utilizar a *app* e se compreendem com clareza cada funcionalidade da mesma.

Para além disso, os utilizadores serão segmentados por idade e dispositivo utilizado. Com esta segmentação, é possível entender como a idade pode influenciar a sensação de desempenho e também através do dispositivo, perceber se existe algum grupo com mais tendência a ter erros de *performance/bugs*.

Além deste inquérito aos utilizadores/consumidores finais da *app* irá ser feito um outro questionário às pessoas envolvidas neste processo de desenvolvimento: *developers; testers; product owner*, de forma a perceber se a introdução de *mobile analytics* ajudou de alguma forma a melhorar o desenvolvimento e a monitorização da *app*. Este grupo de pessoas está na linha da frente e pode dar um *feedback* importante e mais confiável.

Tendo em conta que o grupo de entrevistados tem funções diferentes na organização e nem todos eles têm conhecimento técnico sobre cada etapa do processo, espera-se que o inquérito tenha questões mais gerais e, ao mesmo tempo, contribua para uma avaliação confiável.

Espera-se que estes inquéritos sejam organizados com respostas de nível (0 a 5, por exemplo) para permitir uma análise mais objetiva do fator em questão. Os resultados do inquérito serão analisados e contribuirão para a compreensão do *feedback* das partes interessadas no processo.

#### ***Testes estatísticos***

Para avaliar a solução pode-se optar pelo teste *One-Tailed Test* (paramétrico) ou *Two-Tailed Test* (não-paramétrico).

Antes de optar por um dos testes deve-se analisar o que se pretende testar. Há duas perguntas principais que se deve fazer [38]:

1. O resultado é maior (ou menor) que um determinado valor?
2. O resultado está dentro ou fora de um determinado intervalo de valores?

A primeira questão conduz a um teste paramétrico, enquanto que a segunda um teste não-paramétrico [38].

### ***Paramétrico***

Para testes paramétricos, existem dois tipos de hipóteses: nulas e alternativas [39].

- $H_0$ : A hipótese nula é a hipótese considerada que o analista espera rejeitar.
- $H_1$ : A hipótese alternativa é a hipótese considerada que é apoiada por rejeitar a hipótese nula.

Um teste paramétrico resulta de uma hipótese alternativa que especifica uma direção, ou seja, quando a hipótese alternativa afirma que o parâmetro é de fato maior ou menor do que o valor especificado na hipótese nula. Um teste paramétrico pode ser à esquerda ou à direita.

Um teste à esquerda é utilizado quando a hipótese alternativa afirma que o valor real do parâmetro especificado na hipótese nula é menor do que a hipótese nula reivindicada.

Um teste à direita é usado quando a hipótese alternativa afirma que o valor real do parâmetro especificado na hipótese nula é maior do que as hipóteses de hipótese nula [39].

### ***Não-Paramétrico***

Um teste não-paramétrico resulta de uma hipótese alternativa que não especifica uma direção, ou seja, quando a hipótese alternativa afirma que a hipótese nula está errada.

A principal diferença entre testes paramétricos e não-paramétricos é que os testes paramétricos só terão uma região crítica, enquanto os testes não-paramétricos terão duas regiões críticas. Se for preciso um intervalo de confiança tem de se fazer alguns ajustes ao usar um teste não-paramétrico.

O intervalo de confiança deve permanecer num tamanho constante, portanto, se se está a realizar um teste não-paramétrico, pois há duas vezes mais regiões críticas, então essas regiões críticas devem ser metade do tamanho [39].

As respostas ao inquérito, organizadas por níveis (0 a 5), representam uma variável independente esperada para seguir uma distribuição normal e visto que o que se pretende avaliar é se a nova solução é melhor que a solução anterior, o teste estatístico apropriado é o teste paramétrico.

É necessário a definição de um grau de confiança para a aplicação deste tipo de testes sendo comum a definição de um intervalo de confiança de 95%, que reflete um nível de significância de 0.05 ( $\alpha$ ) [40].

### **6.1.2 Detecção de *bugs***

É expectável uma diminuição do número de *bugs* com a introdução da nova solução, sendo que a ferramenta de *mobile analytics* ajuda a detetar a presença de *bugs*.

Será calculado o número de *bugs* detetados pela equipa de *developers* e *testers* na solução anterior e, na nova solução, o número de *bugs* detetados pela ferramenta de *analytics*. Esses números serão analisados e comparados, sendo, portanto, uma análise quantitativa. As hipóteses para testar este fator incluem a hipótese de mais *bugs* serem detetados na antiga solução em comparação com a nova.

## **6.2 Avaliação dos resultados**

Esta secção irá apresentar os resultados obtidos deste estudo de caso de acordo com os fatores definidos anteriormente. Irá ser apresentada primeiramente uma comparação com os resultados obtidos através da implementação da solução, e serão também apresentados os resultados dos inquéritos de satisfação feitos aos inquiridos e suas conclusões.

## 6.2.1 Resultados da Implementação

Nesta subsecção irão ser apresentados resultados acerca das *mobile analytics* monitorizados na solução desenvolvida, e comparados com antigos dados existentes da *app* anterior, de forma a perceber se existiu alguma mudança significativa com esta implementação.

### Países/Regiões

Na Figura 26 estão representados os países em que a *app* desenvolvida está disponível. Como se pode comprovar pela Figura 26, a *app* desenvolvida encontra-se disponível em todo o mundo, como se pode observar pela cor azul, sendo que tem utilizadores em 222 países/regiões. A cor azul mais forte indica os países onde tem mais utilizadores, sendo estes a Alemanha e o Reino Unido.

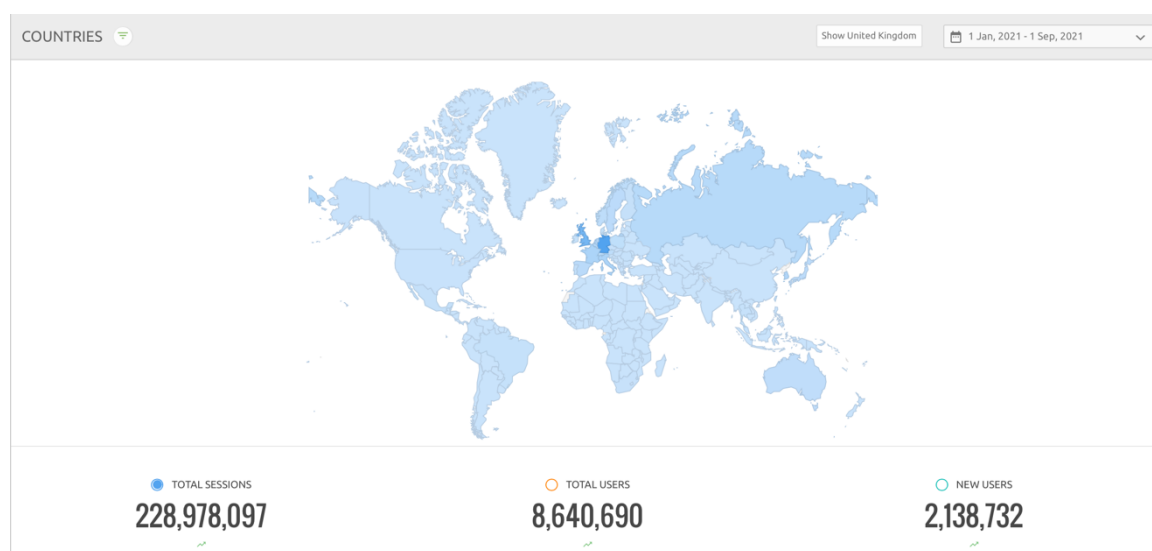


Figura 26 - Países da *app* desenvolvida

Na Tabela 14 estão representados os países/regiões em que a *app* antiga estava disponível.

Tabela 14 - Países/Regiões da *app* antiga

| <b>Países/Regiões</b> |                 |
|-----------------------|-----------------|
| <b>Europa</b>         |                 |
| Alemanha              | Áustria         |
| Bélgica               | Bulgária        |
| Croácia               | Chipre          |
| Dinamarca             | Espanha         |
| Eslováquia            | Eslovénia       |
| Estónia               | Finlândia       |
| França                | Grécia          |
| Holanda               | Hungria         |
| Irlanda               | Itália          |
| Letónia               | Lituânia        |
| Luxemburgo            | Malta           |
| Noruega               | Polónia         |
| Portugal              | República Checa |
| Reino Unido           | Roménia         |
| Rússia                | Suécia          |
| Suíça                 |                 |
| <b>África</b>         |                 |
| África do Sul         |                 |
| <b>América</b>        |                 |
| Brasil                | Canadá          |
| Estados Unidos        | México          |
| <b>Ásia</b>           |                 |
| China continental     |                 |

Por outro lado, como se observa na Tabela 14, a *app* antiga disponibilizava suporte apenas em 37 países, sendo que são os países com mais mercado para os utilizadores dos automóveis da empresa X.

## Utilizadores

Na Figura 27 está representado um gráfico com a crescente dos utilizadores da *app* ao longo do ano de 2021

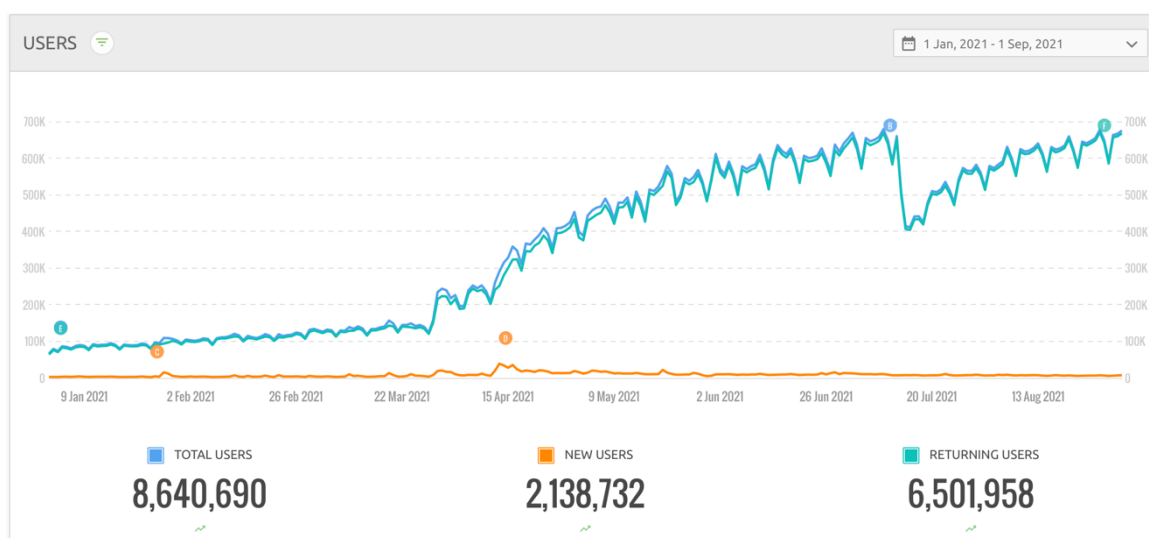


Figura 27 - Utilizadores da *app*

Como se pode observar, o total de utilizadores da *app* é de 8.640.690 sendo que 6.501.958 são utilizadores da *app* anterior. A *app* teve uma crescente de 2.138.732 utilizadores novos, o que é um indicador positivo acerca da nova *app*.

Na Figura 28 é ilustrado o mapa do mundo com o top de países com maior número de utilizadores da *app*, sendo que se observa que o mercado alemão é o que retém o maior número de utilizadores com 2.394.786. Outro ponto que se pode observar também é que a Coreia do Sul tem um número considerável de utilizadores (526.913) que não existiam na versão anterior da *app* pois esta não estava disponível nesse mercado.

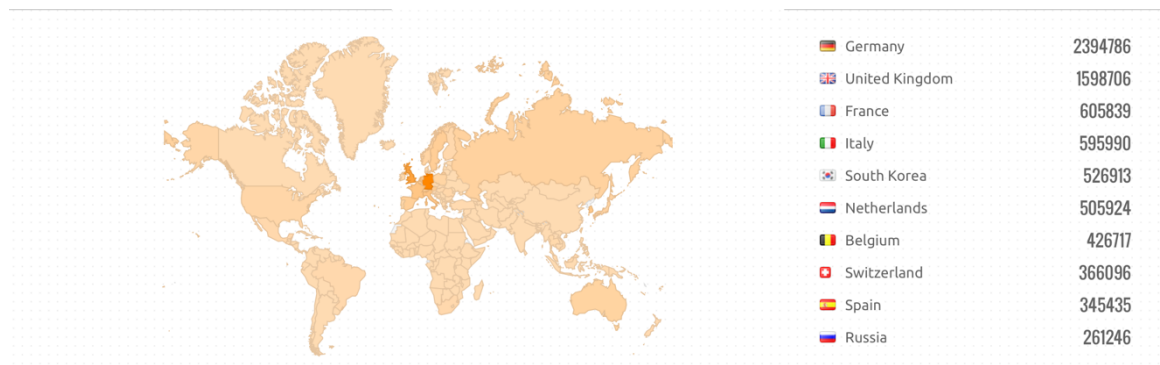


Figura 28 - Países com maior número de utilizadores

Outra métrica que foi monitorizada foi a quantidade de utilizadores que estão ativos no momento, assim como a sua média por hora, como se pode observar na Figura 29.

Os números de *Daily Active Users* devem estar sempre numa trajetória ascendente, caso contrário, é preciso utilizar ferramentas como *push notifications* para trazer os utilizadores de volta.

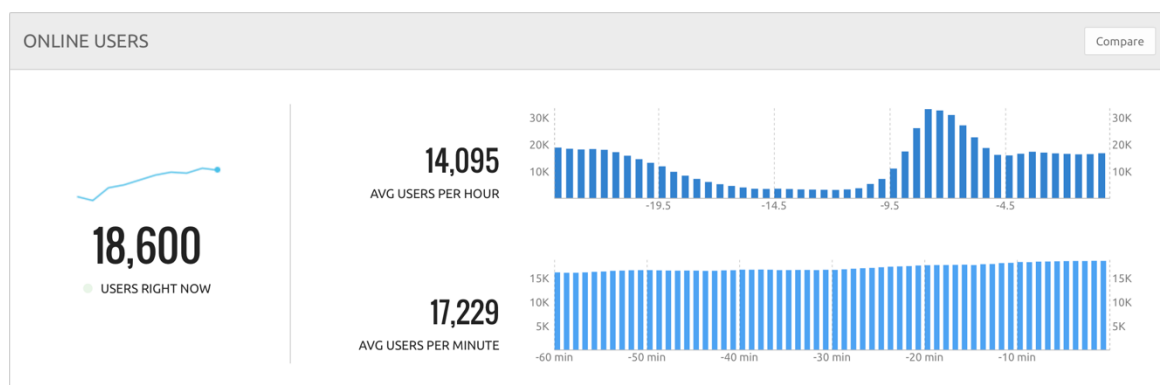


Figura 29 - Utilizadores ativos

Ainda acerca dos utilizadores, foi monitorizada uma importante métrica (Figura 30) para perceber qual a frequência de sessão dos utilizadores. Desta forma consegue-se perceber se os utilizadores estão a retornar à *app* com frequência, ou se só a utilizam esporadicamente.

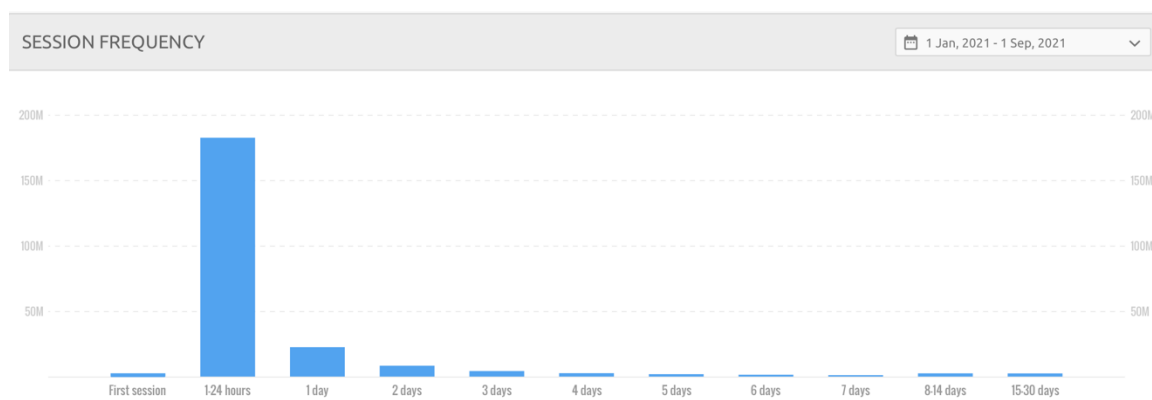


Figura 30 - Frequência de sessão dos utilizadores

Pode-se retirar da Figura 30 que a grande maioria dos utilizadores volta à *app* passado 1-24 horas, sendo que a quantidade de utilizadores que volta apenas passado 15-30 dias é muito baixa. Esta métrica é importante pois dá uma ideia do quão valiosa a *app* é para os seus utilizadores.

## Dispositivos

Na Tabela 15 está representado o número de dispositivos que estão a ser utilizados, divididos por tipo de dispositivo: telemóvel ou *tablet*.

Tabela 15 - Tipo de dispositivos

| Tipo de dispositivo | Sessões totais | Utilizadores totais | Novos utilizadores |
|---------------------|----------------|---------------------|--------------------|
| Telemóvel           | 227.350.218    | 8.725.577           | 2.084.936          |
| Tablet              | 621.726        | 256.026             | 15.928             |

Como se pode observar pela Tabela 15, a maior parte dos dispositivos utilizados são telemóveis, apesar de que os *tablets* também são usados e por isso, devem ser tomados em conta durante o desenvolvimento da *app*.

Além do tipo de dispositivo, também são monitorizadas métricas como o tipo de sistema operativo mais utilizada: iOS ou Android, representada na Figura 31, assim como os dispositivos mais utilizados na *app*, representados na Figura 32, resoluções de ecrãs mais utilizados e ainda o número da versão dos sistemas operativos mais utilizados.

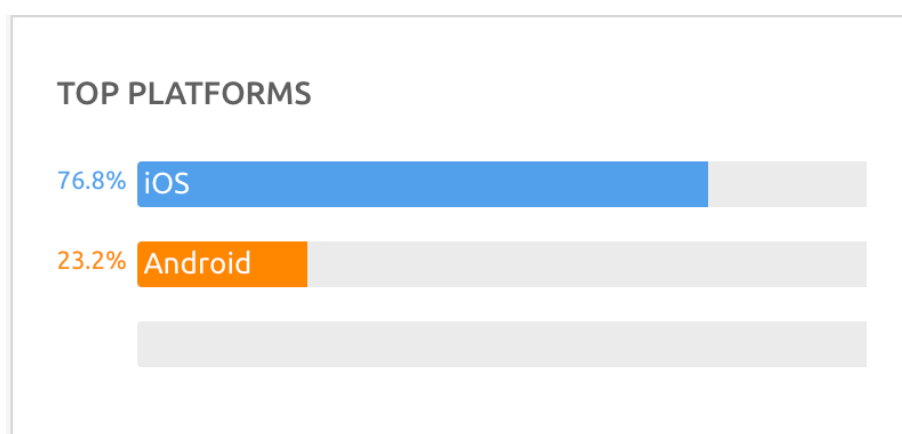


Figura 31 - Sistemas operativos utilizados

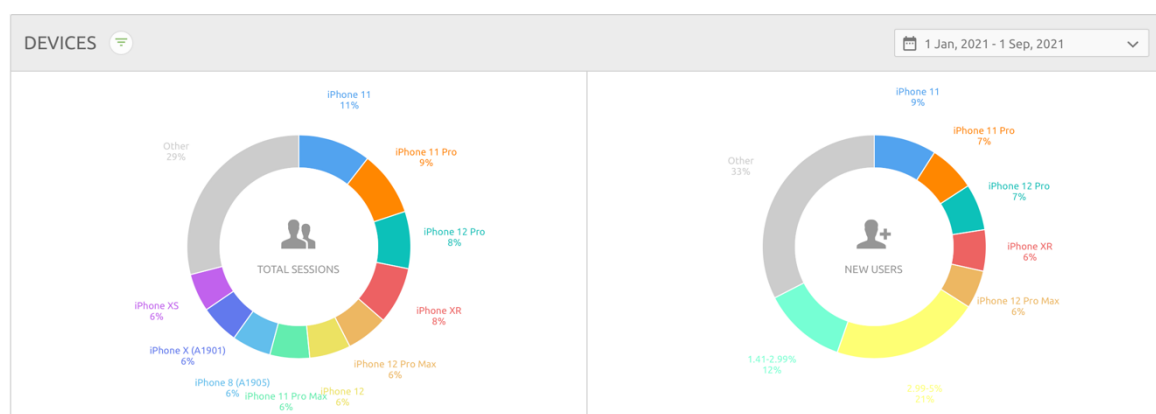


Figura 32 - Dispositivos mais utilizados

Pode-se concluir com apoio nas figuras 31 e 32, que grande parte dos utilizadores (76.8%) utiliza o sistema operativo iOS, enquanto apenas 23.2% utilizam Android. É importante conhecer o dispositivo a partir do qual a app é usada tanto para questões de marketing como para priorizar novas versões e ajudar os *developers* a focar-se mais nesta plataforma.

### Feedback de Utilizadores

Em relação ao feedback dos utilizadores irá ser utilizada a pontuação e *reviews* deixadas na Store de cada *app* para demonstrar o agrado destes.

Na Figura 33 está representada a pontuação atribuída pelos utilizadores à *app* antiga, que é de 1.9 em 5 estrelas. Esta pontuação é bastante má e comprova que os utilizadores não estavam satisfeitos com a *app*. De seguida irão ser também apresentados alguns comentários a esta mesma *app*, que irão ser traduzidos para português:



Figura 33 - Pontuação *app* antiga: 1.9

- “A aplicação e o serviço por trás dela são muito pobres. O *remote-control* (por exemplo, para destrancar portas) na sua maioria não funciona. Quando funciona é muito lento e ainda não atualiza corretamente o estado da aplicação. A localização do carro na *app* está geralmente horas ou dias desatualizada”.
- “Nenhuma informação ou *feedback* sobre o estado atual, ou seja, não pode ver se o carro está trancado ou não. Extremamente má qualidade, *bugs* em funcionalidades padrão que fazem constantemente ter que reiniciar a aplicação. No geral, muito pouco tempo de resposta para funções padrão, por exemplo, desbloquear ou iniciar o aquecedor, geralmente entre 30 e 90 segundos!”
- “A aplicação congela e não está em contacto com o carro.”

Na Figura 34 está representada a pontuação atribuída pelos utilizadores à *app* desenvolvida de novo, que é de 4.6 em 5 estrelas. Esta pontuação como se pode observar é boa e comprova que de facto a implementação de uma nova *app*, desenvolvida em torno do utilizador e da sua satisfação, medindo todos os fatores importantes para este, realmente suscitou uma diferente opinião e satisfação por parte dos utilizadores. De seguida irão ser também apresentados alguns comentários a esta *app*, que irão ser traduzidos para português:



Figura 34 - Pontuação *app* desenvolvida

- “É bom ver uma UI renovada e parece ter um desempenho melhor do que a *app* anterior.”
- “Esta nova atualização da *app* é muito superior à versão mais antiga, basicamente funciona e funciona bem.”
- “Embora esta aplicação faça coisas boas, e a interface do utilizador é intuitiva, não há uma maneira de ver se o carro está trancado ou desbloqueado. À parte disso, funciona muito bem, é uma pena que algumas das grandes funcionalidades da antiga aplicação tenham sido perdidas.”
- “A aplicação tem uma nova sensação mais rápida, mas ainda falta algumas funcionalidades a que estou habituado da anterior *app*.”
- “A aplicação é boa e é melhor do que a aplicação anterior, embora esteja a perder uma das minhas funcionalidades favoritas.”

Como se pode comprovar pelos comentários acima, os utilizadores no geral estão mais contentes com a *performance* e *design* da nova *app*, embora com alguns comentários menos positivos dado que a *app* antiga tinha bastantes mais funcionalidades que a nova *app*.

## 6.2.2 Satisfação do Utilizador

Nesta subsecção irão ser apresentados os resultados dos questionários feitos aos utilizadores da app e à equipa de desenvolvimento. Estes questionários foram feitos com ajuda da plataforma de questionários *Survicate* e encontram-se disponíveis nos **Anexos A e B**, respetivamente.

### **Questionário 1: Utilizador da app**

No total, 60 pessoas responderam a este questionário. Todas estas pessoas são de outra nacionalidade, daí o questionário se encontrar em inglês, e foram escolhidas ao acaso para testar a usabilidade da *app*.

No fim desses testes de usabilidade foi-lhes pedido para responder as seguintes perguntas, com respostas organizadas por níveis de 1 a 5, à exceção das duas primeiras:

- Questão 1: How old are you?
  - 18-25
  - 26-35
  - 36-50
  - 50+
- Questão 2: What is your operating system?
  - iOS
  - Android
- *Questão 3: How would you rate the ease of use of the app?*
- *Questão 4: How would you rate the stability of the app?*
- *Questão 5: How would you rate the look and feel of the app?*
- *Questão 6: How would you rate the overall quality of the app?*
- *Questão 7: Would you recommend this app to your friend or colleague?*

## Questão 1

Na Figura 35 é apresentada a matriz de respostas por parte dos inquiridos para a primeira questão, acerca da sua idade.

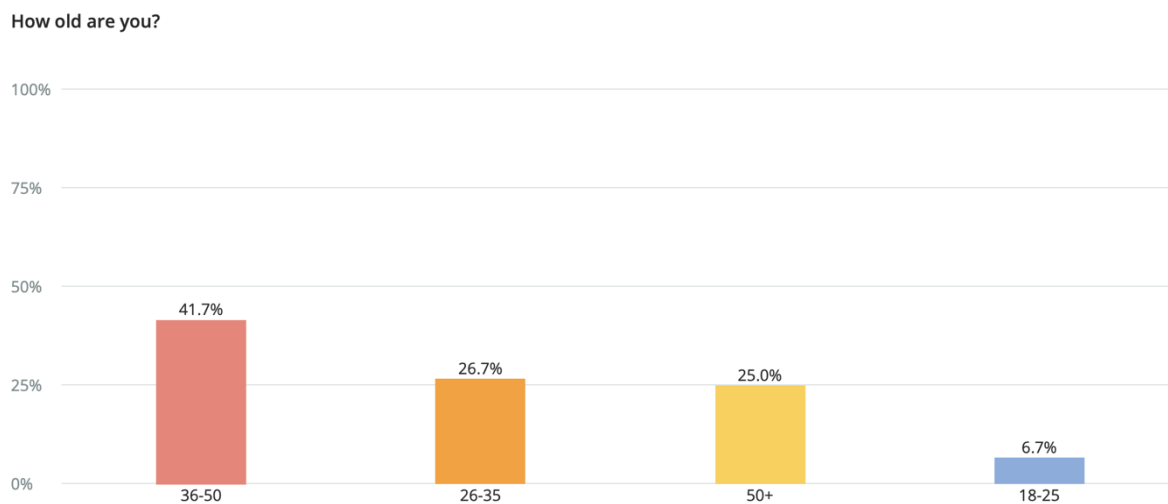


Figura 35 - Matriz Respostas Questão 1

## Questão 2

Na Figura 36 é apresentada a matriz de respostas por parte dos inquiridos para a segunda questão, acerca do seu sistema operativo.

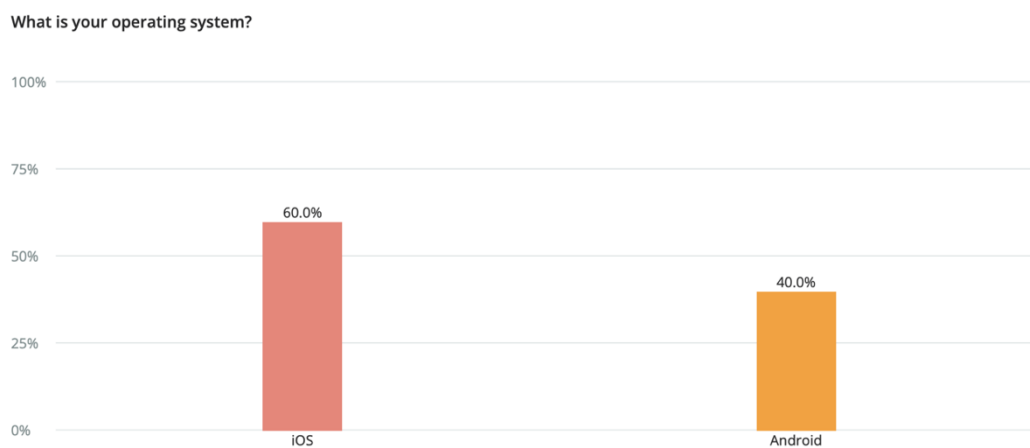


Figura 36 - Matriz Respostas Questão 2

Como se pode verificar pela Figura 35, a maior percentagem dos inquiridos (41.7%) pertence à faixa etária dos 36-50, enquanto as faixas dos 26-35 e 50+ se encontram muito equiparadas, a faixa dos 18-25 tem uma muito baixa percentagem de utilizadores (6.7%). Em relação ao sistema operativo mais utilizado, o iOS ganha ao Android com 60% para 40%, como se comprova na Figura 36.

### Questão 3

Na Figura 37 é apresentada a matriz de respostas por parte dos inquiridos para a terceira questão, acerca da facilidade de uso da *app*.

1. How would you rate the ease of use of the app?

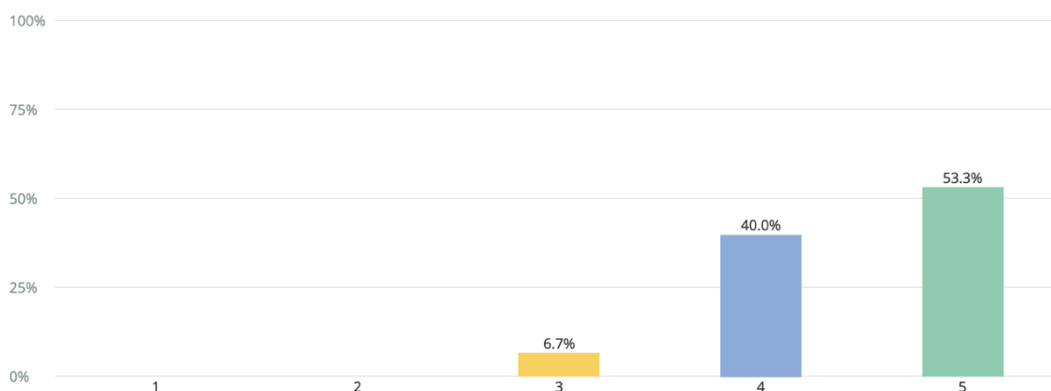


Figura 37 - Matriz Respostas Questão 3

A Figura 37 mostra que em termos de facilidade de utilização da *app*, os utilizadores consideraram-na bastante simples de utilizar, dado que a maioria destes votou no nível 4 e 5. Apesar disto, ainda assim, 6.7% dos inquiridos votaram o nível 3, o que comprova que nem toda a gente tem facilidade de uso a utilizar a *app* e que poderiam existir melhorias.

### Questão 4

Na Figura 38 é apresentada a matriz de respostas por parte dos inquiridos para a quarta questão, acerca da estabilidade da *app*.

2. How would you rate the stability of the app?

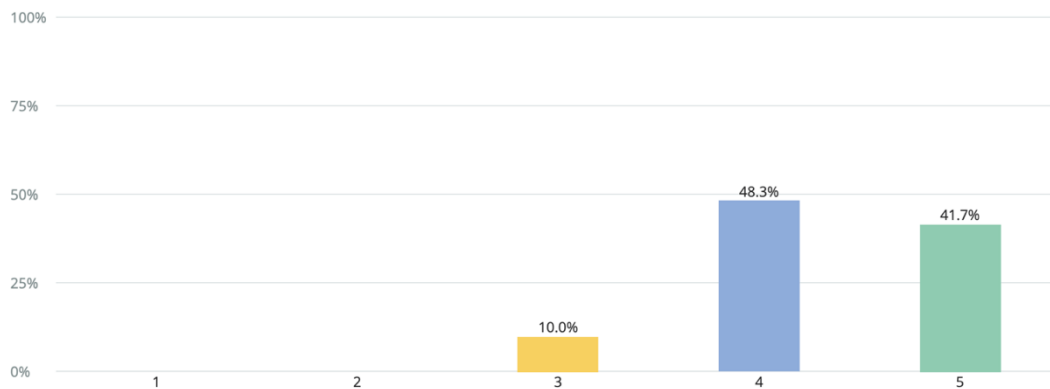


Figura 38 - Matriz Respostas Questão 4

A questão 4 remete para a estabilidade da *app*, sendo este um ponto bastante importante para os utilizadores. Como comprova a Figura 38, a maioria dos utilizadores como na questão anterior votou nos níveis 4 e 5, sendo que o nível 4 é o mais acentuado. Tal como na questão anterior, 10% dos inquiridos considerou o nível 3, o que indica que a *app* ainda tem alguma instabilidade e que com certeza os utilizadores já experienciaram algum tipo de erro ou *bug*.

### Questão 5

Na Figura 39 é apresentada a matriz de respostas por parte dos inquiridos para a quinta questão, acerca do aspeto e *design* da *app*.

3. How would you rate the look and feel of the app?

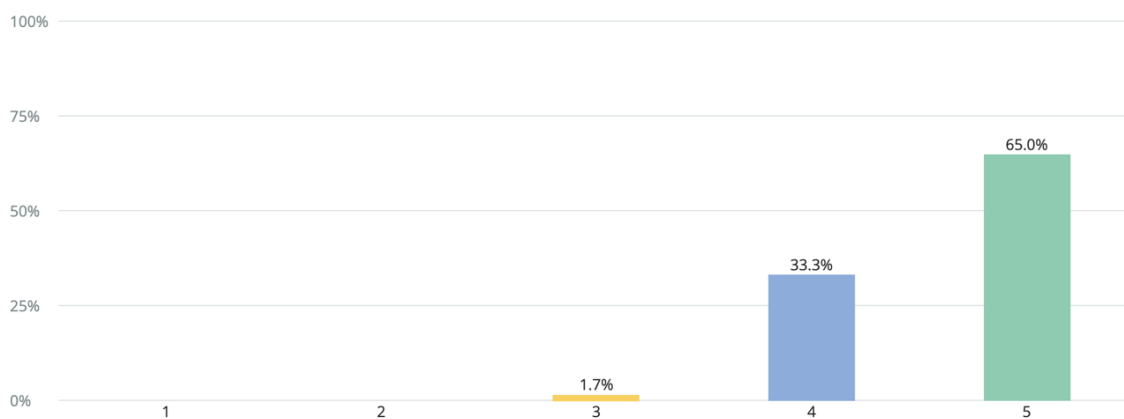


Figura 39 - Matriz Respostas Questão 5

### Questão 6

Na Figura 40 é apresentada a matriz de respostas por parte dos inquiridos para a sexta questão, acerca da qualidade geral da *app*.

4. How would you rate the overall quality of the app?

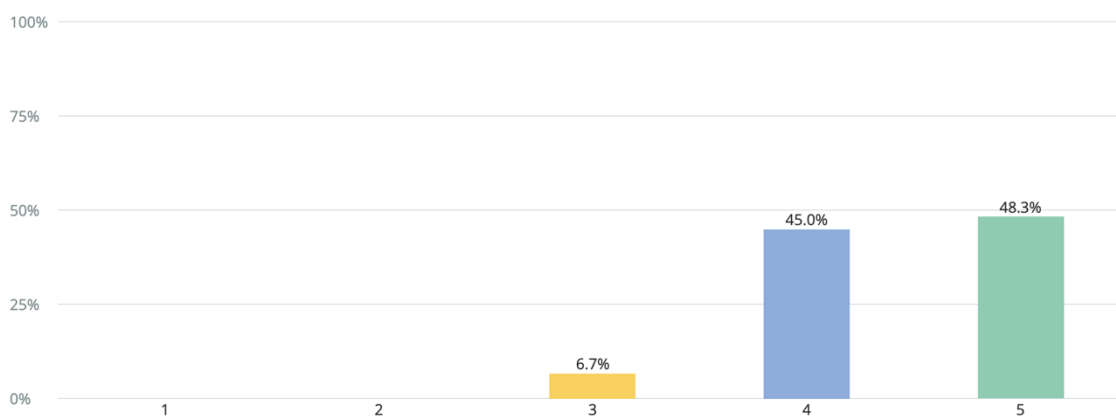


Figura 40 - Matriz Respostas Questão 6

Como se pode verificar pela Figura 39, a grande percentagem dos inquiridos (65%) considera o aspeto e *design* da *app* excelente, dado que votaram no nível 5 e apenas 1.7%

dos utilizadores votou no nível 3, o que é uma percentagem mínima, mas que indica que ainda assim existe pessoas a não gostar deste aspeto.

Em relação à qualidade geral da *app*, presente na Figura 40, 48.3% dos inquiridos votou no nível 5 e 45% no nível 4, o que comprova que os utilizadores estão satisfeitos com a qualidade que a *app* fornece.

### Questão 7

Na Figura 41 é apresentada a matriz de respostas por parte dos inquiridos para a sétima questão, acerca da recomendação que o utilizador daria da *app*.

5. Would you recommend this app to your friend or colleague?

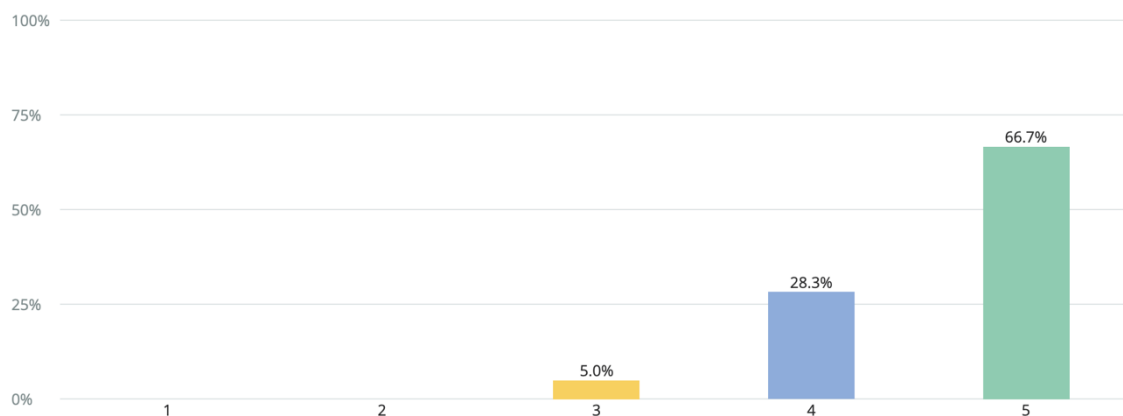


Figura 41 - Matriz Respostas Questão 7

Por fim, na Figura 41 está representada a matriz de respostas correspondente à questão se o utilizador recomendaria a *app* a algum amigo ou colega. A maioria dos inquiridos (66.7%) também respondeu o nível 5 o que indica que os utilizadores no geral gostam da *app* e a indicavam aos seus conhecidos para uso.

## Questionário 2: Equipa de Desenvolvimento

No total, 7 pessoas responderam ao questionário, sendo que todas elas responderam também a todas as questões. Perante este cenário, pode-se considerar que todas as amostras são quantitativas, independentes, mas não se pode assumir uma distribuição normal, sendo a amostra inferior a 30 resultados.

Para avaliar este indicador, foram utilizadas as seguintes questões do questionário realizado aos membros da equipa:

- Questão 1 - Em termos de dificuldade, como classificarias a aprendizagem de implementação de *mobile analytics* no projeto?
- Questão 2 - Sentes que a adoção de *mobile analytics* melhorou o processo de deteção de erros da *app*?
- Questão 3 - Sentes que desde a adoção de *mobile analytics*, têm diminuído o número de *bugs* detetados?
- Questão 4 - Sentes que a adoção de *mobile analytics* ajudou no desenvolvimento e conseqüente monitorização da *app*?

### Questão 1

Na Figura 42 é apresentada a matriz de respostas por parte dos inquiridos para a primeira questão.

|       |       |             |
|-------|-------|-------------|
| 1     | 71.4% | 5 responses |
| 2     | 28.6% | 2 responses |
| 3     | 0.0%  | 0 responses |
| 4     | 0.0%  | 0 responses |
| 5     | 0.0%  | 0 responses |
| Total |       | 7 responses |

Figura 42 - Matriz Respostas Questão 1

Como se pode observar na matriz, conclui-se que a maioria dos inquiridos (71.4%) não sentiram dificuldade na aprendizagem de *mobile analytics*. Os 28.6% restantes responderam opção 2.

Como já referido anteriormente, apenas 7 respostas compõem a lista de resultados, logo não se pode assumir que se está perante uma distribuição normal. Irá ser utilizado um teste *Shapiro-Wilk* para verificar o tipo de distribuição:

$H_0$ : A população segue uma distribuição normal

$H_1$ : A população não segue uma distribuição normal

O Excerto de Código 6 apresenta o teste de Shapiro-Wilk realizado com a amostra de respostas relativas à questão 1.

```
1 > questao1<-c(1,1,1,1,1,2,2)
2 > shapiro.test(questao1)
3
4      Shapiro-Wilk normality test
5
6 data:  questao1
7 W = 0.6004, p-value = 0.0002752
```

#### Excerto de Código 6 - Teste Shapiro-Wilk Questão 1

De acordo com o Excerto de Código 6, com intervalo de confiança de 95%, pode-se concluir que como o **p-value** é inferior a 5% (nível de significância), a amostra não segue uma distribuição normal, logo um *t-test* não pode ser usado, sendo que irá ser aplicado um *Wilcoxon* teste, que é um teste não paramétrico.

Para avaliar os resultados da questão 1 do inquérito, define-se que a média deve ser inferior a 1,75. Por conseguinte, as hipóteses para o teste atual são:

$H_0$ : Média de respostas é menor ou igual a 1,75

$H_1$ : Média de respostas é maior que 1,75

```

1 > wilcox.test(questao1, mu = 1.75, alternative = "less")
2
3     Wilcoxon signed rank test with continuity correction
4
5 data:  questao1
6 V = 3, p-value = 0.03249
7 alternative hypothesis: true location is less than 1.75

```

#### Excerto de Código 7 - Teste Wilcoxon Questão 1

Após analisar o Excerto de código 7, como o **p-value** é inferior a 0.05, a hipótese nula ( $H_0$ ) é rejeitada, logo com 95% de confiança pode-se concluir que a equipa de desenvolvimento considera que não existe dificuldade na aprendizagem da implementação de *mobile analytics*.

#### Questão 2

Na Figura 42 é apresentada a matriz de respostas por parte dos inquiridos para a segunda questão.

|       |       |             |
|-------|-------|-------------|
| 1     | 0.0%  | 0 responses |
| 2     | 0.0%  | 0 responses |
| 3     | 0.0%  | 0 responses |
| 4     | 28.6% | 2 responses |
| 5     | 71.4% | 5 responses |
| Total |       | 7 responses |

Figura 43 - Matriz Respostas Questão 2

Irá ser utilizado um teste *Shapiro-Wilk* para verificar o tipo de distribuição, pois não se pode assumir que se está perante uma distribuição normal:

$H_0$ : A população segue uma distribuição normal

$H_1$ : A população não segue uma distribuição normal

O Excerto de Código 8 apresenta o teste de *Shapiro-Wilk* realizado com a amostra de respostas relativas à questão 2.

```
1 > questao2<-c(5,5,5,5,5,4,4)
2 > shapiro.test(questao2)
3
4      Shapiro-Wilk normality test
5
6 data:  questao2
7 W = 0.6004, p-value = 0.0002752
```

#### Excerto de Código 8 - Teste Shapiro-Wilk Questão 2

Pode-se concluir que como o **p-value** é inferior a 5% (nível de significância), a amostra não segue uma distribuição normal, logo um *t-test* não pode ser usado, sendo que irá ser aplicado um *Wilcoxon* teste, que é um teste não paramétrico.

Para avaliar os resultados da questão 2 do inquérito, define-se que a média deve ser superior a 4,0. Por conseguinte, as hipóteses para o teste atual são:

$H_0$ : Média de respostas é menor ou igual a 4,0

$H_1$ : Média de respostas é maior que 4,0

```
1 > wilcox.test(questao2, mu = 4.0, alternative = "greater")
2
3      Wilcoxon signed rank test with continuity correction
4
5 data:  questao2
6 V = 15, p-value = 0.01844
7 alternative hypothesis: true location is greater than 4
```

#### Excerto de Código 9 - Teste Wilcoxon Questão 2

Após analisar o Excerto de código 9, como o **p-value** é inferior a 0.05, a hipótese nula ( $H_0$ ) é rejeitada, logo com 95% de confiança pode-se concluir que a equipa de desenvolvimento considera que a adoção de *mobile analytics* na *app* melhorou o processo de deteção de erros.

### Questão 3

Na Figura 43 é apresentada a matriz de respostas por parte dos inquiridos para a terceira questão.

|       |       |             |
|-------|-------|-------------|
| 1     | 0.0%  | 0 responses |
| 2     | 0.0%  | 0 responses |
| 3     | 28.6% | 2 responses |
| 4     | 42.9% | 3 responses |
| 5     | 28.6% | 2 responses |
| Total |       | 7 responses |

Figura 44 - Matriz Respostas Questão 3

Irá ser utilizado um teste *Shapiro-Wilk* para verificar o tipo de distribuição, pois não se pode assumir que se está perante uma distribuição normal:

$H_0$ : A população segue uma distribuição normal

$H_1$ : A população não segue uma distribuição normal

O Excerto de Código 10 apresenta o teste de *Shapiro-Wilk* realizado com a amostra de respostas relativas à questão 3.

```
1 > questao3<-c(5,5,4,4,4,3,3)
2 > shapiro.test(questao3)
3
4      Shapiro-Wilk normality test
5
6 data:  questao3
7 W = 0.85771, p-value = 0.1444
```

Excerto de Código 10 - Teste Shapiro-Wilk Questão 3

Pode-se concluir que como o **p-value** é inferior a 5% (nível de significância), a amostra não segue uma distribuição normal, logo um *t-test* não pode ser usado, sendo que irá ser aplicado um *Wilcoxon* teste, que é um teste não paramétrico.

Para avaliar os resultados da questão 3 do inquérito, define-se que a média deve ser superior a 4,0. Por conseguinte, as hipóteses para o teste atual são:

$H_0$ : Média de respostas é menor ou igual a 4,0

$H_1$ : Média de respostas é maior que 4,0

```
1 > wilcox.test(questao3, mu = 4.0, alternative = "greater")
2
3     Wilcoxon signed rank test with continuity correction
4
5 data:  questao3
6 V = 5, p-value = 0.5793
7 alternative hypothesis: true location is greater than 4
```

#### Excerto de Código 11 - Teste Wilcoxon Questão 3

Após analisar o Excerto de código 11, como o **p-value** é superior a 0.05, a hipótese nula ( $H_0$ ) não pode ser rejeitada, logo com 95% de confiança pode-se concluir que a equipa não considera que desde a adoção de *mobile analytics*, o número de *bugs* detetados tenha diminuído.

#### Questão 4

Na Figura 44 é apresentada a matriz de respostas por parte dos inquiridos para a quarta questão.

|       |        |             |
|-------|--------|-------------|
| 1     | 0.0%   | 0 responses |
| 2     | 0.0%   | 0 responses |
| 3     | 0.0%   | 0 responses |
| 4     | 0.0%   | 0 responses |
| 5     | 100.0% | 7 responses |
| Total |        | 7 responses |

Figura 45 - Matriz Respostas Questão 4

Relativamente à questão 4, todos os inquiridos responderam com o valor 5, sendo a amostra questão 4(5,5,5,5,5,5) está-se perante uma distribuição discreta, com um único valor com probabilidade 1, logo pode-se concluir que a equipa de desenvolvimento considera, sem qualquer dúvida, que a adoção de *mobile analytics* na *app* ajudou no desenvolvimento e monitorização da *app*.

### 6.3 Sumário

Neste capítulo foi apresentada a metodologia de avaliação da solução, assim como os fatores a serem avaliados, os testes de hipóteses e quais os testes a serem usados para avaliar esses fatores.

Neste capítulo, assim como proposto, foram respondidas às seguintes questões:

#### **Que métodos e parâmetros irão ser usados para avaliar a solução?**

Os métodos que irão ser usados para avaliar a solução serão: satisfação do utilizador e deteção de *bugs*.

#### **Que procedimentos/ferramentas irão ser usadas para avaliar esses parâmetros?**

Os procedimentos a ser utilizados para avaliar estes parâmetros serão inquéritos de satisfação e testes estatísticos.

#### **A adoção de *mobile analytics* melhorou o funcionamento/monitorização da *app*?**

Após analisar os resultados da implementação da solução, pode-se concluir que a nova *app* com monitorização melhorou bastante em relação à anterior, como também se pode comprovar com o *feedback* dos utilizadores e suas *reviews*.

De acordo com os questionários efetuado à equipa de desenvolvimento, pode-se concluir que esta considerou que a implementação de *mobile analytics* trouxe uma melhoria na monitorização e conseqüente desenvolvimento da *app*. Em relação aos questionários efetuados aos utilizadores, também se pode concluir que a grande maioria dos utilizadores dá *feedback* positivo da *app* e da qualidade desta.

# 7 Conclusões

O presente capítulo tem como objetivo apresentar as conclusões finais de todo o trabalho desenvolvido ao longo deste projeto, os resultados obtidos e objetivos realizados, limitações e por fim, o trabalho futuro.

## 7.1 Síntese

Com o desenvolvimento desta dissertação foi analisado um problema que se trata do insucesso das *apps*.

As métricas das aplicações móveis são fundamentais porque, sem elas, as empresas não conseguem perceber se as suas *apps* estão a ter sucesso ou a falhar. As empresas que carecem de métricas são obrigadas a confiar em indicadores amplos, como o total de receitas ou o total de *downloads*. Estes indicadores de alto nível podem ocultar o facto de que a aplicação pode estar a ficar sem utilizadores mais rapidamente do que os atrai. Com ferramentas de métricas, as empresas podem ver estes problemas e tomar medidas [41]. O objetivo deste estudo foi perceber se a monitorização das *apps*, seria uma solução viável para o melhor desempenho e sucesso das mesmas. Para testar a viabilidade deste estudo foi construída uma nova *app* e lançada no mercado para tirar conclusões acerca da mesma.

Um dos importantes objetivos desta dissertação seria conseguir responder à questão colocada na Secção 1.3: **Serão as *mobile analytics* a solução para o desempenho e sucesso de uma *app*?**

Há uma série de fatores que são responsáveis pelo desempenho de uma *app*, que vai desde pesquisa, desenvolvimento, marketing, etc., mas o que se pode concluir deste estudo, é que o fator mais decisivo para os utilizadores é a sua qualidade e estabilidade.

Com uma ferramenta de *analytics* que deteta os erros e os seus pontos fracos, os *developers* poupam tempo, evitam erros críticos e entregam *apps* sem falhas, tornando os seus utilizadores satisfeitos, como se pôde concluir através dos resultados obtidos da solução.

## 7.2 Objetivos realizados

Os objetivos desta dissertação seriam estudar e avaliar as métricas mais eficazes para monitorizar a *app*, desenvolver uma *app* implementando estas métricas e, por fim, avaliar e comparar resultados com uma antiga solução pré-existente.

Todos estes objetivos definidos inicialmente foram concluídos com sucesso, a parte do último que foi maioritariamente concluído, dado que não existindo ferramenta de monitorização na *app* anterior, foi complicado obter dados suficientes sobre esta mesma *app* pré-existente.

Assim sendo, na Tabela 16 são apresentados os objetivos e respetivo grau de realização.

Tabela 16 - Realização de Objetivos

| OBJETIVO  | RESULTADO                  |
|---|----------------------------|
| Estudar e avaliar as métricas mais eficazes para monitorizar a <i>app</i> e melhorar o seu desempenho | Realizado                  |
| Desenvolver uma <i>app</i> com a implementação de <i>mobile analytics</i>                             | Realizado                  |
| Avaliar e comparar resultados com uma antiga solução pré-existente                                    | Realizado maioritariamente |

### **7.3 Limitações e Trabalho futuro**

Independentemente do empenho colocado no desenvolvimento deste projeto, nem tudo foi realizado com completo sucesso. Como tal, nesta secção são expostas as limitações aferidas ao longo do projeto, assim como trabalho futuro que poderá ser realizado.

Em termos de limitações ao longo do projeto, um aspeto a melhorar seriam a quantidade de informações disponíveis que se conseguiu obter acerca da *app* anteriormente desenvolvida. Como esta *app* não foi desenvolvida pelos colaboradores da empresa X, tornou-se mais complicado obter informação acerca desta devido a também não existir qualquer tipo de monitorização na solução.

Apesar de se ter chegado à conclusão de que a implementação de uma ferramenta de monitorização na *app*, melhorou o desempenho da mesma, existiram ainda bastantes comentários negativos acerca da falta de funcionalidades que existiam na versão anterior da *app*.

Posto isto, como qualquer outra *app*, a desenvolvida precisa de evolução e suporte, sendo que já estão planeados mais lançamentos da mesma com novas funcionalidades para o futuro.

### **7.4 Apreciação final**

A dissertação desenvolvida foi uma mais-valia para a sua autora, uma vez que teve a oportunidade de estudar e implementar uma solução para resolver um problema que tem vindo a ser cada vez mais recorrente e que ajudará a autora deste documento, a ter em conta em futuros desenvolvimentos de *apps*.



## Referências

- [1] «Aplicação móvel». [https://pt.wikipedia.org/wiki/Aplicativo\\_móvel](https://pt.wikipedia.org/wiki/Aplicativo_móvel) (acedido Dez. 05, 2020).
- [2] MPhil, «Tudo o que se precisa de saber sobre mobile analytics», 2019. <https://www.guiadeinvestimento.com.br/mobile-app-analytics/> (acedido Dez. 05, 2020).
- [3] A. A. Julian Harty, A. A., *The Mobile Analytics Playbook: A practical guide to better testing*. Hewlett Packard Enterprise, 2016.
- [4] A. Dossey, «The Most Prominent Reasons Apps Fail & How To Avoid Them», *Clear Bridge Mobile*, 2019. <https://clearbridgemobile.com/reasons-why-apps-fail/> (acedido Dez. 15, 2020).
- [5] «5 Reasons Why Your App Needs an Error Monitoring System», 2020. <https://devops.com/5-reasons-why-your-app-needs-an-error-monitoring-system/> (acedido Dez. 15, 2020).
- [6] A. GOYAL, «Top reasons why mobile apps fail to make a mark in the market», *Octal It Solut.*, 2019.
- [7] «Tudo o que precisa de saber sobre aplicações móveis», 2019. <https://setup.technology/tudo-o-que-precisa-de-saber-sobre-aplicacoes-moveis> (acedido Dez. 10, 2020).
- [8] G. Esteves, «20 Métricas para medir o sucesso da sua APP». <https://metricasboss.com.br/20-metricas-para-medir-o-sucesso-do-seu-app.html> (acedido Dez. 27, 2020).
- [9] G. Jonathan Meltzer, «Linking analytics and ads, The solution to today's marketing challenges», *Google Mark. Platf.*, pp. 1–18, 2018.
- [10] Marketeam, «Avalie o desempenho da sua app», 2020. <https://usemobile.com.br/mobile-app-analytics/> (acedido Jan. 24, 2021).

- [11] N. Patel, «Push Notification: O Que é, Como Aplicar e 3 Ferramentas Para Usar». <https://neilpatel.com/br/blog/push-notification/> (acedido Jan. 24, 2021).
- [12] A. Blair, «Push Notification Analytics», 2020.  
<https://learn.buildfire.com/en/articles/2056551-push-notification-analytics> (acedido Jan. 24, 2021).
- [13] P. Han, «A Beginner’s Guide to Cohort Analysis: the Most Actionable (and Underrated) Report on Google Analytics», *Humanlytics*, 2017.
- [14] K. Ferreira, «Como fazer um teste A/B e ver exemplos de como isso pode influenciar na sua estratégia», 2019.  
<https://rockcontent.com/br/blog/teste-ab-no-google-analytics/> (acedido Jan. 24, 2021).
- [15] Wikipedia, «Firebase». <https://en.wikipedia.org/wiki/Firebase> (acedido Jan. 24, 2021).
- [16] J. Kurzweg, «Top 10 Mobile App Analytics Tools», 2020.  
<https://uxcam.com/blog/top-10-analytics-tool-for-mobile-in-2018/> (acedido Dez. 15, 2020).
- [17] E. G. Haije, «Mobile App Analytics Tools», 2019.  
<https://mopinion.com/mobile-app-analytics-tools/> (acedido Dez. 05, 2020).
- [18] N. Patel, «Mobile App Analytics». <https://neilpatel.com/blog/mobile-app-analytics/> (acedido Dez. 05, 2020).
- [19] A. Velykyy, «Flutter: a full introduction to the framework», *Axon Dev. Gr.*, 2020.
- [20] P. Gomes, «Flutter: A revolução mobile da gigante Google», *Pplware*, 2020.  
<https://pplware.sapo.pt/internet/flutter-a-revolucao-mobile-da-gigante-google/> (acedido Fev. 13, 2021).
- [21] K. Ratra, «BLOC Architecture in Flutter», 2021.  
<https://medium.com/gradeup/bloc-architecture-explained-10d54c236df4> (acedido Ago. 10, 2021).

- [22] K. Galvão, «Analyzing BLoC and MobX», 2020.  
<https://blog.flutterando.com.br/analyzing-bloc-and-mobx-7118053e913a>  
(acedido Ago. 10, 2021).
- [23] K. H. Muntefering, «Value analysis», 1985.  
[https://www.urenio.org/tools/en/value\\_analysis.pdf](https://www.urenio.org/tools/en/value_analysis.pdf) (acedido Dez. 27, 2020).
- [24] P. A. C. M. Pierry Tezaa, Gertrudes Dandolinia, João Artur de Souzaa, Viviane Brandão Miguez, Roberto Fabiano Fernandes, «Modelos de front end da inovação: similaridades, diferenças e perspectivas de pesquisa», 2015.
- [25] P. Belliveau, A. Griffin, S. Somermeyer, e Product Development & Management Association., *The PDMA toolbook for new product development*. R & D Management, 2002.
- [26] K. W. Peter Koen, Greg Ajamian, Robert Burkart, Allen Clamen, Jeffrey Davidson, Robb D'Amore, Claudia Elkins, Kathy Herald, Michael Incorvia, Albert Johnson, Robin Karol, Rebecca Seibert, Aleksandar Slavejkov, «Providing clarity and a common language to the “Fuzzy Front End”», *Res. Manag.*, vol. 44, n. 2, pp. 46–55, 2001.
- [27] «Valor». [https://pt.wikipedia.org/wiki/Valor\\_\(marketing\)](https://pt.wikipedia.org/wiki/Valor_(marketing)) (acedido Dez. 27, 2020).
- [28] A. I. Tânia Modesto Veludo-de-Oliveira, «O conceito de valor para o cliente», 2005.
- [29] W. U. and A. Eggert, «Value-Based Differentiation in Business Relationships: Gaining and Sustaining Key Supplier Status», *Sage Publ. Inc*, vol. 70, n. 1, pp. 119–136, 2006.
- [30] T. Woodall, «Conceptualising “Value for the Customer”: An Attributional, Structural and Dispositional Analysis», *Acad. Mark. Sci. Rev.*, 2003.
- [31] E. R. SCHARF, «Proposta de Valor e Capital Humano: práticas estratégicas de marketing», *Rev. Bras. GESTÃO NEGÓCIOS*, vol. 14, n. 43, pp. 216–233, 2012.

- [32] P. KOTLER, *Administração de Marketing*. Prentice Hall, 2000.
- [33] M. A. Z. Fábio Luiz Zandoval Bonazzi, «Inovação e Modelo de Negócio: um estudo de caso sobre a integração do Funil de Inovação e o Modelo Canvas», *Rev. Bras. Gest. Negócios*, vol. 16, n. 53, pp. 616–637, 2014.
- [34] Wikipedia, «Requisito Funcional», 2012. [https://pt.wikipedia.org/wiki/Requisito\\_funcional](https://pt.wikipedia.org/wiki/Requisito_funcional) (acedido Jun. 21, 2021).
- [35] ReQtest, «Why is the difference between functional and Non-functional requirements important?», 2012. <https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/> (acedido Jun. 21, 2021).
- [36] B. A. T. in H. COEPD, «What is FURPS+?», 2014. <https://businessanalysttraininghyderabad.wordpress.com/2014/08/05/what-is-furps/> (acedido Jun. 21, 2021).
- [37] E. C. Marie Baehr, «Evaluation Methodology», 2004.
- [38] U. of T. Dr. David C. Stone & Jon Ellis, Department of Chemistry, «Instrumental Analysis and Calibration». <https://sites.chem.utoronto.ca/chemistry/coursenotes/analsci/stats/12tailed.html> (acedido Fev. 07, 2021).
- [39] N. University, «One-tailed and Two-tailed Tests», 2018.
- [40] M. N. Graeme D. Ruxton, «When should we use one-tailed hypothesis testing?», *J. Compil. © 2010 Br. Ecol. Soc.*, 2010.
- [41] J. Booth, «What are mobile app analytics metrics?», *The Signal*, 2020. <https://mixpanel.com/blog/mobile-app-analytics-metrics/> (acedido Jun. 21, 2021).
- [42] H. P. Automoveis, «O que são carros híbridos? Saiba como funcionam e vantagens», 2021. <https://honda-automoveis.pt/blog/o-que-sao-carros-hibridos/> (acedido Ago. 10, 2021).

# Anexos

## Anexo A - Questionário 1: Utilizador da *app*

How old are you?

18-25

26-35

36-50

50+

What is your operating system?

iOS

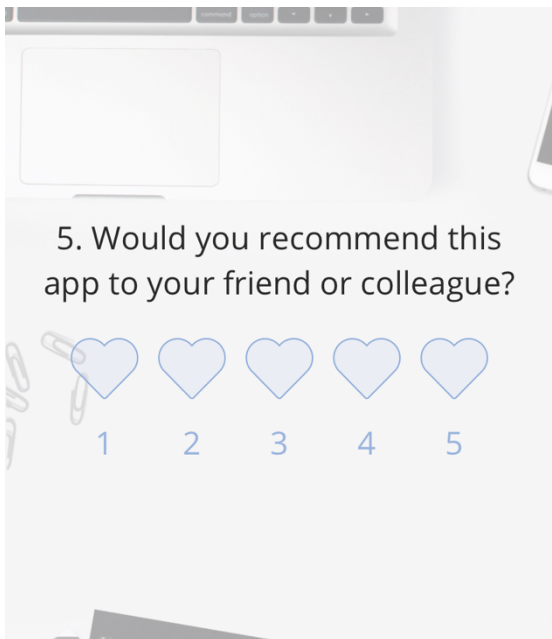
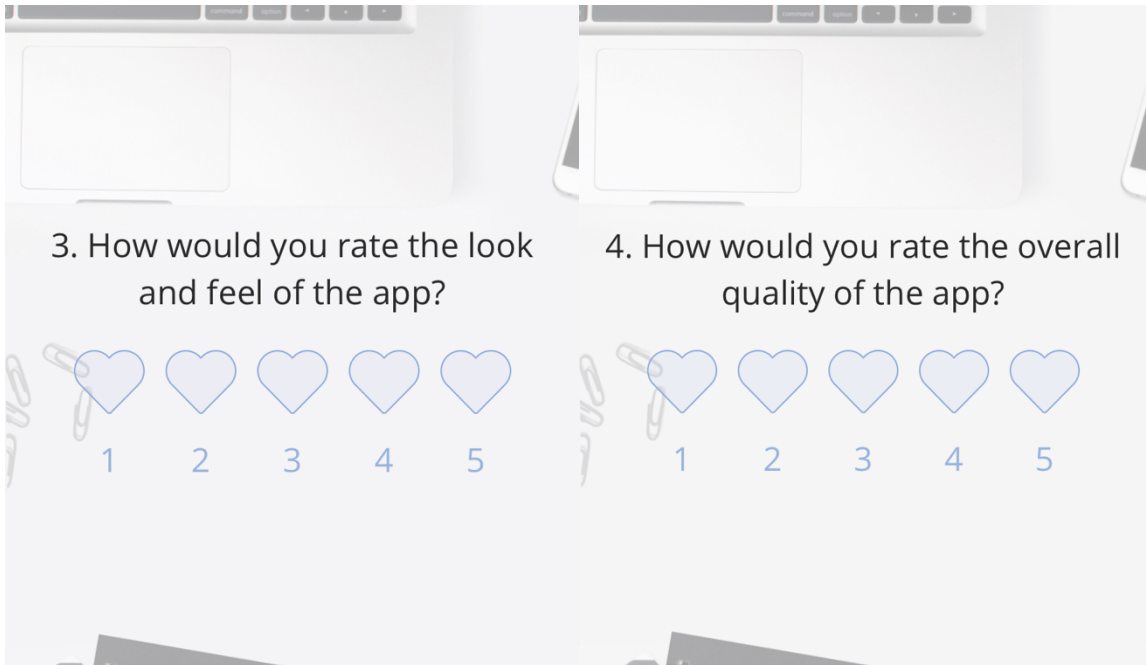
Android

1. How would you rate the ease of use of the app?

1  2  3  4  5

2. How would you rate the stability of the app?


1  2  3  4  5



## Anexo B – Questionário 2: Equipa de Desenvolvimento


1. Em termos de dificuldade, como classificarias a aprendizagem de implementação de Mobile Analytics no projeto?

1 2 3 4 5




2. Sentes que a adoção de Mobile Analytics melhorou o processo de deteção de erros da app?

1 2 3 4 5



3. Sentes que desde a adoção de Mobile Analytics, têm diminuído o número de bugs detetados?

1 2 3 4 5



4. Sentes que a adoção de Mobile Analytics ajudou no desenvolvimento e consequente monitorização da app?

1 2 3 4 5

