



Mecanismo de Refer a Friend

Mestrado em Engenharia Informática – Computação Móvel

Rúben Ricardo Monge Lauro

Leiria, outubro de 2021



Mecanismo de Refer a Friend

Mestrado em Engenharia Informática – Computação Móvel

Rúben Ricardo Monge Lauro

Estágio realizado sob a orientação da Professora Doutora Sónia Maria Almeida da Luz e sob supervisão do Paulo Sousa

Leiria, outubro de 2021

Originalidade e Direitos de Autor

O presente relatório de estágio é original, elaborado unicamente para este fim, tendo sido devidamente citados todos os autores cujos estudos e publicações contribuíram para o elaborar.

Reproduções parciais deste documento serão autorizadas na condição de que seja mencionado o Autor e feita referência ao ciclo de estudos no âmbito do qual o mesmo foi realizado, a saber, Mestrado em Engenharia Informática – Computação Móvel, no ano letivo 2020/2021 da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, Portugal, e, bem assim, à data das provas públicas que visaram a avaliação destes trabalhos (se aplicável).

Agradecimentos

Em primeiro lugar, gostaria de agradecer a todas as pessoas que de alguma forma tiveram impacto no meu percurso académico.

Gostava ainda de agradecer à WIT pela oportunidade e pela excelente integração na equipa. Da WIT, queria deixar um especial obrigado à Aline Santos que me acompanhou durante grande parte do estágio e que contribuiu imenso para o sucesso do projeto.

Ao Politécnico de Leiria, gostava de agradecer à professora orientadora atribuída, Sónia Luz, por todo o apoio e orientação prestado ao longo deste ano.

Um agradecimento especial àqueles que sempre me apoiaram e que vão estar sempre ali quando eu precisar, os meus pais e irmão.

Por último, mas não menos importante, um grande obrigado à minha namorada, por tudo.

Resumo

O presente relatório foi realizado no âmbito da unidade curricular de Estágio do Mestrado em Engenharia Informática – Computação Móvel da Escola Superior de Tecnologia e Gestão, a qual pertence ao Instituto Politécnico de Leiria. O Estágio teve duração de nove meses e foi realizado de forma totalmente remota na WIT Software, empresa que se dedica ao desenvolvimento de *software*, principalmente, para a área de telecomunicações.

Esta unidade curricular pretende oferecer a oportunidade de experienciar o contexto empresarial adquirindo novos conhecimentos e aplicando os diversos conceitos aprendidos durante o percurso académico, para que seja possível atingir os resultados esperados dentro do tempo estimado. O estágio teve como objetivo o desenvolvimento de uma prova de conceito de *Refer a Friend*, para a qual foi idealizada e desenvolvida uma arquitetura constituída por vários componentes. Foi introduzida numa aplicação móvel a possibilidade de enviar convites para que outros utilizadores se juntassem à aplicação. Para além disto, foi ainda integrada na plataforma web de gestão do cliente a possibilidade de gerir as campanhas de *Refer a Friend* e ainda observar a evolução e o sucesso das mesmas.

Do estágio resultou uma plataforma pronta a ser testada para posterior integração no ecossistema do cliente, com o objetivo de adquirir mais clientes mantendo-os ligados à plataforma o maior tempo possível. Surgiu ainda, a oportunidade de iniciar o processo de desenvolvimento de uma plataforma genérica utilizando tecnologias *cross-platform* que permitem um desenvolvimento mais rápido.

Palavras-chave: *Refer a friend*, aplicação móvel, plataforma web, campanhas

Abstract

This report was produced under the scope of the curricular unit of Internship of the Master's in Computer Engineering - Mobile Computing of the School of Technology and Management, which belongs to the Polytechnic Institute of Leiria. The internship lasted nine months and was carried out entirely remotely at WIT Software, a company dedicated to software development, mainly for the telecommunications area.

This curricular unit intends to offer the opportunity to experience the business context, acquiring new knowledge and applying the various concepts learned during the academic course, so that it is possible to achieve the expected results within the estimated time. The internship aimed to develop a proof of concept of *Refer a Friend*, for which an architecture consisted in several components was designed and developed. It was introduced in a mobile application the possibility of sending invitations so that other users could join the application. In addition to this, it was also integrated in the customer's web management platform the possibility of managing *Refer a Friend* campaigns and even observe their evolution and success.

The internship resulted in a platform ready to be tested for further integration into the customer's ecosystem, with the goal of acquiring more customers and keeping them connected to the platform as long as possible. Furthermore, there was an opportunity to start the development process of a generic platform using cross-platform technologies that allow for faster development.

Keywords: Refer a friend, mobile application, web management platform, campaigns

Índice

Originalidade e Direitos de Autor	I
Agradecimentos.....	II
Resumo.....	III
Abstract.....	IV
Lista de Figuras.....	VIII
Lista de Tabelas.....	X
Lista de Siglas e Acrónimos	XI
1. Introdução	1
1.1. Enquadramento e Problema.....	2
1.2. Motivação e Objetivos	2
1.3. Estrutura do documento	3
2. Caracterização da Entidade de Acolhimento.....	4
3. Estado da arte	5
3.1. Código.....	5
3.1.1. Uber Eats	5
3.1.2. TheFork	6
3.1.3. Tropical Smoothie Cafe.....	7
3.1.4. Yorn.....	8
3.2. Link.....	9
3.2.1. Revolut.....	9
3.2.2. Airbnb.....	11
3.2.3. Dropbox.....	12
3.2.4. Tesla.....	13
3.2.5. Evernote.....	14
3.2.6. World of warcraft	15
3.2.7. Fiverr	17

3.2.8.	Humble bundle	18
3.2.9.	T-Mobile.....	19
3.2.10.	Three.....	20
3.2.11.	Mint Mobile.....	21
3.3.	Link & Código	22
3.3.1.	Uber.....	22
3.3.2.	N26.....	23
3.3.3.	Lyft.....	24
3.3.4.	Wish	25
3.3.5.	Hotel Tonight	26
3.4.	Resumo	28
4.	Planeamento e Metodologia de desenvolvimento	29
4.1.	Planeamento.....	29
4.2.	Metodologia de desenvolvimento	30
4.3.	Resumo	34
5.	Análise de requisitos e protótipos	35
5.1.	Análise de requisitos.....	35
5.2.	Protótipos	42
5.3.	Resumo	45
6.	Desenvolvimento.....	46
6.1.	Arquitetura do sistema	46
6.2.	Ferramentas utilizadas	48
6.2.1.	Intellij IDEA.....	48
6.2.2.	Android studio.....	49
6.2.3.	Visual studio code	49
6.2.4.	Docker	49
6.2.5.	Bpmn.io	50

6.3.	Aplicação Móvel.....	50
6.4.	Backoffice	60
6.5.	Backend	69
6.5.1.	Microserviço.....	70
6.5.2.	Desenvolvimento inicial dos <i>endpoints</i>	71
6.5.3.	Melhorias	77
6.6.	Resumo	79
7.	Conclusão	80
7.1.	Trabalho futuro	81
	Bibliografia	82
	Anexo A – Protótipos de baixa fidelidade.....	89
	Anexo B – Processos de negócio.....	101
	Anexo C – Plano de estágio	106
	Anexo D – UI em React Native	109

Lista de Figuras

Figura 1 – Uber Eats <i>refer a friend flow</i>	6
Figura 2 – TheFork <i>refer a friend flow</i>	7
Figura 3 – Tropical Smoothie Cafe <i>refer a friend flow</i>	8
Figura 4 – Yorn Shake it <i>refer a friend flow</i>	9
Figura 5 – Revolut <i>refer a friend flow</i>	10
Figura 6 – Airbnb <i>refer a friend flow</i>	11
Figura 7 – Dropbox <i>refer a friend flow</i>	12
Figura 8 – Página de encomenda do veículo	13
Figura 9 – Evernote <i>refer a friend flow</i>	15
Figura 10 – Interface de <i>refer a friend</i> do jogo World of Warcraft	16
Figura 11 – Fiverr <i>refer a friend flow</i>	18
Figura 12 – Humble bundle <i>refer a friend flow</i>	19
Figura 13 – Three <i>refer a friend flow</i>	21
Figura 14 – Mint Mobile <i>refer a friend flow</i>	22
Figura 15 – Seguimento de um link partilhado pela aplicação Uber condutor.....	23
Figura 16 – Seguimento do link no website do N26.....	24
Figura 17 – Lyft <i>refer a friend flow</i>	25
Figura 18 – Wish <i>refer a friend flow</i>	26
Figura 19 – Hotel Tonight <i>refer a friend flow</i>	27
Figura 20 – Hotel Tonight <i>refer a friend social sharing flow</i>	28
Figura 21 – Plano de estágio resumido.....	30
Figura 22 – Representação do ciclo scrum.....	32
Figura 23 – Diagrama de processo de negócio de <i>Refer a friend</i>	35
Figura 24 – Protótipo de baixa fidelidade da página inicial das campanhas de um produto.....	42
Figura 25 – Representação da arquitetura geral.....	47
Figura 26 – Fragmentação do mercado Android	52
Figura 27 – Elemento preenchido sem contacto presente	54
Figura 28 – Elemento por preencher	54

Figura 29 – Elemento preenchido com contacto com foto	54
Figura 30 – Elemento preenchido com contacto sem foto.....	54
Figura 31 – Exemplo link antes da conversão para <i>short</i> URL.....	55
Figura 32 – Exemplo JSON de mensagem do Firebase Cloud Messaging.....	56
Figura 33 – Exemplo lista para remetente sem prémio ou sem limite.....	59
Figura 34 – Exemplo lista com limite.....	60
Figura 35 – Elemento com <i>badge</i>	60
Figura 36 - Excerto de código representativo da comunicação criação de uma campanha.....	63
Figura 37 – Protótipo de baixa fidelidade da analítica	65
Figura 38 – Secção dos gráficos	67
Figura 39 – Secção dos cartões	68
Figura 40 – Representação da solução de <i>reverse-proxy</i> implementada	69
Figura 41 – Mecanismo de funcionamento do microserviço	70
Figura 42 – Representação do mecanismo de <i>fallback</i>	72

Lista de Tabelas

Tabela 1 – <i>User stories</i> da aplicação.....	39
Tabela 2 – <i>User stories</i> do <i>Backoffice</i>	40
Tabela 3 – <i>User stories</i> da aplicação definidas para desenvolvimento.....	43
Tabela 4 – <i>User stories</i> do <i>Backoffice</i> definidas para desenvolvimento.....	43

Lista de Siglas e Acrónimos

API	<i>Application Programming Interface</i>
BI	<i>Business intelligence</i>
BPM	<i>Business process mapping</i>
CSV	Valores separados por vírgulas
IDE	<i>Integrated Development Environment</i>
JAR	<i>Java ARchive</i>
JVM	<i>Java virtual machine</i>
KPI	<i>Key performance indicators</i>
MMORPG	<i>Massively Multiplayer Online Role-Playing Game</i>
QA	<i>Quality Assurance</i>
SDK	<i>Software development kit</i>
UI	Interface do utilizador
UIT	União Internacional de Telecomunicações
URL	<i>Uniform Resource Locator</i>
WOM	<i>Word-of-Mouth</i>

1. Introdução

Com um mundo completamente invadido pela era tecnológica, são nos trazidos diversos benefícios, mas também algumas adversidades. Assim sendo, também o mundo do marketing sofre mudanças. As estratégias de marketing tradicionais, como a publicidade nas televisões, jornais, etc. perderam o impacto que tinham e a influência na compra de produtos de uma determinada marca. Assim como, a redução do principal objetivo, a aquisição de novos clientes.

A introdução das redes sociais, fez com que um produto conseguisse atingir mais clientes de uma maneira bastante mais rápida, no entanto, atualmente, existe uma frequência elevada de apresentação intrusiva de determinadas publicidades, que por vezes pode não trazer benefícios, saturando o cliente final e provocando perda de confiança no produto ou marca anunciada. No entanto, existe uma técnica de marketing que prevalece há bastantes anos, na qual os consumidores continuam a confiar bastante, é o *word-of-mouth* (WOM).

O WOM consiste em marketing de um cliente para outro, partilhando informações sobre uma marca ou produto, de forma espontânea sem que nenhum receba qualquer recompensa [1]. É com base neste conceito que surgiram os *Referral Programs* no mundo digital, no entanto, estes podem ser considerados um tipo diferente de WOM, pois pode existir algum tipo de estimulação quer seja para quem está a divulgar um produto ou marca ou por quem está a receber esta informação [2].

A estimulação utilizada por parte da empresa tem de ser algo bem calculado, para que seja sustentável, sendo necessário ter em conta o valor da aquisição de um cliente para superar o valor da estimulação oferecida, de forma a ser gerado algum lucro. Para além disso, ao ser oferecido algum tipo de estimulação, surge a possibilidade de existir fraude seja por quem está a divulgar ou por quem recebeu a informação sobre o produto ou marca, tornando-se bastante importante a definição das regras do programa.

Existem diversos tipos de estimulação, sendo que um *Referral Program* pode ter muitas variantes, tendo estas muito impacto no sucesso do mesmo. No estudo de Ryu e Feick [3], em 2007, sobre probabilidade de um cliente divulgar um produto ou marca, com a influência de diversos fatores, tais como, a presença de estímulo, assim como, a quantidade

e o destinatário, foi possível concluir que os clientes preferem a equidade, com exceção de quando existe uma relação de proximidade entre eles. Estes estudos, auxiliam a definição dos objetivos necessários para a elaboração de *Referral Program* de sucesso.

1.1. Enquadramento e Problema

O presente relatório foi elaborado no âmbito da unidade curricular de Estágio do Mestrado em Engenharia Informática – Computação Móvel da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria. Tendo como objetivo apresentar o trabalho desenvolvido ao longo do estágio na empresa WIT Software, o qual decorreu de forma totalmente remota.

A problemática do estágio consiste principalmente na necessidade da aquisição de novos clientes para uma determinada empresa a qual possui um produto digital, como por exemplo, um *website* ou uma aplicação móvel, através do envio de um convite por parte de um utilizador da plataforma, para outro utilizador, o qual ainda não é cliente daquela plataforma.

1.2. Motivação e Objetivos

A aquisição de clientes através de técnicas que apresentam investimento baixo e alguma retenção dos clientes, tem sido um facto muito estudado pelos *Marketers*. Sendo o WOM um dos fatores mais confiáveis pelos *Marketers* e considerando-o como o fator primário de 20% a 50% das opções de compra [4], uma empresa que adote uma estratégia baseada no WOM, como por exemplo, os *Referral Programs*, pode vir a obter clientes que se traduzem num possível lucro. É com base nesta opção que foi elaborado o presente estágio e para o qual foram definidos alguns objetivos.

Os objetivos foram divididos em objetivos técnicos e objetivos gerais. Tendo como principais objetivos técnicos, a construção de uma plataforma de *refer a friend* com uma experiência *seamless*, ou seja, sem que sejam necessários passos manuais para que o utilizador aceite um convite, por último, deve ser possível efetuar o *tracking* do sucesso do programa.

Quanto aos objetivos gerais, pretende-se experienciar o contexto empresarial aplicando conceitos aprendidos ao longo do percurso académico, demonstrando autonomia no desenvolvimento e na resolução de problemas, aproveitando para aprender novas tecnologias e conceitos.

1.3.Estrutura do Documento

O presente relatório está estruturado em sete capítulos:

- No capítulo dois, é apresentada a caracterização da entidade de acolhimento, assim como, a descrição de alguns pontos específicos do estágio;
- No capítulo três, são apresentadas as diferentes plataformas no mercado que utilizam *Referral Programs* para a aquisição de novos clientes, constituindo assim, o estado da arte;
- No capítulo quatro, é explicado o plano e metodologia de desenvolvimento utilizado no estágio;
- No capítulo cinco, o processo de levantamento de requisitos e prototipagem é detalhado, sendo apresentadas as diversas fases atravessadas.
- No capítulo seis, são descritos os processos e o ambiente de desenvolvimento utilizados nos diferentes componentes constituintes da arquitetura do sistema, assim como, é explicado o processo de integração entre os mesmos.
- No capítulo sete, são expostas as conclusões do trabalho realizado e apresentada uma perspectiva de um possível trabalho futuro que possa ser realizado.

2. Caracterização da Entidade de Acolhimento

O presente estágio teve duração de 9 meses, tendo início em setembro 2020 e término em junho de 2021, foi realizado na WIT Software, de forma totalmente remota. A WIT Software [5] é uma empresa de software fundada em 2001, como uma empresa *spin-off* da Universidade de Coimbra. Esta empresa apresenta uma divisão flexível em 3 áreas principais, a área da Lógica de Negócio, na qual estão presentes funções de estudo de mercado e tomada de decisões em relação ao software a desenvolver, a área de Desenvolvimento de Software, que tal como o nome indica possui a função de desenvolver software, mas também por exemplo, garantir a qualidade do mesmo, e por último, a área dos Serviços de Suporte, na qual se encontram os serviços de finanças, recursos humanos, design, entre outro. Encontra-se sediada em Lisboa e conta com cerca de 350 colaboradores, tendo ainda escritórios em Leiria, Coimbra, Aveiro e Porto.

Os principais valores da empresa emergem como resultado da combinação de ideais de quatro figuras notáveis, Fernão de Magalhães, Mahatma Gandhi, Albert Einstein e Steve Jobs. Fernão de Magalhães representa a ambição e conquista, Mahatma Gandhi representa a valorização dos valores humanos e a ajuda mútua, Albert Einstein representa a inteligência e a capacidade para resolver problemas e por último Steve Jobs representa a persistência e inovação [6].

O principal objetivo da empresa é criar produtos e soluções de marca branca para as principais operadoras de telecomunicações. O leque de clientes que a WIT Software possui é constituído por operadoras nacionais tais como, a Vodafone e a NOS, como também, internacionais como, por exemplo, a T-Mobile [7].

Ao longo dos 20 anos de existência da WIT, esta tem vindo a evoluir através da adoção de uma perspetiva de inovação contínua, adaptando-se aos diferentes mercados e consequentemente aos diferentes clientes. Existe ainda uma estratégia de motivação a partir de dentro, ou seja, é transmitida a ideia da necessidade constante da melhoria do software produzido, para isso cada um tem de competir consigo próprio de modo que a qualidade de produção melhore de dia para dia. Assim, é criado software com a melhor qualidade possível, traduzindo-se, numa enorme satisfação do cliente. Consequentemente, existe a criação de uma relação de confiança entre os clientes e a WIT, permitindo uma evolução contínua de ambos.

3. Estado da Arte

Neste capítulo apresentam-se diversas aplicações/plataformas que possuem programas de *refer a friend*, assim como, é feita uma análise dos mesmos, considerando diversos aspetos, tais como: método, tipo de convite, tipo de recompensas e a quem são atribuídas, entre outros. Após a análise, foi possível estabelecer uma divisão por método do programa, sendo esta:

- Código;
- Link;
- Link e código.

3.1. Código

Nas aplicações que utilizam códigos como método dos seus programas de *refer a friend* é comum ver um certo tipo de características, tais como, introdução manual do código partilhado a partir de contas de utilizador que ainda não consumiram o primeiro serviço.

3.1.1. Uber Eats

A aplicação Uber Eats [8] permite aos utilizadores usufruírem das suas refeições prediletas, confeccionadas em restaurantes, na sua própria casa de forma muito fácil, através de um serviço de entregas globalizado.

Para além disso, esta aplicação possui ainda um programa de *refer a friend* com as seguintes características:

- Recompensas para ambos;
- 10€ de desconto no primeiro pedido;
- Sem limite de recompensas;
- Não existe *tracking* dos convites;
- Partilha nos diferentes canais de comunicação.

A Figura 1 representa o *flow* para enviar um convite na aplicação Uber Eats.

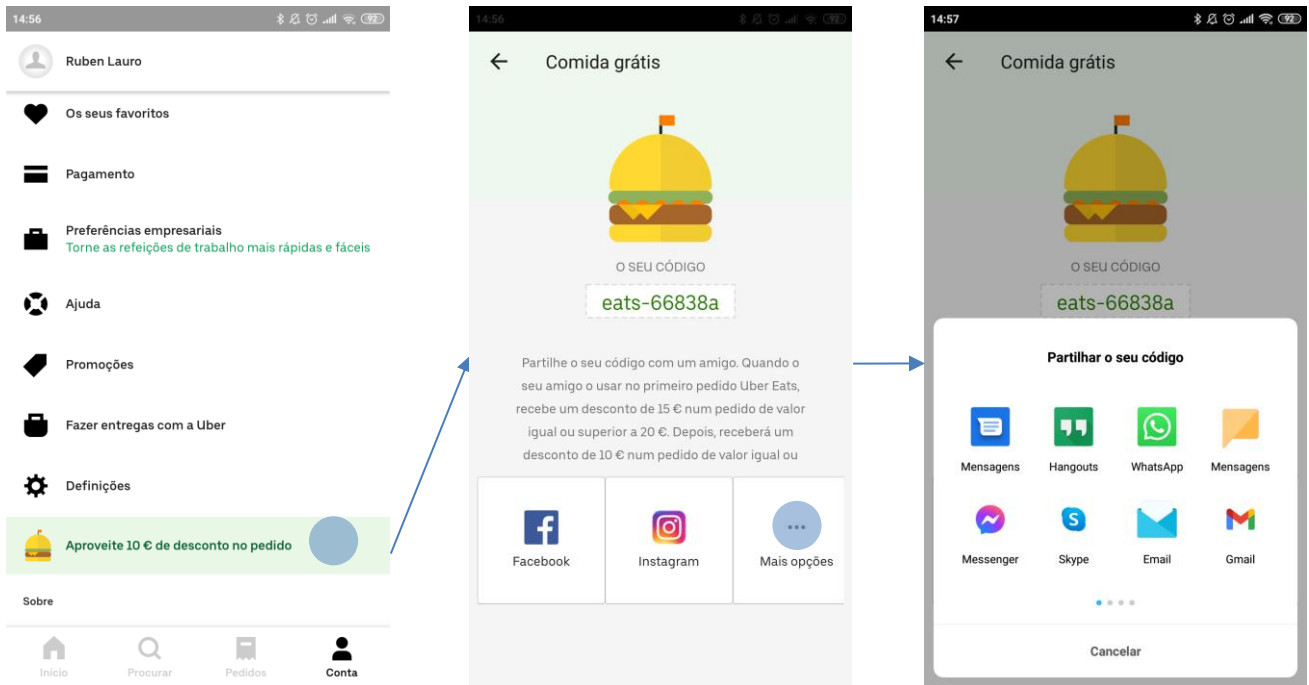


Figura 1 – Uber Eats *refer a friend flow*

3.1.2. TheFork

O TheFork [9] é uma aplicação que possibilita aos seus utilizadores a realização de reservas em restaurantes à sua escolha, em tempo real. Esta aplicação dá também ao utilizador diversos benefícios, tais como, ofertas exclusivas, menus especiais e muitos descontos. Para além disso, esta aplicação possui um programa de *refer a friend* com as seguintes características [10]:

- Recompensas para ambos;
- 500 Yums para quem envia o convite (Crédito na aplicação);
- 1000 Yums para quem recebe o convite (Crédito na aplicação);
- Partilha nos diferentes canais de comunicação;
- Não existe *tracking* dos convites.

A Figura 2 representa o *flow* para enviar um convite na aplicação TheFork.

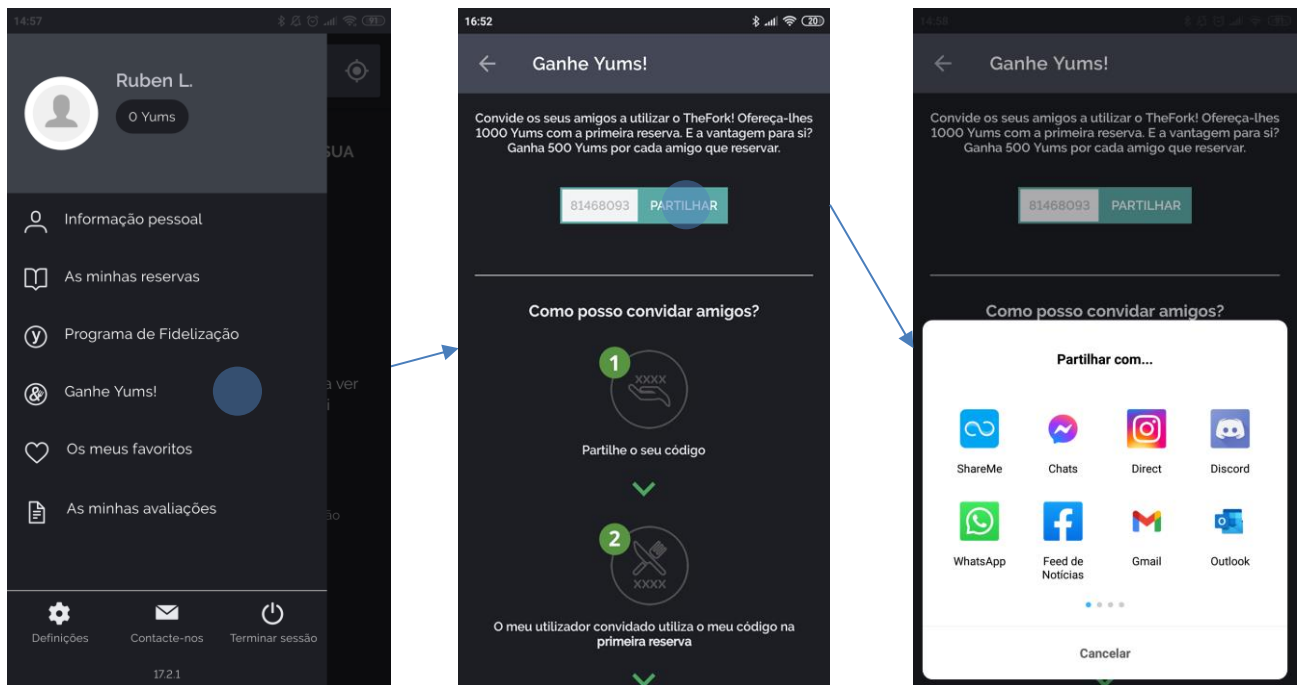


Figura 2 – TheFork refer a friend flow

3.1.3. Tropical Smoothie Cafe

O Tropical Smoothie Cafe [11] é uma cadeia *franchise* de restauração, situada nos Estados Unidos. A qual possui uma aplicação que proporciona aos seus utilizadores a habilidade para encomendar os seus produtos e receber ofertas exclusivas. Dentro destas ofertas, encontra-se um programa de *refer a friend* com as seguintes características [12]:

- Recompensa para quem convida;
- 1 *smoothie* grátis;
- Partilha nos diferentes canais de comunicação;
- Não existe *tracking* dos convites.

A Figura 3 representa o *flow* para enviar um convite na aplicação do Tropical Smoothie Cafe.

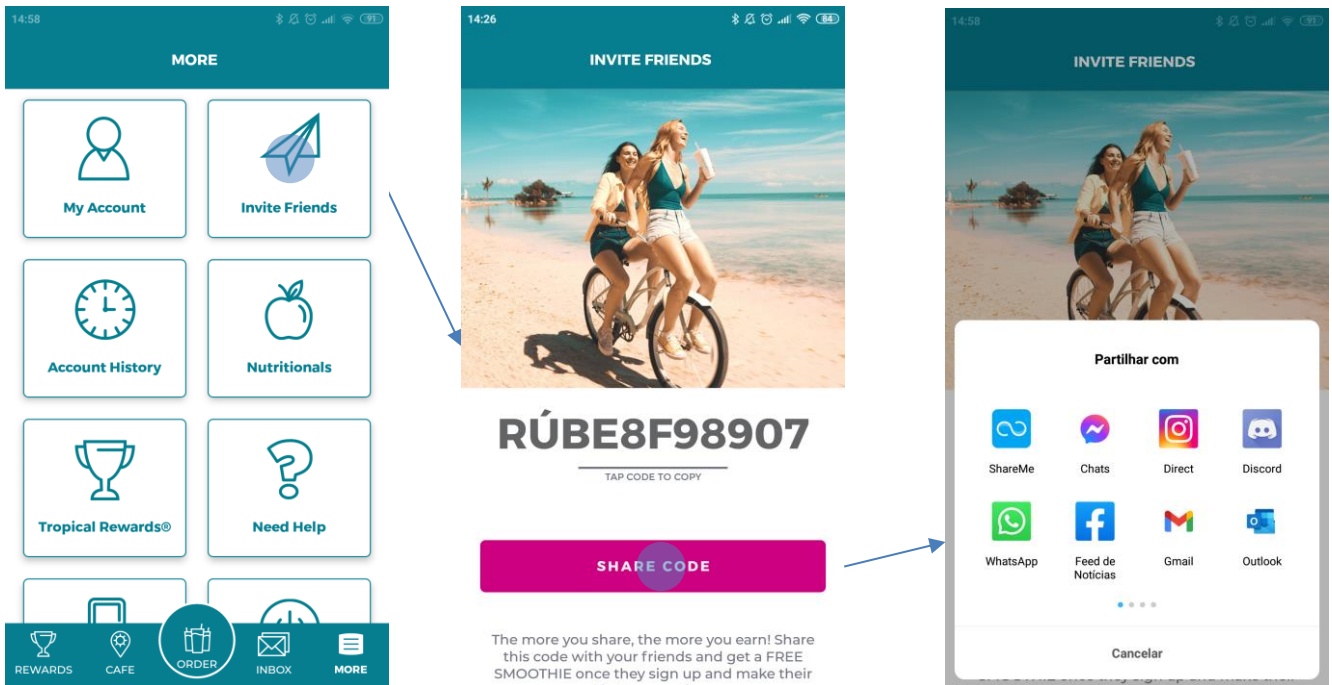


Figura 3 – Tropical Smoothie Cafe refer a friend flow

3.1.4. Yorn

A Yorn [13] é uma marca que pertence a um dos principais operadores de telecomunicações de Portugal, a Vodafone. Sendo assim, a Vodafone tem um tarifário chamado Yorn que é exclusivo a pessoas com menos de 25 anos ou estudantes.

Para além das muitas outras ofertas deste tarifário, este tem uma campanha chamada Yorn Shake it [14], a qual consiste em fazer *Shakes*, ou seja, abanar o *smartphone* para ganhar prémios. É então neste programa que está presente o sistema de *refer a friend* com as seguintes características [15]:

- Recompensa para ambos;
- 1 *shake*;
- Partilha em diferentes canais de comunicação;
- Não existe *tracking* dos convites;
- Limite de 10 pessoas convidadas.

A Figura 4 representa o *flow* para enviar um convite na aplicação da My Vodafone nomeadamente na secção do Yorn Shake it.

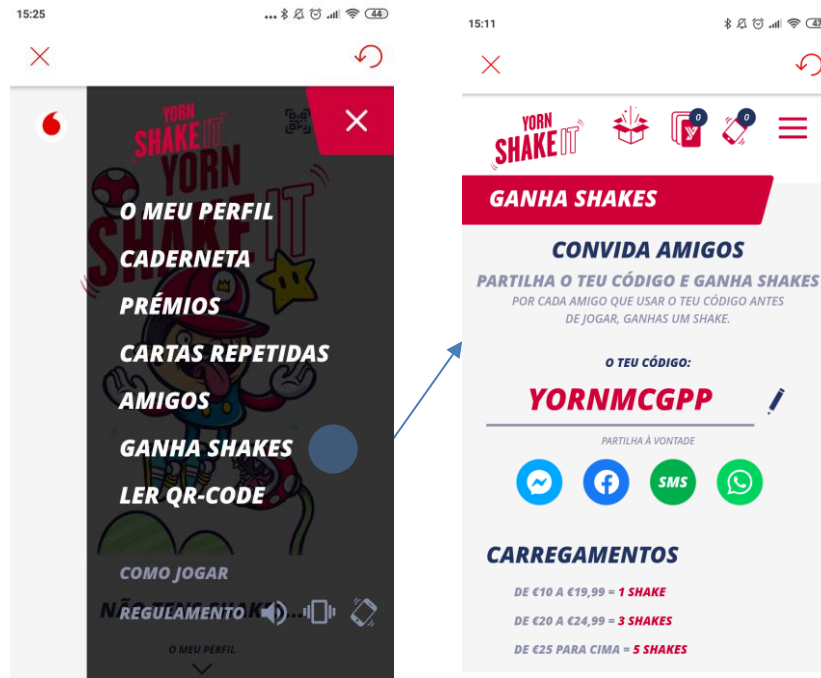


Figura 4 – Yorn Shake it refer a friend flow

3.2.Link

A utilização de *links* como método dos programas de *refer a friend* é o mais comum nas diferentes aplicações/plataformas. Este método é mais vantajoso pois permite aos utilizadores uma melhor experiência devido à inexistência de uma tarefa manualmente, no entanto também apresenta as suas desvantagens, nomeadamente, o facto de um utilizador que já tenha criado uma conta ter de criar uma nova para beneficiar do programa.

3.2.1. Revolut

O Revolut [16] é um banco digital. Este possui uma aplicação que permite fazer as mais diversas operações financeiras, como se de um banco físico se tratasse. Permite aos utilizadores, não só, cambiarem dinheiro sem a aplicação de qualquer taxa, de forma rápida e em poucos segundos, mas também, a realização de investimentos em criptomoedas ou em ações, execução de transações para a maioria dos bancos a nível mundial, entre outras. Possui ainda um programa de *refer a friend* com as seguintes características:

- Partilha em diferentes canais de comunicação;
- Recompensas para quem convida;
- Subscrição do plano “Metal” por um ano;

- *Tracking* dos convites;
- Limite de 5 pessoas;
- Recompensas atribuídas depois dos 5 convidados, fazerem o registo, fazerem um depósito no mínimo de 10€, verificarem a identidade e pedirem o cartão físico.

A Figura 5 representa o *flow* para enviar um convite na aplicação do Revolut.

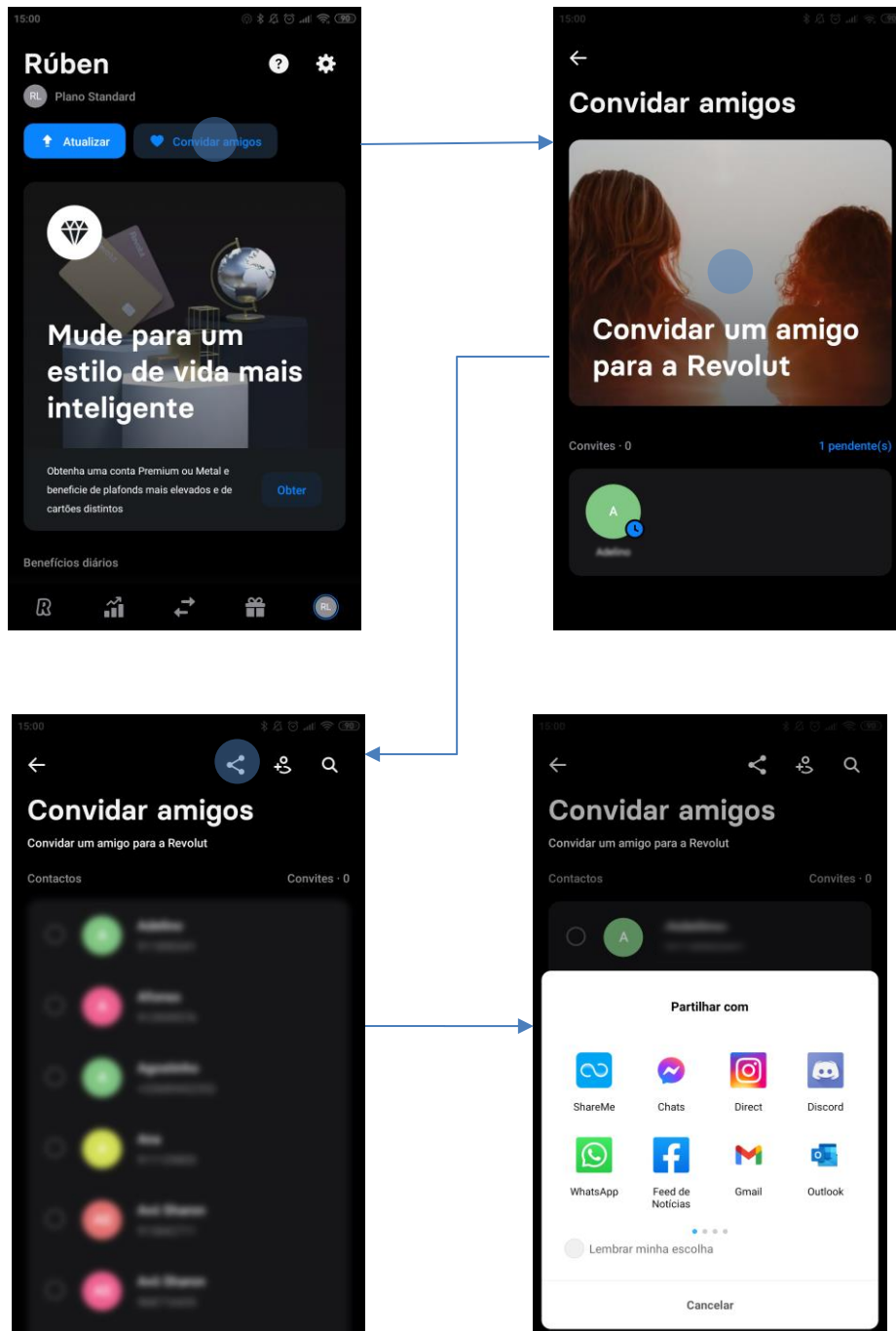


Figura 5 – Revolut refer a friend flow

3.2.2. Airbnb

A plataforma do Airbnb [17] tem como conceito base uma plataforma de reservas de hotéis ou estalagens. No entanto, introduziram uma grande inovação, através do conceito de partilha de casa própria entre os proprietários, com o papel de *hosts*, e convidados, com o papel de *guests*. Sendo assim, surge uma opção para além dos convencionais hotéis e propicia a oportunidade a pessoas que tenham quartos ou casas com espaço livre de obterem algum rendimento.

O Airbnb conta agora, com mais de 4 milhões de *hosts* e mais de 800 milhões de chegadas de convidados. Esta plataforma oferece ainda um programa de *refer a friend* com as seguintes características [18]:

- Partilha em diferentes canais de comunicação;
- Recompensa para ambos;
- 50€ de desconto para o convidado;
- Montante baseado na reserva feita para quem convida;
- *Tracking* dos convites;
- Limite de 5000€ ganhos.

A Figura 6 representa o *flow* para enviar um convite na aplicação do Airbnb.

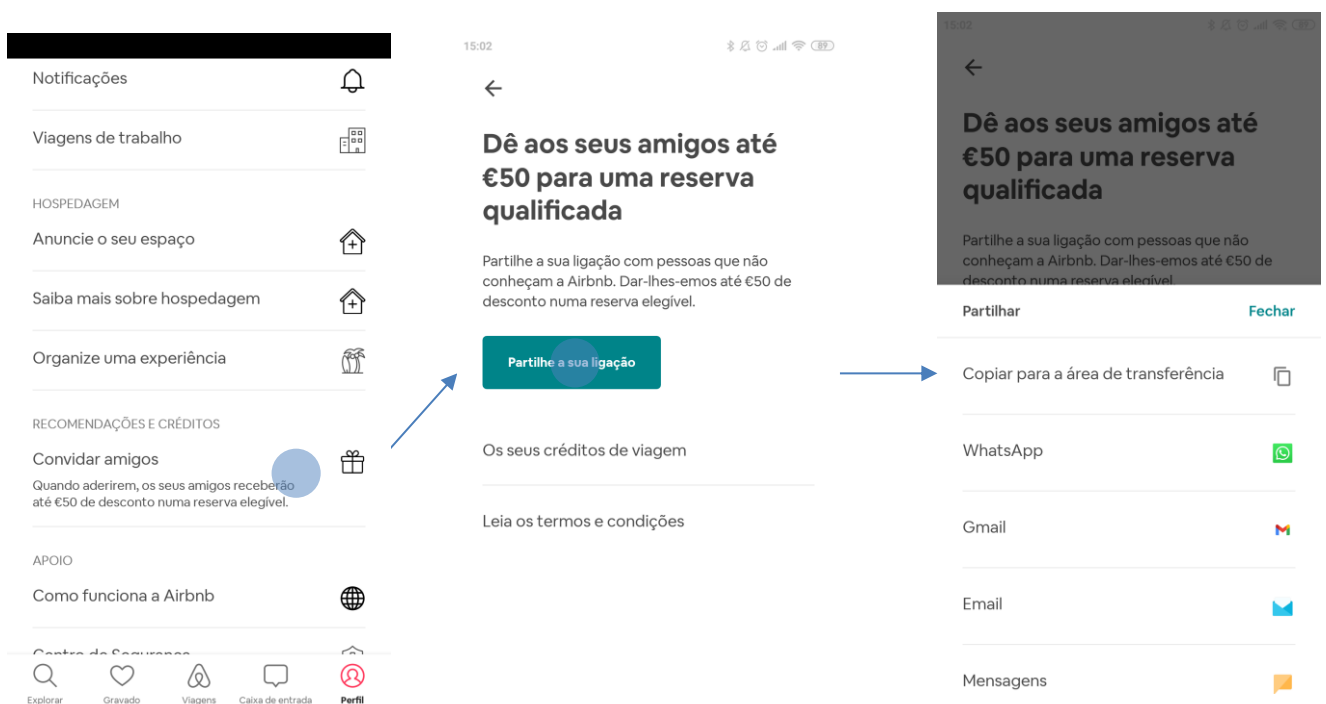


Figura 6 – Airbnb *refer a friend* flow

3.2.3. Dropbox

Baseado no *cloud computing*, a Dropbox [19] é considerada uma das pioneiras relativamente ao armazenamento de ficheiros em *cloud*, ou seja, num lugar remoto. Assim, possibilitando o acesso a ficheiros a partir de qualquer sítio com ligação à internet, tornando obsoletas as formas de armazenamento físico. Esta inovação permite ter diversas cópias de ficheiros em diferentes dispositivos e estando estes sempre sincronizados entre eles. A plataforma tem ainda um programa de *refer a friend* com as seguintes características [20]:

- Partilha por e-mail;
- Recompensa para ambos;
- Mais 500 MB de armazenamento para ambos;
- Não existe *tracking* dos convites;
- Limite para plano básico de 16GB e para premium de 32GB;
- Recompensas atribuídas depois do convidado, fazer o registo, instalar a aplicação, fazer login e validar o e-mail.

A Figura 7 representa o *flow* para enviar um convite na aplicação da Dropbox.

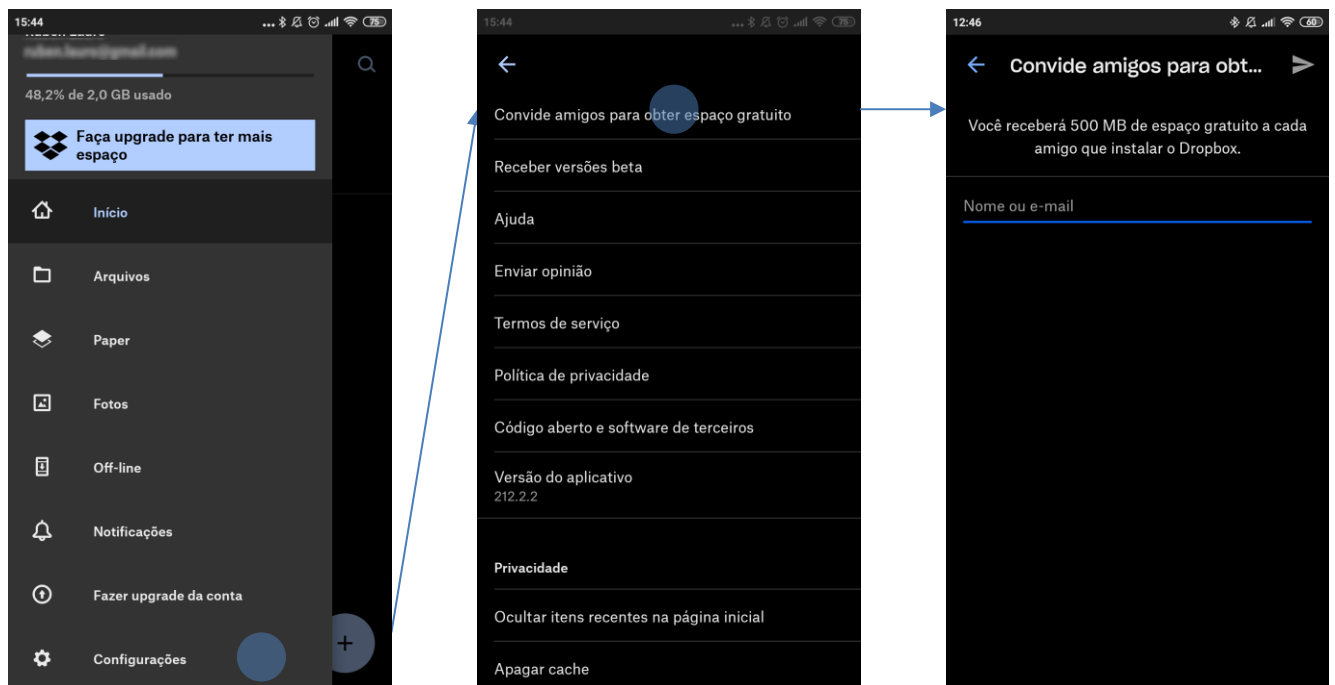


Figura 7 – Dropbox *refer a friend* flow

3.2.4. Tesla

A Tesla [21] é uma das empresas do bilionário Elon Musk, fundada em 2003, que tem como objetivo a redução do consumo de combustíveis fósseis, através da produção de veículos elétricos.

Para além dos veículos, a Tesla dedicou-se também, à produção de produtos de fabrico e armazenamento de energia, conseguindo assim, ter ainda mais impacto na condução do planeta em direção às energias limpas. Para quem adquire um veículo Tesla é lhe oferecida a vantagem de poder usufruir de um programa de *refer a friend* através da aplicação, com as seguintes características [22]:

- Partilha em diferentes canais de comunicação;
- Encomendas têm de ser feitas através do link partilhado;
- Recompensa para ambos;
- 1500 km quilómetros gratuitos de carregamentos nos tesla *superchargers*;
- Recompensas atribuídas depois do veículo ser entregue.

A Figura 8 representa a página de encomenda do veículo no seguimento do link partilhado.



Figura 8 – Página de encomenda do veículo

3.2.5. Evernote

Stepan Pachikov, criador de plataforma Evernote [23], tinha em mente três conceitos aquando da criação da mesma: lembranças, construção de ligações e criação de novas ideias. Assim surgiu o Evernote, uma plataforma, que permite aos utilizadores anotar informações. Esta simples plataforma tem mais de 225 milhões de utilizadores que tiram partido dos seus produtos/funcionalidades.

Para além de guardar notas, o Evernote permite ainda guardar imagens, ideias, projetos, etc., e tudo isto em apenas um lugar, o qual está sincronizado com todos os dispositivos para que nada se perca. A plataforma dá ainda a oportunidade aos utilizadores de participarem no programa de *refer a friend* com as seguintes características [24]:

- Partilha em diferentes canais de comunicação;
- Não existe *tracking* dos convites;
- Recompensas para ambos;
- Plano premium para o convidado;
- Pontos para comprar um plano premium para quem convida;
- Recompensas atribuídas depois do convidado fazer o registo através do *link*, instalar a aplicação e fazer o login;
- Os primeiros três convites completos dão 10 pontos depois disso são dados apenas 5 pontos.

A Figura 9 representa o *flow* para enviar um convite na aplicação do Evernote.

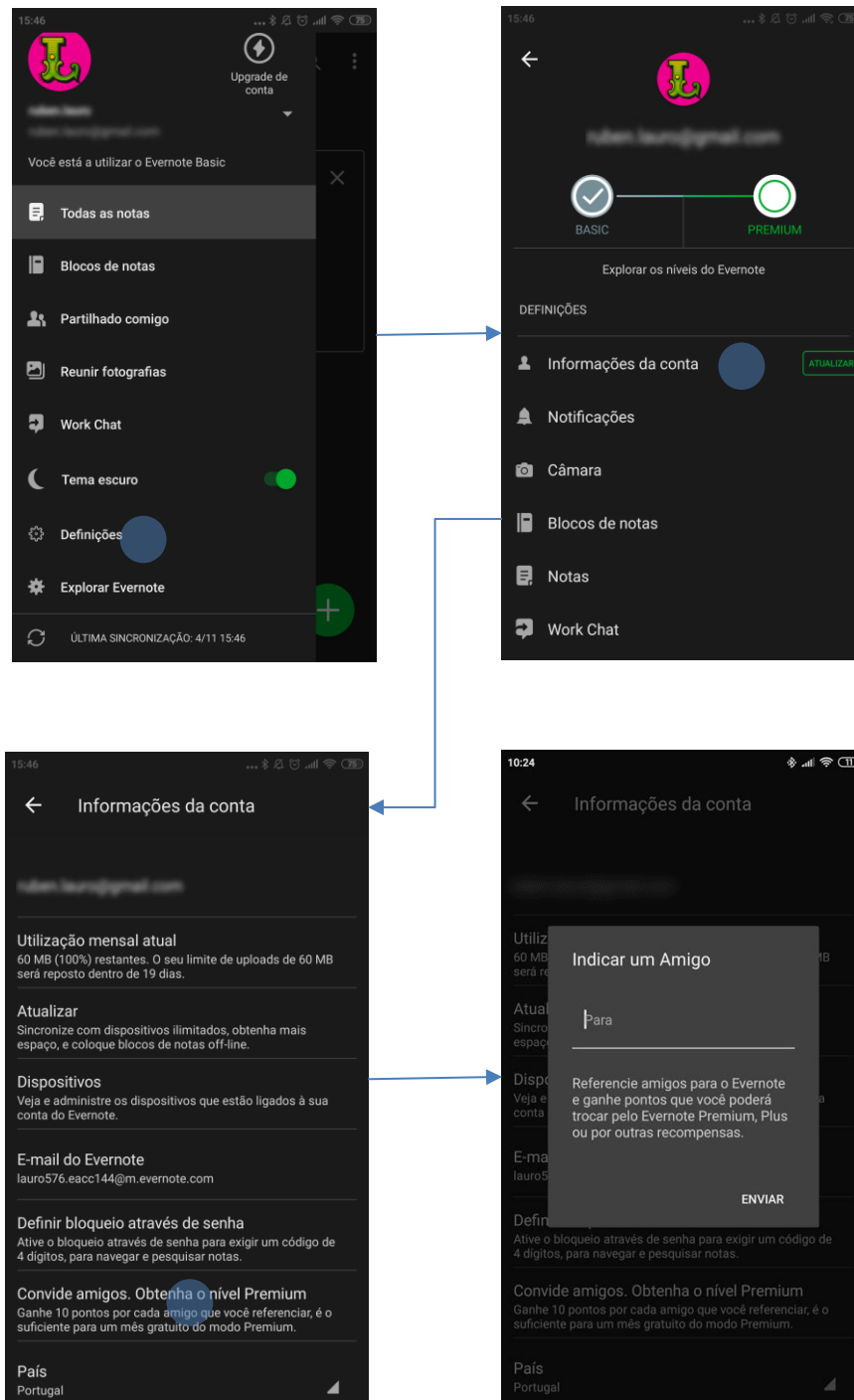


Figura 9 – Evernote refer a friend flow

3.2.6. World of warcraft

O World of Warcraft [25] é um jogo online do tipo *Massively Multiplayer Online Role-Playing Game* (MMORPG), que foi lançado em 2004 pela Blizzard Entertainment.

Em 2020, com mais de 113 milhões de assinantes e com cerca de 6 milhões de utilizadores diariamente, é assim considerado o MMORPG mais popular a nível mundial.

A Blizzard implementou uma estratégia que lhes permite a obtenção de mais utilizadores, produzindo ainda mais receitas, através de um programa de *refer a friend* que tira partido das qualidades do jogo, com as seguintes características [26]:

- Partilha manual do convite;
- Quem convida pode gerar um *link* (Expira a cada 30 dias);
- Cada *link* apenas pode conectar 4 pessoas por mês;
- Recompensas para ambos;
- Sistema de progressão para quem convida, ou seja, por cada mês de jogo que o convidado compra, dá a quem convida uma recompensa diferente até aos 12 meses. Depois disso, a cada 3 meses quem convida recebe um mês de jogo grátis;
- Um jogador convidado a comprar vários meses é igual a múltiplos convidados comprarem um mês;
- Limite de 10 pessoas.

A Figura 10¹ representa a interface de *refer a friend* do jogo World of Warcraft.

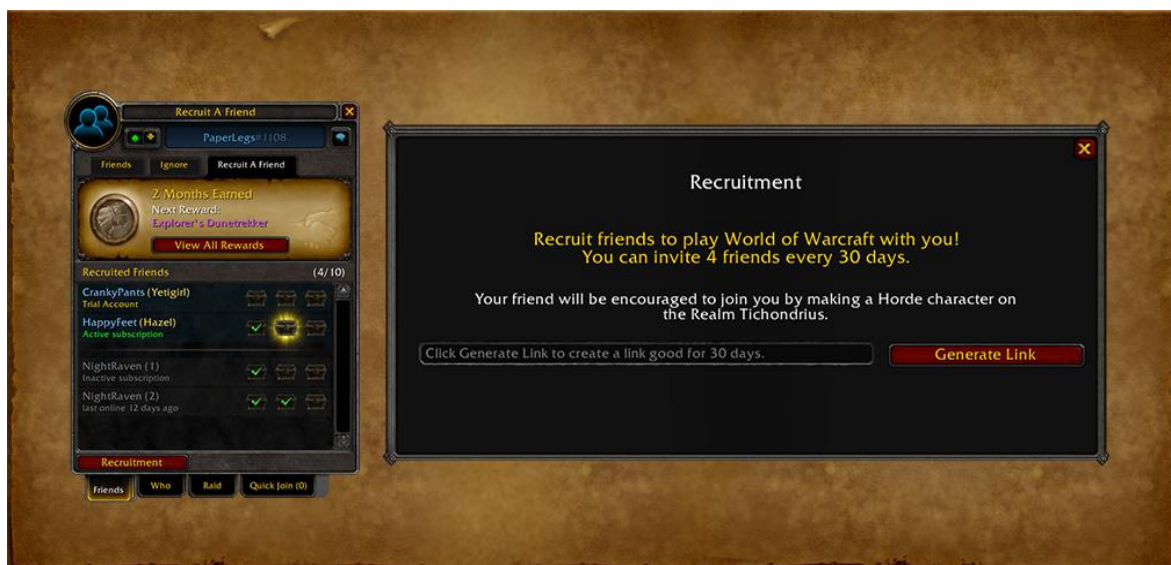


Figura 10 – Interface de *refer a friend* do jogo World of Warcraft

¹<https://images.blz-contentstack.com/v3/assets/blt3452e3b114fab0cd/bltc4f0ebf3d93fb9e1/5d8d1a441f27654da6f8eee2/enUS.jpg>

3.2.7. Fiverr

Em fevereiro de 2010, Micha Kaufman e Shai Wininger decidiram criar, utilizando o conceito de *marketplace*, uma plataforma para o comércio de serviços digitais, com o nome de Fiverr [27].

Esta plataforma é constituída por um modelo de taxas, ou seja, para cada serviço pago, existe uma parte que fica para a plataforma. Mais de 800 mil pessoas oferecem os seus serviços de *freelancing*, contando com cerca de 2.4 milhões de compradores diários.

Os utilizadores desta plataforma, têm ainda a oportunidade de ganhar e oferecer recompensas aos novos utilizadores com o programa de *refer a friend* com as seguintes características [28]:

- Partilha em diferentes canais de comunicação;
- Recompensas para ambos;
- Não existe *tracking* dos convites;
- 20% de desconto para o convidado numa compra até 100€;
- 20% do valor da primeira compra para quem convida;
- Recompensas atribuídas depois do convidado fazer o registo e fazer a primeira compra.

A Figura 11 representa o *flow* de *refer a friend* na aplicação do Fiverr.

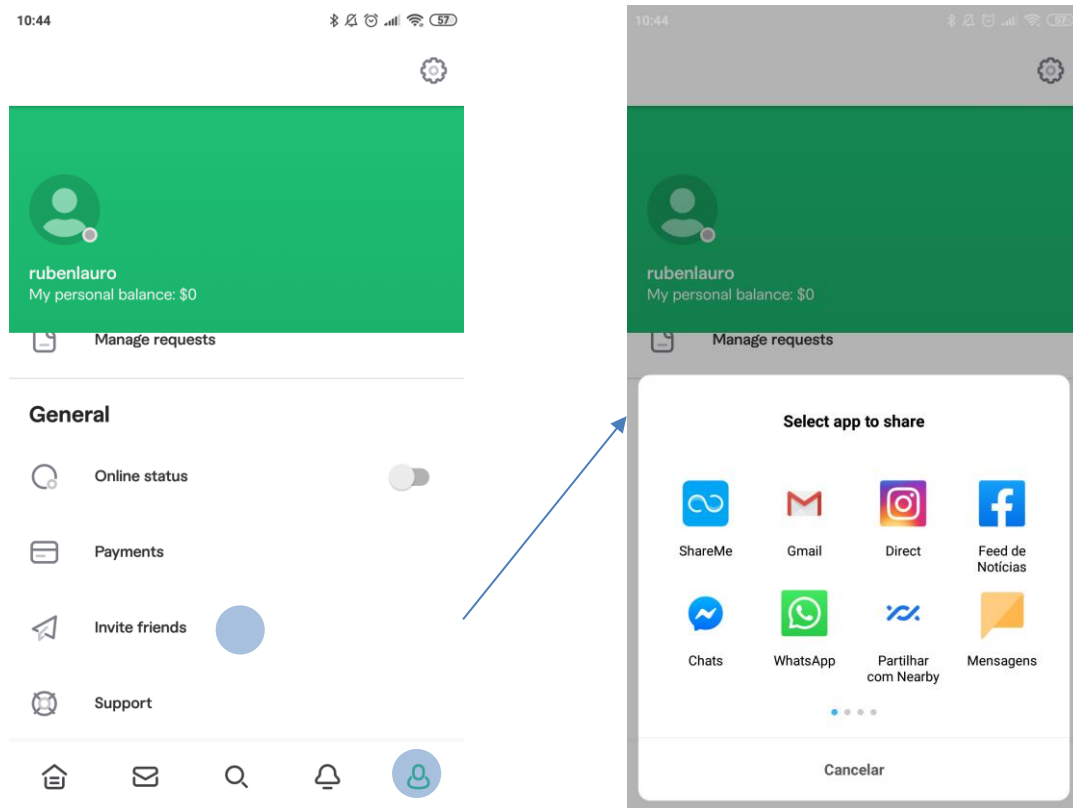


Figura 11 – Fiverr *refer a friend flow*

3.2.8. Humble bundle

O Humble bundle [29] é uma plataforma de comércio de *bundles*, como, por exemplo, *bundles* de jogos, *ebooks*, software e outros conteúdos digitais, que tem atualmente mais de 12 milhões de clientes.

Os criadores têm como principal objetivo, a redução de preços e ainda a atribuição de uma percentagem dos lucros para caridade, valor este que já atingiu 192 milhões de euros, desde 2010. Os preços dos *bundles* são atribuídos de acordo com os produtos que neles constam. A divisão dos lucros pode ser personalizada, através da seleção de diferentes percentagens atribuídas aos criadores, caridade e *humble bundle*.

Esta plataforma oferece ainda um plano de *refer a friend* com as seguintes características [30]:

- Partilha em diferentes canais de comunicação;
- Recompensa para quem convida;

- 50% do valor da primeira subscrição mensal ou 9% da subscrição anual;
- Recompensas atribuídas após o convidado subscrever um plano;
- Limite máximo de 30 pessoas;
- *Tracking* do número total de convites completos.

A Figura 12 representa o *flow* de *refer a friend* no website do Humble bundle.



Figura 12 – Humble bundle *refer a friend* flow

3.2.9. T-Mobile

A T-Mobile [31] é uma marca alemã, criada em 2002, que opera na área das telecomunicações. A marca atua em várias partes da europa, mas é nos Estados Unidos que possui mais clientes. Atingiu no terceiro quadrimestre de 2020 os 100 milhões de clientes.

O seu plano “Un-carrier” tornou-se responsável pela aquisição de muitos clientes, pelo facto de facilitar a mudança de operadora. Atualmente, a operadora aposta fortemente nas redes de nova geração, o 5G.

Esta possui ainda um plano de *refer a friend* como forma de tentar chegar a mais clientes com as seguintes características [32]:

- Partilha em diferentes canais de comunicação;
- *Tracking* de convites;

- Recompensas para quem convida;
- Cartão pré-pago com 50€;
- Recompensas atribuídas depois do registo e manter a conta em bom estado durante 30 dias;
- Limite máximo de 500€ por ano (10 *referrals*).

3.2.10. Three

A empresa de telecomunicações Three [33] foi criada em 2003, com início no Reino Unido, utilizando apenas a rede 3G. O conceito desta empresa baseia-se em criar uma rede que nasceu para ser diferente.

Atualmente, esta empresa conta com mais de 10 milhões de clientes, sendo líder em relação ao espectro 5G, tendo sido esta a única que conseguiu atingir os requisitos estabelecidos pela União Internacional de Telecomunicações (UIT), em 2020.

Esta empresa, possui um programa de *refer a friend* com as seguintes características [34]:

- Partilha em diferentes canais de comunicação;
- *Link* único para cada convite;
- As encomendas têm de ser feitas através do *link* enviado para serem elegíveis;
- Quem envia o convite pode escolher para quem vão as recompensas;
- Recompensa é um montante base do produto comprado;
- Recompensas atribuídas depois do registo e da primeira fatura ser paga por completo e dentro do prazo estipulado.

A Figura 13 representa o *flow* de *refer a friend* no *website* da Three.

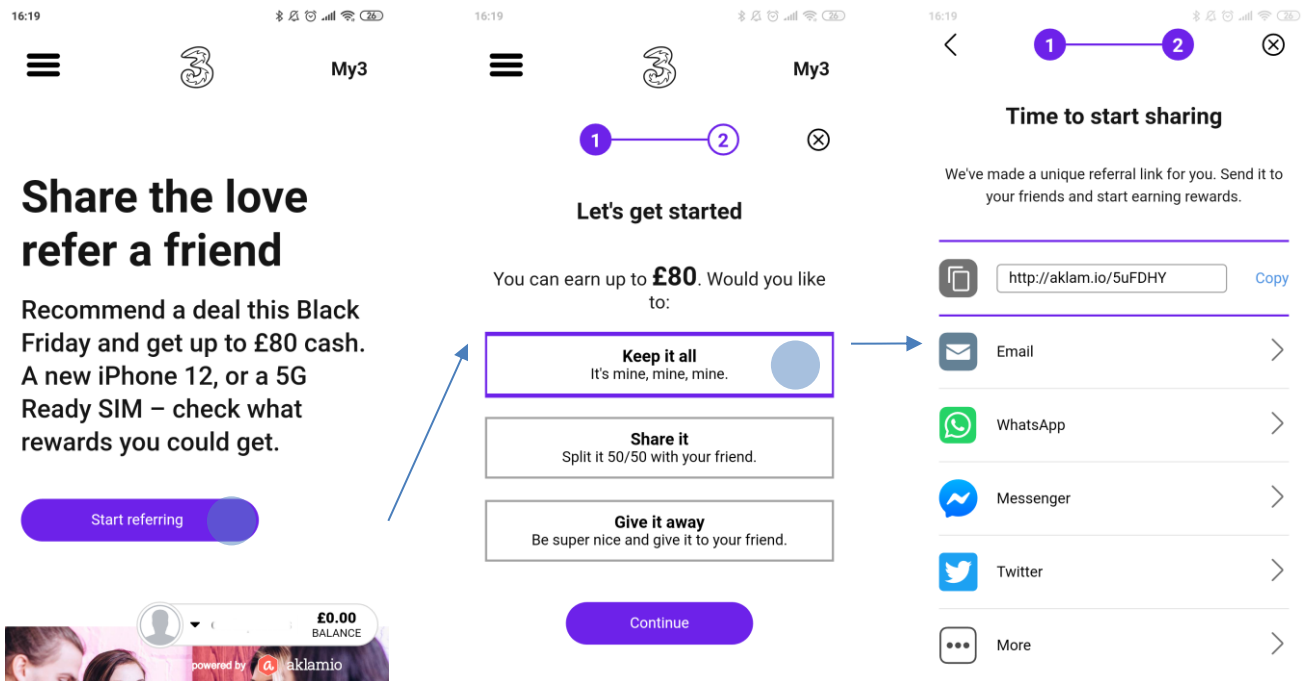


Figura 13 – Three refer a friend flow

3.2.11. Mint Mobile

A Mint Mobile [35] é uma empresa de telecomunicações americana, criada em 2015. Esta empresa, tem uma característica própria marcada pelo facto de ser uma operadora que não tem uma rede própria, ou seja, esta opera perante a rede de outra operadora, que neste caso é a T-Mobile. Assim permite, por exemplo, a utilização da infraestrutura 5G da T-Mobile, o que lhe traz um valor acrescido.

Os seus clientes são incentivados a efetuar pagamentos em massa por um preço mais baixo, ou seja, os clientes podem subscrever um plano por 3 meses, em vez de ser apenas por um mês.

Para além disto, os clientes podem beneficiar de um plano de *refer a friend* com as seguintes características [36]:

- Partilha em diferentes canais de comunicação;
- Recompensas para ambos;
- Crédito de 15\$ para o convidado;
- Sistema progressivo para quem convida (1 = 25\$, 2 = 30\$, 3 = 35\$, 4 = 40\$, 5 = 110\$);

- Limite máximo de 10 convites por ano de calendário;
- *Tracking* de convites completos por e-mail.

A Figura 14² representa o *flow* de *refer a friend* na aplicação da Mint Mobile.

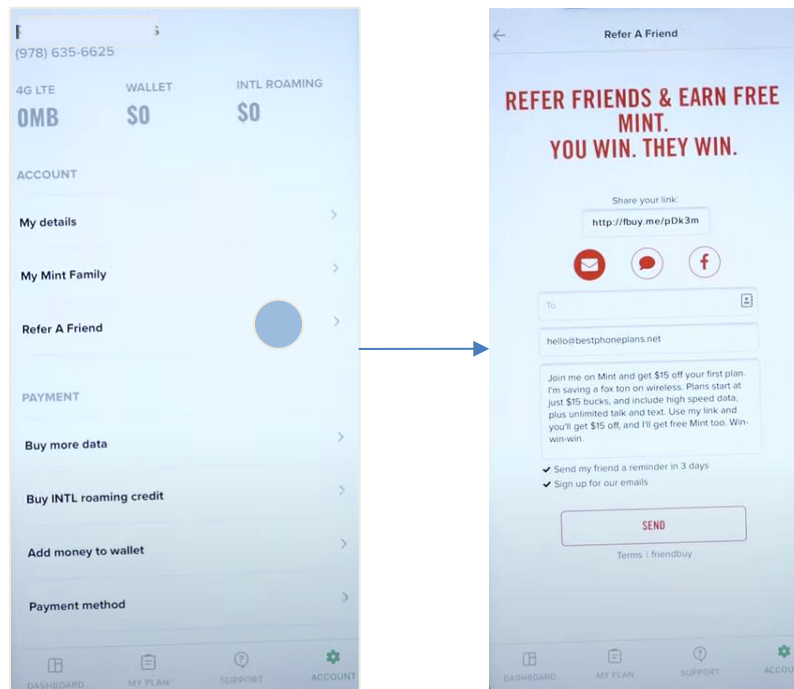


Figura 14 – Mint Mobile *refer a friend* flow

3.3.Link & Código

A combinação dos dois métodos, *Link* e *Código*, é adotada por algumas plataformas devido a aumentar o número de utilizadores abrangidos pelos programas. Por exemplo, um utilizador que já tenha uma conta criada, com a introdução manual do código, pode continuar a beneficiar dos programas sem ter de ser obrigado a criar uma conta nova. No entanto, o facto de existir uma combinação de métodos, faz com que a complexidade de desenvolvimento aumente, o que se traduz em maior tempo despendido e consequentemente um aumento nos custos. A existência dos dois métodos permite a introdução manual do código, no entanto, se for seguido o *link*, o código é associado automaticamente.

3.3.1. Uber

A Uber [37] tem sede nos Estados Unidos e foi uma das empresas mais revolucionárias no que toca a partilhas de carro. Apesar de mensalmente ter cerca de 78

² <https://www.youtube.com/watch?v=AOmkDILLWvE&t=1s>

milhões de utilizadores, foi muito criticada devido à perturbação do funcionamento normal dos antigos táxis.

A plataforma da Uber deu oportunidade a pessoas de usarem os seus carros pessoais para transportarem passageiros. Oferecendo, um sistema constituído por duas aplicações, uma para os condutores e outra para os passageiros. Os passageiros estão à distância de poucos toques para conseguirem uma viagem do ponto A para o ponto B.

A aplicação para os condutores, possui um programa de *refer a friend* com as seguintes características:

- Partilha em diferentes canais de comunicação;
- Recompensas para quem convida;
- Recompensas são oferecidas depois de o condutor completar as primeiras viagens;
- *Tracking* de convites.

A Figura 15 representa o seguimento de um link partilhado pela aplicação de condutores [38].

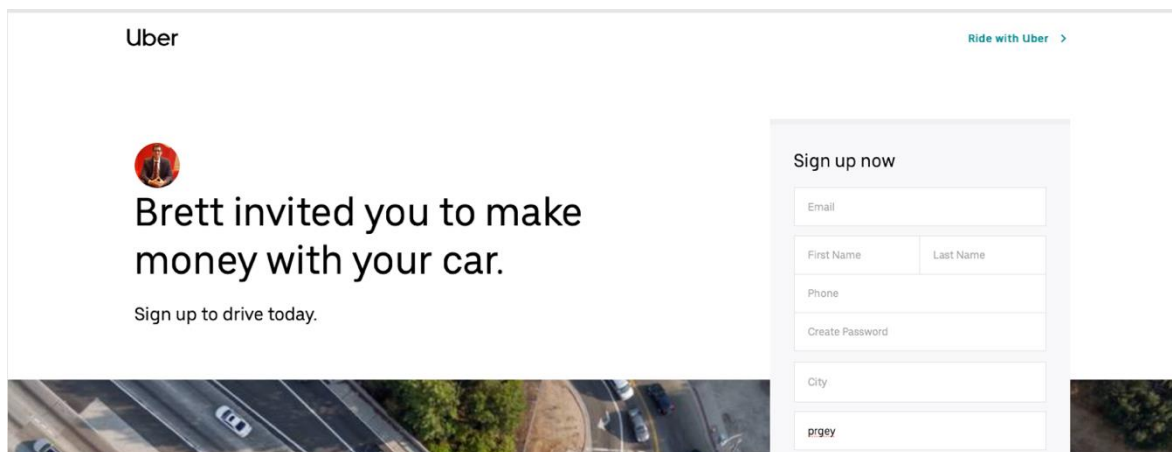


Figura 15 – Seguimento de um link partilhado pela aplicação Uber condutor

3.3.2. N26

O N26 [39] é uma solução bancária 100% digital. Com cerca de 7 milhões de clientes, a plataforma foi criada com o objetivo de criar um banco o mais descentralizado possível. Esta solução tem como serviço base para qualquer cliente, uma conta corrente com a possibilidade de fazer qualquer tipo de transação e ainda com a oferta de um cartão físico com cor, a qual pode ser escolhida pelo cliente.

A aplicação permite ainda uma configuração adaptada ao cliente, podendo este criar as próprias categorias nas quais pretende gastar dinheiro, pode ainda partilhar estas categorias com outros clientes de forma a, por exemplo, dividir a despesa de um jantar. Existe ainda um espaço de estatísticas que permite ao cliente ter a perceção de onde gastou o seu dinheiro ao longo de um determinado período de tempo.

Para além de todas as funcionalidades, os clientes têm ainda a oportunidade de tirar partido de um programa de *refer a friend*, com as seguintes características [40]:

- Partilha em diferentes canais de comunicação;
- Recompensas para quem convida;
- Recompensas atribuídas depois de o convidado completar o registo, ativar o cartão físico e fazer uma transação;
- *Tracking* dos convites;
- Limite de 1500€.

A Figura 16 representa o registo no seguimento do link, no entanto é possível alterar o código.

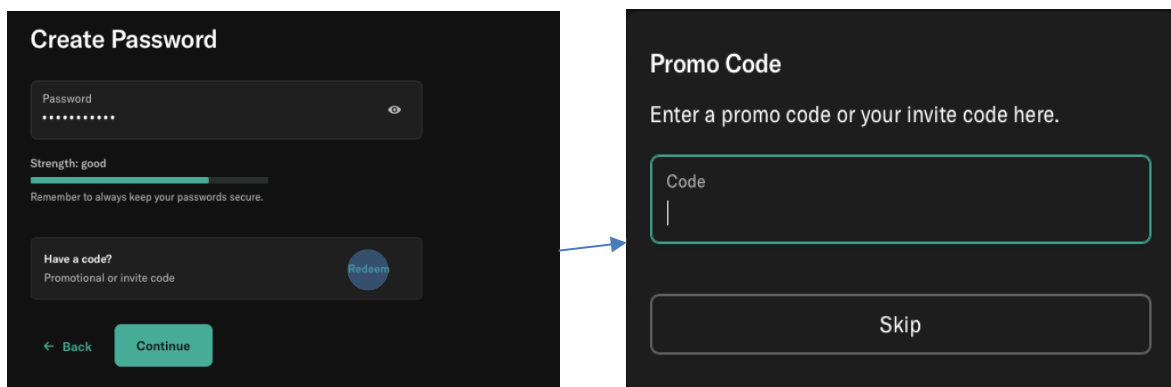


Figura 16 – Seguimento do link no website do N26

3.3.3. Lyft

Depois da Uber surgiram muitas empresas com o mesmo propósito, transportar pessoas do ponto A para o ponto B, sendo condutores autónomos.

A Lyft [41] opera principalmente nos Estados Unidos, contando com cerca de 20 milhões de clientes ativos. Os clientes e os condutores estão ligados através de uma aplicação, que lhes permite pedir uma viagem através de poucos toques.

A aplicação para os clientes possui um programa de *refer a friend*, programa este que também já esteve presente na aplicação da Uber, neste caso chama-se Lyft *free rides* e tem as seguintes características [42]:

- Partilha em diferentes canais de comunicação;
- Recompensas para ambos;
- Crédito na aplicação baseado em diversos fatores tais como a localização do utilizador;
- Limite de 2000€ por semana.

A Figura 17 representa o *flow* de *refer a friend* na aplicação da Lyft.

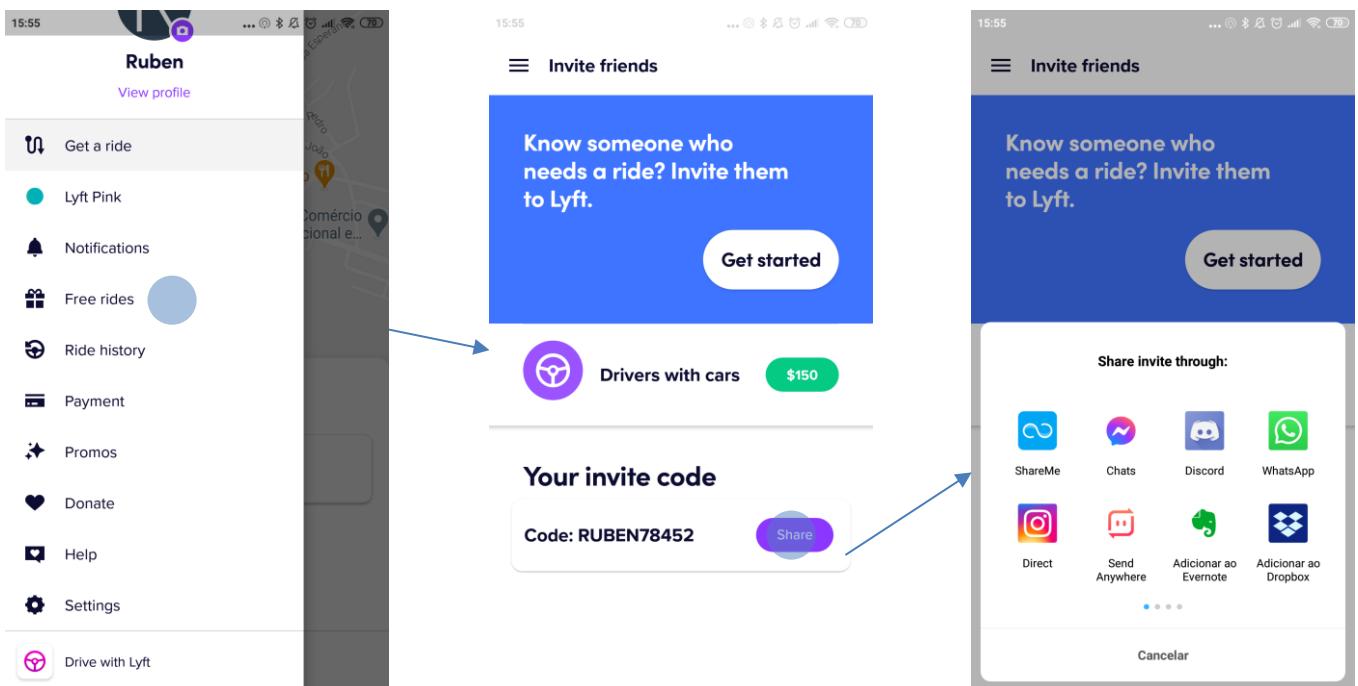


Figura 17 – Lyft *refer a friend* flow

3.3.4. Wish

A Wish [43] tem como modelo de negócio o e-commerce. Foi criada em 2010 e atualmente encontra-se em mais de 100 países. É uma das maiores lojas *online* operando apenas em formato *mobile*, vende uma quantidade avultada de produtos, nomeadamente mais de 150 milhões de produtos. Muitos deles com preços muito reduzidos.

Ao longo da utilização da aplicação, o sistema de sugestões vai ficando mais apurado recomendando aos utilizadores os produtos que estes querem ver.

É também na aplicação que os utilizadores podem tirar partido de um programa de *refer a friend* com as seguintes características:

- Partilha em diferentes canais de comunicação;
- Campanhas diárias;
- Recompensas para ambos;
- 5€ de crédito para o convidado;
- 9€ de crédito para quem convida;
- Recompensa instantânea após o registo para o convidado;
- Recompensa para quem convida só depois do convidado completar a primeira encomenda;
- Limite de 10 pessoas por campanha.

A Figura 18 representa o *flow* de *refer a friend* na aplicação da Wish.

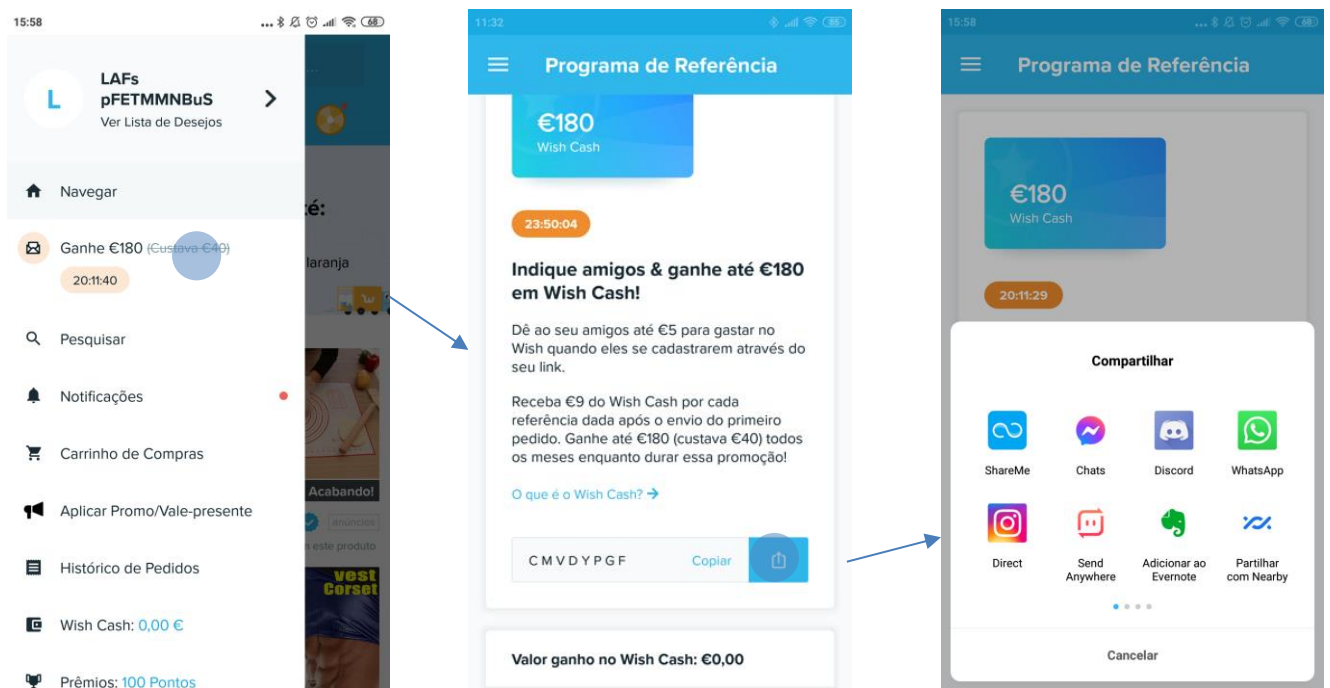


Figura 18 – Wish *refer a friend* flow

3.3.5. Hotel Tonight

A plataforma Hotel Tonight [44] foi fundada em 2010, inicialmente operava apenas nos Estados Unidos e somente para utilizadores com dispositivos Apple. É conhecida por praticar preços baixos para reservas de última hora em hotéis. Em 2019, foi comprada pela Airbnb, tornando-se assim uma subsidiária da mesma.

Os utilizadores desta plataforma podem ter a oportunidade de convidar um amigo para se juntar à aplicação através do seu programa de *refer a friend* com as seguintes características [45]:

- Partilha em diferentes canais de comunicação;
- Recompensas para ambos;
- 20€ para ambos;
- Recompensa atribuída a quem convida depois de o convidado fazer uma reserva no valor mínimo de 120€;
- *Tracking* de convites concluídos.

A Figura 19 e a Figura 20 representam os *flows* de *refer a friend* na aplicação do Hotel Tonight.

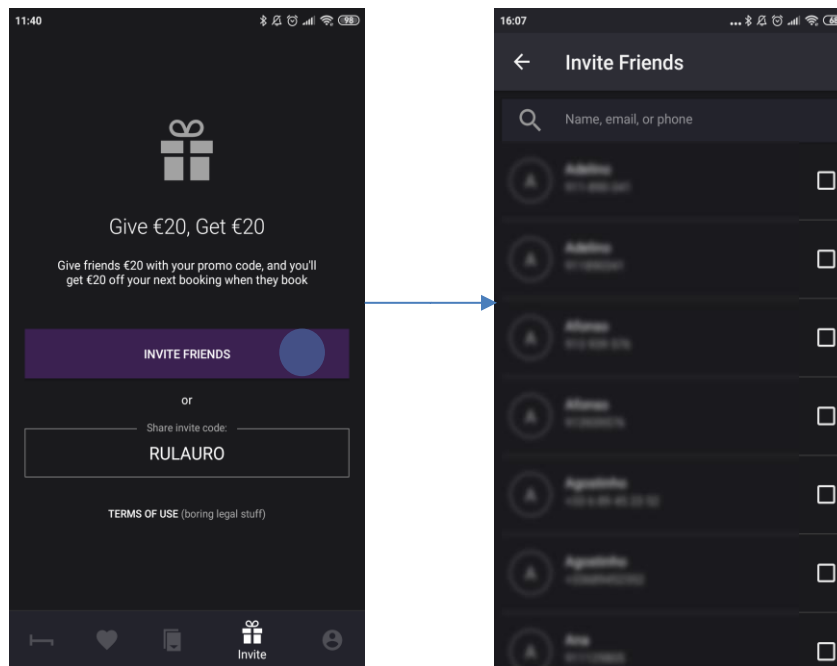


Figura 19 – Hotel Tonight *refer a friend* flow

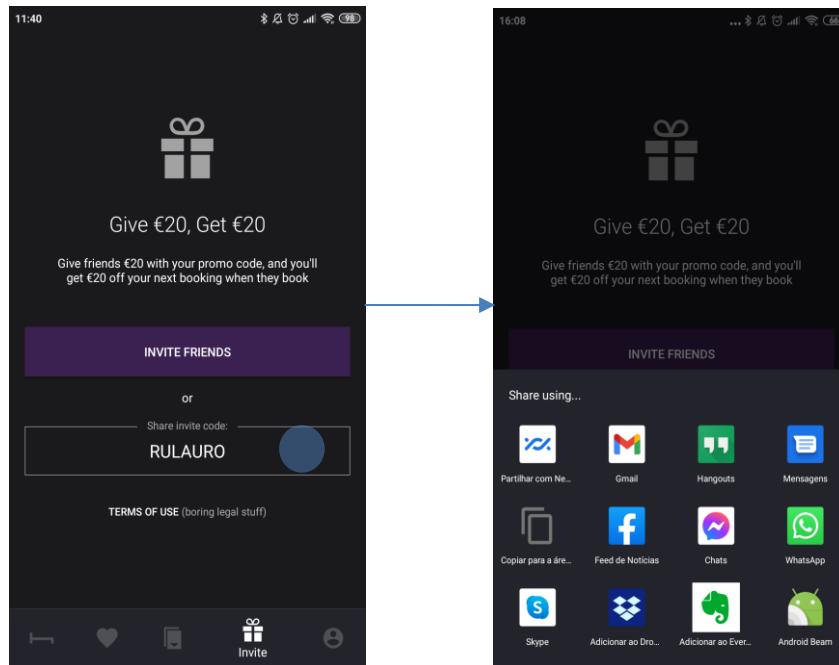


Figura 20 – Hotel Tonight refer a friend social sharing flow

3.4. Resumo

Após a análise das diferentes aplicações/plataformas foi possível concluir que os programas de *refer a friend* variam em diversos parâmetros. Não só no método como foram divididas as aplicações/plataformas inicialmente, mas também, na forma de partilha, forma de recompensa (só para o convidado, só para quem convida ou para ambos), tipo de recompensa, mecanismo de recompensa (prémios únicos, agregados e progressivos), existência de *tracking* dos convites, existência de limite e também os diferentes passos até serem atribuídos os prémios.

Assim surge a necessidade da existência de diferentes campanhas, tendo estas diversos parâmetros de configuração, sendo a maior parte deles mutáveis mesmo que a campanha já se encontre a decorrer. As alterações podem trazer uma maior ou menor adesão ao programa.

Por fim, o estudo do estado da arte permitiu concluir que a solução a desenvolver deveria incluir um grande número de parâmetros configuráveis pelo cliente, de modo a que estes pudessem criar campanhas e alterá-las quando lhes fosse conveniente e sem necessidade de recorrer a novos desenvolvimentos, o que se demonstrou ser uma grande mais valia na implementação do projeto.

4. Planeamento e Metodologia de Desenvolvimento

Neste capítulo será descrito tanto o planeamento, como a metodologia de desenvolvimento utilizada na implementação do programa de *refer a friend* que se pretende que seja utilizado por qualquer aplicação, seja ela Android ou iOS.

4.1. Planeamento

O presente estágio teve início com a ambientação ao contexto da WIT como empresa, tendo sido iniciadas as apresentações dos elementos principais de acompanhamento ao estágio, assim como, o contexto do projeto a ser desenvolvido.

Em seguida foi iniciado o estudo de mercado, estudo este apresentado anteriormente no capítulo 3. O estudo permitiu extrair diversas conclusões em relação ao funcionamento de diversos programas de *refer a friend*, o que foi levado para o próximo passo, o levantamento de requisitos.

Foi no levantamento de requisitos, que foram definidas as principais funcionalidades da plataforma, assim como, os principais mecanismos de funcionamento. Para uma melhor compreensão, nesta fase foram elaborados os protótipos de baixa fidelidade, representados no Anexo A – Protótipos de baixa fidelidade, e modelados os processos de negócio, representados no Anexo B – Processos de negócio. Estes apresentam-se divididos por aplicação, a aplicação móvel e o *Backoffice*.

É importante realçar que os protótipos produzidos contemplavam uma plataforma isolada e genérica, de maneira a servir as necessidades de qualquer cliente, podendo posteriormente ser integrada com as restantes plataformas do mesmo. Para além disso, os protótipos apresentavam ainda a possibilidade de a plataforma ser gerida por parte da WIT ou pelo cliente.

Os protótipos desenvolvidos foram posteriormente utilizados pela equipa de *design* da WIT para serem criados os protótipos de alta fidelidade. Estes foram concebidos tendo como base uma plataforma específica em desenvolvimento na empresa, tendo como consequência, a remoção de algumas das funcionalidades especificadas no levantamento de requisitos. Esta alteração teve como finalidade a adequação das funcionalidades a desenvolver ao tempo de estágio e à plataforma em questão.

Após a etapa de conceção dos protótipos de alta fidelidade da aplicação móvel iniciou-se a fase de desenvolvimento da interface do utilizador (UI). Inicialmente, foi planeado um desenvolvimento *cross platform* como por exemplo, utilizando as *frameworks* Flutter ou React Native, no entanto, devido ao suporte por parte do projeto já em desenvolvimento optou-se pelo desenvolvimento nativo.

A próxima etapa, teve como objetivo implementar os mecanismos necessários para tornar mais real o mecanismo de *refer a friend*, este foi alcançado através do desenvolvimento da *Application Programming Interface* (API) que dá suporte à UI implementado. Ao longo deste desenvolvimento, foi também implementada a parte funcional da UI que permite a comunicação com a API.

Por fim, foi produzido o protótipo de alta fidelidade do *Backoffice* e de seguida o desenvolvimento do mesmo, para que seja possível a gestão e análise das campanhas pertencentes ao programa de *refer a friend*.

Durante o decorrer do estágio foram utilizadas ferramentas de controlo de versões para que todo o desenvolvimento estivesse sincronizado e disponível para todas as pessoas que estavam a acompanhá-lo.

A Figura 21 representa as tarefas principais elaboradas durante o estágio, apresentando a duração aproximada aos meses, para os quais o estágio decorreu. No Anexo C – Plano de Estágio, encontra-se o plano mais detalhado.

Scope	Status	Set	Out	Nov	Dez	Jan	Feb	Mar	Apr	May	June
Internship setup	Completed										
Benchmark	Completed										
Process mapping	Completed										
UI App development (Android)	Completed										
Understanding of Backend	Completed										
Backend/App (Android)	Completed										
Understanding of Backoffice	Completed										
Backoffice/Backend	Completed										
Improvements	Completed										
App UI em react native	Completed										
Code review	Completed										
Deliverables preparation	Completed										

Figura 21 – Plano de estágio resumido

4.2. Metodologia de Desenvolvimento

Desde o início do desenvolvimento de software sempre existiu uma tentativa para criar software através de uma determinada ordem de eventos. Assim foram usadas diversas metodologias de desenvolvimento para tentar delinear os diferentes passos a seguir para desenvolver software com sucesso. Apesar dos passos estarem definidos, cada empresa pode

decidir como utilizar as metodologias de desenvolvimento, criando muitas vezes uma metodologia única adequada às suas necessidades, através da utilização de diferentes passos de uma ou mais metodologias [46].

As metodologias de desenvolvimento podem ser divididas em dois grupos: tradicionais e ágeis.

As metodologias consideradas “tradicionais” baseiam-se numa arquitetura muito rígida de acontecimentos, existindo a necessidade de completar um determinado acontecimento para ser possível prosseguir para o próximo. Isto faz com que a alteração de requisitos exija um esforço bastante maior, sendo assim, as metodologias tradicionais são aplicadas muitas vezes em projetos para os quais os requisitos são todos definidos inicialmente. O facto de existirem todos os passos obrigatórios tem como consequência uma redução do ritmo do projeto, pois o intervalo temporal até ser atingida a etapa de desenvolvimento é grande [47].

Para estas metodologias pode considera-se, por exemplo, a metodologia em cascata ou *waterfall*, no qual é feito todo um estudo, planeamento e documentação, antes do desenvolvimento, o que faz com que sobressaiam as desvantagens referidas no parágrafo anterior.

As metodologias ágeis, trazem a vantagem de suportarem, de forma mais otimizada, alterações nos requisitos no decorrer do desenvolvimento, devido à utilização de iterações de desenvolvimento. Durante as iterações é requisitada a presença de pessoas importantes para o projeto, de modo, a obter o maior *feedback* possível, para que na iteração seguinte possam ser melhorados os pontos apontados com objetivo de atingir o resultado final, pretendido pelo cliente.

Estas permitem, ainda, uma redução do tempo do retorno do investimento, pois através da utilização destas iterações é possível apresentar algo funcional mais cedo. Todos estes processos, em caso de sucesso, são transformados em satisfação para o cliente, sendo esta considerada um ponto central destas metodologias. No entanto, existe outro aspeto, muitas vezes importante, mas mais desvalorizado neste tipo de metodologias, a documentação [47].

Assim, as empresas tendem a adaptar a metodologia a usar consoante os requisitos do projeto a desenvolver.

Uma das metodologias ágeis bastante utilizada é o Scrum. O Scrum [48] consiste, numa equipa composta por determinados cargos (*Product owner*, *Scrum master* e Equipa de desenvolvimento), eventos (*Sprint*, *Sprint planning*, *Daily scrum*, *Sprint Review*, *Sprint Retrospective*), artefactos (*Product Backlog*, *Sprint Backlog* e *Increment*) e algumas regras. Cada componente, mencionado anteriormente, tem um objetivo específico, operando em conjunto de modo a formar um ciclo, que se encontra representado na Figura 22³.

SCRUM FRAMEWORK

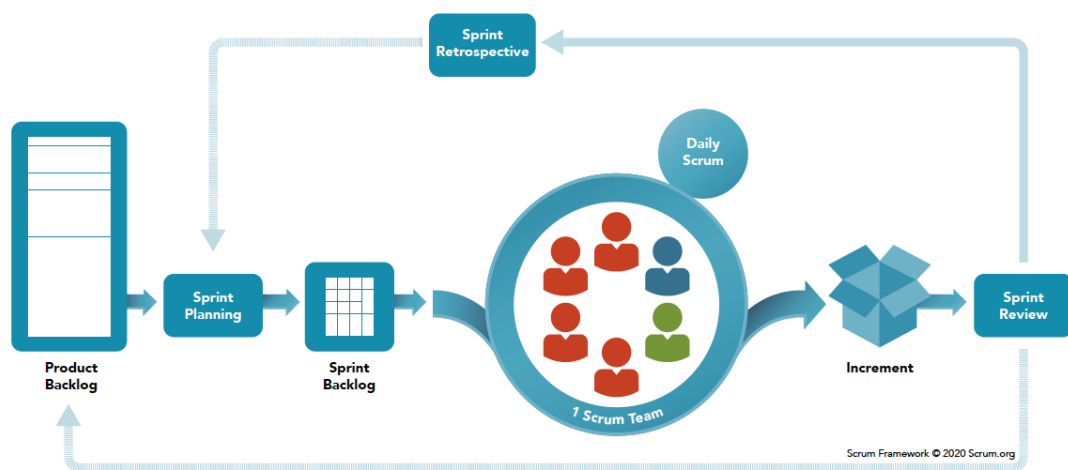


Figura 22 – Representação do ciclo scrum

Resumidamente, o processo começa pelo preenchimento do *Product Backlog* com as funcionalidades a desenvolver para o determinado projeto. Seguidamente, é dado o início do *sprint*, este é o evento principal do Scrum que engloba todos os outros eventos mencionados acima. Todos os eventos têm tempo limite para que não seja desperdiçado tempo numa determinada tarefa.

O *Sprint* tem como principal objetivo criar um incremento para o projeto a desenvolver, para isso para cada *sprint* existe a etapa inicial designada por *Sprint Planning*, que consiste na definição do plano para o incremento a ser entregue no final da próxima *Sprint*. Deste evento resulta o *Sprint Backlog*, que é um conjunto de funcionalidades escolhidas para serem

³ <https://scrumorg-website-prod.s3.amazonaws.com/drupal/2021-01/Scrumorg-Scrum-Framework-tabloid.pdf>

utilizadas pela equipa de desenvolvimento para ser atingido o objetivo planeado para a iteração.

De modo a melhorar a comunicação entre a equipa de desenvolvimento é realizado o *Daily Scrum*. Este é uma reunião diária de curta duração que tem como objetivo fazer o ponto de situação do dia anterior, e como proceder no dia seguinte de forma a ajudar a equipa da melhor maneira.

No final do *Sprint*, são feitos dois eventos de avaliação do trabalho realizado, um deles é o *Sprint Review* no qual participam todos os elementos, a equipa Scrum constituída pelo *Scrum master* e a equipa de desenvolvimento, e o *Product owner*.

O *Sprint Review* permite concluir quais as funcionalidades que devem ser marcadas como “Terminadas” no *Product Backlog* através de demonstrações por parte da equipa de desenvolvimento, para além disso são também debatidos quais os próximos passos. O outro evento é o *Sprint Retrospective*, no qual apenas participa a equipa Scrum e permite criar um plano para as melhorias a serem realizadas na próxima iteração.

A metodologia utilizada pela WIT é uma adaptação do Scrum, sendo assim despromeu-se a necessidade de decisão de entre as demais metodologias, tendo sido aplicada, também, uma adaptação ajustada da metodologia para o contexto de estágio. De modo que, alguns dos eventos não existiram ou tiveram o seu propósito alterado. Os cargos de equipa de desenvolvimento, *Scrum master* e *product owner* foram definidos da seguinte forma:

- A equipa de desenvolvimento era constituída por apenas um elemento, eu;
- O *Scrum master* era o tutor;
- O *Product owner*, era o supervisor do estágio;

Maioritariamente, os *sprints* tinham duração de cerca de 1 semana, exceto alguns *sprints* que tiveram uma duração mais elevada devido à indisponibilidade do *Scrum master* ou do *product owner*.

Os *Sprint Reviews* eram eventos apenas realizados com a presença do *Product owner* e com mais alguns elementos relacionados com o projeto no qual o programa foi inserido. Estes eram realizados quando um certo conjunto de funcionalidades eram marcadas como

“Terminadas”. O principal objetivo era demonstrar o trabalho realizado e definir os próximos passos a seguir.

Depois de cada *sprint* eram realizadas as *Sprint Retrospectives*, na qual participava a equipa de desenvolvimento e o *Scrum master*, onde era efetuada a demonstração do realizado na *sprint* passada e planeada a próxima. Assim, esta etapa substituía o *sprint planning*.

Por fim, os *daily Scrums* não eram realizados, existindo apenas alguns meetings esporádicos se existisse algum problema ou sugestão de alteração relacionada com o *sprint* atual.

4.3. Resumo

Neste capítulo foi demonstrado o plano de estágio, explicitando qual a ordem pela qual as tarefas foram desenvolvidas de modo a atingir o objetivo final e apresentando ainda em formato de tabela, de forma reduzida, as tarefas globais em conjunto com a duração das mesmas. Foi também apresentada uma breve análise das metodologias de desenvolvimento e da forma como se aplicaram ao estágio.

5. Análise de Requisitos e Protótipos

O capítulo seguinte irá conter a especificação da análise de requisitos, com o objetivo de apresentar as principais funcionalidades a desenvolver, seguindo-se a descrição do processo de prototipagem.

5.1. Análise de Requisitos

A etapa de estudo de mercado teve uma grande influência na etapa de análise de requisitos, porque foi através da análise obtida neste estudo que se formulou uma hipótese de possível solução para a implementação do mecanismo de *refer a friend*, para uma determinada aplicação.

Após esta etapa foi possível concluir que seria mais simples se existisse a divisão dos requisitos nos dois papéis dos participantes do processo, ou seja, quem envia o convite, o remetente, e quem recebe, o recetor, na definição dos requisitos para a aplicação móvel. Para além disso, de forma, a ter uma melhor perceção dos requisitos, foram também elaborados os processos de negócio utilizando a ferramenta bpmn.io [49]. A Figura 23 representa um exemplo de processo de negócio, que será explicado de seguida.

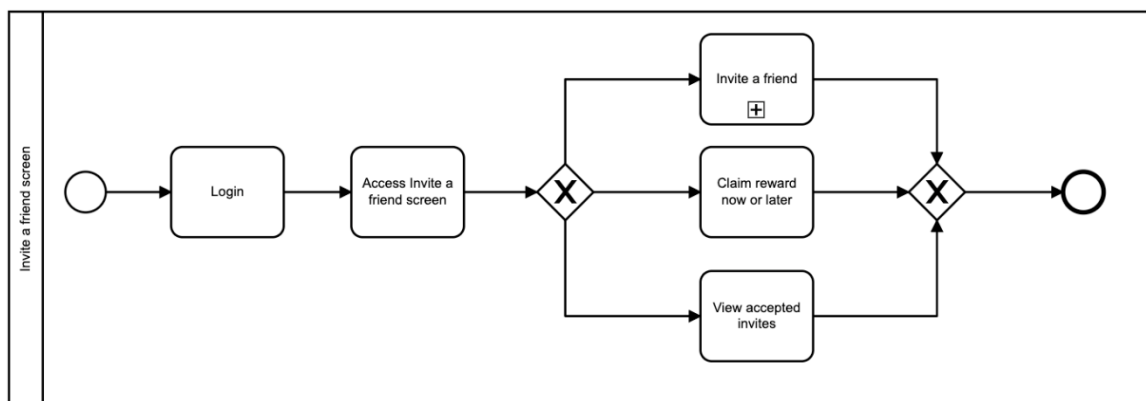


Figura 23 – Diagrama de processo de negócio de *Refer a friend*

Começando pelo remetente, a principal funcionalidade é conseguir enviar um convite através de um determinado canal de comunicação à escolha do utilizador. Para além disto, a aplicação deve dar a possibilidade de fazer o *tracking* dos convites aceites. Deve também possibilitar que este utilizador seja notificado quando o recetor do convite completar os passos para receber o prémio.

Para o recetor, a aplicação deve permitir aceitar um convite recebido ou partilhado num determinado canal de comunicação, deve ainda permitir que o utilizador convidado seja notificado quando completar os passos necessários para receber o prémio associado, se este existir.

Para ambos, deve existir possibilidade de reclamação do prémio. Existem duas formas de reclamar prémios, a forma automática, para a qual o prémio é atribuído quando o convite é concluído, e a forma manual, que dá a oportunidade ao utilizador de fazer uma reclamação, de uma maneira manual, mediante uma data de expiração do prémio. Assim a aplicação deve suportar os dois tipos de prémio e ainda deve possibilitar o preenchimento de informação de envio do prémio, caso este não seja um prémio instantâneo.

O estudo das diferentes plataformas permitiu concluir, não só que diferentes soluções possuem campanhas diferentes, mas também, que dentro da mesma solução podem existir mudanças após um determinado intervalo temporal. As principais mudanças observadas estão relacionadas com os prémios, verificando-se alterações: em a quem estes são atribuídos; no tipo de prémio; e no montante atribuído. Assim, optou-se pela existência de campanhas configuráveis.

Para facilitar a configuração destas campanhas, idealizou-se uma plataforma de gestão, o *Backoffice*, no qual deve ser possível realizar diversas operações relacionadas com as campanhas.

Ao idealizar um *Backoffice* para ser gerido por parte da WIT, propõem-se a apresentação de uma listagem das empresas integradas, permitindo a escolha dos diferentes produtos dentro das mesmas, as quais, conseqüentemente, possuem campanhas de *refer a friend*. Caso contrário, este deve apenas apresentar a listagem de produtos para o cliente onde a plataforma foi implementada. Tomando como exemplo a Vodafone, no papel de empresa, esta possui diferentes produtos, como por exemplo, a Yorn, a Vodafone Tv, entre outros. O conceito de uma empresa possuir diversos produtos foi idealizado em contexto de estágio, com objetivo de uma empresa conseguir aplicar as campanhas nos seus diferentes produtos.

A secção de gestão de campanhas deve possuir uma listagem das campanhas, as quais são definidas para um determinado espaço temporal. Assim, para uma determinada campanha, existem três estados possíveis:

- A decorrer - permite as operações de editar, parar (de forma manual), duplicar, ou ver o relatório da campanha;
- Agendado - permite as operações de editar, começar (de forma manual), eliminar, ou ver o relatório da campanha;
- Terminado - permite as operações de duplicar, eliminar ou ver o relatório da campanha.

É importante realçar que a plataforma foi desenhada de modo a não permitir a sobreposição de campanhas, para que seja possível a gestão automática dos estados das campanhas por parte do *Backend*, ou seja, o sistema começa e termina as campanhas de forma automática.

Uma das funcionalidades presente nesta secção é a criação das campanhas, onde se deve poder definir todos os parâmetros das campanhas configuráveis: detalhes, prémios, aspetos visuais para o remetente e aspetos visuais para o recetor.

Na secção dos detalhes deve ser possível definir:

- Nome;
- Descrição;
- Data de início;
- Data de fim.

Os prémios devem possuir uma secção própria, oferecendo a possibilidade de serem definidos de forma separada, para o remetente e para o recetor. Assim, o utilizador do *Backoffice* pode ter uma maior diversidade no tipo de campanhas e no tipo de prémios a atribuir. Deve ser oferecida a possibilidade de definir:

- Tipo de prémio (Único, Progressivo, Agregado, Nenhum);
- Categoria do prémio;
- Montante do prémio;
- Tipo de reclamação do prémio (Automática ou Manual);
- Tempo de expiração do convite, caso seja manual;
- Número máximo de convites com prémio.

Algumas das opções descritas acima apenas estão disponíveis para o remetente, nomeadamente, os tipos de prémio progressivo e agregado. O tipo progressivo permite

diferentes prémios consoante o número de convites completos, ou seja, quando o remetente possuir um convite concluído deve receber uma t-shirt, no segundo convite deve receber duas canecas, etc.. O tipo agregado tem como objetivo apenas atribuir o prémio após um determinado número de convites, ou seja, quando cinco recetores completarem os passos necessários, então o remetente é recompensado.

Os restantes tipos são partilhados para ambos, apresentando apenas uma diferença mínima no tipo único. Deve existir a possibilidade de definir, para o remetente, o número máximo de convites, no entanto, se não for definido, isto é, ficando sem limite, o remetente deve poder receber prémio sempre que o convite for completo. A definição de um número máximo de convites tem como principal objetivo reduzir a tentativa de fraude, de modo que a empresa consiga ter algum benefício das campanhas.

Para cada utilizador, deve ainda ser possível a definição da parte visual, onde é possível escolher:

- Imagem apresentada ao utilizador;
- Descrição do ecrã.

Por fim, é possível definir os passos necessários para o recetor estar elegível para receber o prémio. Podem ser, por exemplo, registo, primeiro login, primeira transação, se for aplicável a determinada aplicação, validar a conta utilizada para o registo, entre outros.

Ainda na mesma secção deve surgir a opção de editar, na qual é oferecida a possibilidade de alterar os diferentes parâmetros da campanha, exceto a data de início caso a campanha já esteja a decorrer.

Na opção de visualização do relatório das campanhas, deve ser possível observar-se algumas métricas de avaliação da campanha, tais como, por exemplo, convites com sucesso, prémios oferecidos ao remetente, prémios oferecidos ao recetor, sendo apresentadas ao longo de um intervalo de tempo a ser definido pelo utilizador. Devem existir ainda as opções de *download*: *download* dos dados da campanha, caso exista a necessidade de um estudo mais intensivo sobre os dados recolhidos e *download* dos dados de reclamação dos prémios, caso estes sejam prémios manuais. Assim sendo, é necessário o preenchimento de algumas informações, por parte dos utilizadores da aplicação para que lhes possa ser entregue o prémio.

O *Backoffice* deve possuir ainda uma secção de gestão dos prémios, onde é possível criar, duplicar, eliminar, caso não tenha sido utilizado por uma campanha terminada ou esteja a ser utilizado numa campanha a decorrer, e editar, caso não se trate de uma campanha terminada. Todos os parâmetros devem ser editáveis.

A criação de novos prémios deve oferecer a possibilidade de definir:

- Categoria;
- Nome do prémio;
- Montante;
- Tipo do prémio (Instantâneo ou não);
- Url de reclamação (Instantâneo);
- *Username* (Instantâneo);
- *Password* (Instantâneo).

É necessário o preenchimento do url de reclamação, *username* e *password* para que o *Backend* consiga atribuir os prémios de forma automática.

Apesar do principal objetivo do *Backoffice* ser a gestão de campanhas de *refer a friend*, este deve, para que funcione de forma completa, possuir uma secção de gestão de utilizadores. Na qual deve ser possível registar, editar e eliminar utilizadores, definindo:

- E-mail;
- Cargo (editável);
- Produtos (editável).

De forma a resumir as funcionalidades necessárias serão apresentadas, de forma separada por Aplicação e *Backoffice*, em formato de *User Story* em conjunto com o valor de negócio associado, na Tabela 1 e na Tabela 2, respetivamente.

Tabela 1 – User stories da aplicação

<i>User Story</i>	Valor de negócio
Como recetor, eu quero aceitar um convite depois de carregar no <i>link</i> que abre a aplicação.	9
Como utilizador, quero poder navegar até ao ecrã de <i>Refer a friend</i> .	1

Como utilizador, quero ver um ecrã onde posso enviar convites, ver os meus convites aceites ou reclamar o prémio.	5
Como utilizador, quero ver/reclamar o prémio ou guardá-lo para mais tarde.	6
Como utilizador, quero preencher as informações necessárias para receber um prémio que não seja instantâneo.	3

Tabela 2 – *User stories do Backoffice*

<i>User Story</i>	Valor de negócio
Como utilizador não autenticado, quero fazer <i>login</i> .	5
Como utilizador autenticado, quero fazer <i>logout</i> .	5
Como super utilizador WIT, quero ter a possibilidade de criar, editar e apagar empresas, desde que não possuam campanhas a decorrer, ou então escolher uma para ver os seus produtos.	3
Como super utilizador WIT, para criar uma empresa posso definir o logótipo e o nome da empresa.	3
Como super utilizador WIT, para editar uma empresa posso redefinir o logótipo e o nome da empresa.	3
Como administrador, posso criar, editar ou apagar produtos, sem campanhas a decorrer, ou escolher um para ver as suas campanhas.	4
Como administrador, posso criar novos produtos definindo o logótipo e o nome.	4
Como administrador, posso editar um produto redefinindo o logótipo e o nome.	4
Como utilizador autenticado posso ver a página inicial onde está a lista de campanhas, podendo pesquisar, filtrar, ordenar as colunas e criar novas campanhas.	6
Como utilizador autenticado, posso seleccionar o filtro de campanhas criadas, as quais posso começar, ver ou alterar informação, apagar ou duplicar a campanha.	6

Como utilizador autenticado, posso seleccionar o filtro de campanhas em progresso, as quais posso terminar, ver ou alterar informação, ver os relatórios ou duplicar a campanha.	6
Como utilizador autenticado, ao seleccionar o filtro das campanhas terminadas posso duplicar, ver a informação, ver o relatório e apagar a campanha.	6
Como utilizador autenticado quero criar uma nova campanha, onde defino os detalhes, os prémios e aspetos visuais.	9
Como utilizador autenticado quero ver a informação ou fazer modificações à campanha. (Campanhas não terminadas)	5
Como utilizador autenticado quero ver estatísticas sobre as campanhas terminadas ou a decorrer, tais como, número de convites com sucesso, prémios dados aos remetentes, prémios dados aos recetores, número de utilizadores registados, número de utilizadores que concluiu os passos para adquirir o prémio, downloads da aplicação. Quero ainda filtrar por data de início e final.	4
Como administrador, quero ver a lista de utilizadores onde posso criar, apagar e editar utilizadores.	2
Como administrador, quero criar um novo utilizador definindo o e-mail, cargo e produtos.	2
Como administrador, quero editar o cargo ou os produtos do utilizador.	2
Como utilizador autenticado, vejo a lista de prémios onde posso filtrar, pesquisar, criar, editar (usada - em progresso, não usada), duplicar e apagar prémios (não usados).	7
Como utilizador autenticado, quero criar novos prémios definindo categoria, nome, montante, tipo de prémio (instantâneo ou não), url de reclamação (instantâneo), <i>username</i> (instantâneo), <i>password</i> (instantâneo).	4
Como utilizador autenticado, quero editar um prémio redefinindo a categoria, o nome, o montante, tipo de prémio (instantâneo ou não), url de reclamação (instantâneo), <i>username</i> (instantâneo), <i>password</i> (instantâneo),	7

5.2. Protótipos

A análise de requisitos foi acompanhada com a elaboração de diversos protótipos, de modo a ser obtida uma melhor percepção visual. A elaboração dos mesmos tem como benefício uma melhoria contínua dos requisitos até que estes estejam numa fase de definição desejada para que possa ser iniciada a fase de desenvolvimento. É importante realçar, que os requisitos são mutáveis, mesmo após o início da fase de desenvolvimento.

Inicialmente, foram elaborados os protótipos de baixa fidelidade. Estes, assim como os diagramas resultantes da modelação dos processos de negócio, são utilizados para representar uma determinada funcionalidade de forma simples, para que qualquer tipo de pessoa dentro da equipa possa perceber o que se quer representar.

Os protótipos têm como principal objetivo clarificar, perante a equipa, a funcionalidade de uma forma bastante rápida e com um custo bastante reduzido. Estes foram elaborados com apenas “um lápis e papel”, como está representado na Figura 24, podendo ser rapidamente adaptados consoante o *feedback* recebido. Apesar de a representação visual não ser muito fidedigna, o conteúdo da funcionalidade está presente e permite o ajuste das funcionalidades, antes de serem elaborados os protótipos de alta fidelidade.

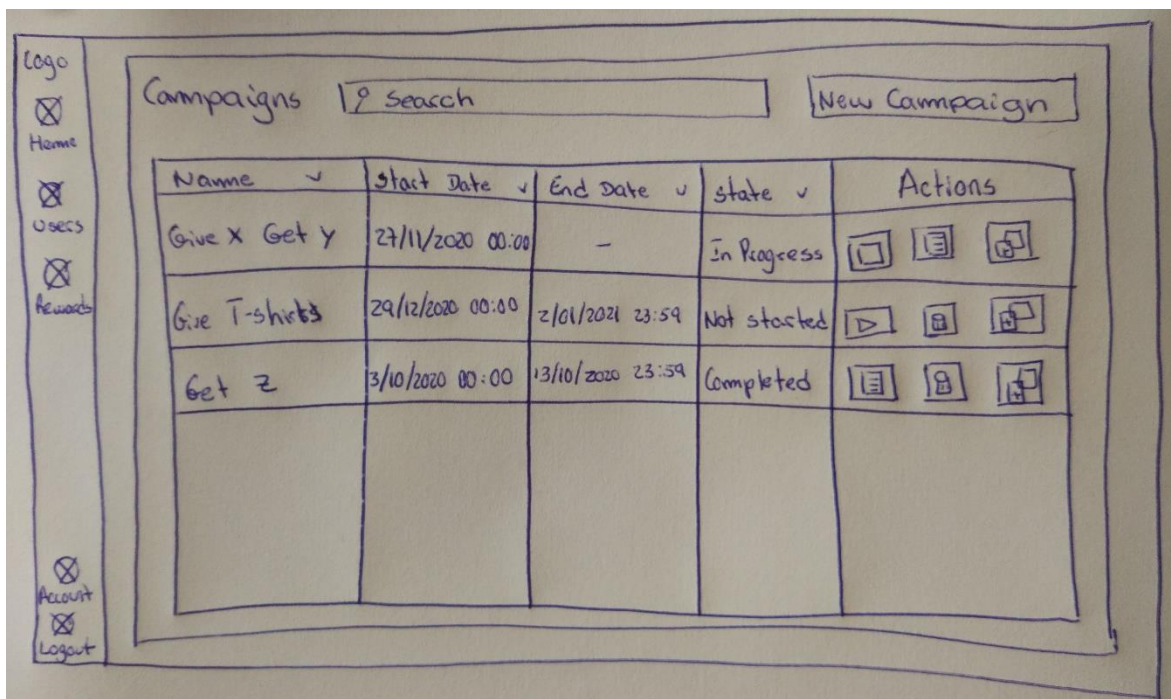


Figura 24 – Protótipo de baixa fidelidade da página inicial das campanhas de um produto

De seguida, e após várias iterações na fase dos protótipos de baixa fidelidade, seguiu-se a fase de elaboração dos protótipos de alta fidelidade, por parte da equipa de design da WIT,

sendo estes baseados nos protótipos anteriores, mas adequados à plataforma específica para a qual seria implementado o programa de *refer a friend*.

Estes tipos de protótipos são concebidos de forma a estarem mais próximos da realidade, assim, ao contrário dos protótipos de baixa fidelidade em que o objetivo é apenas representar a conteúdo/contexto da funcionalidade com pouco detalhe, estes são mais apelativos visualmente e o nível de detalhe é bastante mais elevado. Portanto, transmitem a ideia do aspeto e funcionamento do produto final ao cliente, podendo demonstrar elementos chave, que com os protótipos de baixa fidelidade não seria possível observar. É nestes protótipos de alta fidelidade que o desenvolvimento é baseado.

Devido a especificidade da plataforma e de forma a adequar as funcionalidades a implementar ao tempo de estágio, existiram algumas alterações nos requisitos, sendo alguns removidos ou modificados e outros adicionados. Assim, a Tabela 3 e a Tabela 4 representam as *user stories* referentes aos requisitos estabelecidos para o desenvolvimento.

Tabela 3 – *User stories* da aplicação definidas para desenvolvimento

<i>User Story</i>	Valor de negócio
Como recetor, quero aceitar um convite depois de carregar no <i>link</i> que abre a aplicação.	9
Como utilizador, quero poder navegar até ao ecrã de <i>Refer a friend</i> .	1
Como recetor, quero ter um sinal de notificação sempre que possuir um novo convite aceite.	4
Como utilizador, quero ver um ecrã onde posso enviar convites, ver os meus convites aceites.	5
Como utilizador, quero ver o prémio recebido.	6

Tabela 4 – *User stories* do *Backoffice* definidas para desenvolvimento

<i>User Story</i>	Valor de negócio
Como utilizador não autenticado, quero fazer <i>login</i> .	5
Como utilizador autenticado, quero fazer <i>logout</i> .	5
Como utilizador autenticado posso ver a página inicial onde está a lista de campanhas, podendo pesquisar, filtrar, ordenar as colunas e criar novas campanhas.	6

Como utilizador autenticado, posso seleccionar o filtro de campanhas criadas, as quais posso começar, ver ou alterar informação, apagar ou duplicar a campanha.	6
Como utilizador autenticado, posso seleccionar o filtro de campanhas em progresso, as quais posso terminar, ver ou alterar informação, ver os relatórios ou duplicar a campanha.	6
Como utilizador autenticado, ao seleccionar o filtro das campanhas terminadas posso duplicar, ver a informação, ver o relatório e apagar a campanha.	6
Como utilizador autenticado quero criar uma nova campanha, onde defino os detalhes, os prémios e aspetos visuais.	9
Como utilizador autenticado quero ver a informação ou fazer modificações à campanha. (Campanhas não terminadas)	5
Como utilizador autenticado quero ver estatísticas sobre as campanhas terminadas ou a decorrer, tais como, número de convites com sucesso, prémios dados aos remetentes, prémios dados aos recetores, número de utilizadores registados, número de utilizadores que concluiu os passos para adquirir o prémio. Quero ainda filtrar por data de início e final.	4
Como administrador, quero ver a lista de utilizadores onde posso criar, apagar e editar utilizadores.	2
Como administrador, quero criar um novo utilizador definindo o e-mail e cargo.	2
Como administrador, quero editar o cargo.	2

Na criação das campanhas foi introduzido um novo conceito que tem impacto no restante sistema, trata-se da possibilidade de existência de duas campanhas a decorrer, sendo uma delas considerada uma campanha *default*. A definição de uma campanha *default*, tem como objetivo a substituição entre campanhas, ou seja, sempre que não exista uma campanha a decorrer, é a campanha *default* que fica visível para os utilizadores finais da aplicação.

A adaptação (remoção/modificação) das *user stories* deveu-se ao facto de estas não se enquadrarem no contexto atual da plataforma, que agora se insere no contexto de um cliente específico.

5.3. Resumo

Neste capítulo foram descritas as funcionalidades necessárias para a implementação de um programa de *refer a friend* numa determinada aplicação e com campanhas configuráveis controladas pelo *Backoffice*, o qual deve permitir uma análise estatística da campanha, se esta estiver a decorrer ou tiver terminado. Por fim, de forma a complementar o levantamento de requisitos foi explicitado como foi elaborada a etapa de prototipagem e o impacto que esta tem no levantamento de requisitos.

6. Desenvolvimento

O presente capítulo tem como objetivo demonstrar todo o desenvolvimento realizado ao nível da aplicação móvel, *Backoffice* e *Backend*. Para além disso, é demonstrada a arquitetura do sistema para uma melhor compreensão do contexto onde foi inserido o programa de *refer a friend* e apresentada a forma como a plataforma foi validada e verificada. Em seguida, são descritas as ferramentas utilizadas para o desenvolvimento da plataforma.

6.1. Arquitetura do Sistema

A arquitetura geral do sistema, presente na Figura 25, foi baseada na arquitetura já existente dos projetos integrantes do mesmo. Assim, o diagrama demonstra as ligações que foram implementadas para que fosse possível integrar os diferentes componentes desenvolvidos.

O sistema implementado centra-se no microserviço criado para o tratamento das campanhas de *refer a friend* e o *tracking* dos convites. Apesar da arquitetura de microserviços trazer mais complexidade, pela necessidade da introdução de componentes como a *facade*, que faz o encaminhamento dos pedidos para os microserviços corretos. Por outro lado, traz vantagens, como, por exemplo, o desacoplamento do serviço, facilitando a integração em outros sistemas ou plataformas.

O microserviço encontra-se conectado a três bases de dados. A base de dados Mongo que guarda os dados das campanhas e dos convites. A base de dados de *business intelligence* (BI) em Postgres, que é mantida e criada pela equipa da WIT, na qual são registados os diversos eventos desencadeados pelos utilizadores na aplicação e os quais permitem a validação dos passos para a atribuição dos prémios. Por último, a base de dados que utiliza a tecnologia Elastic Search, onde são guardados os dados de analítica, utilizados pela plataforma de visualização de dados, o Kibana.

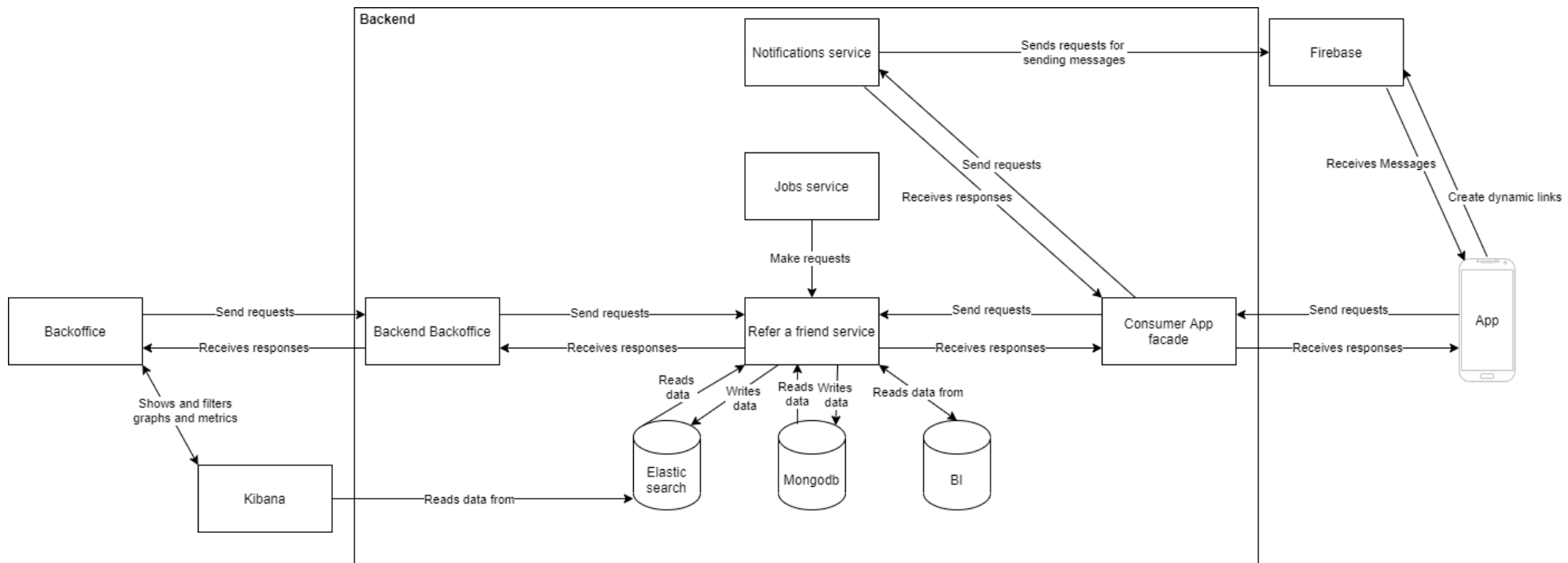


Figura 25 – Representação da arquitetura geral

A comunicar com o microserviço, temos a *facade* e o *Backend* do *Backoffice*, sendo que ambos têm funções semelhantes, ou seja, ambos recebem o pedido do cliente, a aplicação e o *Backoffice*, respetivamente, e reencaminham a resposta. A *facade* coordena os pedidos e ainda comunica com os restantes serviços para o bom funcionamento da plataforma, no caso de estudo, comunica com o microserviço das notificações, que conseqüentemente irá comunicar com o *firebase*, para que o remetente e recetor sejam notificados, na aplicação, quando recebem prémios.

Existe outro serviço a comunicar com o microserviço criado, o serviço de *jobs*, este controla a atualização dos dados, nomeadamente, a manutenção do estado da campanha e a atualização dos dados de estatística.

A aplicação possui ainda outra ligação ao *firebase*, para que possam ser criados os *links* que permitiram aos utilizadores aceitar convites.

Por fim, o *Backoffice* comunica com o Kibana de forma a criar os diversos gráficos e as diferentes métricas de avaliação das campanhas.

6.2.Ferramentas Utilizadas

Este subcapítulo pretende enunciar as ferramentas que tiveram um papel essencial na ajuda ao desenvolvimento das diferentes plataformas.

6.2.1. IntelliJ IDEA

O IntelliJ IDEA [50] é um *Integrated Development Environment* (IDE), desenvolvido pela JetBrains, destinado ao desenvolvimento das linguagens que podem ser compiladas para *java virtual machine* (JVM), no entanto apresenta suporte para outras linguagens. Foi utilizado para o desenvolvimento do microserviço, *Backend* do *Backoffice*, no serviço de *jobs* e na *consumer app facade*. Possui várias funcionalidades como:

- Navegação inteligente;
- Assistência na codificação;
- *Debugger*;
- *Profiler*;
- Terminal;
- *Build tools*;

- Controlo de versões.

6.2.2. Android studio

O Android studio [51] é o IDE oficial para o desenvolvimento Android e foi utilizado para o desenvolvimento e integração do programa de *refer a friend* na aplicação móvel. É baseado no IntelliJ IDEA. Para além das principais funcionalidades do IntelliJ IDEA, possui mais funcionalidades que favorecem o desenvolvimento em Android, tais como:

- Sistema de compilação baseado em *gradle*;
- Criação de emuladores e utilização dos mesmos para testar as aplicações criadas;
- Compatibilidade integrada com o Google Cloud Platform.

6.2.3. Visual studio code

O Visual studio code [52] é um IDE *open source*, desenvolvido pela Microsoft, para Windows, macOS e Linux, com repositório no github. Apresenta suporte nativo para linguagens baseadas em Javascript, como Typescript e Node.js, no entanto, possui um *marketplace* repleto de extensões que ajudam o programador a desenvolver em muitas outras linguagens. No contexto do estágio, foi utilizado para desenvolver o *Backoffice*.

6.2.4. Docker

O Docker [53] é uma plataforma *open source*, que permite uma abstração do ambiente de desenvolvimento onde a aplicação está a ser implementada. É possível ter diversas aplicações em funcionamento para um determinado *host*, através do conceito de *Container*.

Os *containers* contêm apenas os recursos necessários para que uma determinada aplicação funcione de forma correta. A abstração do ambiente, tem como principal vantagem possibilitar a partilha de *containers* entre equipas e ambientes. Por exemplo, se toda a equipa utilizar um mesmo *container* para o desenvolvimento, são excluídos os possíveis problemas que possam ser introduzidos pela utilização de diferentes versões de software, assim como, a utilização das mesmas versões em ambiente de desenvolvimento local e ambiente de produção. Permite, igualmente, uma redução significativa dos problemas quando uma determinada aplicação é colocada em ambiente de produção. No decorrer do estágio, foram utilizados diversos *containers*, nomeadamente:

- Kibana;
- Elastic Search;
- Postres;
- PgAdmin (Cliente de gestão da base de dados Postgres);
- Mongo;
- Mongo express (Cliente de gestão da base de dados Mongo);
- Hazelcast;
- Config server (Criado pela WIT e utilizado pelo *Backend*);
- Portainer;
- Rabbitmq;
- Nginx.

6.2.5. Bpmn.io

O Bpmn.io [49] é uma ferramenta disponível através do browser, *open source* e grátis, desenvolvida pela Camunda, a qual tem um longo histórico em *business process mapping* (BPM). A ferramenta usa o BPMN 2.0, que é o standard para a modelação de processos de negócio, define, ainda, a formatação necessária dos ficheiros para que estes possam ser partilhados pelas diferentes plataformas de modelação.

Esta ferramenta foi utilizada no contexto do estágio, tal como foi referido anteriormente na secção do planeamento e da análise de requisitos, para ajudar na perceção visual, dos diferentes processos de negócio necessários para a implementação do programa de *refer a friend*. Sendo o principal objetivo atingir todos os intervenientes, de forma simples e eficaz, para que possam compreender e dar *feedback* sobre as diferentes necessidades para a implementação da plataforma.

6.3. Aplicação Móvel

Este subcapítulo tem como objetivo descrever o processo de desenvolvimento do programa de *refer a friend* para ser introduzido numa aplicação. Serão descritas as funcionalidades, os mecanismos implementados ou ajustados, a arquitetura de software utilizada e as integrações feitas com as diferentes plataformas. Apesar de ser descrito o processo de desenvolvimento, não serão apresentadas figuras com *layouts* criados por questões de confidencialidade.

Como foi referido na secção do planeamento, inicialmente idealizou-se um desenvolvimento *cross platform* para que pudesse ser reutilizado de forma simples em outras aplicações móveis.

O desenvolvimento *cross platform*, tal como o nome indica, consiste no desenvolvimento multi-plataforma, neste caso para Android e iOS, utilizando a mesma *code base*. Atualmente, existe uma diversidade de *frameworks*, como por exemplo, React Native, Flutter, Xamarin, entre outras.

Cada uma apresenta diversas particularidades, no entanto, todas apresentam objetivos e vantagens comuns. Nomeadamente, como principais vantagens temos a partilha da mesma *code base*, que como consequência oferece a possibilidade da mesma equipa de desenvolvimento produzir de forma mais rápida e com menos recursos, tornando-se mais eficiente e obtendo-se um *time to market* menor. O *time to market* é a diferença temporal desde que foi iniciado o desenvolvimento até que este esteja disponível para venda.

No entanto, devido à falta de suporte para integração do código *cross platform* com o código nativo por parte da aplicação escolhida pela WIT para a integração do programa de *refer a friend*, optou-se pelo seguimento do desenvolvimento daquela aplicação, o nativo em Android. Por outro lado, o facto de esta utilizar tecnologias como *push notifications* e *DeepLinking* necessárias para a integração do programa de *refer a friend*, possibilitando um aceleramento do desenvolvimento, foi preponderante na tomada de decisão.

Atualmente, a nível mundial, o Android como sistema operativo móvel, possui a maior quota de mercado, contando com, cerca de 72% [54]. Apesar de possuir a grande parte do mercado, esta plataforma tem uma grande desvantagem, em comparação com iOS, porque apresenta uma enorme fragmentação, como pode ser observado na Figura 26⁴. A última versão, 11.0, lançada em Setembro de 2020, apenas está disponível em 9% dos dispositivos móveis [55]. Em comparação com iOS, a versão 14.4 lançada em Março de 2021, para cerca de 75% dos dispositivos móveis já está disponível [56].

⁴ <https://gs.statcounter.com/android-version-market-share/mobile/worldwide/#monthly-202004-202104-bar>

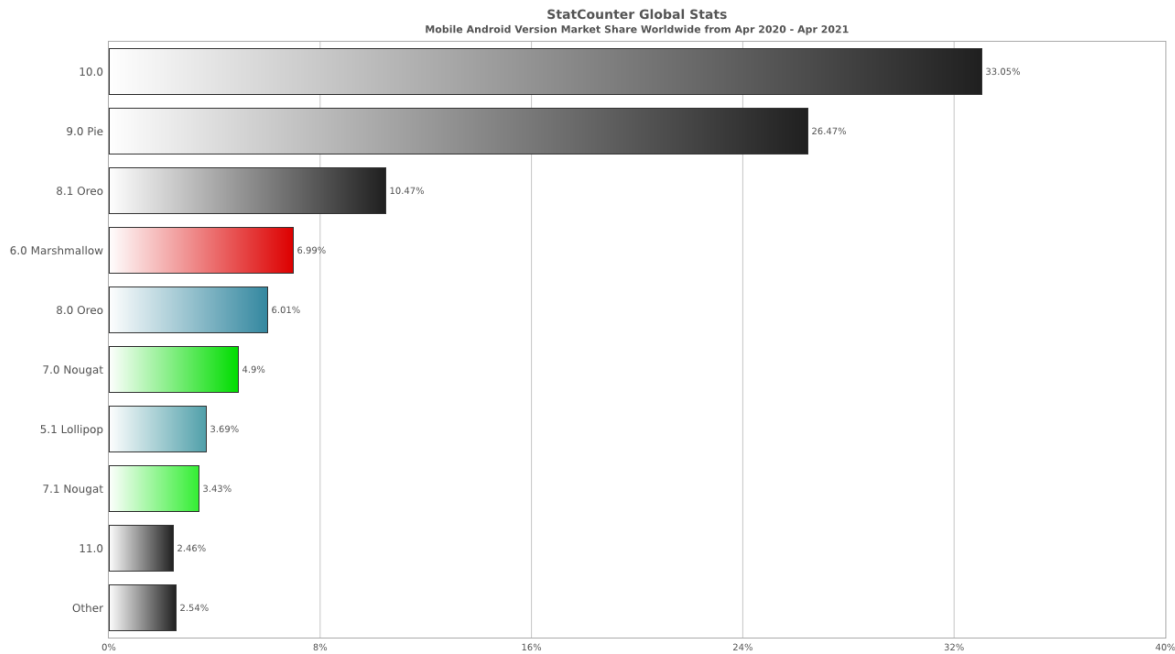


Figura 26 – Fragmentação do mercado Android

Esta fragmentação faz com que os *developers* tenham, frequentemente, de adaptar o seu código às diversas versões Android, exigindo um maior esforço. Podendo existir determinadas funcionalidades que só estão disponíveis para uma determinada versão, ou, por outro lado, surgir a necessidade de desenvolver e manter pedaços de código diferentes para que a mesma funcionalidade esteja disponível em versões diferentes.

Atualmente a linguagem oficial para desenvolvimento Android é o Kotlin, que foi criada pela JetBrains em 2010, no entanto, só em 2019 foi reconhecida como tal pela Google [57], antes, a linguagem reconhecida era o Java. Segundo a Google, 60% das apps no top 1000 já utilizam Kotlin.

O programa de *refer a friend*, foi introduzido numa aplicação com cerca de um ano de desenvolvimento pela equipa da WIT, pelo que a arquitetura da aplicação já estava bem definida. A aplicação faz parte de um conjunto de aplicações que partilham diversos módulos, fazendo com que parte do código possa ser reaproveitado.

A aplicação assenta na arquitetura de desenvolvimento Model-View-ViewModel (MVVM). Os três componentes comunicam entre si, sendo que a *View* comunica com o *ViewModel* e comunicando este com o *Model* [58]. No caso de estudo, foram criadas duas *Activities* que estão ligadas às suas *Views*, uma para os recetores e outra para os remetentes, ambas com acesso ao seu *ViewModel* que comunica com o *Repository* que obtém a informação do modelo. Os *ViewModels* podem estar ligados a diversos *Repositories*, de

modo a obter diferentes tipos de informação, assim como, os *Repositories* podem estar ligados a outros, com o mesmo propósito.

Inicialmente, a primeira etapa passou por adicionar um ponto de entrada para que os utilizadores pudessem ter acesso ao ecrã de *refer a friend*, como remetentes. Assim sendo, foi adicionada uma opção à lista de opções no espaço do utilizador.

Em seguida, procedeu-se ao desenvolvimento da UI com recurso a dados em memória estáticos. Desenvolveu-se o ecrã dos remetentes, considerando apenas uma campanha com o tipo de prémio Único e que implica alterações na UI quando o utilizador já atingiu o limite de convites definido. Para além destas alterações, consoante o tipo de prémio, são realizadas ligeiras alterações na lista de convites aceites e no seu título, no entanto, este funcionamento só foi implementado mais tarde, quando surgiram os protótipos para os diferentes tipos de prémio. Neste ecrã, são apresentadas ao utilizador, diversas informações sobre a campanha, nomeadamente:

- Imagem;
- Título;
- Descrição;
- Lista de convites aceites (título e lista horizontal);
- Botão para enviar convites;
- Texto para termos e condições e data de fim, se existir.

Todos os elementos, acima descritos, são alterados quando o utilizador atingir o número limite de convites, com exceção da lista de convites que se mantém igual. Esta lista é criada através da utilização de um componente *RecyclerView* [59] que permite criar os elementos da lista dinamicamente. Toda a lógica visual é gerida por um *Adapter* que é associado à *RecyclerView*, sendo na classe do *Adapter* que a UI é adaptada de forma dinâmica consoante o tipo de prémio definido e número de convites completos.

Existem dois tipos de elementos na lista, o elemento que representa um convite por preencher, Figura 28, e o elemento preenchido, Figura 29. O primeiro tipo é apresentado ao utilizador até este atingir o limite de convites aceites, por exemplo, tendo um convite aceite e com limite de cinco, serão apresentados 4 convites por preencher. O segundo tipo, pode ser apresentado com diferentes figuras, consoante o utilizador possuir ou não o contacto do convite aceite no *smartphone*. Se possuir o contacto da pessoa que aceitou o convite, então

será apresentada a foto, tal como representado na Figura 29, se esta estiver definida, senão irão ser apresentadas as iniciais do nome definido, semelhante ao representado na Figura 30. Se não possuir o contacto será apresentado um *placeholder*, representado pela Figura 27.

Este comportamento foi decido de forma a proteger a privacidade dos utilizadores convidados, pois, um convite pode ser partilhado com qualquer pessoa. Assim, ao aceitar o convite fica excluída a hipótese de a foto ou nome ficarem expostos para qualquer pessoa.



Figura 28 – Elemento por preencher



Figura 29 – Elemento preenchido com contacto com foto



Figura 30 – Elemento preenchido com contacto sem foto



Figura 27 – Elemento preenchido sem contacto presente

O botão para enviar convites utiliza o *Android Sharesheet* [60] para se obter um comportamento *standard* para as diferentes versões Android e com o objetivo de partilhar os convites nos diferentes canais de comunicação.

O convite partilhado contém uma mensagem definida na campanha e um *link*. O *link* foi a implementação escolhida para a aquisição de novos utilizadores no programa de *refer a friend*. Foram utilizados *Firebase Dynamic Links* [61], não só por a aplicação cliente já

possuir suporte, mas também, por trazerem uma vantagem enorme pelo facto de serem persistidos mesmo se a aplicação não se encontrar instalada, e ainda, possibilitar a obtenção de algumas estatísticas em relação aos diversos *links* criados.

A utilização dos mesmos em conjunto com os *DeepLinks* [62] de Android permitem ao remetente carregar no botão de enviar convites e ser gerado um *link* através da adição dos parâmetros, id da campanha e id do utilizador, parâmetros estes que irão ser utilizados mais tarde quando o utilizador carregar no *link* e for aberta a aplicação. A biblioteca dos *Firestore Dynamic Links* [61] permite converter o *link* em *short URL*, de forma a esconder alguma informação sensível que possa constar no URL original. A Figura 31 representa um *link* antes de ser convertido em *short URL*.

```
https://example.com/invite?campaignId=x&userId=y
```

Figura 31 – Exemplo link antes da conversão para *short URL*

Quando o recetor do convite pressiona o *link*, irá ser redirecionado para a *play store* ou *app store* se não possuir a aplicação. Se já possuir a aplicação, esta irá ser aberta porque o *setup* dos *DeepLinks* na aplicação permite reconhecer o padrão do URL e respetivamente abrir a aplicação. A partir deste momento, é utilizada a biblioteca *Firestore Dynamic Links* [61] para descodificar o URL e redirecionar para o *handler* correto, em seguida é realizado o *parse* que após ser concluído com sucesso valida se o remetente não se convidou a si próprio. Caso isso não aconteça, então o convite é guardado, através de um mecanismo explicado mais à frente no relatório.

O ecrã dos recetores foi desenvolvido da mesma forma que o anterior, no entanto, este não é modificado ao longo do tempo. O ecrã é constituído por:

- Título principal;
- Imagem;
- Título;
- Descrição;
- Botão de redirecionamento para a atividade do remetente.

Este é apresentado aos utilizadores quando recebem uma *push notification*. O funcionamento das *push notifications* já estava implementado pela WIT, apenas foi adaptado para os diferentes tipos de notificação definidos no *Backend*. A equipa da WIT utiliza o

Firebase Cloud Messaging [63] para o envio e receção de notificações, a Figura 32 representa um exemplo do formato JSON da mensagem recebida pela aplicação.

```
{
  "message": {
    "token": "bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMExUdFQ3P1...",
    "data": {
      "Nick" : "Mario",
      "body" : "great match!",
      "Room" : "PortugalVSDenmark"
    }
  }
}
```

Figura 32 – Exemplo JSON de mensagem do *Firebase Cloud Messaging*

O utilizador ao realizar a operação de login, é feita uma comunicação inicial com o *firebase*, na qual é devolvido um *token* de identificação para o *smartphone* utilizado. Esta informação é depois guardada no *Backend*, e utilizada, por exemplo, para o envio das *push notifications*.

Estas notificações são enviadas em formato de mensagem semelhante ao apresentado na figura anterior (Figura 32), sendo elas constituídas por alguns campos específicos necessários para o tratamento correto da mensagem enviada pelo *firebase*.

As mensagens quando chegam à aplicação, passam por um processo de *parse* dos conteúdos da mensagem e um posterior tratamento, caso seja necessário algum tipo de operação relacionado com a mensagem recebida, como num dos exemplos implementados para o programa de *refer a friend*. Um dos campos principais é a identificação do tipo da notificação, pois é com base nele que a aplicação consegue distinguir os diferentes tipos de mensagens, desencadeando o comportamento correto para o utilizador final.

Quando o tipo de notificação é *REFER_FOR_REWARDS_COMPLETED_STEPS* e o utilizador abre a notificação será apresentado o ecrã dos recetores com as informações preenchidas através do conteúdo *parsed* da notificação, com exceção da imagem, que por ser transformada no formato Base64, necessita de ser apresentada através da informação da campanha pois ultrapassa o tamanho limite definido para as mensagens no *Firebase*. Este formato é utilizado pela WIT para manter a segurança na aplicação do cliente, diminuindo o número de pedidos a serviços externos.

De forma a alertar o remetente relativamente ao facto de um recetor ter aceiteado o seu convite foi definido um outro tipo, *REFER_FOR_REWARDS_ACCEPTED_INVITE*, o qual para além de exibir uma *push notification* tem uma funcionalidade adjacente que irá ser explicada mais à frente.

Após o desenvolvimento do UI, procedeu-se ao desenvolvimento funcional, através da implementação dos *ViewModels* para cada atividade e do *Repository*. Estes componentes utilizam uma biblioteca de *dependency injection*, Koin [64], permitindo a injeção das dependências exigidas pelas diferentes classes em *runtime*.

As *Activities* injetam a dependência do *ViewModel* e através do método *observe* da classe *LiveData* [65] detetam as alterações do modelo. Quando as alterações são detetadas, é invocado o método que foi passado na função e que irá ser responsável por fazer o tratamento dos dados, bem como proceder às alterações necessárias na UI.

O *ViewModel* comunica com o *Repository*, também este injetado, possuindo, o mesmo, várias dependências, nomeadamente uma dependência para o módulo *Storage*, que permite aceder ao armazenamento local oferecendo as operações de leitura e escrita de dados e outra para o módulo da API, que permite comunicações com os serviços externos. Ambos os módulos já existiam no projeto, foram criados e mantidos pela equipa de desenvolvimento da WIT e adaptados para o caso de estudo.

De forma a tirar partido do armazenamento local a aplicação utiliza uma base de dados local com a tecnologia *Realm* [66], facilitando a construção de uma aplicação *offline-first*. Este conceito oferece a possibilidade de utilização da aplicação mesmo não possuindo internet ou ficando, por momentos, sem acesso à rede. Para a utilização desta tecnologia é necessário a criação do modelo com os diversos campos obrigatórios e a criação de uma migração que permita à base de dados *Realm* ter conhecimento do objeto do modelo para que possam ser executadas as operações de leitura e escrita.

A comunicação dos *ViewModels* com o *repository* é protegida através da utilização de *Coroutines* [67], estas têm como objetivo principal fazer o processamento assíncrono num contexto diferente do principal, permitindo que o utilizador continue a utilizar a aplicação sem que esta bloqueie a execução. O *repository* utiliza *Flows* [68], pois os acessos à *Storage* acontecem de forma assíncrona, a utilização de funções *suspend* oferece uma ajuda extra visto que a função espera pelo resultado da leitura da base de dados.

Após a ligação dos componentes da parte funcional, procedeu-se à implementação das restantes funcionalidades e da comunicação com o microserviço. A comunicação com o microserviço é realizada através da utilização da biblioteca Retrofit [69], sendo este um cliente HTTP que oferece diversas características que facilitam a comunicação com serviços externos. O *repository* tem implementadas três comunicações com o microserviço, sendo elas, o pedido da informação da campanha, o pedido que guarda um convite e o pedido que atualiza a lista de convites visualizados.

Tanto o primeiro pedido como o segundo são invocados após o login com sucesso, sendo que o primeiro pedido é sempre realizado e após ser obtida uma resposta com sucesso são guardados na *Storage* os dados que depois são consultados pela *Activity*, o que possibilita a construção do programa de *refer a friend* tendo em conta o conceito *offline-first*. O segundo pedido só é realizado se na base de dados local constar um convite aceite. Este convite aceite é guardado na base de dados local quando um utilizador segue um *link* partilhado. Este é persistido até que o utilizador conclua o login, permitindo que este siga o *link* não concluindo o login, no entanto, mais tarde possa voltar à aplicação, concluir o login e ganhar o prémio, se o convite ainda se encontrar válido.

Na *Activity* para os *senders*, é requisitada a permissão de acesso aos contactos pela necessidade de preenchimento do nome ou foto dos convites aceites, e após receber a informação da campanha através do *ViewModel* é realizado o terceiro pedido.

No modelo definido, cada convite tem uma *flag*, com valor *true* ou *false*, que permite concluir se o utilizador tem convites por visualizar ou não, sendo que, esta *flag* é também utilizada para mostrar o *notification sign* no ponto de entrada dos detalhes da conta do utilizador. Esta funcionalidade necessitou da criação de uma classe que regista um *LifecycleObserver* [70], o qual fica à escuta de um evento.

Neste caso, o evento é o *Messaging Event* do Firebase, pois é através da receção de uma *push notification* do tipo *REFER_FOR_REWARDS_ACCEPTED_INVITE*, que é adicionado o convite à lista de convites na base de dados local com a *flag* de valor *false*, fazendo com que seja necessário mostrar o *notification sign*. Este comportamento permite poupar um novo pedido ao serviço externo por novas informações da campanha.

A *Storage* devolve em tempo real, fazendo uma consulta imediata, sem necessitar de esperar por outros eventos que possam estar a ocorrer, um *boolean* como *LiveData* [65], para

o qual irá ser registado na classe anterior um *observer* que irá controlar o estado do *notification sign*.

Assim sendo, se existirem convites não visualizados é realizado o terceiro pedido, que irá atualizar a lista de convites de um determinado utilizador no microserviço. No entanto, quando a *activity* comunica com o *ViewModel* e conseqüentemente passa o pedido ao *Repository*, este faz com que a *Storage* atualize a lista de convites local, fazendo com que o *notification sign* seja ocultado.

Após serem elaborados os protótipos dos diferentes tipos de campanhas relativas aos diferentes tipos de prémio associado ao utilizador remetente, foram implementadas as alterações visuais necessárias de forma a suportar os diferentes tipos de campanha, dinamicamente. Existem essencialmente dois comportamentos, um representado pela Figura 33, em que na primeira posição da lista se situa sempre um elemento por preencher, aplica-se nos casos em que não existe prémio ou o prémio não tem limite. Este comportamento foi escolhido para o utilizador final conseguir perceber que pode continuar a enviar convites, pois a campanha não possui limite de convites, podendo ganhar sempre prémio, caso se trate de uma campanha com prémio sem limite.

O outro comportamento, para as campanhas com prémios agregados ou com limite, foi descrito anteriormente, representado pela Figura 34, foi apenas adicionado o suporte das campanhas com tipos de prémio agregado, em que um conjunto de convites apenas oferece um prémio. Este comportamento pretende que o utilizador tenha precisamente a perceção contrária do comportamento anterior, transmitindo ao utilizador que este pode continuar a convidar, mas que não irá receber prémio.

Foi ainda acrescentado um limite total à lista representado por *badge* no último elemento que apresenta o número total de convites, tal como representado na Figura 35.



Figura 33 – Exemplo lista para remetente sem prémio ou sem limite



Figura 34 – Exemplo lista com limite

Figura 35 – Elemento com *badge*

6.4.Backoffice

Este subcapítulo tem como objetivo descrever o processo de desenvolvimento do *Backoffice* e da parte da analítica contida no mesmo, serão descritas as funcionalidades principais e as adaptações implementadas. Tal como no capítulo anterior, não serão apresentadas quaisquer figuras do *layout* implementado por questões de confidencialidade.

O *Backoffice* do cliente para o qual o programa de *refer a friend* foi desenvolvido, tem a sua *code base* partilhada com outro cliente. Alterando apenas o tema visual e partilhando grande parte das funcionalidades. Uma das funcionalidades já implementadas pela equipa da WIT, era a gestão das *referral campaigns*, o que proporcionou um desenvolvimento substancialmente mais acelerado, tornando-o mais simples.

A *framework* utilizada no desenvolvimento foi o *React* [71], criada pelo Facebook em 2013. Esta *framework* tem como base o *Javascript*, no entanto, no projeto foi utilizado *Typescript* [72], linguagem criada e mantida pela *Microsoft*. Esta linguagem baseia-se em *Javascript* mas acrescenta tipos de dados estáticos como existem em muitas outras linguagens, como por exemplo, *boolean*, *string*, *number*, entre outros. Desta forma, esta linguagem permite mitigar uma das desvantagens da utilização de *Javascript* puro.

O projeto encontra-se dividido por funcionalidade, no entanto, existem componentes que são partilhados pelas diversas funcionalidades. A equipa da WIT utiliza um *Storybook* [73] como ferramenta de desenvolvimento e teste de componentes de UI partilhados, procedendo depois à integração no projeto, como biblioteca de componentes.

Atualmente, o código pertencente às *referral campaigns* encontra-se na secção de configurações, dentro desta secção estão contidas muitas outras funcionalidades. Para a

funcionalidade, existem essencialmente três páginas, nomeadamente, a página principal, a página de criação das campanhas e a página de edição.

A página principal contém a lista de campanhas, ordenadas por estado, primeiro são apresentadas as campanhas a decorrer, em seguida as campanhas agendadas e por fim as campanhas terminadas. Para o estado a decorrer, está apenas disponível a ação de duplicar, enquanto, para os restantes estados, está também disponível a opção de eliminar.

O utilizador do *Backoffice*, consegue observar o estado da campanha, o nome, tipo de prémio para o recetor, o montante para o recetor e data de início e fim. Para além disto, é possível filtrar por nome e estado da campanha.

Se o utilizador decidir criar uma campanha, este tem de pressionar o primeiro elemento da lista, o qual não é uma campanha, mas sim um botão de acesso à página de criação de campanhas. Ao exercer a mesma ação em qualquer uma das outras linhas, irá ser redirecionado para a página de edição da campanha.

A equipa da WIT encontrava-se a desenvolver ao mesmo tempo que foi iniciada a integração para o cliente, onde estava a ser implementado o programa de *refer a friend*, assim, algumas funcionalidades foram acrescentadas e outras sofreram melhorias, melhorias estas que irão ser apresentadas, mais à frente no capítulo. Para que as melhorias ficassem disponíveis para o cliente e tudo continuasse a funcionar como era suposto, estas foram integradas mais tarde, trazendo algumas mudanças no *Backend*, que serão descritas no capítulo seguinte.

Na criação de uma campanha, é necessário completar quatro passos sequenciais de preenchimento de informação, que serão descritos de seguida.

Primeiramente, a informação geral da campanha, na qual é preenchido o nome, data de início, data de fim, evento de conversão que no caso de estudo apenas foi considerado, o primeiro login, e a mensagem de convite. Esta mensagem, no seu conteúdo, tem de conter obrigatoriamente a *tag* “[URL Link]” que será utilizada pela aplicação, para a substituir pelo *short URL* gerado pelo *firebase* quando o remetente pretende enviar o convite.

Em segundo lugar, será necessário o preenchimento da informação dos prémios do remetente e do recetor. Inicialmente, é necessário que o utilizador preencha o tipo da campanha, para serem apresentados os restantes campos a preencher. Para o remetente, é

possível escolher os tipos único, agregado ou sem prémio, sendo que para o recetor, existem apenas o tipo único ou sem prémio. Nos tipos único e agregado, é possível definir o limite, sendo que o tipo agregado tem limite obrigatório, mas no tipo único pode não existir limite. Estes tipos, exigem, ainda, o preenchimento da categoria do prémio (*Cash, Discount, Airtime, Bundle*) e o seu montante.

Os dois passos seguintes, destinam-se à definição de alguns elementos da UI para o remetente e o recetor. No terceiro passo, existe uma variante relacionada com o tipo de prémio escolhido anteriormente, se for definido um tipo de prémio com limite, como por exemplo, o tipo único limitado ou tipo agregado, é possível definir dois ecrãs diferentes, sendo o primeiro até atingir o limite e o segundo depois de atingi-lo. Em ambos, o utilizador pode escolher o título, descrição e imagem. Caso o tipo de prémio seja nenhum ou único sem limite, é apenas possível preencher as informações do primeiro ecrã.

Por fim, no quarto passo, o utilizador irá preencher as informações do ecrã dos recetores, os quais o irão ver após ser recebida a notificação, que só será enviada quando os mesmos seguirem um *link* e completarem o primeiro login. Neste passo, o utilizador tem de preencher o título principal, título, descrição e imagem. Para ambos os passos, são apresentados *previews* dos ecrãs, no entanto, este componente não contemplava as alterações necessárias para o tema do cliente. Sendo assim, de forma a tornar a solução a mais próxima da realidade possível realizaram-se as alterações necessárias, tendo sido alterados o *preview* para o tema da aplicação do cliente e os *placeholders* dos convites aceites.

Após completar os quatro passos, o utilizador pode criar a campanha. Assim é enviado um pedido *HTTP* para o *Backend* com todas as informações da campanha através do *HTTP client Axios* [74]. Caso a campanha seja criada com sucesso, o utilizador é redirecionado para a página principal e é apresentada uma mensagem de sucesso, caso contrário, é apresentada uma mensagem de erro. Esta comunicação, assim como, o código que possibilita este comportamento encontra-se representado na Figura 36.

```

ReferralCampaignsApi.methods.createReferralCampaign(values, config && config.client).then(
  () => {
    showAlertProps({
      type: AlertTypeEnum.SUCCESS,
      title: t("pages.referralCampaigns.createCampaign.getCreateCampaignSuccess"),
    });
    showAlert();
    history.push(RoutesEnum.REFERRAL_CAMPAIGNS);
  },
  () => {
    showAlertProps({
      title: t("pages.referralCampaigns.createCampaign.getCreateCampaignError"),
      type: AlertTypeEnum.ERROR,
    });
    showAlert();
  },
);

```

Figura 36 - Excerto de código representativo da comunicação criação de uma campanha

Quando o utilizador realiza a ação de *hover* numa campanha da lista, podem ser exibidas duas ações, a ação de duplicar ou a ação de eliminar. A ação de duplicar irá criar uma cópia da campanha selecionada para um período a definir pelo utilizador, ou seja, é pedido ao utilizador que introduza uma data de início, esta irá ser validada pelo *Backend* e se o intervalo estiver disponível irá ser realizada a cópia da campanha com sucesso. A ação de eliminar, é antecedida por um alerta que informa o utilizador, de que está a proceder a uma ação destrutiva que não pode ser revertida, se o utilizador mesmo assim desejar prosseguir então é enviado o pedido para o *Backend*.

Por fim, o utilizador ao selecionar uma das campanhas da lista irá ser redirecionado para a página de edição da campanha. Esta página possui toda a informação da campanha, separada por *tabs*, possuindo as quatro *tabs* iguais às da criação.

Sendo estas, a informação geral da campanha, os prémios, UI para o remetente e UI para o recetor, pode ainda possuir outra *tab*, caso a campanha se encontre a decorrer ou se já estiver terminada, a *tab* da analítica.

O desenvolvimento da analítica das campanhas, não tinha sido planeado para a equipa da WIT que desenvolveu a restante UI, pelo que todo o desenvolvimento foi realizado em contexto de estágio. A utilização de componentes reutilizáveis, permitiu que a adaptação feita nos *previews* se refletisse, retirando a necessidade de acrescentar, novamente, o suporte para o cliente do contexto de estágio.

Nesta página, caso a campanha esteja a decorrer, não é possível editar a data de início e os prémios, sendo, ainda, oferecida a possibilidade de terminar a campanha de forma manual. Caso a campanha esteja terminada, apenas pode consultar as informações da campanha, caso contrário, pode alterar todos os parâmetros e começar a campanha de forma manual. É importante realçar que as alterações são todas validadas pelo *Backend*, dando especial relevância ao intervalo escolhido pelo utilizador, caso tenha sido alterado, incluindo as ações de começar e terminar a campanha de forma manual, que editam a data de início e de fim.

Para garantir o correto funcionamento de todas as funcionalidades, foi necessário ajustar os pedidos enviados para o *Backend*, pois era necessário distinguir a partir de que cliente estavam a ser realizados os pedidos, de forma que estes fossem reencaminhados para o microserviço. Assim, a cada pedido foi adicionada a identificação do cliente, como *query parameter*, ao URL do pedido. Caso o utilizador comece ou termine a campanha de forma manual foram adicionadas *flags* de controlo ao *body* do pedido em questão, para que o *Backend* tivesse conhecimento destas ações e pudesse desencadear o comportamento correto, o qual será explicado no próximo capítulo.

Após a primeira fase de integração estar completa, procedeu-se ao desenvolvimento da *tab* de analítica, o que requereu o desenvolvimento do UI, integração com a ferramenta de visualização de dados, *Kibana* [75], a qual foi sugerida por elementos da WIT de forma a acelerar o desenvolvimento da analítica e demonstrar as *key performance indicators* (KPI). Sendo que, foi ainda adicionado um novo *endpoint* que permite fazer o *download* dos dados da campanha.

O *Kibana* [75] está ligado ao *Elastic Search* [76], que permite armazenar os dados de forma indexada, para poderem ser utilizados pelos utilizadores através de consultas complexas na ferramenta de visualização de dados. O *Elastic Search* [76] está integrado com o microserviço, criando o fluxo de dados dos convites das campanhas do microserviço para o *Elastic Search*, oferecendo ao utilizador a possibilidade de visualizar as KPI no *Backoffice*.

Visto que não foi desenvolvido o protótipo de alta fidelidade pela equipa de design para esta funcionalidade, o UI foi desenvolvido tendo em conta o protótipo de baixa fidelidade, presente na Figura 37, no entanto, ao longo do desenvolvimento do UI existiu ajuda de um elemento da equipa de design, na revisão do mesmo para que ficasse de acordo com o tema definido.

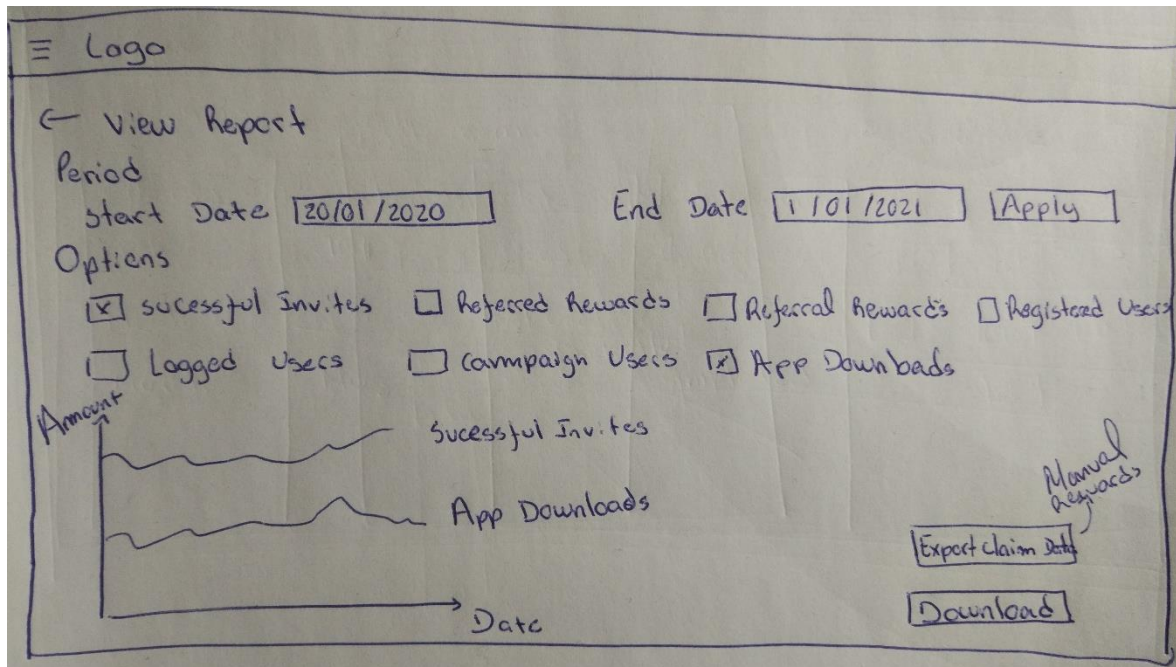


Figura 37 – Protótipo de baixa fidelidade da analítica

Ao UI já desenvolvido da página de edição, ao seleccionar a *tab* da analítica foi adicionada a operação de *download* dos dados da campanha e o conteúdo da *tab* foi desenvolvido em duas secções, a secção dos gráficos e a secção dos cartões.

Ao seleccionar a operação de *download* dos dados, é realizado um pedido a um novo *endpoint*, o qual possibilita ao utilizador a obtenção de um ficheiro de valores separados por vírgulas (CSV) com toda a informação das campanhas, permitindo-lhe fazer consultas mais avançadas e criação de gráficos mais personalizados.

A secção dos gráficos, representada na Figura 38, é constituída por três *tabs*, sendo elas as três KPI definidas, novos convites aceites, número de prémios oferecidos e montante de prémios oferecidos. Em cada *tab* existe uma *dropdown* com filtros temporais, um botão de

download e uma *tooltip*, para ajudar o utilizador a compreender o que está a ser apresentado no gráfico. A *dropdown* possui diversas opções sendo elas:

- Última semana;
- Último mês;
- Últimos três meses;
- Últimos seis meses;
- Intervalo personalizado;
- Limpeza dos filtros.

Todos os intervalos são calculados a partir da data atual, com exceção do intervalo personalizado, que utiliza uma data de início e fim personalizadas. Para a realização das operações com os intervalos temporais foi utilizada a biblioteca Moment.js [77], que oferece um auxílio importante na realização dos diferentes cálculos.

Os filtros são aplicados a todos os gráficos sempre que o utilizador seleciona um intervalo. O componente dos filtros foi desenvolvido pela equipa da WIT, que desenvolve o *Backoffice*, e desta forma reutilizado na análise das campanhas.

A opção de *download*, junto do gráfico, dá oportunidade ao utilizador de fazer *download* do conteúdo do gráfico, aplicando os filtros escolhidos por ele. Foi utilizado o mesmo *endpoint* criado para o *download* dos dados da campanha, sendo apenas acrescentados os *query parameters* necessários para que o *Backend* consiga devolver o ficheiro CSV com os dados corretos, para o intervalo escolhido pelo utilizador. Todos os *endpoints* irão ser explicados em mais detalhe no próximo capítulo.

Por fim, temos o componente do gráfico que é um *iframe* do *Kibana*. Todo o conteúdo é controlado através do URL com diversos parâmetros e, para que fosse possível atingir o resultado final foram necessários diversos testes, de forma a concluir como aplicar os filtros

através do URL. Os gráficos são histogramas, representados na Figura 38, os quais acumulam os dados para os diversos dias.

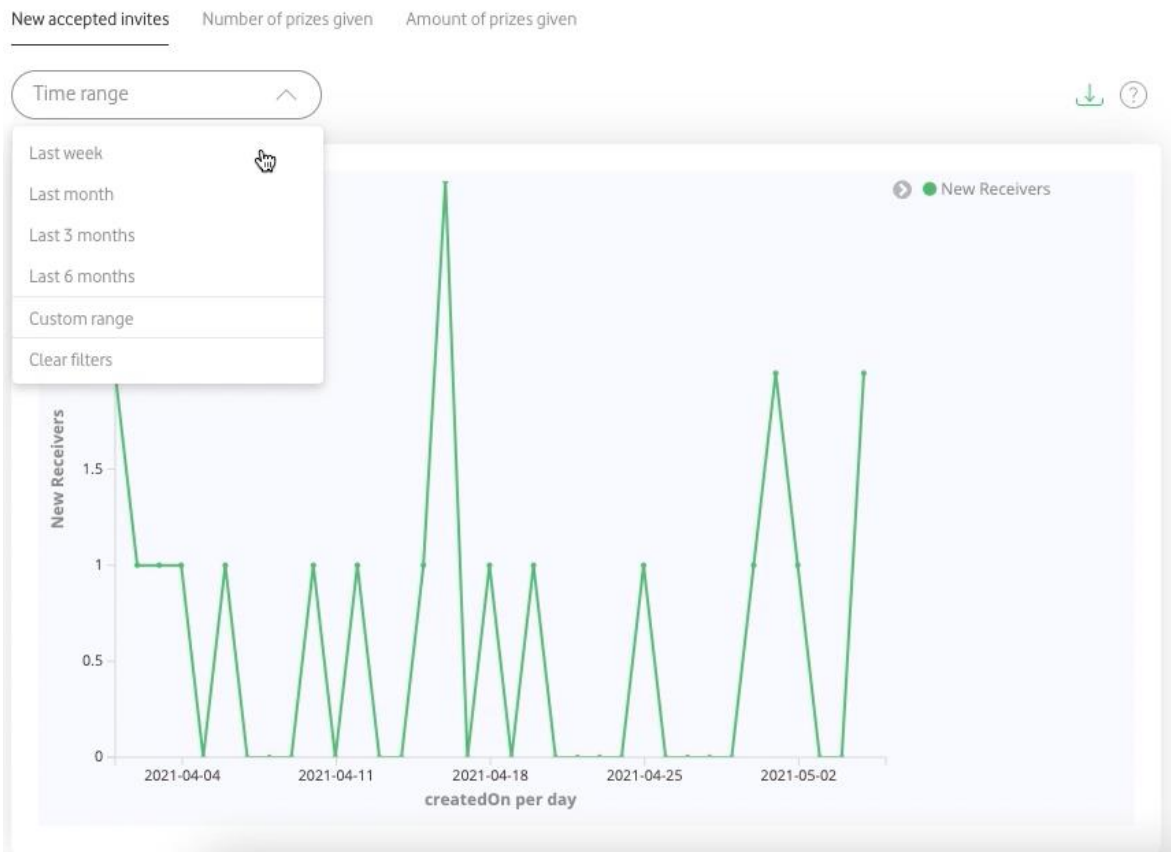


Figura 38 – Secção dos gráficos

O *Kibana* apenas representa dados para os dias para os quais tem convites. Por exemplo, uma campanha teve início no dia um, mas apenas, no dia cinco existiram convites aceites, assim, o gráfico terá início na data que teve convites e não na data de início da campanha, o mesmo acontece para o final da campanha. No entanto, de modo a mitigar esta situação quando é utilizada a funcionalidade de *download*, os dados começam no primeiro dia da campanha e terminam na data atual ou data de fim da campanha, caso a campanha esteja a decorrer ou já tenha terminado, respetivamente.

Na secção dos cartões, representada na Figura 39, são apresentadas as mesmas KPI, mas apresentadas em forma de métrica, ou seja, são mostrados os dados de forma cumulativa para a campanha. É possível, ainda, filtrar da mesma forma que na secção dos gráficos, possibilitando ao utilizador observar, de forma cumulativa, para o mesmo intervalo dos

gráficos, tirando conclusões mais concretas ou para outro intervalo em que o utilizador tenha interesse.

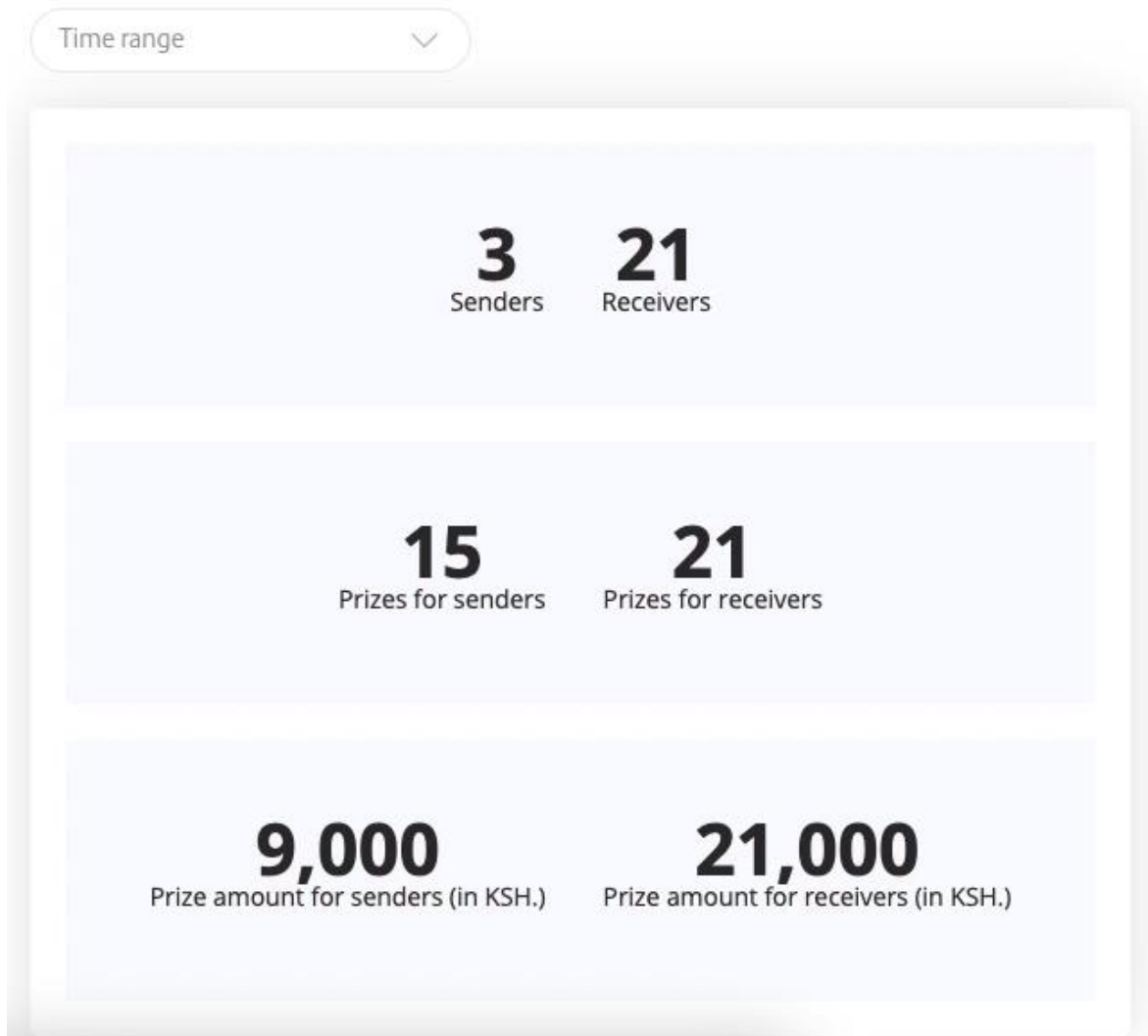


Figura 39 – Secção dos cartões

Após a analítica estar completa, procedeu-se à integração das melhorias desenvolvidas pela equipa da WIT, as quais trouxeram algumas alterações. Nomeadamente, foi adicionada uma data final ao pedido de duplicação de campanha, fazendo com que o utilizador introduza uma data de início e de fim e, conseqüente, adição deste campo ao *Backend*. A validação do intervalo temporal escolhido para a campanha passou a ser realizada no primeiro passo da criação das campanhas sendo adicionado um novo pedido ao *Backend*. Por fim, foi adicionado um novo conceito de campanhas *default*.

As campanhas *default*, são campanhas que coexistem com as campanhas a decorrer, servindo como opção caso não exista nenhuma campanha a decorrer. Ou seja, caso não exista

nenhuma campanha a decorrer e o utilizador tiver definido uma campanha *default* então esta é a que irá ser apresentada aos utilizadores finais. Foi criado um *endpoint*, que valida a possibilidade de criar uma campanha *default*, também utilizado para tornar uma campanha *default* na página de edição. Só é possível criar ou tornar uma campanha *default* se não existir nenhuma deste tipo.

Por último, foi adicionada uma camada extra de segurança ao *Elastic Search*, necessitando de *username* e *password* para realizar operações no *Kibana*. O *iframe* nativo do *Kibana* não possui um mecanismo de autenticação através do URL. De forma a mitigar este problema, foi criada uma solução que permite que os utilizadores não tenham de se autenticar duas vezes, na plataforma de gestão e no *Kibana*.

A solução consistiu na criação de um *reverse-proxy*, pelo qual são passados os URLs dos *iframes*, o qual adiciona o *header* de autenticação ao pedido permitindo o acesso às KPI de forma segura. O *header* passado apenas tem permissões de leitura, o que previne alguma ação indesejável por parte de um utilizador malicioso. A solução implementada encontra-se representada na Figura 40.

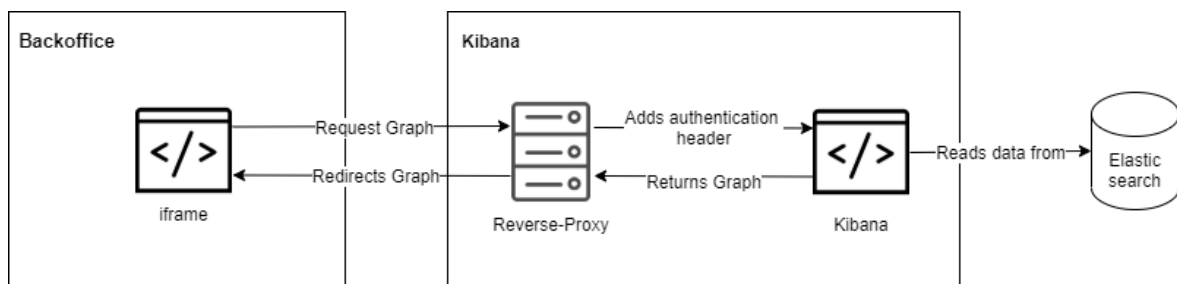


Figura 40 – Representação da solução de *reverse-proxy* implementada

6.5.Backend

Este subcapítulo tem como objetivo descrever o processo de desenvolvimento do *Backend*, o qual é constituído por diversos componentes, sendo o principal constituinte o microserviço. São ainda descritas as principais interações, assim como, a funcionalidade de cada *endpoint* criado.

A linguagem utilizada no *Backend* é Java, no entanto, existem componentes que foram criados por diferentes equipas de desenvolvimento da WIT o que implica diferentes arquiteturas e diferentes *frameworks*, assim, o *Backend* do *Backoffice* utiliza a *framework*

Jersey [78] com um pouco de *Spring Boot* [79], que possibilita a integração dos componentes das outras equipas, os quais são todos desenvolvidos utilizando *Spring Boot* [79].

6.5.1. Microserviço

Começando pelo microserviço, como ponto principal da arquitetura, sendo este imprescindível para o bom funcionamento do programa de *refer a friend*, é baseado na arquitetura utilizada pela equipa da WIT que desenvolve o *Backend*, para o cliente referente ao presente estágio. O seu *core* é constituído essencialmente por controladores, modelo, repositório e serviços.

Os controladores recebem os pedidos e devolvem uma resposta dos serviços, é nestes que está localizada toda a lógica de negócio, esta pode necessitar de fazer consultas à base de dados, para isso, são utilizados os repositórios que por omissão já possuem algumas consultas, no entanto, é possível criar consultas mais personalizadas. O modelo é mapeado para a base de dados e posteriormente associado aos repositórios, para serem executadas as consultas. Este mecanismo encontra-se representado na Figura 41.

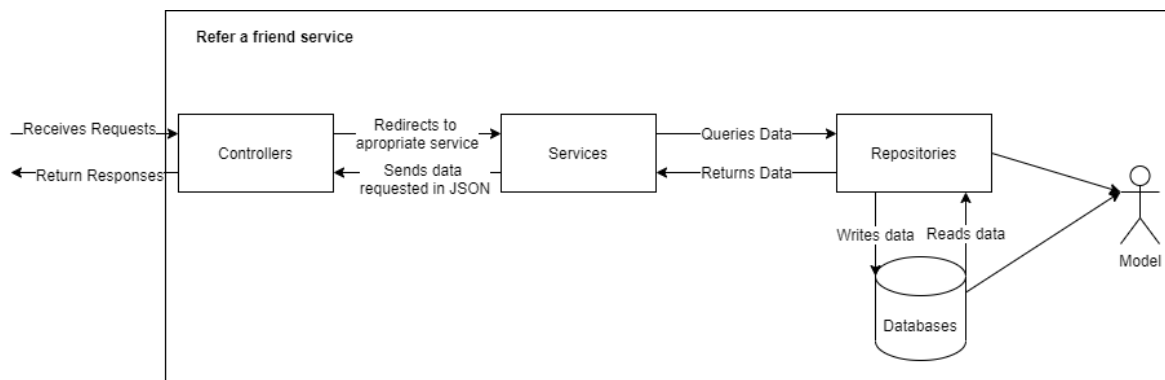


Figura 41 – Mecanismo de funcionamento do microserviço

Assim, foi iniciado o desenvolvimento dos *endpoints* necessários para a aplicação. O primeiro *endpoint* desenvolvido, foi do tipo GET e devolvia a lista de convites para um determinado utilizador passando o id do utilizador através do URL. No entanto, este *endpoint* deixou de existir, quando foi ajustado do modelo, tendo em conta o já implementado pela equipa da WIT, no *Backend* do *Backoffice*. Este ajuste teve como objetivo facilitar a integração do microserviço com o *Backend* do *Backoffice*, diminuindo o esforço de desenvolvimento.

A comunicação feita com o microserviço, é realizada através da integração com um *software development kit* (SDK), o qual é implementado em conjunto com o microserviço e

tem o comportamento semelhante ao de um *proxy*. Esta forma de comunicação é a adotada pela equipa da WIT, sendo assim adotada para o microserviço. O SDK é compilado para *Java ARchive* (JAR), sendo posteriormente adicionado como dependência aos projetos em que é necessário comunicar com o microserviço criado.

A app comunica com o microserviço através da *Consumer app facade*, que na arquitetura de microserviços funciona como *gateway*. A *Consumer app facade*, tem as diversas dependências dos SDKs, usando-as para comunicar com os diferentes microserviços de forma sequencial. Esta possui um mecanismo de *cache* para os pedidos e um *endpoint* de *fallback*, que permite à aplicação continuar a execução mesmo que um pedido ainda não esteja completo, este mecanismo foi utilizado e irá ser explicado mais à frente no capítulo.

6.5.2. Desenvolvimento inicial dos *endpoints*

Em seguida, foi implementado o *endpoint*, do tipo POST, que guarda um *referral*. No *body* do pedido é passada a informação do convite, nomeadamente o id da campanha, id do remetente e do recetor e a *flag* utilizada no controlo das visualizações dos convites. Este *endpoint* foi sofrendo algumas alterações, começando por devolver apenas o id do *referral* guardado, para garantir o sucesso do pedido.

Adicionaram-se as validações que apuram se o convite é legítimo, verificando se o recetor do convite ainda não aceitou outro convite ou se o convite pertence a uma campanha já terminada.

Ao integrar o envio das notificações na *facade*, foi necessário acrescentar uma *flag* de controlo que permitisse distinguir o conteúdo da notificação do remetente. Nomeadamente, caso este já tivesse atingido o limite de convites o conteúdo seria, por exemplo, “Another friend accepted your invitation”, caso contrário seria “You won a reward!”. Com esta integração, o pedido passou a demorar mais alguns segundos, assim, tornou-se o pedido assíncrono, oferecendo a possibilidade à app de continuar a execução, comunicando com o *endpoint* de *fallback* para atualização do estado do convite.

Quando é enviado o pedido é gerado um id e devolvida a resposta marcada como pedido pendente. Com o id, a app comunica com o *fallback*, para obter o estado do mesmo, ao ser dado o pedido como terminado é enviada a resposta para a app. Este mecanismo já se encontrava implementado pela equipa da WIT tendo sido apenas adaptado para o caso de

uso. A app faz pedidos ao *fallback* por um determinado número de milissegundos, sendo a duração entre pedidos aumentada gradualmente até atingir o limite máximo, ficando o pedido marcado como não sucedido e com erro de *timeout*, caso seja atingido este limite. Para uma melhor compreensão, o mecanismo de *fallback* está representado na Figura 42.

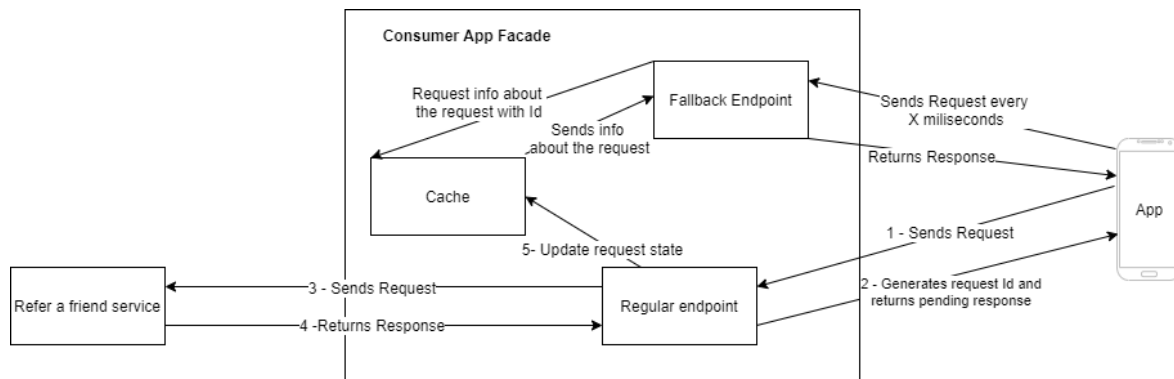


Figura 42 – Representação do mecanismo de *fallback*

Com a adaptação do modelo ao criado no *Backend* do *Backoffice*, surgiu a informação do ecrã do recetor, a qual era necessária no preenchimento do *payload* das notificações, sendo assim, foi criado um objeto que contém o id do *referral*, a informação do ecrã do recetor e a *flag* que distingue o conteúdo da notificação enviada ao remetente.

Na *facade*, após receber a resposta do pedido ao microserviço, é criada a notificação do recetor e enviada. Após isto desenrola-se o mesmo mecanismo para a notificação do remetente, dando por fim como terminado o pedido. As notificações são enviadas pelo *firebase*, as quais são requisitadas pelo microserviço das notificações.

Por fim, ao ser adicionada a ligação à base de dados de BI, foi adicionada a validação do passo de conversão do remetente, através de uma consulta que verifica se o evento definido já existe na base de dados.

No *endpoint* de atualização dos convites visualizados, do tipo PUT, é validada a existência de uma campanha a decorrer, se existir é realizada uma consulta que devolva os convites de um determinado id de um utilizador e a *flag* de controlo destes utilizadores é atualizada para *true*. Caso alguma atualização falhe, o pedido devolve *false*, caso contrário, devolve *true*.

Seguiu-se o desenvolvimento do controlador das campanhas, o qual possui grande parte dos *endpoints* e foi introduzido quando se iniciou a integração com o *Backoffice*, trazendo

as mudanças ao modelo desenvolvido. Um dos *endpoints* é utilizado pela aplicação, os restantes são utilizados pelo *Backoffice* ou pelo serviço de *jobs*.

O *endpoint* utilizado pela app, valida a existência de uma campanha a decorrer, se existir é feita uma consulta, utilizando o repositório dos *referrals*, para preencher os convites do utilizador identificado pelo id no URL e para a campanha que se encontra a decorrer naquele momento. Para manipular o conteúdo da resposta, são criados *resources* com apenas os campos necessários à aplicação, e, de forma a evitar converter todos os campos de forma manual, utiliza-se a biblioteca *ModelMapper* [80], que faz o mapeamento de forma automática.

Após os *endpoints* necessários para a aplicação estarem prontos, iniciou-se a integração com o *Backoffice*, para isto recorreu-se ao *Backend* do mesmo. Começou por se adicionar o SDK como dependência e injetar o serviço, que irá tratar das comunicações com o SDK. Em seguida, foi ajustado o modelo do SDK e do microserviço.

As comunicações que requerem objetos complexos no *body*, necessitam a conversão dos mesmos de forma manual para os objetos conhecidos pelo SDK. Este foi o principal desenvolvimento exercido no *Backend* do *Backoffice*, com algumas exceções em determinados *endpoints* que implicam a introdução de alguma lógica que irá ser explicada em seguida.

Todos os *endpoints* recebem como *query parameter*, o cliente, que enviou o pedido para que seja possível distinguir o serviço que irá fazer o tratamento. Para além deste, existem outros parâmetros relevantes que irão ser mencionados, para o *endpoint* em que são utilizados.

Começando pelo *endpoint* que devolve todas as campanhas, do tipo GET, é um *endpoint* simples, no qual é apenas invocado o SDK que permite a comunicação com o *endpoint* correspondente. No microserviço é realizado o mapeamento para o *resource* criado, que conseqüentemente irá ser convertido para o modelo conhecido pelo *Backoffice*.

O *endpoint* que permite obter os detalhes de uma campanha é bastante semelhante ao anterior, a única diferença é que este recebe por parâmetro do URL, o id da campanha, fazendo com que o microserviço execute a pesquisa por este id e devolva a campanha com os seus detalhes.

Quanto à criação e edição de campanhas, apesar de ser utilizado um *endpoint* diferente, com tipos POST e PUT, respectivamente, estes utilizam o mesmo método do serviço para o tratamento de dados. Pelo que, para alguns campos é necessário tratamento diferente caso se esteja a editar ou a criar. No entanto, este tratamento específico só foi necessário quando foram introduzidas as ações de início e fim manuais, as quais foram introduzidas com as melhorias no *Backoffice*.

É de notar que as imagens são convertidas para Base64, o que faz com que quando é requerida informação, é necessário comparar as imagens presentes no servidor com a imagem recebida, para que possa ser encontrada a corresponde àquele Base64 associando assim a um URL para ser apresentado no *Backoffice*.

É importante também referir que, como no *Backoffice*, para as datas de início e fim, não é definida a parte temporal da data, (sendo esta definida no *Backend* do *Backoffice*), define-se automaticamente que o início seja às zero horas da data de início escolhida e o final seja no último minuto da data de fim escolhida.

Após todos os parâmetros serem convertidos para o formato conhecido pelo SDK, é feita a comunicação com o microserviço, neste é realizado um passo bastante importante que permite que seja possível a gestão automática das campanhas, a validação do intervalo de datas escolhido.

Existem múltiplos casos que são validados para que tudo corra como esperado, sendo eles:

- Data de início antes da atual;
- Data de fim antes da data de início;
- Data de início dentro de uma campanha;
- Data de fim dentro de uma campanha;
- Data de início de outra campanha dentro do intervalo definido;
- Data de fim de outra campanha dentro do intervalo definido.

Se estivermos a fazer uma edição, nenhum dos passos anteriores é necessário, se o utilizador escolher um intervalo dentro do intervalo já definido para aquela campanha, caso contrário, é necessário validar todos os passos referidos anteriormente.

Dos *endpoints* iniciais, antes das melhorias do *Backoffice* e do início de desenvolvimento da analítica, faltam o *endpoint* de apagar campanhas, do tipo DELETE, o qual recebe o id da campanha como parâmetro do URL, usado no microserviço para encontrar a campanha e apagá-la.

O *endpoint* de duplicar campanhas, que recebe, por parâmetro, o id da campanha a duplicar, inicialmente apenas recebia a data de início no *body* do pedido, sendo a duração da campanha calculada no microserviço através da diferença entre a data de início recebida e a data de fim da campanha a duplicar.

Após a implementação das melhorias do *Backoffice*, o *endpoint* passou a receber a data de fim, retirando a necessidade do comportamento descrito anteriormente, passando a calcular o intervalo através das duas datas recebidas. É de realçar que neste *endpoint*, também são utilizados os passos acima descritos, para a validação do intervalo escolhido pelo utilizador.

Por fim, antes de ser descrita a parte do desenvolvimento da analítica e das melhorias implementadas, existe o *endpoint* que permite a gestão automática das campanhas, o qual é utilizado pelo serviço de *Jobs*.

O serviço de *Jobs* foi criado e é mantido pela equipa da WIT responsável pelo desenvolvimento do *Backend* do cliente referente ao estágio, tendo como objetivo criar *Jobs* que ocorrem para um intervalo definido de forma indeterminada. Para este *endpoint*, foi decidido que seria de minuto a minuto, pois podem existir campanhas a começar a qualquer minuto do dia caso sejam começadas de forma manual, assim como terminadas. Esta decisão teve também em conta uma perspetiva futurista, na qual seria possível introduzir a altura do dia para a qual se queria iniciar ou terminar a campanha.

Assim, é validado a cada minuto se existe alguma campanha com data de início menor que a atual, o que significa, que tem de passar ao estado a decorrer, e também se existe alguma campanha com data de fim menor que a atual para que possa ser terminada.

Após toda a integração com o *Backend* do *Backoffice*, desenvolveu-se o *endpoint* necessário para o fluxo de dados da analítica para o *Kibana* e o *endpoint* usado para fazer o *download* dos dados das campanhas.

O *endpoint* que controla o fluxo de dados da analítica é utilizado por um *Job*, executado de hora a hora. Como a atualização da analítica não tinha como requisito, ser em tempo real, decidiu-se este intervalo de tempo de forma a diminuir o *overhead*. É de realçar que estes tempos podem ser alterados de forma bastante simples no ficheiro de configurações.

Neste *endpoint*, é validado se o *Elastic Search* já possui o convite, caso não exista irá ser criado o objeto com base nas configurações da campanha a que pertence. Para cada tipo de utilizador (remetente e recetor) foi necessário verificar se estavam elegíveis para receber prémio possibilitando o preenchimento correto das informações necessárias.

Assim, caso o tipo de prémio for NENHUM, o prémio do remetente é marcado como zero, caso o tipo de prémio for AGREGADO, foi elaborada uma consulta que permite contar o número de convites aceites para uma determinada campanha até à data do convite em questão, quando este número for igual ao limite, o prémio do remetente é marcado com o montante do prémio definido, caso contrário é marcado como zero. Por último, para o tipo de prémio ÚNICO, caso não exista limite, é sempre atribuído o montante do prémio, caso exista, foi utilizada a mesma consulta anterior, no entanto, a condição é ligeiramente diferente, pois até ser atingido o limite, o prémio do remetente é marcado com o montante, caso contrário, o prémio é marcado como zero.

Para o recetor, é mais simples, pois caso o prémio seja do tipo NENHUM, este irá ter o montante atribuído como zero, caso contrário sendo o tipo ÚNICO, é atribuído o montante do prémio. É importante realçar que para o *endpoint*, a ser descrito de seguida é utilizada uma lógica muito semelhante à descrita anteriormente, sendo assim, facilitada a sua reutilização.

O outro *endpoint*, que permite o *download* dos dados, recebe como parâmetro o id da campanha, e recebe como *query parameters* opcionais a data de início e de fim e a KPI pelo qual foi filtrada. Caso não receba a KPI, significa que o utilizador selecionou a opção de fazer *download* de todos os dados da campanha. É devolvido um *array* de *bytes*, que representa um ficheiro no formato CSV e para auxílio nesta funcionalidade foi utilizada a biblioteca Apache Commons CSV [81].

Cada linha do ficheiro representa um convite aceite. Na construção do ficheiro, foi necessário a utilização da lógica descrita anteriormente para o preenchimento de campos

mais personalizados, como por exemplo, o campo que permite saber se o remetente do convite está elegível para receber o prémio.

Caso seja recebida a KPI, como *query parameter*, cada linha do ficheiro representa um dia da campanha, assim é necessário calcular o intervalo para o qual o ficheiro irá ser preenchido, podendo este ser recebido através dos *query parameters*, que representam a filtragem dos dados feita pelo utilizador no *Backoffice*. Caso não seja definida data de início é utilizada a data de início da campanha. Se não existir data de fim e a campanha tenha sido terminada é utilizada a data de fim da campanha, caso contrário é utilizada a data atual. Após serem definidas as datas de início e fim é calculado o número de dias entre elas. É importante realçar que para cada KPI, a estrutura do ficheiro é diferente.

Para o ficheiro de número de novos convites aceites por dia, é percorrido o número de dias calculado anteriormente e realizada a consulta que devolve o número de convites para o determinado dia, com início às zero horas e fim no último minuto do dia.

Na KPI de número de prémios oferecidos é utilizada a mesma consulta para obter os convites de um determinado dia, após esta operação é utilizada a lógica descrita anteriormente para concluir se o remetente e recetor estão elegíveis para receber o prémio, fazendo a soma para o determinado dia.

Por fim, o montante de prémios oferecidos, tem um comportamento muito semelhante ao da KPI anterior, com uma pequena mudança, em que são somados os montantes ao invés de serem somados os números de utilizadores.

Quando o pedido é invocado pelo *Backend* do *Backoffice* e é recebida a resposta, são adicionados os *headers* de *content-disposition* com o *filename* para que o *Backoffice* consiga reconhecer o ficheiro.

6.5.3. Melhorias

Por último, foram integradas as melhorias feitas no *Backoffice*, o que fez com que fosse necessário rever alguma lógica implementada. Foram ainda adicionados dois novos *endpoints*, um que valida se existe uma campanha *default* configurada, e para isto foi apenas adicionada uma consulta simples que verifica se existe uma campanha com a *flag* de controlo das *default campaigns* com o valor *true*. O outro *endpoint*, valida um determinado período

temporal, no entanto, esta funcionalidade já estava implementada e foi apenas necessário criar o *endpoint*.

A introdução das campanhas *default* requereu uma atenção extra, pois estas não podem entrar na validação dos intervalos temporais, tendo ainda de ser adicionada uma verificação extra nos *endpoints* que utilizam a campanha a decorrer para algum tipo de ação, pois só podem utilizar a campanha *default*, caso não exista mais nenhuma a decorrer naquele momento.

Estas melhorias, trouxeram também algumas alterações em relação ao *Backend* do *Backoffice*, nomeadamente no serviço de criação das campanhas. Tal como referido anteriormente, este é utilizado para edição e criação das campanhas, sendo assim, com a introdução das ações de início e fim das campanhas de forma manual, foi necessário acrescentar duas *flags* que controlam estas ações, de forma a adotar o comportamento correto quando forem realizadas.

Caso o utilizador escolha iniciar a campanha de forma manual, a data de início, não pode ser a meia-noite desse dia, mas sim o exato momento, o mesmo acontece com o término da campanha, a campanha não pode terminar no final daquele dia, mas no exato momento.

6.6. Validação e Verificação

Este subcapítulo tem como objetivo descrever o processo efetuado para a verificação e validação das funcionalidades elaboradas ao longo do estágio. Ao longo do desenvolvimento foram realizados diversos testes para garantir que a comunicação entre os diversos componentes estava a decorrer da forma correta para que a solução funcionasse como um todo. Para testar os *endpoints* do microserviço de controlo das campanhas de *refer a friend*, foi utilizada a ferramenta Postman [82] que é adotada por bastantes pessoas para realizar testes a APIs, permitindo concluir se os *endpoints* estão disponíveis e se a resposta recebida, é a esperada. Para além do *happy path*, também é possível testar os *edge cases* implementados, assim como os casos de erro de forma mais simples, sem ser necessário a interação com as aplicações clientes, como a aplicação Android ou plataforma de gestão.

Apesar de serem realizados estes testes é necessário testar os diferentes casos através da aplicação e plataforma de gestão pois a existência dos intermediários, como o *Backend* do

Backoffice e a *Facade*, possuem alguns dos processos, como o de validação da autenticação que podem interferir na resposta obtida.

No final das iterações, o código elaborado era submetido a uma etapa de *Code Review* onde algum membro da equipa de desenvolvimento da WIT revia o código, onde colocava sugestões de melhoria. Estas sugestões foram sempre tidas em consideração para que o código produzido tivesse a qualidade desejada. Para além desta etapa, era realizada uma demonstração onde estavam presentes alguns membros da WIT com objetivo de apresentar as funcionalidades implementadas, concluindo através do *feedback* obtido se as funcionalidades estavam concluídas ou se ainda necessitavam de algum trabalho adicional.

Por fim, a plataforma antes de ser colocada em ambiente de produção é necessário que esta seja sujeita a um ambiente de *Quality Assurance (QA)*, no qual a equipa da WIT faz testes exploratórios de modo a relatar erros, medir a performance e a escalabilidade. Os relatórios obtidos são posteriormente analisados para que possam ser melhorados estes parâmetros de forma que o cliente fique satisfeito com a solução implementada. É importante realçar que esta etapa não estava incluída nos objetivos de estágio pelo que não foi atingida durante o período de estágio.

6.7. Resumo

Neste capítulo, dividido em seis subcapítulos, Arquitetura, Ferramentas Utilizadas, Aplicação, *Backoffice*, *Backend*, Validação e Verificação foi descrito todo o processo de desenvolvimento e teste, enunciando as principais tecnologias utilizadas, assim como alguns conceitos e bibliotecas. Este permitiu, também, explicar de que forma foram integrados os diferentes componentes da arquitetura e qual o fluxo de dados ao ser utilizada a plataforma como um todo. Permitindo, ainda, conhecer o ambiente de desenvolvimento da plataforma, através de uma breve explicação das diferentes ferramentas utilizadas para a possível conclusão da mesma.

7. Conclusão

O presente estágio teve como objetivo o estudo e concepção uma plataforma de *refer a friend* que oferecesse a possibilidade de integração em diferentes plataformas e diferentes clientes.

Com a evolução do mundo digital e a criação de uma dependência a este mundo através dos nossos *smartphones*, uma empresa que deseje adquirir clientes de uma forma simples e eficaz, pode recorrer ao tipo de estratégia implementada para o presente estágio.

No estágio, foi implementada uma solução funcional *end-to-end*, no entanto, é possível a redução desta solução, apenas à componente aplicacional, possuindo por exemplo, apenas uma campanha que não necessita de qualquer tipo de gestão. Em contexto de estágio, decidiu-se a construção completa para se retirar o maior partido possível, das campanhas de *referral*, correspondendo a uma prova de conceito preparada para ser integrada no ecossistema do cliente.

Todos os objetivos técnicos definidos, foram atingidos em tempo útil e como planeado. Devido a algumas funcionalidades terem sido desenvolvidos mais rápido do que o esperado, foi, ainda, possível, a integração de melhorias feitas por parte de outras equipas da WIT, assim como melhorias ao nível de código nos componentes elaborados de forma autónoma e também mediante *feedback* de elementos das equipas da WIT. Foram também utilizadas tecnologias diferentes, as quais nunca tinha utilizado antes, trazendo bastantes benefícios para o futuro.

Quanto aos objetivos gerais, foram também estes atingidos com sucesso, houve uma integração muito boa por parte dos elementos da WIT, estando sempre disponíveis quando existisse algum problema, foi possível experienciar a forma de trabalhar numa empresa como a WIT, aumentando não só capacidades técnicas, mas também capacidades de comunicação, muito importantes nos dias de hoje. Para além disto, a ajuda da *Business Analyst*, como tutora foi essencial para que o desenvolvimento tivesse uma visão menos técnica e mais ampla, o que conseqüentemente possibilitou a idealização da solução *end-to-end*.

É importante realçar que todo o estágio foi realizado de forma remota o que proporcionou algumas necessidades diferentes, tais como, a exigência de um desenvolvimento mais autónomo, apesar da disponibilidade por parte dos elementos da WIT para ajudar em

qualquer problema que existisse. Para além disso, foi necessária uma maior concentração para manter o foco, continuando a ser produtivo, de modo a seguir o plano apresentando os resultados em tempo útil, como esperado.

7.1. Trabalho futuro

Uma das principais características do desenvolvimento de *software*, é que não existe *software* perfeito, existindo sempre algum *bug* ou algum *edge case* que não foi previsto e que é necessário fazer *fix* ou implementação. O mesmo se reflete no estágio, apesar de a solução estar funcional *end-to-end*, existem pontos de melhoria, como por exemplo, a otimização do código, reduzindo, tempos de resposta do *Backend*.

Na aplicação é possível melhorar algumas mensagens de erro, assim como, alguns casos de erro que se encontrarem sem *feedback* visual.

No *Backend* do *Backoffice*, a conversão para Base64 e respetiva procura pela imagem em Base64 recebida, não é a mais otimizada, podendo por exemplo incluir a imagem em formato URL e em Base64.

Igualmente, a criação de um componente no *Backoffice*, que permita a comunicação com o Kibana de forma mais simples pode trazer muitos benefícios, podendo este ser reutilizado em outros projetos.

Existe, ainda, uma outra perspetiva que devido à otimização do tempo de estágio, foi possível iniciar, trata-se de uma generalização completa da plataforma de *refer a friend*, podendo esta ser utilizada para qualquer cliente. Foi iniciado o desenvolvimento do UI da aplicação utilizando a ferramenta *Cross-Platform*, React Native, presente no Anexo D – UI em React Native, que, apesar de comunicar com o microserviço, ficou a faltar ainda a parte funcional permitindo aceitar convites e receber notificações.

Por fim, será também importante a criação de um novo *Backoffice* permitindo a gestão das campanhas para diferentes clientes e produtos e ainda, o isolamento do microserviço e recriação dos serviços necessários do cliente utilizado para contexto de estágio.

Bibliografia

- [1] E. Sirma and N. Walter, “WORD-OF-MOUTH MARKETING FROM A GLOBAL PERSPECTIVE,” 2009.
- [2] P. Schmitt, B. Skiera, and C. Van Den Bulte, “Referral programs and customer value,” *J. Mark.*, vol. 75, no. 1, pp. 46–59, 2011, doi: 10.1509/jmkg.75.1.46.
- [3] R. Gangseog and L. Feick, “A penny for your thoughts: Referral reward programs and referral likelihood,” *J. Mark.*, vol. 71, no. 1, pp. 84–94, Jan. 2007, doi: 10.1509/jmkg.71.1.84.
- [4] “A new way to measure word-of-mouth marketing | McKinsey.” <https://www.mckinsey.com/business-functions/marketing-and-sales/our-insights/a-new-way-to-measure-word-of-mouth-marketing#> (accessed Jun. 18, 2021).
- [5] “WIT Software – Witness the difference.” <https://www.wit-software.com/> (accessed Jun. 22, 2021).
- [6] “Careers – WIT Software.” <https://www.wit-software.com/careers/> (accessed Jul. 24, 2021).
- [7] “Expertise – WIT Software.” <https://www.wit-software.com/expertise/> (accessed Jul. 24, 2021).
- [8] “Uber Eats | Take-away e entrega de comida ao domicílio.” <https://www.ubereats.com/pt> (accessed Jun. 25, 2021).
- [9] “TheFork - Reserve nos melhores restaurantes da Europa.” <https://www.thefork.pt/> (accessed Jun. 25, 2021).
- [10] “Sponsorship - TheFork Referral Program.” <https://www.thefork.com/sponsorship> (accessed May 01, 2021).
- [11] “Tropical Smoothie Cafe.” <https://www.tropicalsmoothiecafe.com/> (accessed Jun. 25, 2021).
- [12] “Mobile App FAQs | Tropical Smoothie Cafe.” <https://www.tropicalsmoothiecafe.com/app-faq-app/> (accessed Jun. 17, 2021).

- [13] “Yorn És Tu.” <https://www.yorn.net/yorn/index.html> (accessed May 01, 2021).
- [14] “Shake It - Joga e Ganha - Yorn.” <https://www.yorn.net/yorn/shakeit.html#> (accessed May 01, 2021).
- [15] “Regulamento shake it.” https://www.yorn.net/content/dam/digital-yorn-legacy/doc/yorn_shake_it_regulamento.pdf?v=333 (accessed Jun. 25, 2021).
- [16] “Uma forma mais simples de gerir o seu dinheiro | Revolut PT.” <https://www.revolut.com/pt-PT> (accessed Jun. 25, 2021).
- [17] “Airbnb: alojamentos de férias, cabanas, casas de praia, casas e experiências únicas.” <https://www.airbnb.pt/> (accessed Jun. 25, 2021).
- [18] “Airbnb Referral Program Terms and Conditions - Airbnb Help Center.” <https://www.airbnb.com/help/article/2269/airbnb-referral-program-terms-and-conditions> (accessed May 01, 2021).
- [19] “Dropbox Business.” <https://www.dropbox.com/?landing=dbv2> (accessed Jun. 25, 2021).
- [20] “Invite friends to Dropbox to earn storage space | Dropbox Help.” <https://help.dropbox.com/accounts-billing/space-storage/earn-space-referring-friends> (accessed Jun. 17, 2021).
- [21] “Veículos elétricos, energia solar e energia limpa | Tesla Portugal.” https://www.tesla.com/pt_pt (accessed Jun. 25, 2021).
- [22] “Programas de referência | Tesla.” https://www.tesla.com/pt_PT/support/referral-program (accessed May 01, 2021).
- [23] “Melhor aplicação para tirar notas - Organize as suas notas com o Evernote.” <https://evernote.com/intl/pt/> (accessed Jun. 25, 2021).
- [24] “How to refer a friend to Evernote – Evernote Help & Learning.” <https://help.evernote.com/hc/en-us/articles/209005427-How-to-refer-a-friend-to-Evernote> (accessed May 01, 2021).
- [25] “World of Warcraft.” <https://worldofwarcraft.com/pt-br/> (accessed Jun. 25, 2021).

-
- [26] “Recruit A Friend - World of Warcraft.” <https://worldofwarcraft.com/en-us/game/recruit-a-friend> (accessed Jun. 17, 2021).
- [27] “Fiverr - Freelance Services Marketplace for Businesses.” <https://www.fiverr.com/> (accessed Jun. 25, 2021).
- [28] “Give your friends a free Gig | Fiverr.” https://www.fiverr.com/referral_program (accessed Jun. 17, 2021).
- [29] “Humble Bundle | game bundles, book bundles, software bundles, and more.” <https://www.humblebundle.com/> (accessed Jun. 25, 2021).
- [30] “Humble Choice - Referral Program – Humble Bundle.” <https://support.humblebundle.com/hc/en-us/articles/115000170207-Humble-Monthly-Referral-program> (accessed Jun. 17, 2021).
- [31] “T-Mobile & Sprint merged to create the leader in 5G.” <https://www.t-mobile.com/> (accessed Jun. 25, 2021).
- [32] “Refer-a-Friend program | T-Mobile Support.” <https://www.t-mobile.com/support/new-to-tmobile/refer-a-friend-program> (accessed Jun. 17, 2021).
- [33] “Three | Phones, Broadband & SIM Only deals.” <http://www.three.co.uk/> (accessed Jun. 25, 2021).
- [34] “Terms & Conditions | Three.” <http://www.three.co.uk/terms-conditions/refer-a-friend> (accessed May 01, 2021).
- [35] “Mint Mobile | Wireless that’s Easy, Online, \$15 Bucks a Month.” <https://www.mintmobile.com/> (accessed Jun. 25, 2021).
- [36] “Mint Mobile Refer-A-Friend Program Terms & Conditions | Mint Mobile.” <https://www.mintmobile.com/referral-terms/> (accessed May 01, 2021).
- [37] “Explore os serviços Uber | Uber.” <https://www.uber.com/pt/pt-pt/> (accessed Jun. 25, 2021).
- [38] “How Referrals Work for Delivery People | Uber.” <https://www.uber.com/us/en/deliver/basics/earnings/how-referrals-work/> (accessed Jun. 17, 2021).

- [39] “N26 The Mobile Bank — N26.” <https://n26.com/en-eu> (accessed Jun. 25, 2021).
- [40] “N26 Referral Program Terms&Conditions.” <https://docs.n26.com/legal/01+DE/01+Account/en/15account-terms-and-conditions-friend-referral-3.0-en.pdf> (accessed May 01, 2021).
- [41] “Seja um motorista ou peça uma corrida já – Lyft.” <https://www.lyft.com/> (accessed Jun. 25, 2021).
- [42] “Lyft Referral Program Rules.” <https://www.lyft.com/terms/referrals> (accessed May 01, 2021).
- [43] “Wish - Comprar ficou mais divertido.” <https://www.wish.com/> (accessed Jun. 25, 2021).
- [44] “Last Minute Hotel Deals at Great Hotels - HotelTonight.” <https://www.hoteltonight.com/> (accessed Jun. 25, 2021).
- [45] “Last Minute Hotel Deals at Great Hotels.” <https://www.hoteltonight.com/terms-of-use/campaign-terms> (accessed May 01, 2021).
- [46] M. L. Despa, “Comparative study on software development methodologies,” 2014.
- [47] A. Chandrababu, “A Comparison between Agile and Traditional Sowane Development Methodologies.” 2005.
- [48] K. Schwaber and J. Sutherland, “The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game,” 2020.
- [49] “About | bpmn.io.” <https://bpmn.io/about/> (accessed May 24, 2021).
- [50] “IntelliJ IDEA overview | IntelliJ IDEA.” <https://www.jetbrains.com/help/idea/discover-intellij-idea.html#next-steps> (accessed May 22, 2021).
- [51] “Conheça o Android Studio | Desenvolvedores Android.” <https://developer.android.com/studio/intro> (accessed May 22, 2021).
- [52] “Documentation for Visual Studio Code.” <https://code.visualstudio.com/docs> (accessed May 23, 2021).

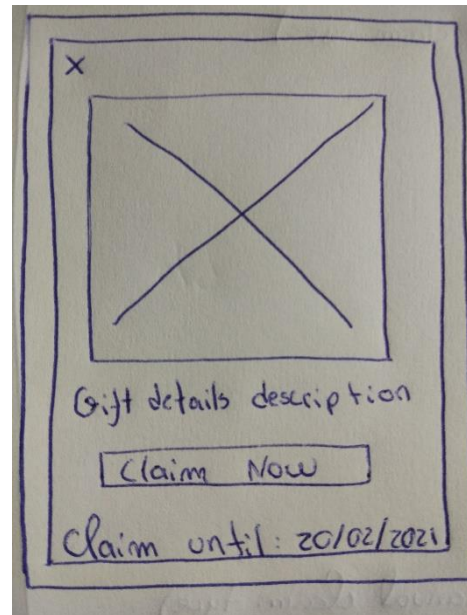
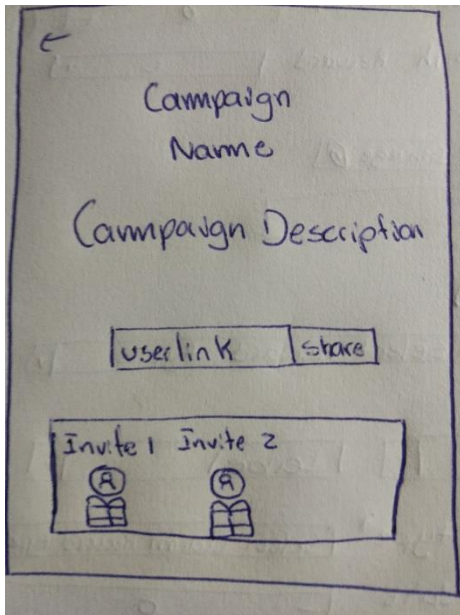
- [53] “Docker overview | Docker Documentation.” <https://docs.docker.com/get-started/overview/> (accessed May 23, 2021).
- [54] “Mobile Operating System Market Share Worldwide | StatCounter Global Stats.” <https://gs.statcounter.com/os-market-share/mobile/worldwide> (accessed May 28, 2021).
- [55] “Mobile Android Version Market Share Worldwide | StatCounter Global Stats.” <https://gs.statcounter.com/android-version-market-share/mobile/worldwide/#monthly-202004-202104> (accessed May 28, 2021).
- [56] “Mobile iOS Version Market Share Worldwide | StatCounter Global Stats.” <https://gs.statcounter.com/ios-version-market-share/mobile/worldwide/#monthly-202004-202104> (accessed May 28, 2021).
- [57] “Kotlin for Android | Kotlin.” <https://kotlinlang.org/docs/android-overview.html> (accessed May 28, 2021).
- [58] T. Lou, “A comparison of Android Native App Architecture MVC, MVP and MVVM,” 2016.
- [59] “Criar listas dinâmicas com o RecyclerView | Desenvolvedores Android.” <https://developer.android.com/guide/topics/ui/layout/recyclerview> (accessed Jun. 05, 2021).
- [60] “Sending simple data to other apps | Android Developers.” <https://developer.android.com/training/sharing/send> (accessed Jun. 06, 2021).
- [61] “Firebase Dynamic Links | Deep link potential users to the right place inside your app.” <https://firebase.google.com/products/dynamic-links> (accessed Jun. 07, 2021).
- [62] “Create Deep Links to App Content | Android Developers.” <https://developer.android.com/training/app-links/deep-linking> (accessed Jun. 07, 2021).
- [63] “Firebase Cloud Messaging.” <https://firebase.google.com/docs/cloud-messaging> (accessed Jun. 06, 2021).
- [64] “Koin - The Kotlin Injection Framework | Koin.” <https://insert-koin.io/> (accessed Jun.

- 06, 2021).
- [65] “LiveData Overview | Android Developers.” https://developer.android.com/topic/libraries/architecture/livedata#observe_livedata_objects (accessed Jun. 06, 2021).
- [66] “Home | Realm.io.” <https://realm.io/> (accessed Jun. 06, 2021).
- [67] “Coroutines | Kotlin.” <https://kotlinlang.org/docs/coroutines-overview.html> (accessed Jun. 06, 2021).
- [68] “Asynchronous Flow | Kotlin.” <https://kotlinlang.org/docs/flow.html#flows> (accessed Jun. 06, 2021).
- [69] “Retrofit.” <https://square.github.io/retrofit/> (accessed Jun. 06, 2021).
- [70] “Handling Lifecycles with Lifecycle-Aware Components.” <https://developer.android.com/topic/libraries/architecture/lifecycle> (accessed Jun. 07, 2021).
- [71] “React – A JavaScript library for building user interfaces.” <https://reactjs.org/> (accessed Jun. 07, 2021).
- [72] “TypeScript: Typed JavaScript at Any Scale.” <https://www.typescriptlang.org/> (accessed Jun. 07, 2021).
- [73] “Storybook: UI component explorer for frontend developers.” <https://storybook.js.org/> (accessed Jun. 08, 2021).
- [74] “Axios.” <https://axios-http.com/> (accessed Jun. 09, 2021).
- [75] “Kibana: explore, visualize, descubra dados | Elastic.” <https://www.elastic.co/pt/kibana> (accessed Jun. 11, 2021).
- [76] “O que é o Elasticsearch? | Elastic.” <https://www.elastic.co/pt/what-is/elasticsearch> (accessed Jun. 11, 2021).
- [77] “Moment.js | Home.” <https://momentjs.com/> (accessed Jun. 11, 2021).
- [78] “Eclipse Jersey.” <https://eclipse-ee4j.github.io/jersey/> (accessed Jun. 14, 2021).

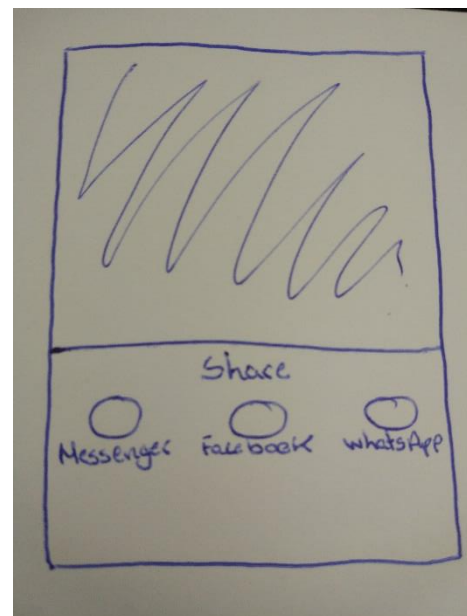
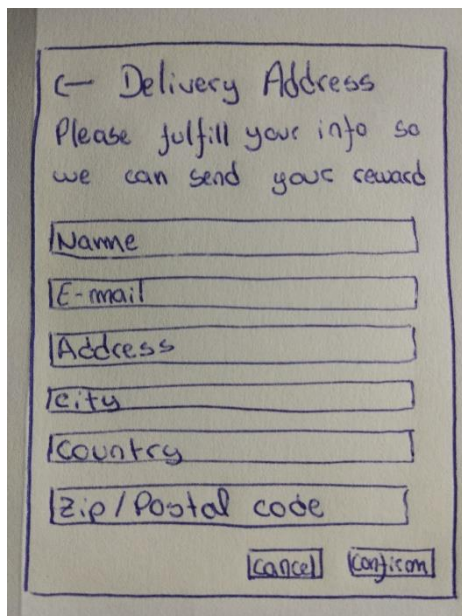
- [79] “Spring Boot.” <https://spring.io/projects/spring-boot> (accessed Jun. 13, 2021).
- [80] “ModelMapper - Simple, Intelligent, Object Mapping.” <http://modelmapper.org/> (accessed Jun. 15, 2021).
- [81] “Commons CSV – Home.” <https://commons.apache.org/proper/commons-csv/> (accessed Jun. 17, 2021).
- [82] “Postman | The Collaboration Platform for API Development.” <https://www.postman.com/> (accessed Jul. 31, 2021).

Anexo A – Protótipos de baixa fidelidade

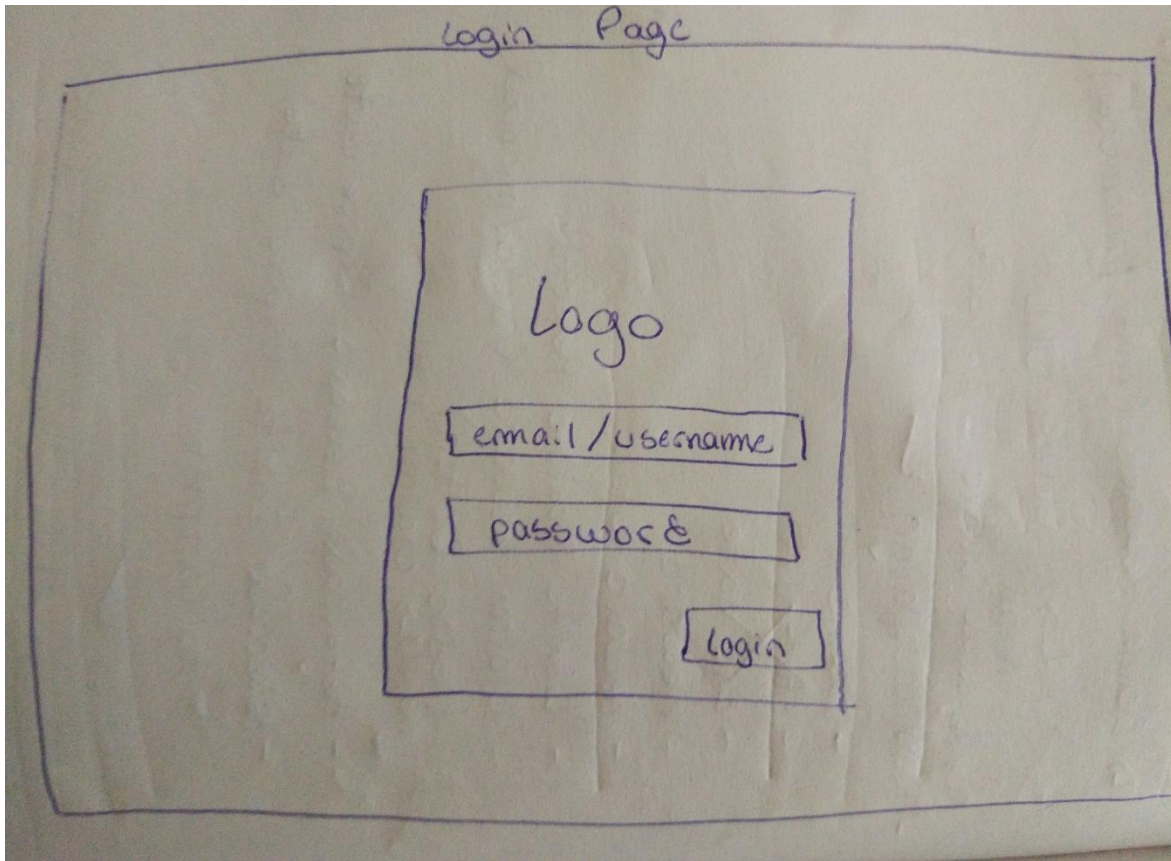
a) Protótipo para ecrã de *refer a friend* e ecrã de reclamação do prémio



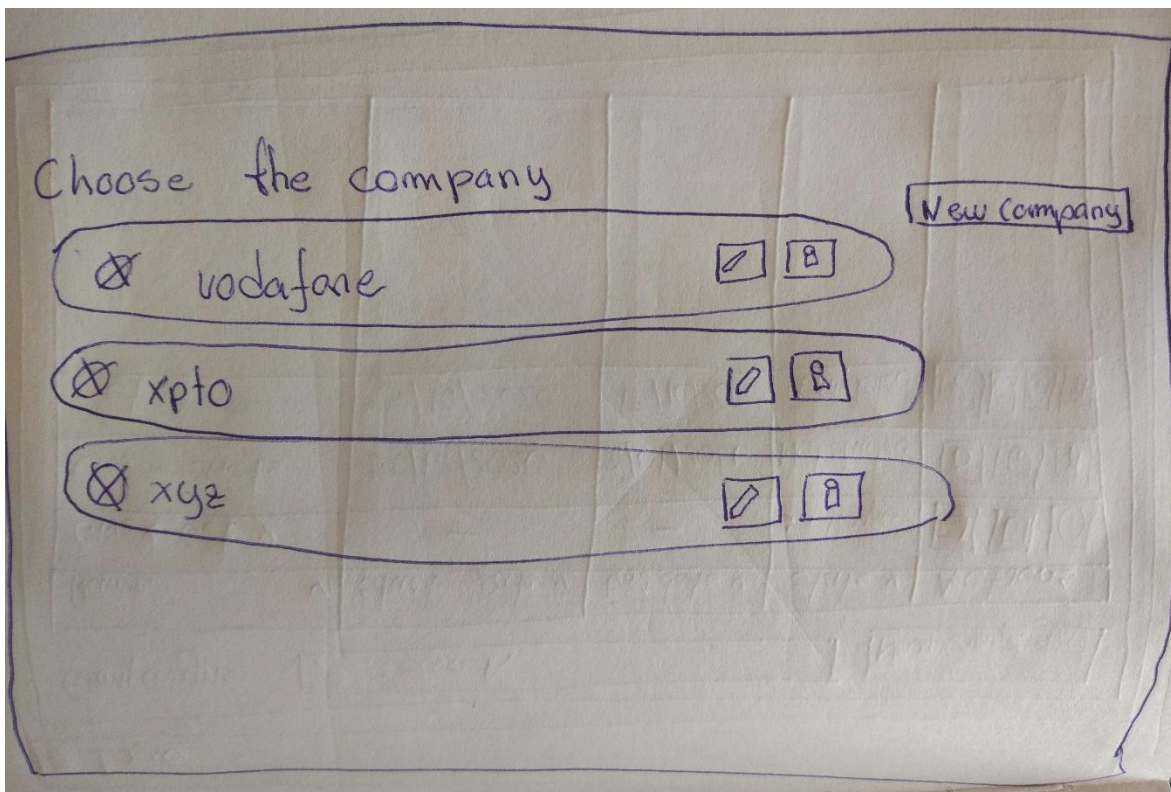
b) Protótipo para o ecrã de preenchimento das informações de envio e o ecrã de partilha do convite



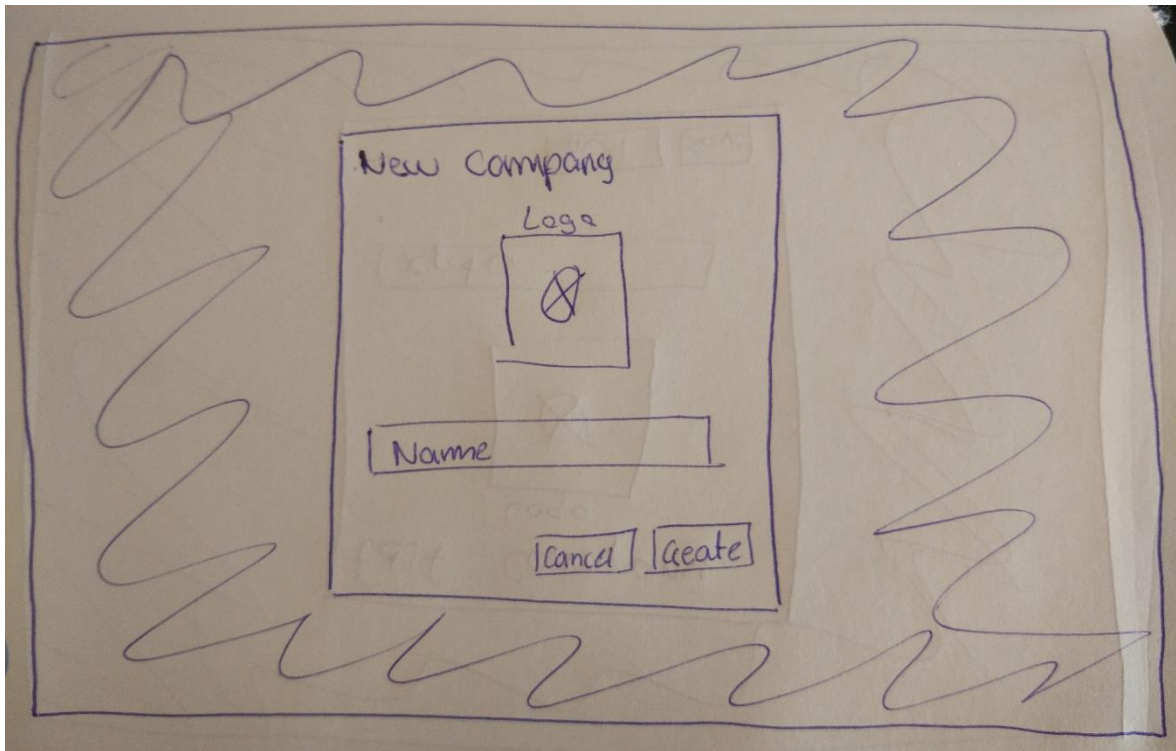
c) Protótipo para a página de login



d) Protótipo para a página das empresas



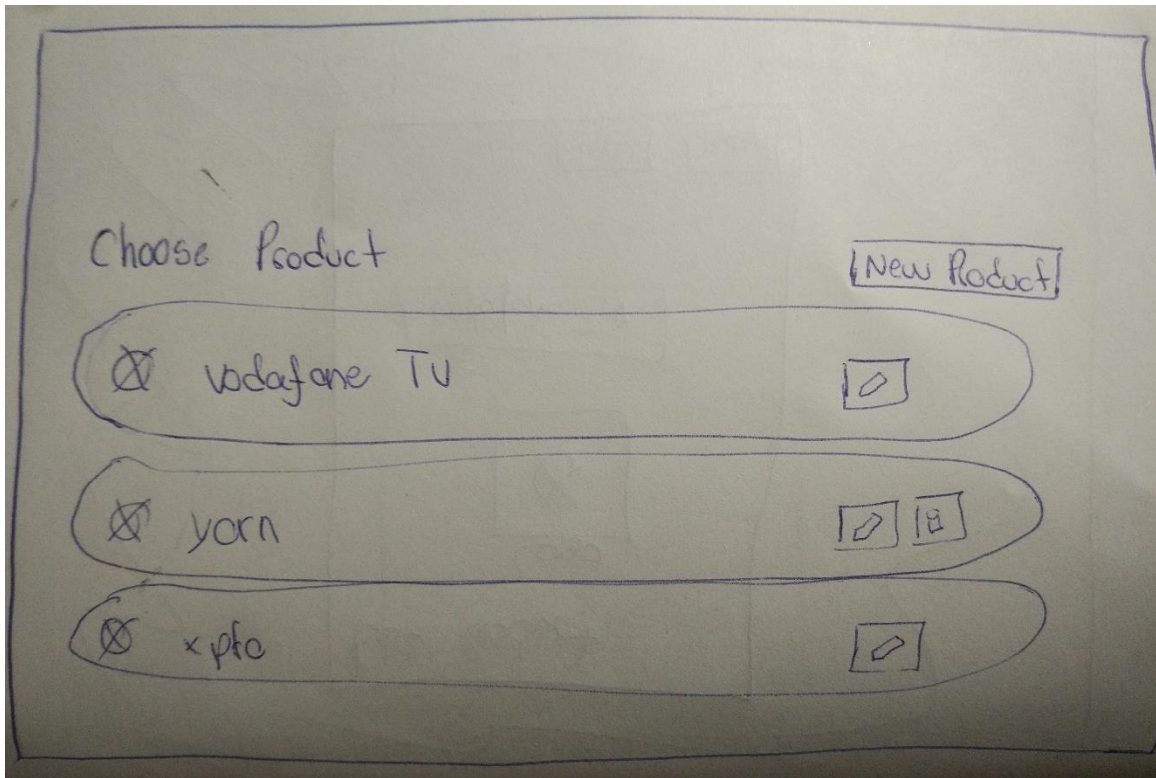
e) Protótipo para a página de criação empresas



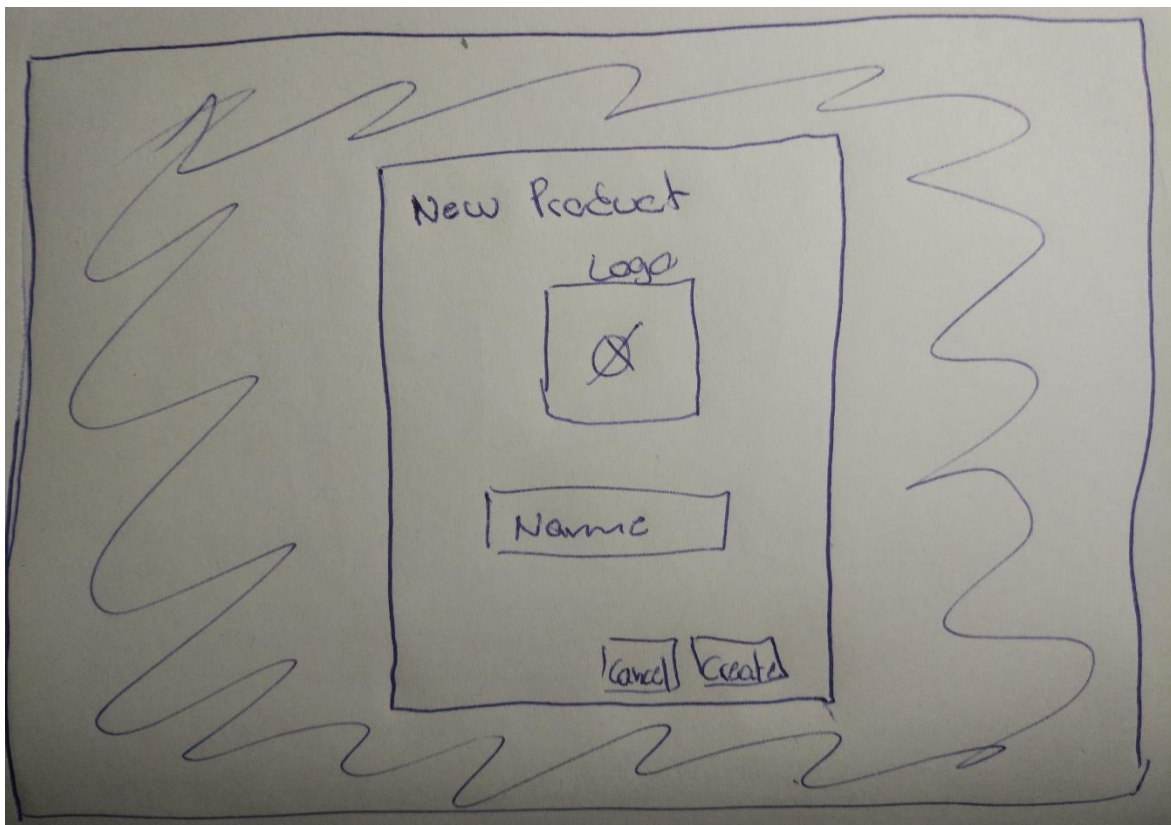
f) Protótipo para a página de edição de empresas



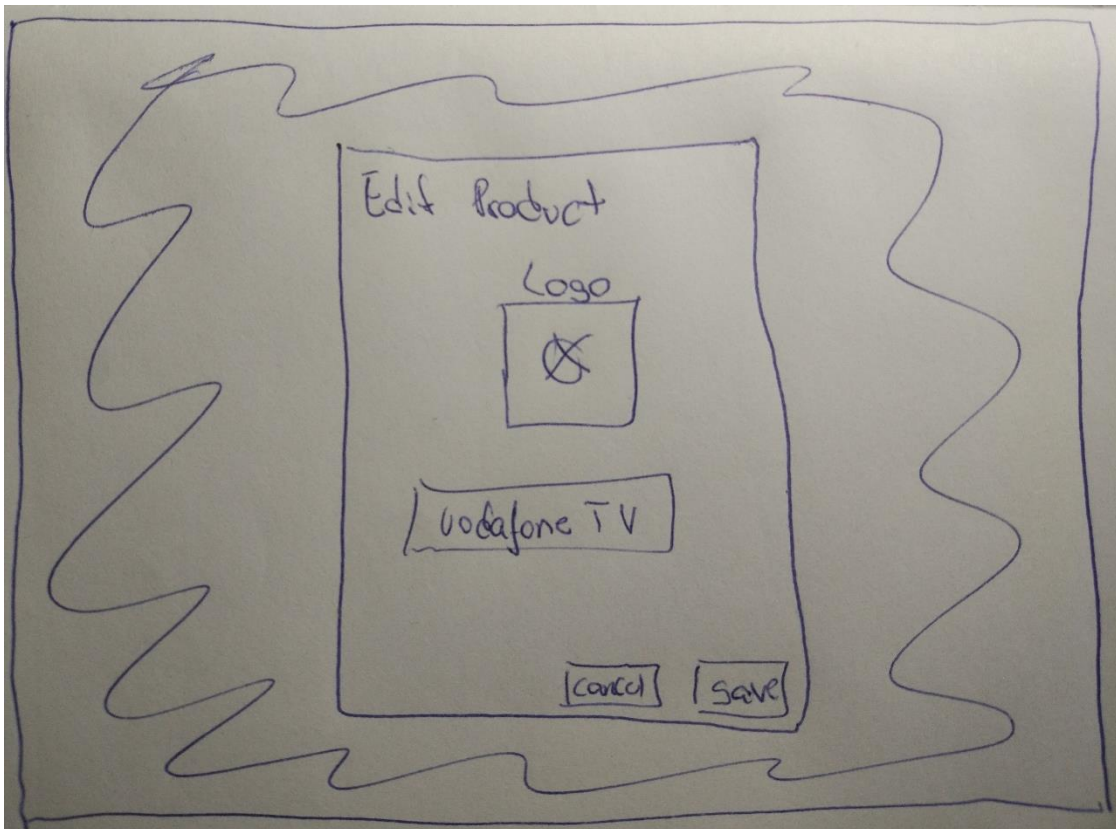
g) Protótipo para a página dos produtos de uma empresa



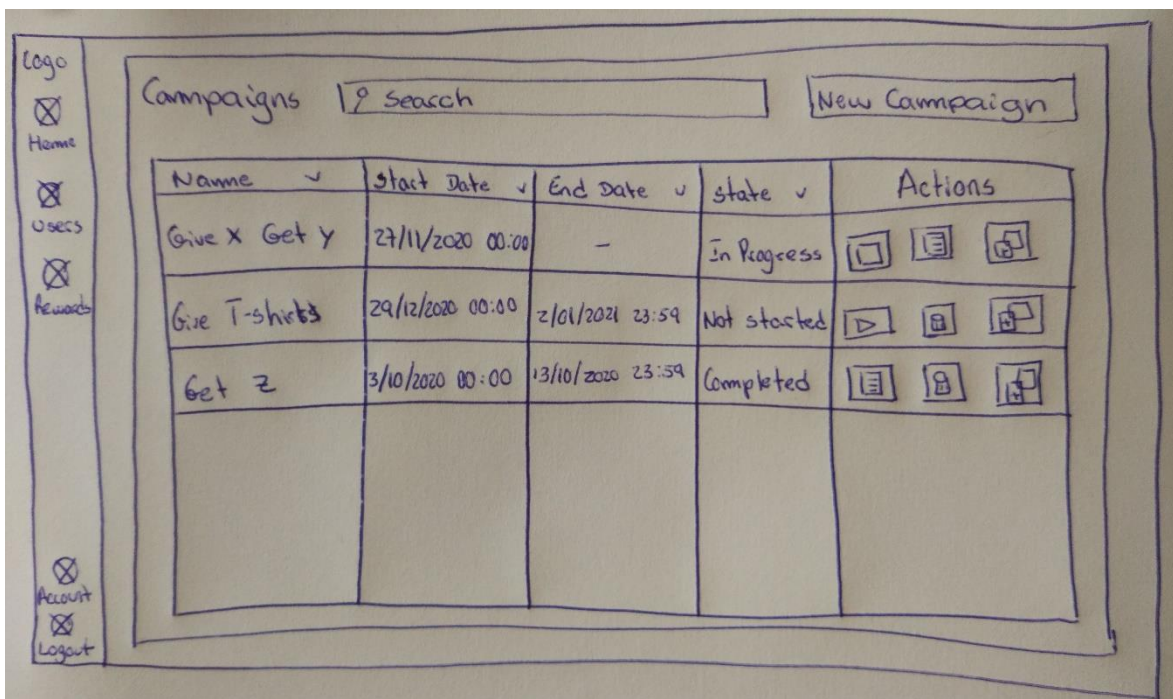
h) Protótipo para a página de criação de um produto



i) Protótipo para a página de edição de um produto



j) Protótipo para a página das campanhas de um produto



k) Protótipo para a página de criação de uma campanha

Logo

← New Reward

Name *

Description *

Time

Start Date*

End Date

Rewards

Referral (sender)

Reward type ⓘ

Reward

Category Label

Claim Reward type

Claim Expiry date (days) (Manual claim type)

Max Invites with Reward

Gift Details

Image

Description

Referred (Receiver)

Reward type ⓘ

Reward

Category Label

Claim Reward type

Claim Expiry date (Manual claim type)

Gift Details

Image

Description

Reward steps ⓘ

Available	Selected
<input type="checkbox"/> Register	<input checked="" type="checkbox"/>
<input type="checkbox"/> Login	<input checked="" type="checkbox"/>
<input type="checkbox"/> Transaction	<input checked="" type="checkbox"/>

1) Protótipo para a página de visualização e edição da informação de uma campanha

≡ Logo

← View info

Give x Get y

Description

Time

start Date 27/11/2020 00:00

End Date 24/12/2020 00:00

Rewards

Referral (sender)

Reward type single

Reward

Cash 10 \$

Claim Reward type Auto

Max invites with Reward 0

Gift Details

Image xpto.png

Description

Preferred (Receiver)

Reward type single

Reward

Cash 10 \$

Claim Reward type Instant

Gift Details

Image xyz.png

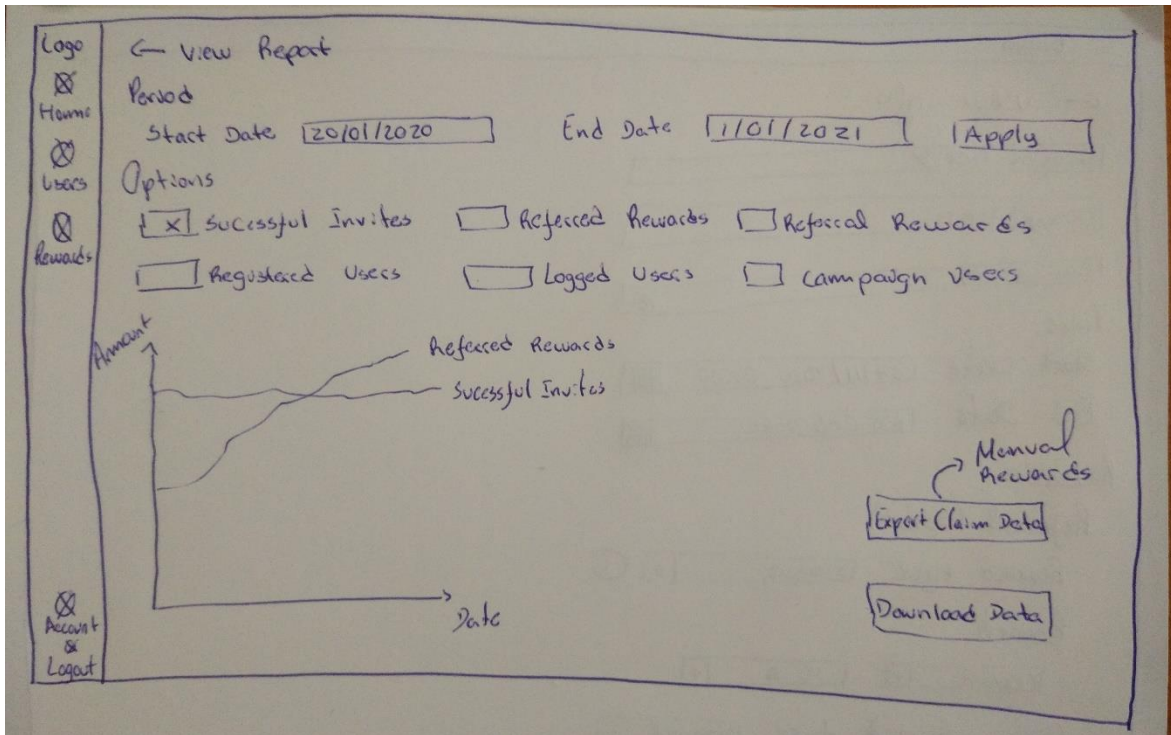
Description

Reward steps

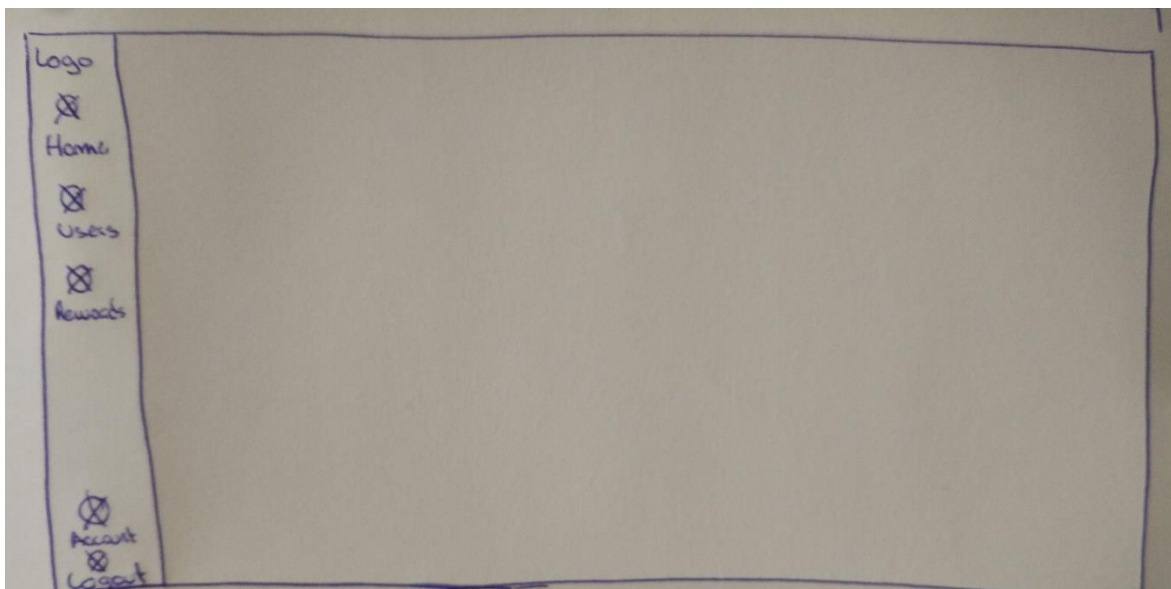
Available	Selected
Login ✓	1 - Register x
Transaction ✓	

Cancel Save

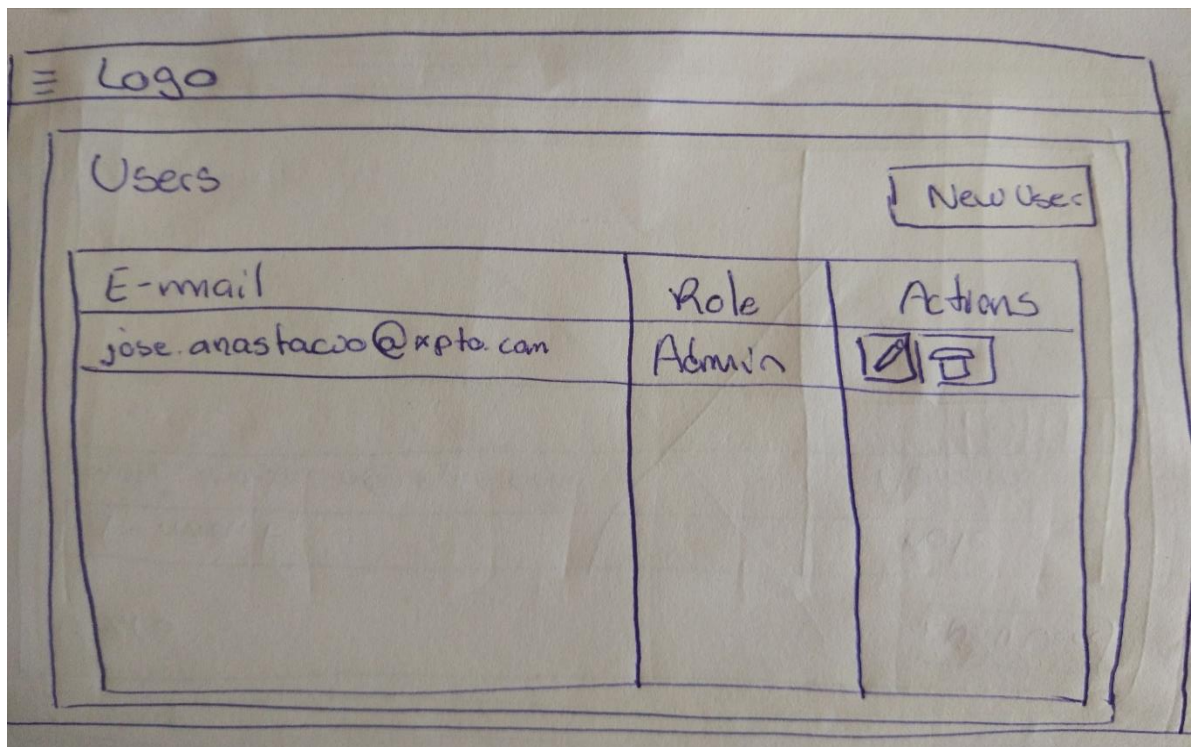
m) Protótipo para a página de estatísticas de uma campanha



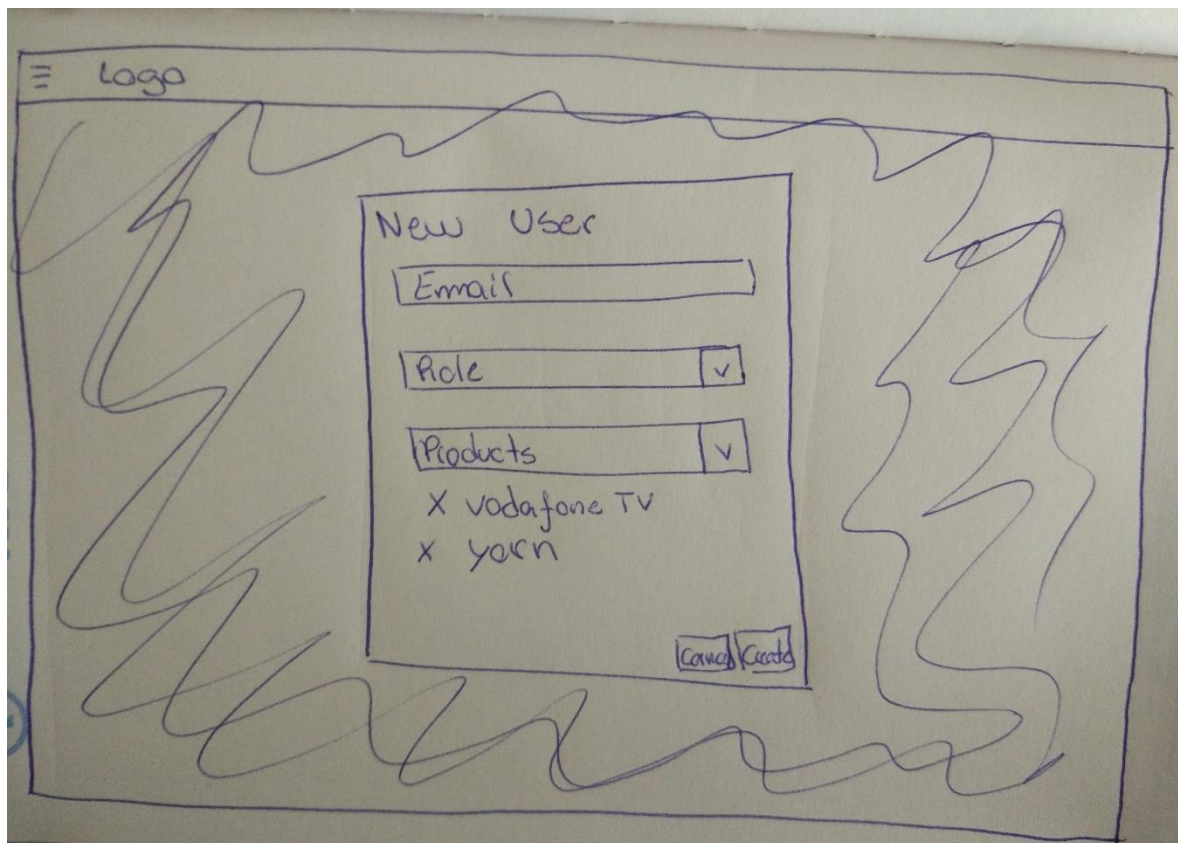
n) Protótipo para o menu de navegação



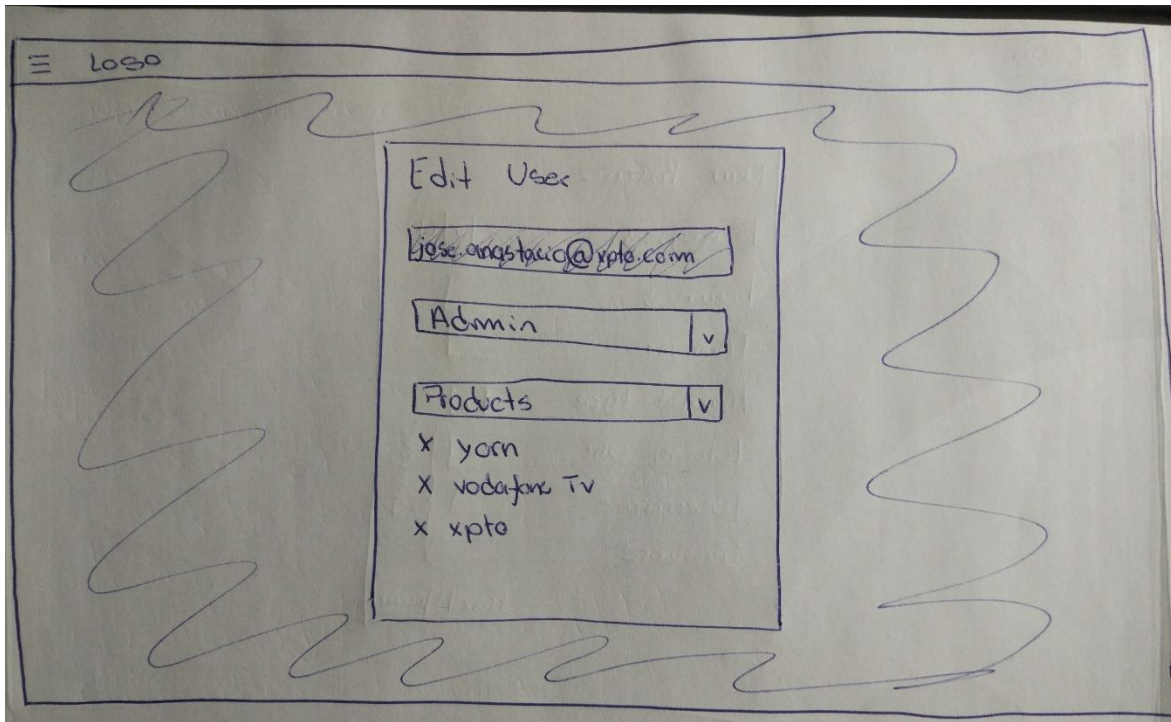
o) Protótipo para a página de gestão de utilizadores



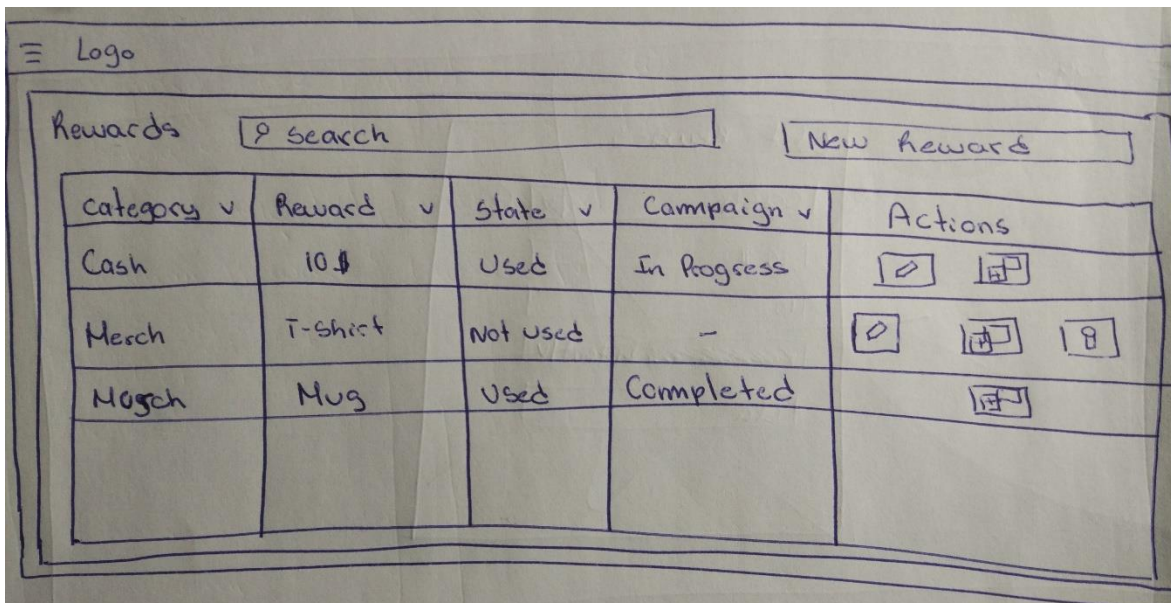
p) Protótipo para a página de criação de um utilizador



q) Protótipo para a página de edição de um utilizador



r) Protótipo para a página de gestão de prémios



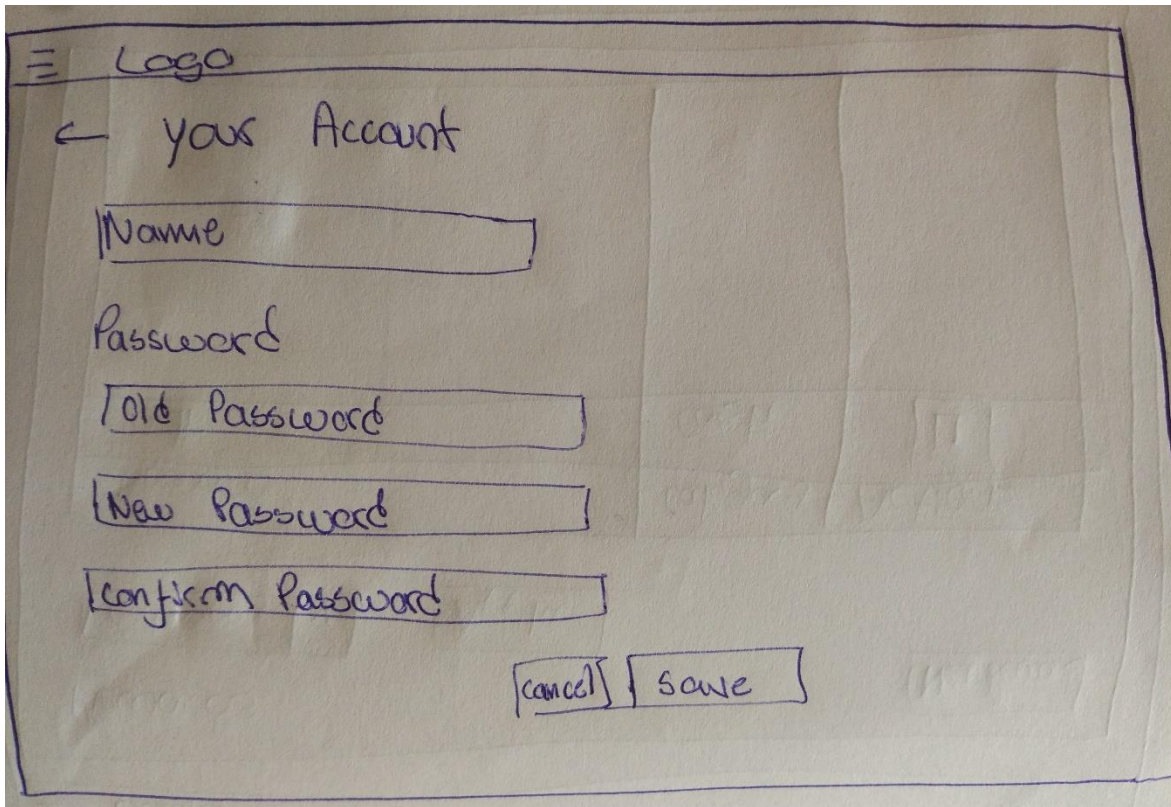
s) Protótipo para a página de criação de um prémio

A hand-drawn prototype for a 'New Reward' creation page. The page has a header with a hamburger menu icon and the text 'Logo'. The main content area contains a form titled 'New Reward' with the following fields: 'Category' (with a dropdown arrow and a plus sign), 'Reward', 'Amount', 'Reward type' (with a dropdown arrow), 'Claim URL', 'Username', and 'Password'. A bracket on the right side of the 'Claim URL', 'Username', and 'Password' fields is labeled 'Instant'. At the bottom of the form are two buttons: 'Cancel' and 'Create'.

t) Protótipo para a página de edição de um prémio

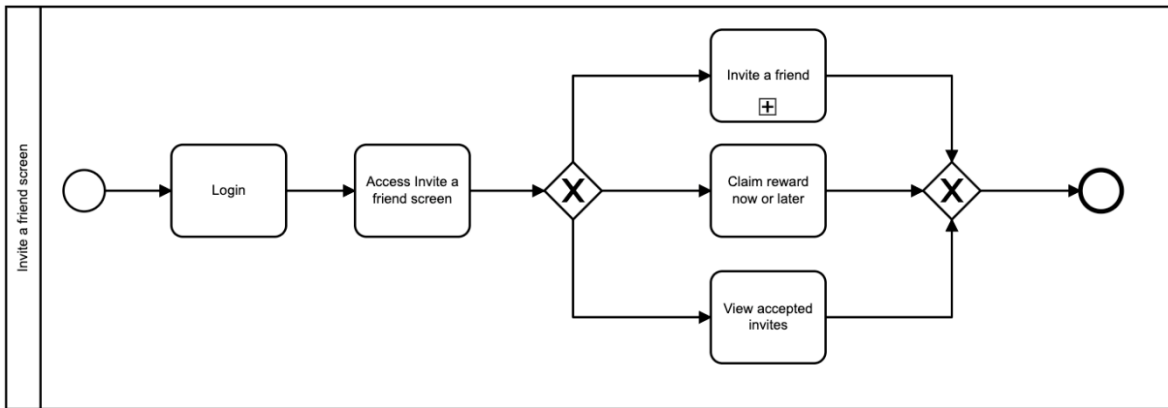
A hand-drawn prototype for an 'Edit Reward' page. The page has a header with a hamburger menu icon and the text 'Logo'. The main content area contains a form titled 'Edit Reward' with the following fields: 'cash' (with a dropdown arrow and a plus sign), '10 \$', '10', 'Instant' (with a dropdown arrow), 'xpto.com.xyz', 'xpto', and '****'. At the bottom of the form are two buttons: 'Cancel' and 'Save'.

u) Protótipo para a página de edição da informação do utilizador

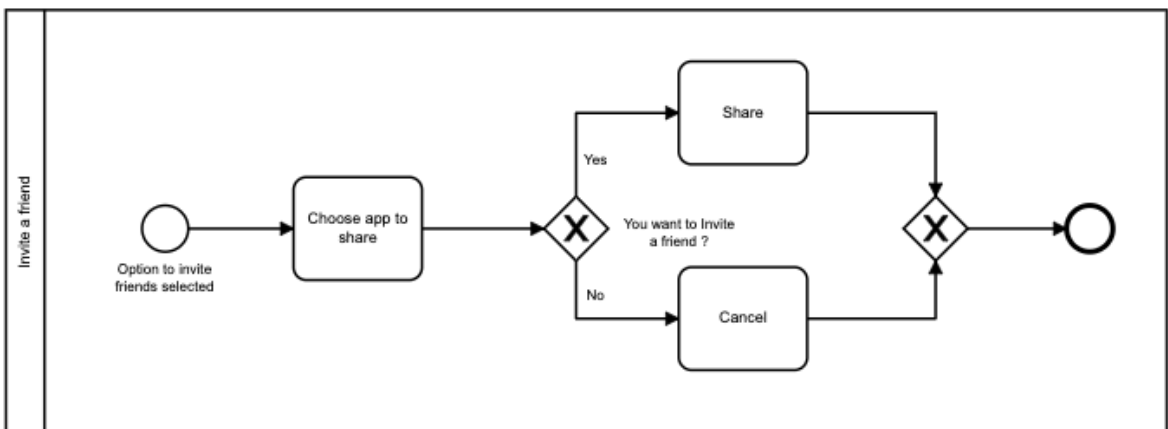


Anexo B – Processos de negócio

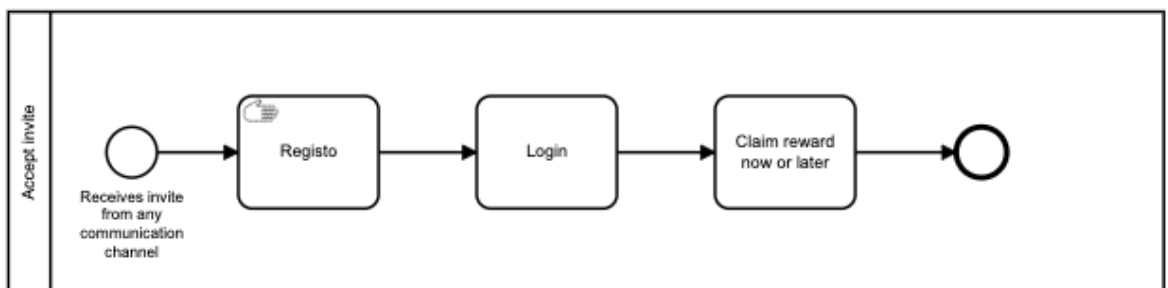
a) Processo de negócio de um remetente



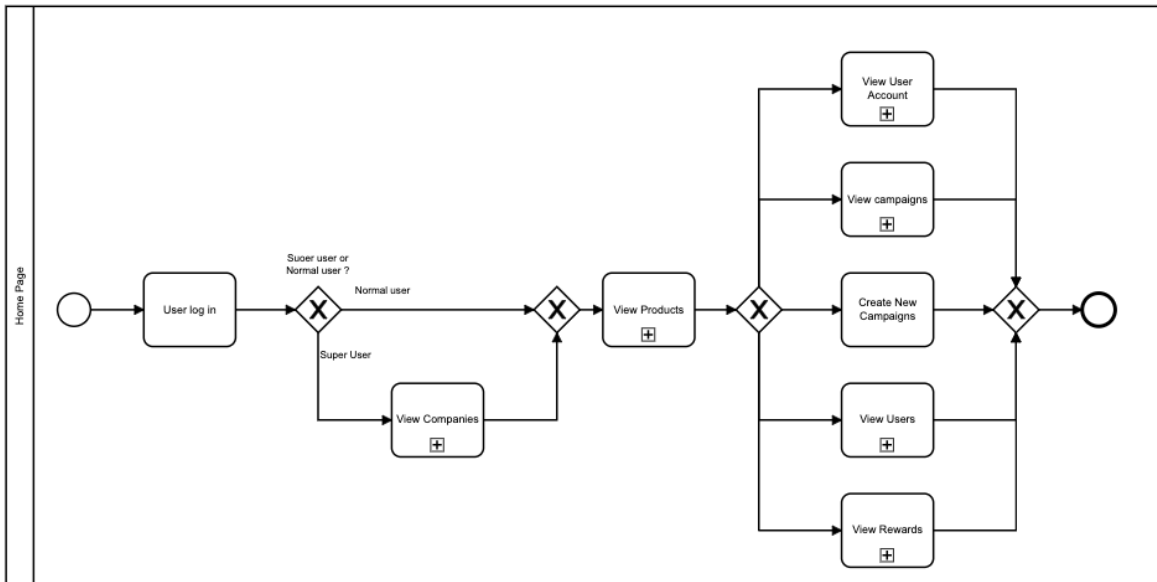
b) Processo de negócio de partilha de um convite



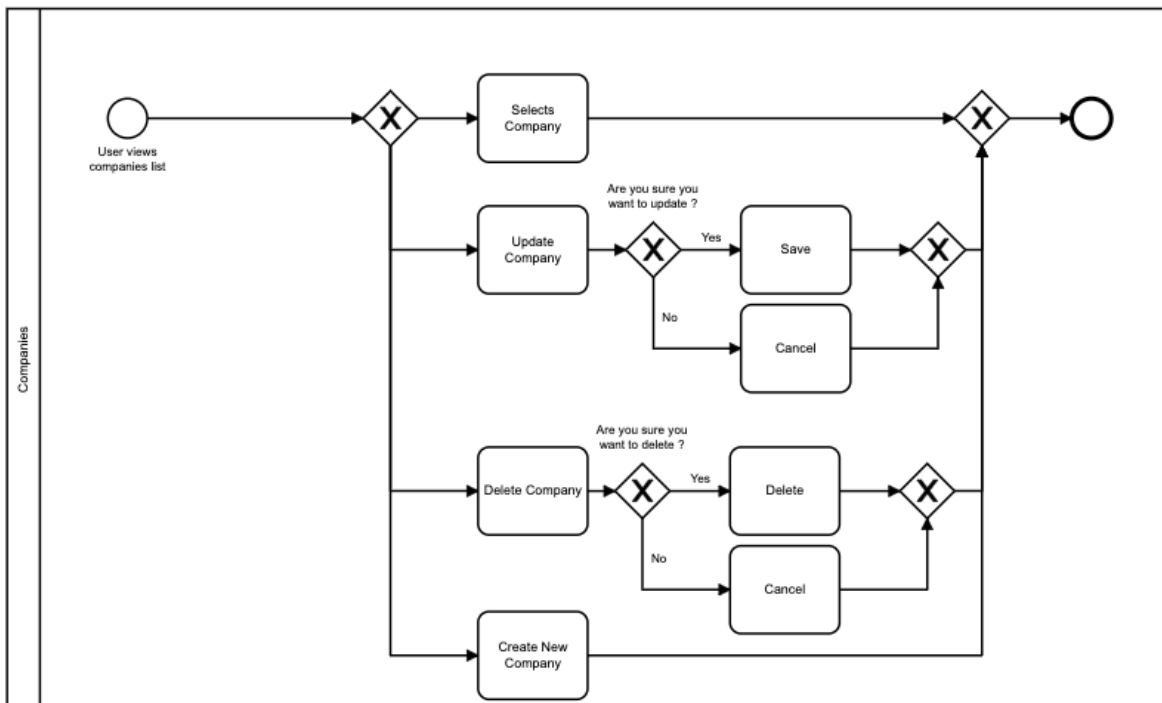
c) Processo de negócio de um recetor



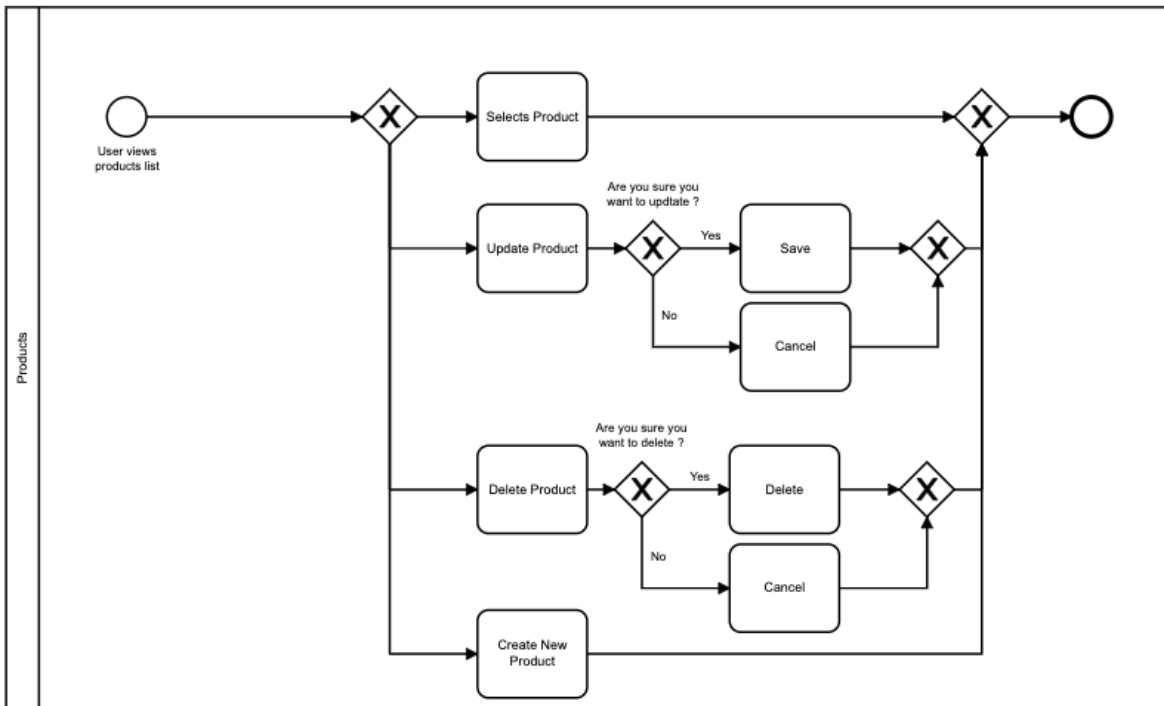
d) Processo de negócio do *backoffice*



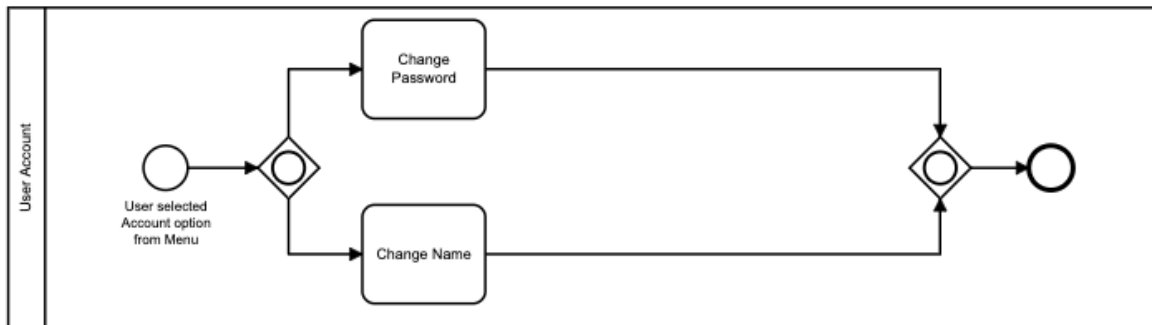
e) Processo de negócio das empresas do *backoffice*



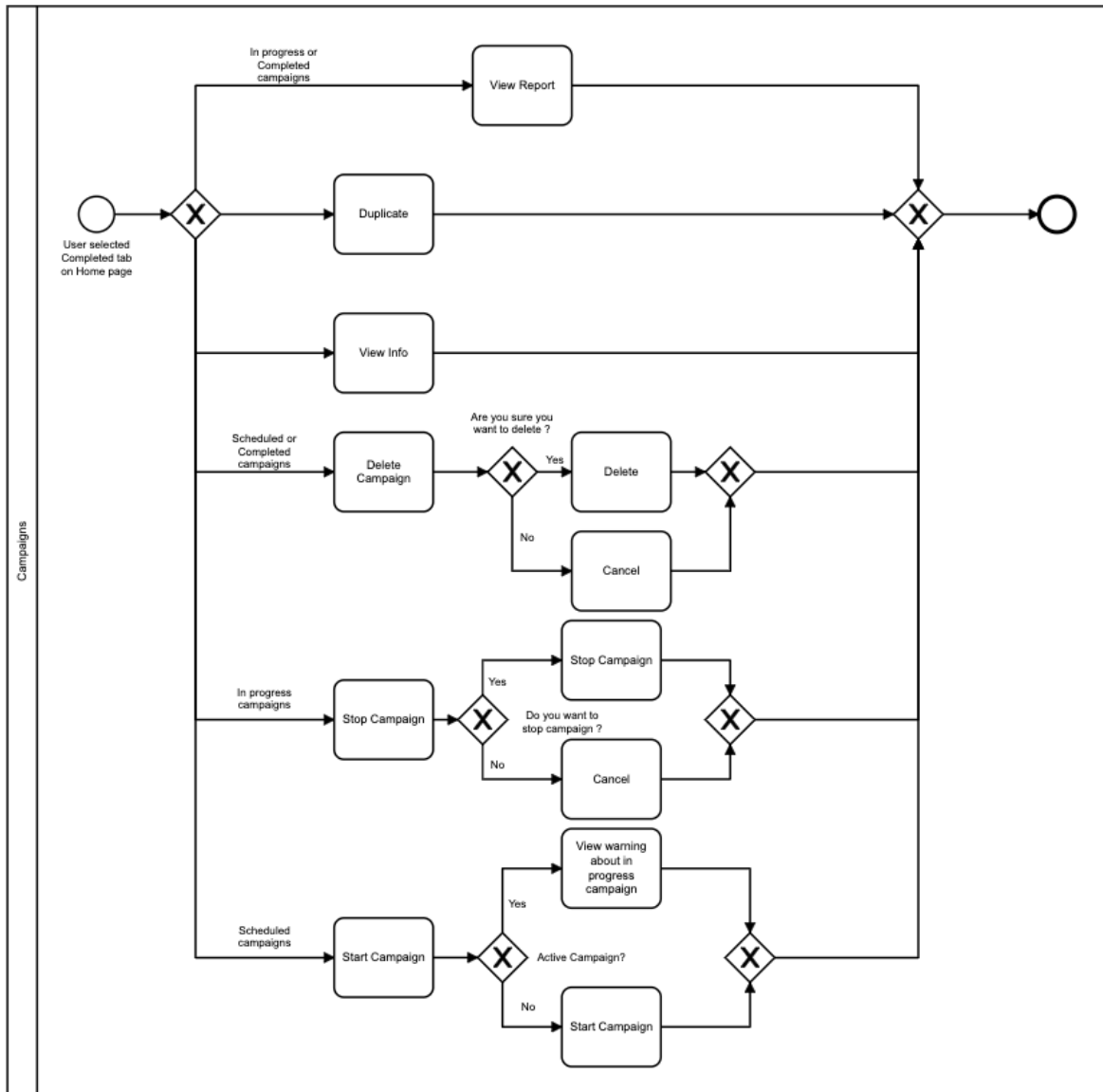
f) Processo de negócio dos produtos do *backoffice*



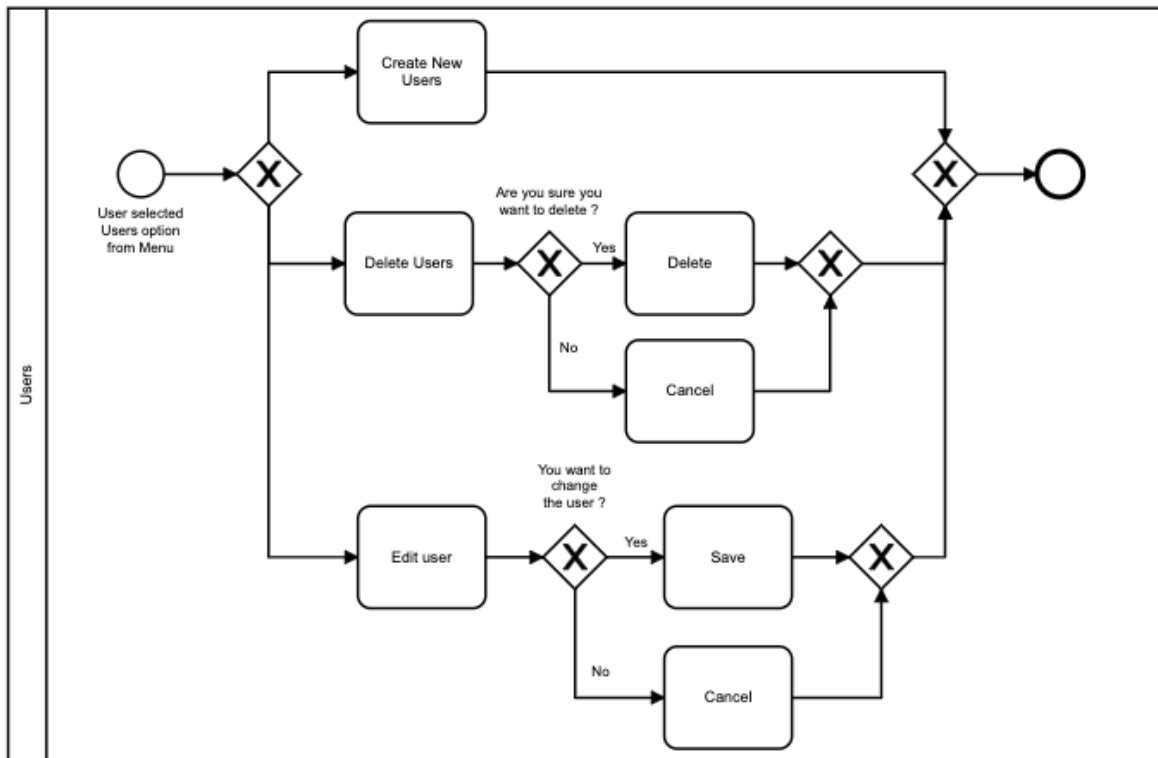
g) Processo de negócio de gestão da conta do utilizador do *backoffice*



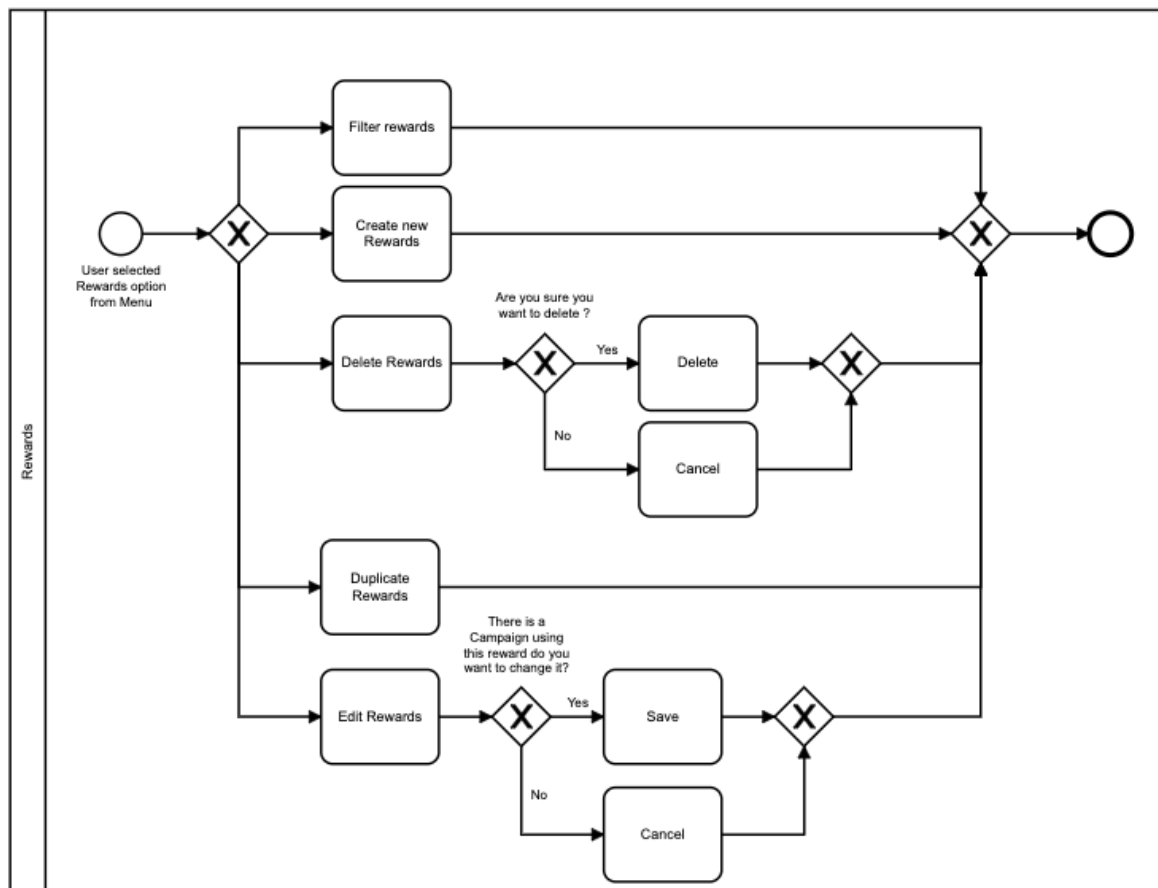
h) Processo de negócio da gestão das campanhas de um produto



i) Processo de negócio da gestão de utilizadores



j) Processo de negócio da gestão de prémios



Anexo C – Plano de Estágio

O plano foi dividido em três partes para uma melhor visualização.

a) Primeira parte do plano de estágio

				Set		Out				Nov					
				39	40	41	42	43	44	45	46	47	48	49	50
Scope	Task	Subtask	Status												
Internship setup	Overview		Completed												
	Understand goals		Completed	x											
	Define highlevel timeline		Completed	x	x										
	Meet the team		Completed	x	x										
Benchmark	Overview		Completed												
	Solutions research		Completed		x										
	Solutions using code		Completed		x	x									
	Solutions using link		Completed			x	x								
	Solutions using code and link		Completed				x	x							
	Solution comparison - pros and cons		Completed							x					
Process mapping	Overview		Completed												
	User flows	Sender flow	Completed												
		Receiver flow	Completed								x				
	Backoffice operator flows	Configure campaigns	Completed									x	x		
		Configure rewards	Completed											x	
		Configure company and users structure	Completed											x	
		Campaign analytics	Completed												x
	Mock-ups	Draft mockup design	Completed							x	x	x	x	x	x
			Completed												

b) Segunda parte do plano de estágio

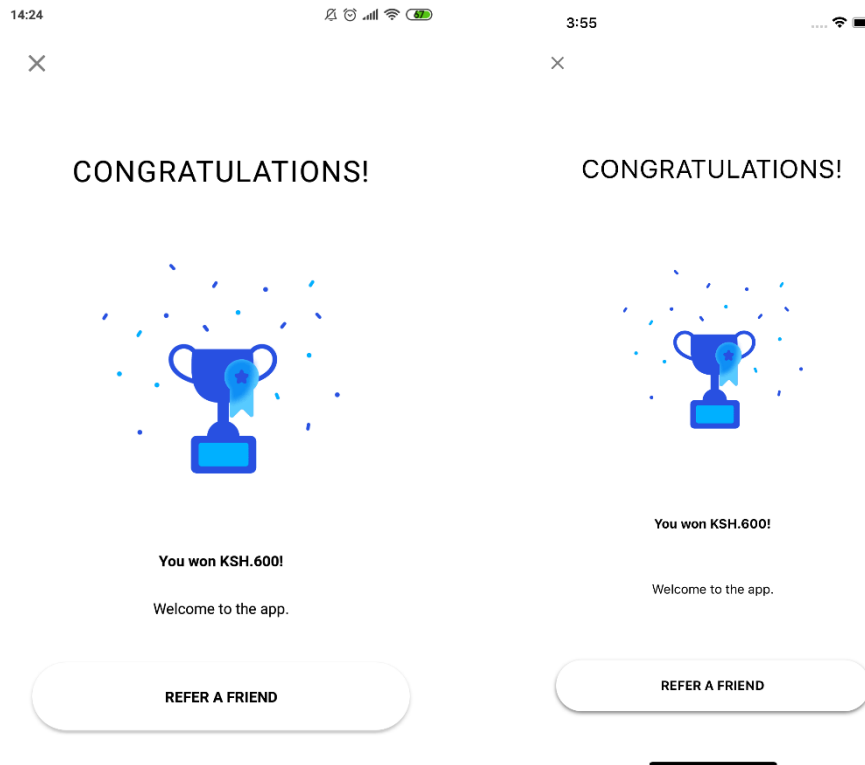
			Weeks	Dez			Jan				Feb				Mar				
Scope	Task	Subtask	Status	51	52	53	1	2	3	4	5	6	7	8	9	10	11	12	13
UI App development (Android)	Overview		Completed	[Orange bar]															
	Understand Consumer project structure		Completed	x															
	UI development		Completed		x	x	x												
	All campaign types		Completed																x
Understanding of Backend	Prepare UI demo video		Completed				x	x											
	Overview		Completed																
	Project structure	Microservices architecture	Completed							x	x								
		Springboot framework	Completed								x								
Backend/App (Android)		Initial database schema	Completed									x							
	Overview		Completed																
	User tracking	Deep links (App path)	Completed										x						
		Send invite with unique link	Completed										x	x					
		New invite accepted indicator on	Completed											x					
		New invite accepted indicator off	Completed												x				
		Send accepted invite push notification - sender	Completed													x			
		Send accepted invite push notification (limit reached) - sender	Completed														x		
		Send conversion step (login) completed push notification - receiver	Completed														x		
		View campaign info - referral area - expiry date + limit	Completed												x				
		View accepted invites - referral area - tracking	Completed											x					
		Firebase Dynamic links (Play store path)	Completed														x		
		Receiver screen - title, description, picture	Completed																x
		View campaign info - referral area - title, description and picture	Completed																x

c) Terceira parte do plano de estágio

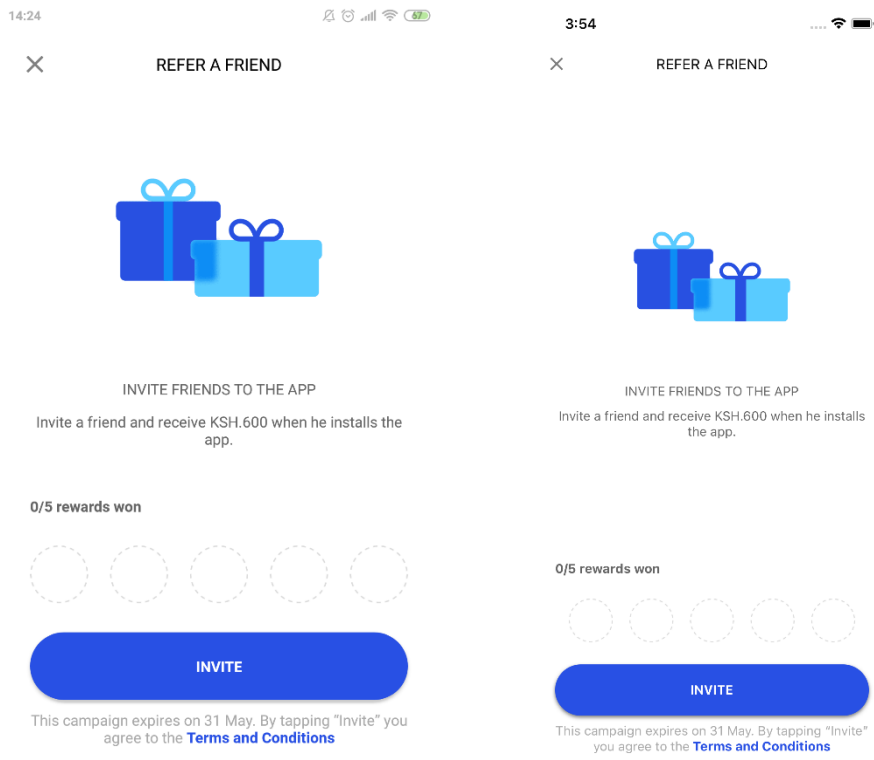
			Weeks	Mar			Apr					May				June		
Scope	Task	Subtask	Status	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Understanding of Backoffice	Overview		Completed															
	Project structure	Backoffice structure	Completed	x	x													
		Backoffice framework	Completed	x	x													
Backoffice/Backend	Overview		Completed															
	Campaigns Implementation	Create campaign	Completed	x														
		List campaigns	Completed		x													
		View campaigns	Completed		x													
		Edit campaigns	Completed		x													
		Delete campaigns	Completed		x													
		Duplicate campaigns	Completed		x													
		Change status of campaigns	Completed				x											
	Campaign analytics	View campaign reports - analytics	Completed						x	x	x							
Intermediate demo preparatio	Overview		Completed				x	x										
Improvements	Overview		Completed															
	Update BO improvements		Completed									x						
	Align BE to BO improvements		Completed									x						
	App improvements - following feedback		Completed									x	x	x				
App UI em react native	Overview		Completed															
Code review	Overview		Completed															
Deliverables preparation	Overview		Completed															
	Final deck preparation		Completed														x	x
	Demo preparation		Completed														x	x
	Report writing		Completed														x	x

Anexo D – UI em React Native

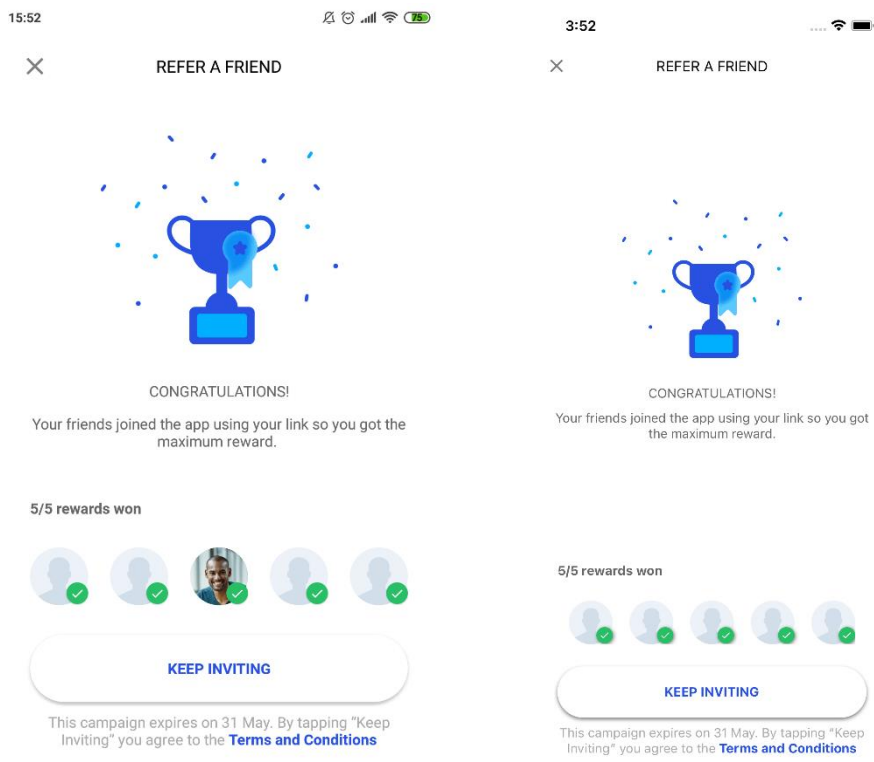
a) UI para o ecrã do recetor em Android e iOS



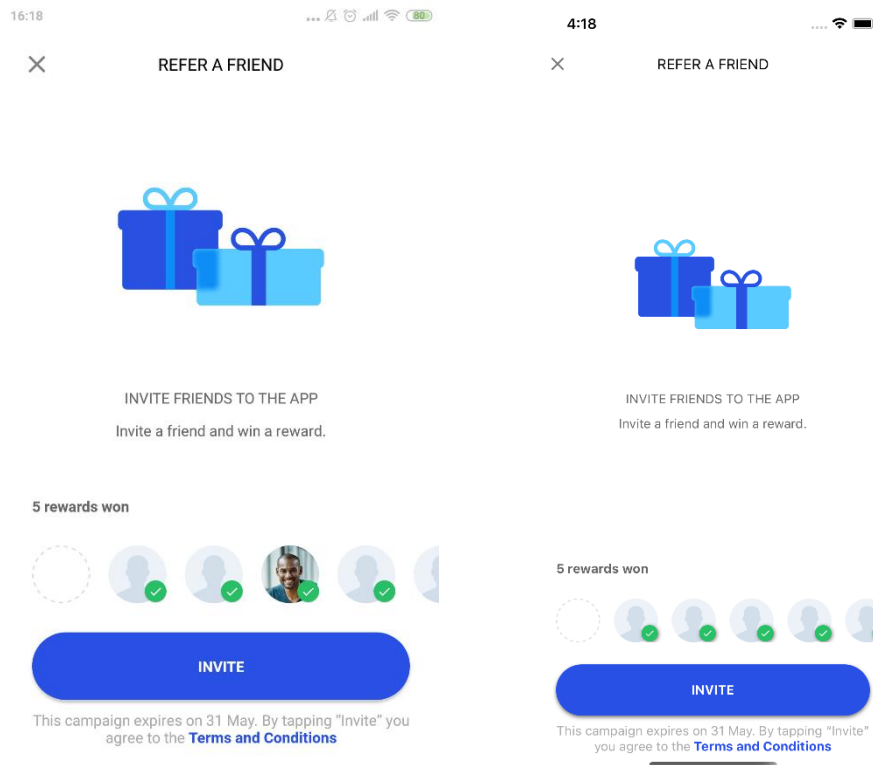
b) UI para o ecrã do remetente com nenhum convite aceite em Android e iOS



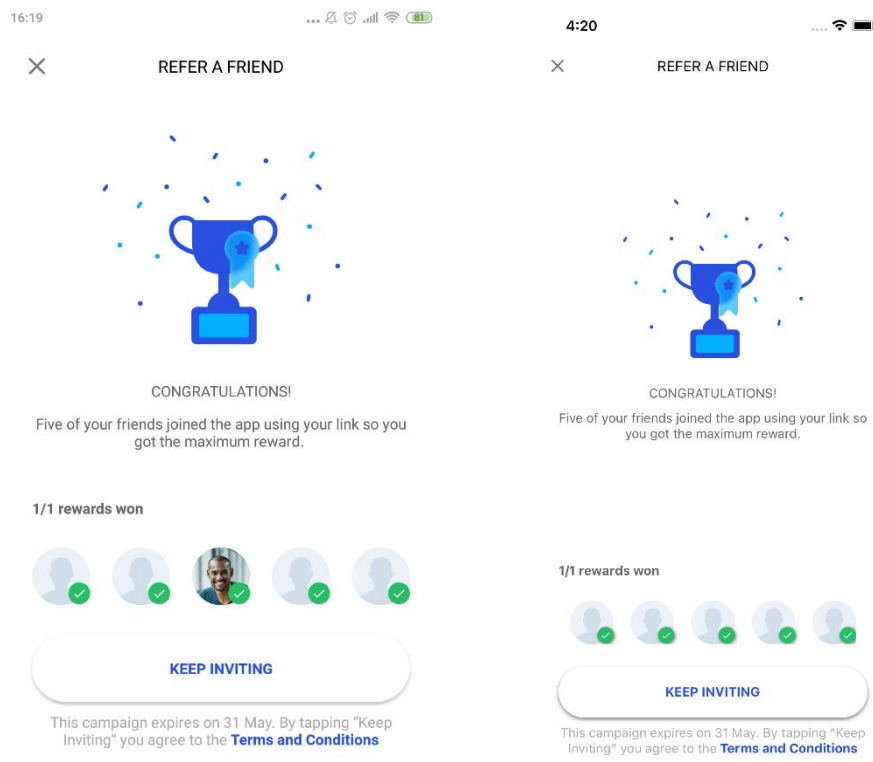
c) UI para o ecrã do remetente para uma campanha com limite de cinco convites



d) UI para o ecrã do remetente para uma campanha sem limite em Android e iOS



a) UI para o ecrã do remetente para uma campanha agregada em Android e iOS



a) UI para o ecrã do remetente para uma campanha sem limite e sem prémio em Android e iOS

