



ISEL – INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
ADEETC – ÁREA DEPARTAMENTAL DE ENGENHARIA DE
ELECTRÓNICA E TELECOMUNICAÇÕES E DE COMPUTADORES

LEIM

LICENCIATURA EM ENGENHARIA INFORMÁTICA E MULTIMÉDIA
UNIDADE CURRICULAR DE PROJETO

Supermercados Inteligentes



Sérgio Oliveira (43744)
Marcelo Marques (43798)

Orientador(es)

Professora [ISEL]

Paula Graça

[DoiTLean]
[DoiTLean]

Inês Contente Jorge
Duarte Ferreira

Julho, 2020

Resumo

No contexto tecnológico atual da *IoT*, aplicado ao sector retalhista de produtos alimentares, a experiência do consumidor comum pode ser melhorada durante uma sessão de compras num supermercado. A utilização de *Beacons* estrategicamente colocados nas várias zonas de produtos, permite providenciar conteúdos contextualizados e ofertas de promoções de acordo com a localização e preferências dos utilizadores, tornando as suas compras mais eficazes. Os *Beacons* são dispositivos *wireless* de pequeno alcance, munidos de uma bateria, que emitem sinais *Bluetooth*, que por sua vez são captados por *smartphones* quando se encontram dentro do seu raio de ação.

O desenvolvimento do projeto Supermercados Inteligentes, teve como principal motivação, a implementação de uma aplicação móvel que facilite a ida ao supermercado, permitindo comprar em menos tempo e com menos custo. Os principais objetivos da aplicação são, a partir da lista de compras criada pelo utilizador, permitir que este seja alertado sempre que passe perto das zonas de produtos respetivas, bem como das promoções oferecidas. O projeto inclui também uma aplicação *web reactive* de *back-office*, para suportar a configuração dos *Beacons*, que inclui a sua localização e promoções associadas.

O projeto foi inteiramente realizado através da plataforma *Outsystems* de *low-code*, utilizando uma metodologia de desenvolvimento ágil adequada, o *Scrum*. Assim, o projeto foi dividido em *Sprints*, incluindo cada um deles um subconjunto de requisitos, que uma vez detalhados e implementados, resultaram na produção de um incremento do produto com um objetivo bem definido que podia ser executado e testado. Ao fim do último *Sprint*, o incremento de produto resultou na aplicação final. Foi também utilizada uma ferramenta de simulação, o *Anylogic*, que permitiu construir alguns cenários para ajudar a compreender o possível comportamento dos consumidores em ambiente de compras utilizando a aplicação móvel.

Consideramos então que a aplicação é bastante útil e fácil de utilizar, mesmo em contexto real. Contudo, teriam de se eliminar ou melhorar alguns fatores de erro, tal como os problemas de sinal que podem haver de telemóvel

para telemóvel, sendo este uma barreira enorme para a implementação de novas funcionalidades. Poderia-se também melhorar a aplicação fornecendo pacotes de promoções personalizados ao utilizador ou até mesmo uma maneira do utilizador saber quantas pessoas estão em cada secção.

Agradecimentos

Ambos: Com a finalização desta fase da nossa vida, não poderíamos deixar de agradecer a várias pessoas que, direta ou indiretamente, nos tenham ajudado e partilhado esta aventura ao longo destes quatro anos.

Gostaríamos de agradecer ao ISEL e a todo o corpo docente pelo esforço realizado para incluir e tratar todos os alunos da mesma maneira, principalmente nesta altura complicada em que tivemos de nos adaptar.

Obrigado aos orientadores do nosso projeto, professora Paula Graça, à Inês Contente Jorge e ao Duarte Ferreira, por toda a ajuda que nos foi dada, principalmente no ínicio quando ainda estávamos a aprender os básicos da plataforma Outsystems.

Gostaríamos também de agradecer a todos os nossos colegas e amigos feitos no ISEL, tenham ajudado ou não nos testes do projeto, visto que são as pessoas com quem nos damos no dia a dia, moldando a nossa personalidade e aprendendo sempre com cada um deles. Para não criar problemas e arriscar faltar alguém, vamos apenas mencionar o grupo que criámos no primeiro dia de aulas no refeitório do ISEL, obrigado à Inês Batista, obrigado à Nádia Nunes, obrigado ao Ruben Silva, obrigado ao Pedro Afonso Gomes e obrigado ao Tiago Santos Fernandes. Mais uma vez, obrigado a todos os outros, ninguém é esquecido.

Ambos participámos na Praxe e daí fizemos também muitas e boas amizades, gostaríamos de agradecer a todos pela experiência, sabedoria e trabalho de equipa proporcionados, um obrigado principalmente à nossa família de praxe.

Sérgio: Um obrigado especial à minha namorada Inês Silva por me atuar durante o dia a dia, especialmente durante a realização deste projeto, o facto de ter testado bastante ambas as aplicações, ajudou-nos a realizar um melhor trabalho. Por fim e não menos importante, um muito obrigado à minha família, principalmente aos meus pais por todas as ideias em relação à realização deste projeto e por todas as conversas e curiosidades sobre o curso ao longo dos anos.

Marcelo: Um obrigado a todas as pessoas ao meu redor que me ajudaram,

tanto amigos como família, e que deram a sua opinião sobre o projeto e nos permitiram realizar um projeto ainda melhor e também me ajudaram a nível pessoal ao longo deste trajeto.

Índice

Resumo	i
Agradecimentos	iii
Índice	v
1 Introdução	1
2 Trabalho Relacionado	3
3 Modelo Proposto	5
3.1 Requisitos	5
3.2 Fundamentos	7
3.2.1 <i>Beacons</i>	7
3.2.2 Plataformas <i>low-code</i>	8
3.2.3 <i>Anylogic</i>	8
3.3 Abordagem	9
4 Implementação do Modelo	11
4.0.1 Mapa do supermercado	11
4.0.2 Modelo de dados	11
4.0.3 Arquitetura Geral	12
4.0.4 Modelo de arquitetura	14
4.0.5 Implementação	15
5 Validação e Testes	29
5.0.1 Testes <i>back-office</i>	29
5.0.2 Testes aplicação <i>mobile</i>	35

5.0.3	Testes <i>Beacons</i>	41
5.0.4	Testes <i>Anylogic</i>	46
6	Conclusões e Trabalho Futuro	49
A	Um Detalhe Adicional	51
	Bibliografia	61

Capítulo 1

Introdução

Imagine-se a criar uma lista de compras através do seu *smartphone* e, no momento em que entra no supermercado, ter a noção de quais os produtos da sua lista estão perto de si? Já pensou no tempo que poderia poupar se usasse aplicação no momento de efetuar as suas compras? É essa a motivação deste projeto, melhorar a experiência de compra do consumidor e permitir que este poupe tempo e dinheiro cada vez que vai às compras.

Esta aplicação permite adicionar produtos a uma lista de compras para posteriormente realizar as suas compras. No supermercado, o consumidor terá sempre a noção de quais os produtos da sua lista que estão por perto através do contributo de dispositivos *wireless* designados de *beacons* associados a cada uma das zonas do supermercado.

Quando o *smartphone* do consumidor detetar o *Beacon* de uma zona onde tenha algum produto que se encontra na sua lista de compras, este vai receber um alerta a notificá-lo de que o produto X se encontra por perto.

Está-se perante uma aplicação que ultrapassa o simples conceito de criar uma lista de compras. O projeto conta com uma rede de dispositivos *wireless* espalhada pelo supermercado e que permite ter-se uma aplicação mais completa e com mais funcionalidades do que apenas uma simples lista de compras.

A maioria do projeto foi desenvolvido em *Outsystems* [out,]. A *Outsystems* é uma plataforma *low-code* que permite criar visualmente uma aplicação, usar ferramentas já criadas e também permite produzir o próprio código quando necessário. No entanto este projeto contou também com uma parte complementar em *Anylogic* [any, b] que permite simular o movimento de pes-

soas dentro de um supermercado e criar um *density map* ao longo do tempo.

Capítulo 2

Trabalho Relacionado

Para a realização deste projeto foi feita uma pesquisa com o intuito de se descobrir o enquadramento do projeto e para se ter a noção de trabalhos realizados no mesmo contexto. Após uma breve pesquisa foi possível constatar-se que existem duas áreas principais no que diz respeito a aplicações deste tipo, as aplicações de listas de compras e as aplicações que usam *Beacons*. Relativamente às aplicações de lista de compras analisaram-se duas aplicações, a *Listonic* [lis,], aplicação que sugere produtos comprados frequentemente e que fornece ao consumidor o valor total da lista de compras, e a *Bring!* [bri,] que é usada em tempo real e em modo *offline*. No que diz respeito a aplicações que usam *Beacons* foram analisadas também duas aplicações, a *Proximitask* [pro,], aplicação que envia lembretes dependendo da localização do utilizador, e a *At The BallPark* [mlb,] que usa *Beacons* em estádios para saber quando o utilizador está por perto.

Após serem estudadas as duas áreas principais foi feita uma pesquisa em que o objetivo foi encontrar aplicações que conjugassem as duas áreas. Neste contexto surgiu uma aplicação chamada *K-supermarket's* [ksu,] que usa 55 *Beacons* nas suas instalações e que permite ao utilizador criar uma lista de compras digital e obter promoções e mensagens dependendo dos seus hábitos. Ainda no contexto das aplicações foi feito um estudo que demonstra que aplicações mobile associadas a supermercados aumentam a adesão do consumidor ao mesmo. Como foi referido no *white paper* do grupo ”*La Comer*”[lac,]:

”Atualmente, 16% dos pedidos que são gerados são feitos através do APP”

Isto mostra uma boa resposta por parte dos clientes a aplicações mobile e que tenderá a aumentar ao longo do tempo.

Para além de testes realizados a aplicações *mobile*, há também testes relativos a *Beacons*. Neste caso foi analisado o *Proximity Directory's Q1 2017 Report* que mostra que, em 2017, os *Beacons* eram a tecnologia de proximidade mais popular da indústria (Figura 2.1) e que até em pequenos negócios

"podem aumentar o lucro em 8% com um retorno de investimento de 365%"

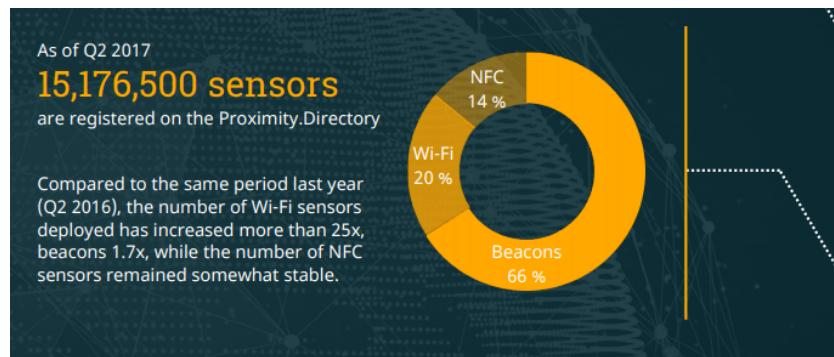


Figura 2.1: Gráfico acerca dos dispositivos wireless

Relativamente ao *Anylogic* foram feitos apenas alguns tutoriais para se conhecer um pouco da biblioteca *Pedestrian* [any, a] e do *Anylogic* em si.

Capítulo 3

Modelo Proposto

3.1 Requisitos

Este projeto inclui requisitos funcionais e requisitos não funcionais. No que diz respeito aos requisitos funcionais, estes são relativos aplicação *backoffice* e aplicação *mobile*. Possui também os requisitos não funcionais.

Relativamente aos requisitos funcionais da aplicação *backoffice* tem-se:

- registar e editar produtos,
- registar e editar promoções,
- registar e editar *beacons*,
- associar *beacons* às respetivas localizações.

No que diz respeito aos requisitos funcionais da aplicação *mobile* tem-se:

- realizar a autenticação,
- enviar mensagem de boas-vindas,
- enviar mensagem de proximidade de produtos,
- visualizar produtos,
- visualizar e gerir a lista de compras.

Como requisitos não funcionais do projeto tem-se:

- aplicação desenvolvida em *Outsystems* e todos os recursos que dela provêm,
- *back-office* desenvolvido através de *web reactive*,
- utilização de *beacons* (70 metros de alcance),
- a aplicação deverá possuir um tempo de resposta/latência baixos, caso contrário, esta vai contra o seu principal propósito, acelerar e melhorar o tempo passado num supermercado.

Na figura 3.1 é possível ver os casos de utilização relativos ao projeto.

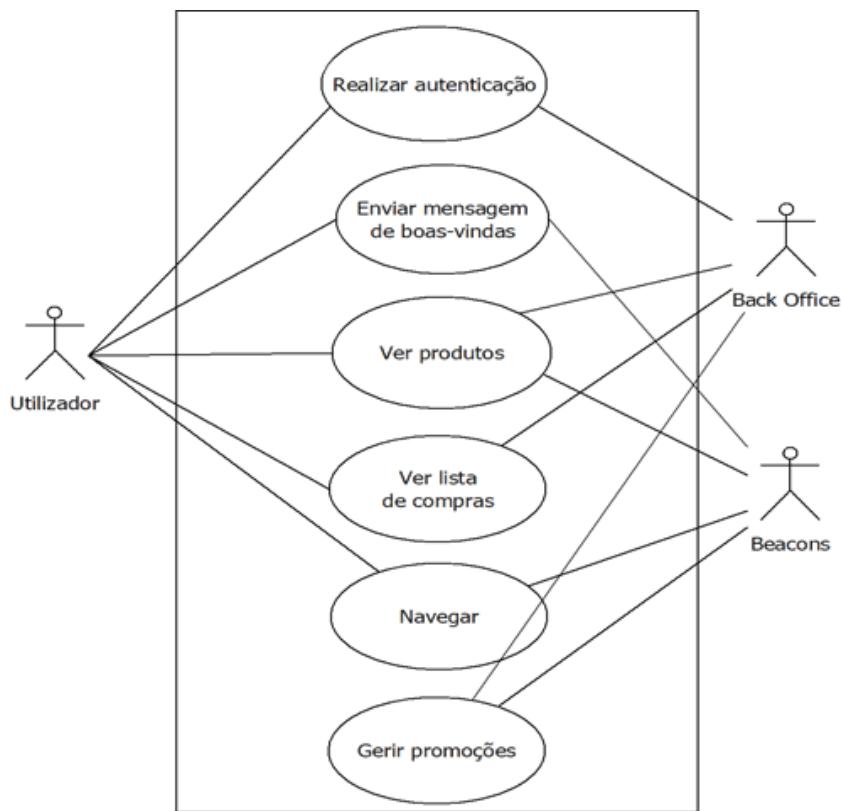


Figura 3.1: Modelo de casos de utilização

Como ordem de prioridade foi definido que se começava por desenvolver o *backoffice*, nomeadamente, a gestão de produtos, promoções e de *beacons*

de modo a que quando se realizasse a aplicação mobile fosse possível testá-la e observar o conteúdo disponível no supermercado.

3.2 Fundamentos

3.2.1 Beacons

Este projeto utiliza *Beacons* do tipo BLE (*Bluetooth Low Energy*). *Bluetooth Low Energy Beacons* são dispositivos que transmitem informação para dispositivos portáteis (*smartphones*, *tablets*, etc) quando estão suficientemente perto para serem detetados. Estes usam um sensor de *Bluetooth* de baixa energia para transmitir o seu identificador único, conhecido por *uuid*, bem como informação relativa ao seu *major* (usado normalmente para distinguir qual a loja em que o *Beacon* se encontra), informação relativa ao *minor* (usado normalmente como identificador do local onde o *Beacon* se encontra numa loja específica) e, por fim, informação relativa ao *signal strength* que deve ser calibrado para cada um dos dispositivos.

Os *Beacons* usados neste projeto são da marca *accent systems* (Figura 3.2) e usam o protocolo *iBeacon* [Inc.,].



Figura 3.2: *Beacons* utilizados

Relativamente à configuração dos mesmos é utilizada a aplicação *iBKS Config* [ibk,]. Esta aplicação encontra-se disponível na *Google Play* e permite-nos configurar todas as informações relativas ao *Beacon*, não só as referidas anteriormente mas também o *Advertising Interval* (definido em ms e permite-nos definir de quanto em quanto tempo o *Beacon* vai enviar um sinal), o *Radio TX Power* (usado para definir o quão intenso o sinal vai ser), assim como o *Calibration Power* (permite calibrar o força do sinal para um metro relativo a cada intensidade do *TX Power*).

3.2.2 Plataformas *low-code*

As plataformas *low-code* são cada vez mais populares a nível departamental e empresarial pelo facto de permitirem realizar, para vários dispositivos, projetos de uma forma mais rápida e fácil. Neste projeto foi usada a plataforma *Outsystems*. Esta encontra-se entre as melhores plataformas *low-code* do mercado tendo até sido nomeada

"Líder no The Forrester Wave: Low-Code Development Platforms for ADD Pros, Q1 2019, posicionando-se no lugar mais elevado na categoria de oferta atual." [Insight,]

A citação abaixo explica de uma forma breve o que é a Outsystems:

"A Outsystems garante que a sua plataforma oferece segurança, escalabilidade, performance, arquitectura e publicações flexíveis. A sua ferramenta de criação de aplicações, o Service Studio, permite aos programadores rapidamente criarem, actualizarem aplicações web e mobile e serviços." [sma,]

A Outsystems conta com um repositório de módulos reutilizáveis chamado de *Forge* [for,]. Desse repositório foram usados dois módulos, o *iBeacon plugin* [iBe,] e a *Firebase Mobile* [fir,] quevão ser explicados mais à frente.

3.2.3 Anylogic

O Anylogic é uma ferramenta de simulação usado em vários tipos de indústria. Esta ferramenta permite a simulação baseada em agentes, eventos discretos e em dinâmica de sistemas.

Neste projeto, como referido anteriormente, foi usada a biblioteca *Pedestrian* para simular o movimento dos consumidores dentro do supermercado e construir um mapa de densidade para se perceber quais as zonas que têm tendência a ser mais movimentadas considerando a posição dos *Beacons*. Esta biblioteca é usada para simular fluxos pedestres em aeroportos, estádios, estações ou em centros comerciais.

3.3 Abordagem

Todo o projeto foi planeado com o uso da *framework Scrum* [scr,]. Esta *framework* usa *sprints* para dividir as várias etapas de um projeto, o que permite um desenvolvimento sequencial e mais organizado do mesmo. Neste caso, o projeto foi dividido em 3 *sprints*. O primeiro *sprint* para a aprendizagem sobre a plataforma e para o desenvolvimento da aplicação *backoffice web*, o segundo para o desenvolvimento da aplicação *mobile* e, por fim, o terceiro para o desenvolvimento do relatório. Foi necessária uma preparação prévia da parte dos alunos relativamente à plataforma *Outsystems* uma vez que nenhum deles tinha tido contacto com ela, nem com outra plataforma *low-code*, anteriormente. Esta preparação consistiu na realização de tutoriais (visuais e práticos) criados pela *Outsystems* o que proporcionou aos alunos um maior conhecimento sobre a plataforma. Após a aprendizagem sobre a plataforma o objetivo é a realização da aplicação *backoffice web*, no entanto, antes da realização do mesmo é necessário desenhar os *wireframes* relativos à mesma com o objetivo de se perceber o que é necessário implementar. Após o desenho dos *wireframes* o foco foi o desenvolvimento do modelo de dados bem como a implementação de toda a aplicação *backoffice web*. De seguida o foco foi a configuração e o teste dos *Beacons* e após estes estarem funcionais segue-se a parte final deste projeto, a aplicação *mobile*.

Capítulo 4

Implementação do Modelo

4.0.1 Mapa do supermercado

Com o objetivo de se tornar percutível o objetivo dos *Beacons* para a aplicação foi desenhado um mapa que simula um supermercado (Figura 4.1). Neste mapa é possível ver os *Beacons* distribuídos pelas categorias, o cliente com a aplicação mobile e a administração que representa a aplicação de *backoffice*.

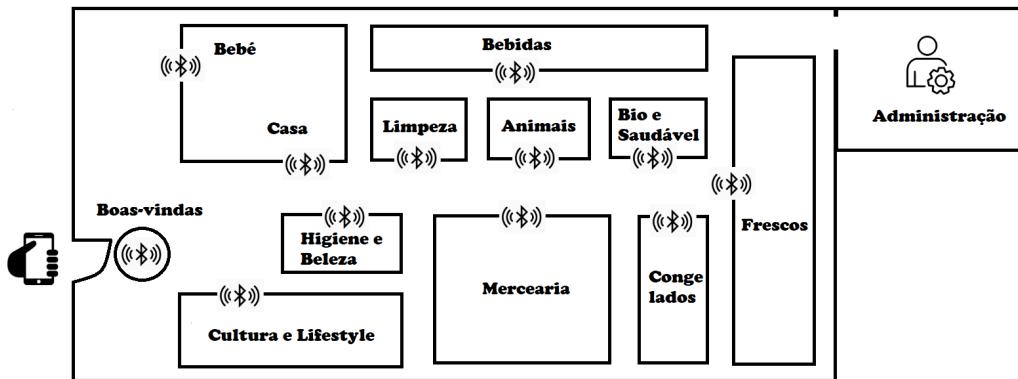


Figura 4.1: Desenho do mapa de um supermercado

4.0.2 Modelo de dados

Na figura 4.2 podemos ver as entidades implementadas no projeto. Em anexo é possível observar uma outra vantagem da plataforma *Outsystems*, esta desenvolve o seu próprio modelo de dados à medida que as entidades

vão sendo criadas. Como é percepível, cada produto bem como cada *Beacon* podem possuir/estar relacionados a uma ou mais promoções. Os produtos têm associados uma categoria e essa categoria está também associada aos Beacons o que permite associar o mesmo à sua localização.

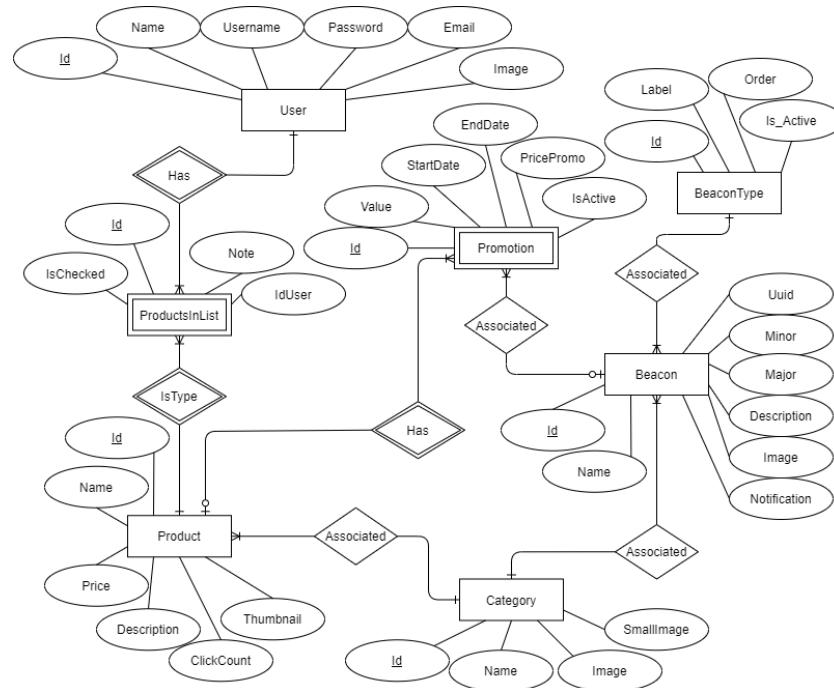


Figura 4.2: Modelo EA

4.0.3 Arquitetura Geral

A arquitetura da plataforma *Outsystems* (Figura 4.3) foi desenhada de modo a maximizar a performance e segurança das aplicações, fornecendo a possibilidade de processos de "*continuous delivery*" (abordagem da engenharia de *software* em que equipas produzem *software* em ciclos curtos, garantindo a fiabilidade do mesmo) e suporta "*continuous innovation*"(processo iterativo de incorporar melhorias nos processos/*designs* de produtos/serviços e tecnologias).

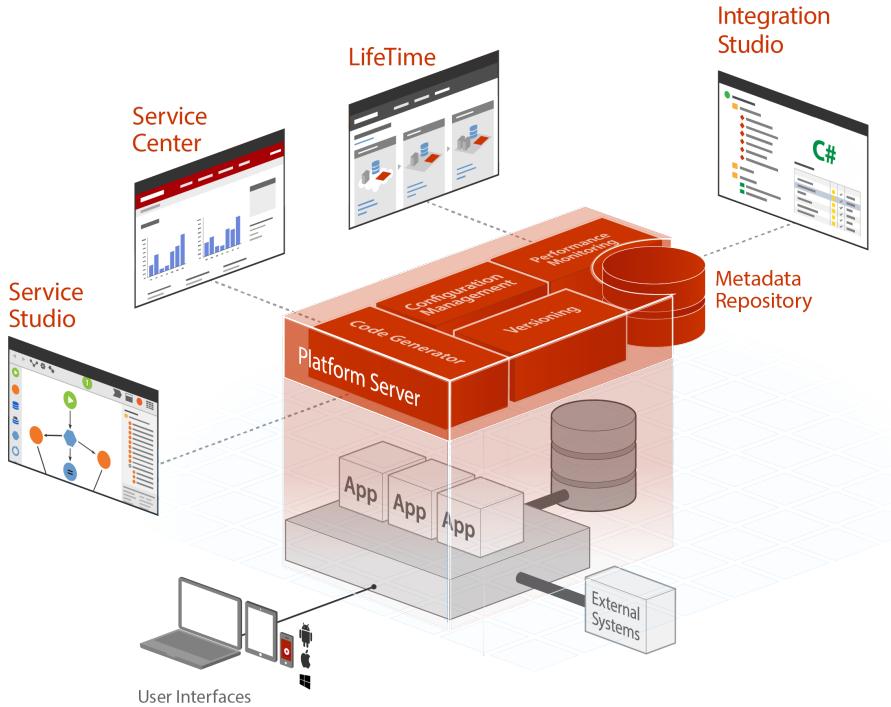


Figura 4.3: Arquitetura geral da *Outsystems*

O *core* da *Outsystems* é o *Platform Server*, é um componente de servidor que complementa uma aplicação *web standard* com um conjunto de serviços. A *OutSystems* não usa um mecanismo de execução próprio, todas as aplicações dependem de arquiteturas *standard* e *frameworks*. Isto é possível por causa da plataforma de servidor da *Outsystems*. Esta trata de todos os passos para gerar, otimizar, compilar e abrir aplicações num servidor de aplicação *web* ou numa aplicação *mobile* nativa. Este servidor da plataforma tem alguns serviços especiais como:

- *Code generator*: Pega na aplicação modelada no editor visual e gera, compila e constroi as aplicações *standard*, já otimizadas para *performance*.
- *Deployment services*: Estes tratam de abrir as aplicações para um servidor de aplicação *web standard*, garantindo que a aplicação é instalada consistentemente em cada *front-end* da infraestrutura.

- *Application services*: Tratam da execução de lotes de processos agendados e fornecem serviços de *logging* assíncronos para guardar eventos como erros ou métricas de performance.
- *Mobile apps build service*: Empacotam a aplicação *mobile* compilada numa aplicação *mobile* nativa que já se encontra preparada para ser instalada nos dispositivos.

4.0.4 Modelo de arquitetura

O *4 Layer Canvas* (4LC) [4la, 2020], (Figura 4.4), é uma ferramenta da arquitetura *Outsystems* para fazer com que o desenho das *Service-Oriented Architecture* (SOA) seja mais simples. Este promove a correta abstração de (micro)serviços reutilizáveis e o correto isolamento de módulos funcionais distintos.

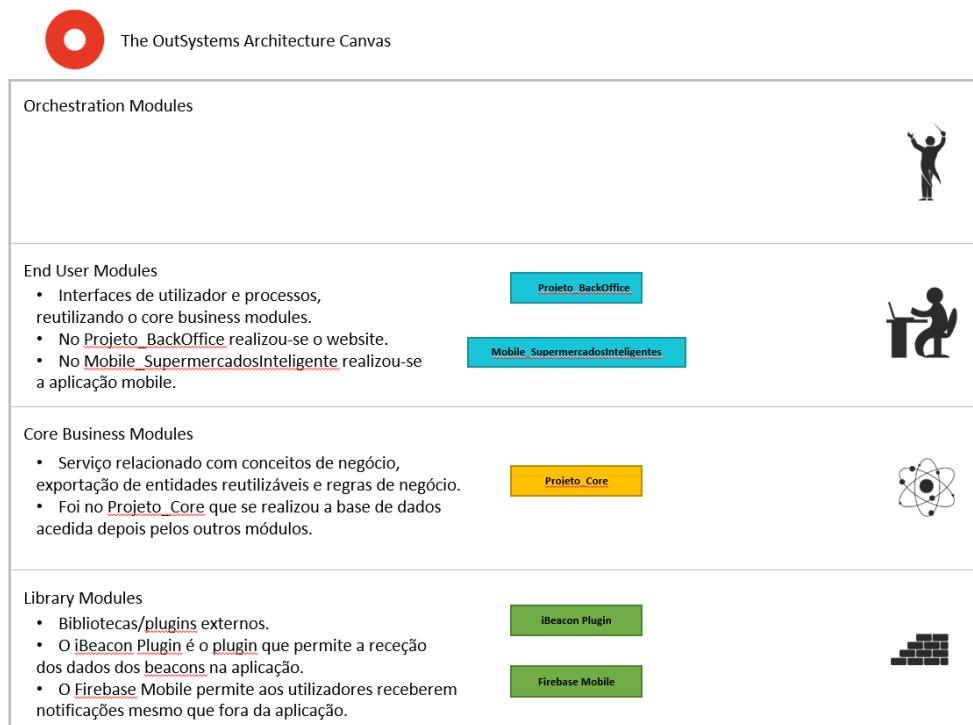


Figura 4.4: 4 Layer Canvas

4.0.5 Implementação

Back-Office

O primeiro passo para o desenvolvimento do *back-office* foi definir as prioridades da implementação. A ordem inicial seria:

- gestão de produtos,
- configuração dos *beacons*,
- gestão de promoções,
- gestão de localização,
- estética do site.

No que diz respeito à gestão de produtos decidiu-se começar por criar visualmente uma tabela (Figura 4.5) com os mesmos.

De modo a simular uma administração de um supermercado criou-se uma opção que permite adicionar produtos ou editá-los como se pode ver na Figura 4.5.

Devido à maneira como o trabalho se desenrolou percebeu-se que seria mais adequado realizar a gestão das promoções no mesmo ecrã (Figura 4.5) que a gestão dos produtos, prosseguindo assim à implementação da adição e edição das mesmas. Na Figura 4.5, na coluna das promoções pode observar-se uma das capacidades da *Outsystems*, que é a implementação de lógica visual, neste caso, um *if* que é usado para apresentar a possibilidade de adicionar uma promoção ou mostrá-la quando é criada. Este tipo de lógica é usada ao longo do projeto.

Nome	Categoria	Preço	Promoção	Remover
	Name	Bio e Saudável	Preço original: € Price (Preço com promoção: € PricePromo)	+
	Name	Bio e Saudável	Preço original: € Price (Preço com promoção: € PricePromo)	+
	Name	Bio e Saudável	Preço original: € Price (Preço com promoção: € PricePromo)	+

Figura 4.5: Tabela de gestão de produtos

De seguida, criou-se um *timer* que utiliza uma *server action* (Figura 4.6) com o objetivo de controlar as datas em que as promoções devem estar ativas, verificando todos os dias à 1h da manhã se seria suposto manter ou apagar as promoções atuais.

Na Figura 4.6 pode ver-se a lógica de servidor implementada. Começou por se obter as promoções existentes (na forma de um *aggregate*) e, de seguida, percorreram-se as mesmas recorrendo a um *for-each*, que analisou cada promoção e verificou se a data a que esta se iniciava era igual à data atual. Caso isto fosse verdade a promoção era colocada no estado ativo e era atualizada.

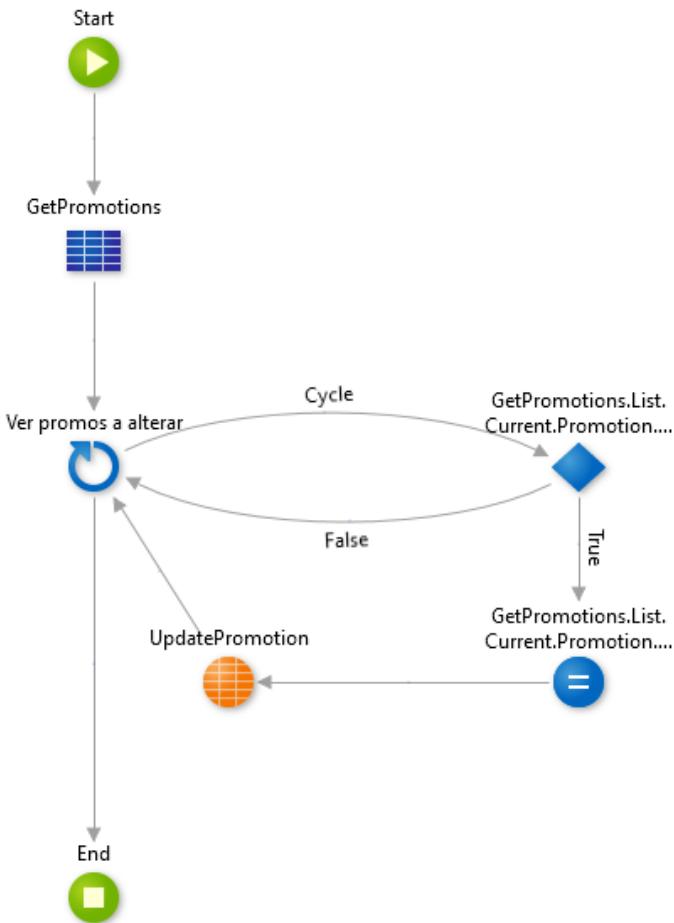


Figura 4.6: *Server Action* para adicionar promoção consoante o dia

Com a gestão dos produtos realizada partiu-se para a implementação do ecrã de configuração dos *Beacons*. Neste ecrã, tal como no ecrã anterior, começou por se criar uma tabela onde se mostram os *Beacons* existentes na base de dados, com as suas características mais importantes. Adicionou-se também a opção de criar novos *Beacons* e/ou editar os *Beacons* já existentes, como se pode observar na Figura 4.7.

The screenshot shows a web-based form titled 'Adicione um beacon' (Add a beacon). The form fields are as follows:

- Nome * (Name): An input field with a placeholder.
- Beacon Type * (Beacon Type): A dropdown menu with a placeholder 'Escolha o tipo do beacon...' (Select beacon type...).
- Uuid (Uuid): An input field.
- Minor (Minor): An input field.
- Major (Major): An input field.
- Description (Description): An input field.
- Image (Image): A file input field with a 'Select file' placeholder and a small icon.

At the bottom left is a 'Save' button.

Figura 4.7: Ecrã de adicionar/editar *Beacons*

No que diz respeito à gestão de localização percebeu-se que não era necessário um ecrã para alcançar esse objetivo, apenas foi necessário associar a categoria da zona onde seria colocado o *Beacon* ao mesmo.

Beacons

Com a aplicação *back-office* finalizada realizou-se uma aplicação para testar a ligação dos *Beacons* ao dispositivo móvel. Considera-se que esta foi a fase mais trabalhosa e que consumiu mais tempo devido a problemas de conectividade que serão explicadas no capítulo seguinte. Esta aplicação consistiu numa aplicação vazia que apenas tratava dos dados dos *Beacons* detetados e mostrava uma lista dos mesmos. Para esta funcionalidade, recorreu-se ao

iBeacon plugin já falado anteriormente.

Tanto neste teste como na aplicação *mobile*, em todos os ecrãs que se pretenda utilizar ou recorrer à funcionalidade dos *Beacons*, é necessário colocar o *iBeaconCentralDispatcher* (Figura 4.8) no ecrã. Este é o *web-block* principal para receber os eventos dos *Beacons*.

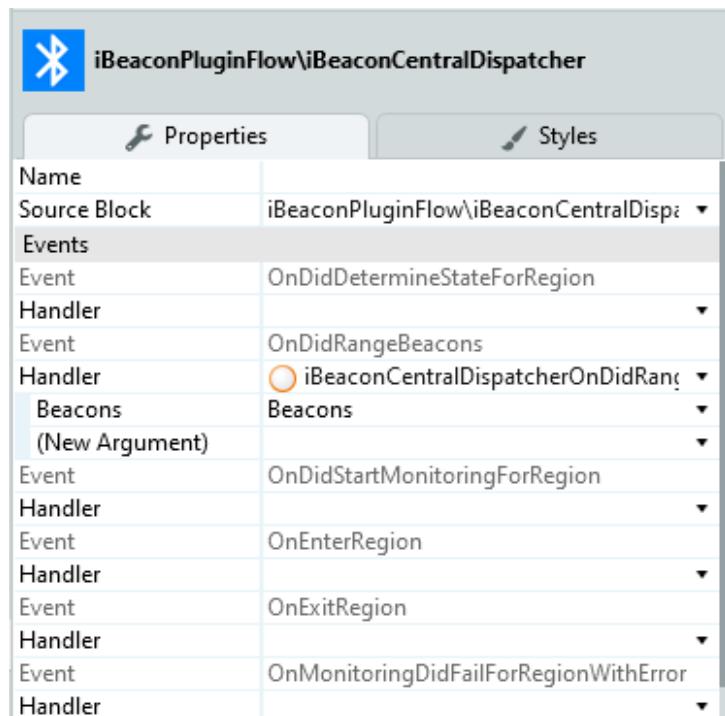
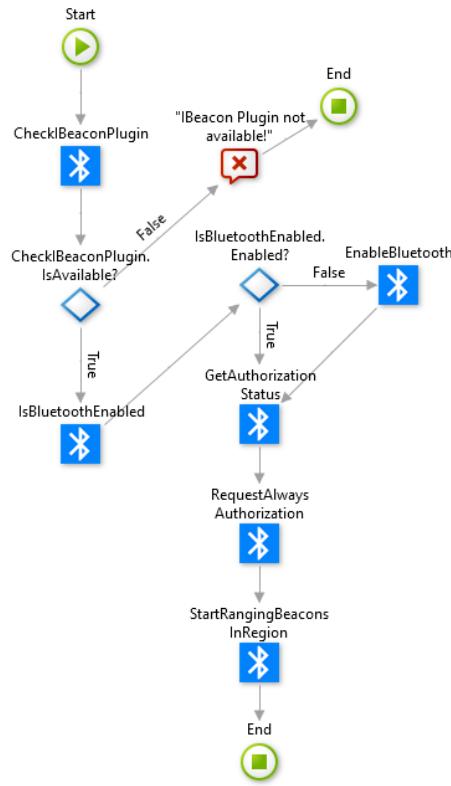
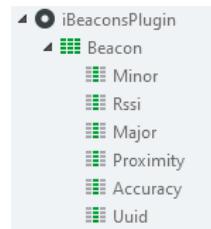


Figura 4.8: *Central Dispatcher*

De modo a garantir que o dispositivo que vai abrir a aplicação está preparado para utilizar esta tecnologia realiza-se um evento *OnInitialize* no primeiro ecrã afetado pelos *Beacons*, este evento possui a seguinte lógica (Figura 4.9):

Figura 4.9: *OnInitialize*

De modo a ver os *Beacons* basta arrastar a estrutura fornecida pelo *plugin* (Figura 4.10), a qual é comparada e adaptada aos *Beacons* existentes na base de dados.

Figura 4.10: *Estrutura fornecida pelo plugin*

Assim como os problemas, os resultados dos testes vão ser explicados mais à frente.

Aplicação *Mobile*

Para começar a aplicação *mobile*, fez-se como no *back-office*, primeiro definiram-se as prioridades (como se pode observar na lista abaixo) e de seguida iniciou-se a implementação.

- escolha de produtos,
- página descritiva do produto,
- lista de compras,
- menu lateral mais inferior,
- *login/registo*,
- perfil de user,
- garantir a independência das diferentes contas,
- notas nos produtos da lista de compras,
- *beacons* mais notificações,
- ecrã de ajuda,
- estética da aplicação.

Após o desenvolvimento do *back-office*, a aplicação (excepto a parte dos *Beacons*) foi relativamente fácil de implementar.

A escolha dos produtos consistiu em criar uma lista a mostrar os mesmos com a possibilidade de os adicionar à lista de compras ou de ver o produto em mais detalhe.

A lista de compras foi criada através de uma entidade que relaciona o produto com uma lista que por sua vez se relacionava com o utilizador (Figura 4.11).

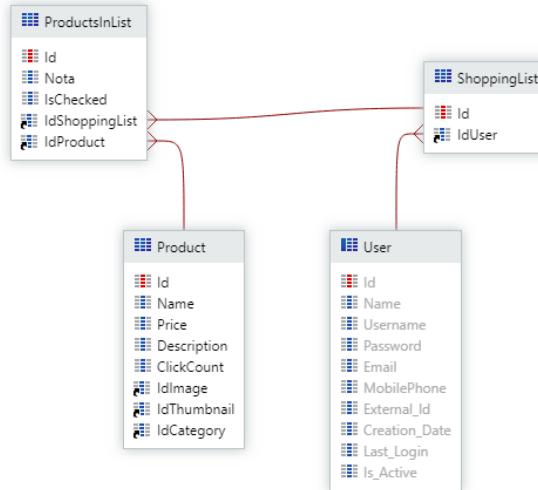


Figura 4.11: Entidades lista de compras

Neste ecrã o utilizador tem a possibilidade de adicionar notas aos produtos, aceder ao detalhe de cada produto, saber se já adquiriu o produto ou até eliminar os mesmos da lista. Tem também a possibilidade de ver o preço total da mesma. Isto tudo pode verificar-se na Figura 4.12.



Figura 4.12: Ecrã lista de compras da aplicação *mobile*

O passo seguinte foi a realização da possibilidade de registo na aplicação permitindo que vários utilizadores tenham a sua própria lista de compras. Isto possibilitou também a criação de uma página de perfil de utilizador (Figura 4.13) onde o mesmo pode alterar os seus dados e a sua foto de perfil e pode encontrar alguns dados sobre a aplicação, como por exemplo, os produtos mais procurados.



Figura 4.13: Ecrã de perfil

Devido a diversas complicações (vão ser realçadas mais tarde) deixou-se a funcionalidade dos *Beacons* para o final. Mais uma vez, recorreu-se ao *iBeacon plugin* que permite receber dados dos *Beacons* que se encontram perto do dispositivo móvel. Para isto teriam de ser comparados os *Beacons* já existentes na base de dados com a estrutura definida pelo *plugin*, como referido anteriormente, caso os dados recebidos correspondessem a algum da base de dados, este seria mostrado no ecrã de zonas de interesse.

Uma das últimas funcionalidades a ser realizada foi a do envio de uma mensagem de boas-vindas sempre que o utilizador entra no supermercado. Para isto, sempre que o utilizador abre a aplicação é verificado se este se encontra perto de algum *Beacon*. Caso isto aconteça este é redirecionado

para o ecrã das zonas de interesse onde é verificado se este se encontra perto do *Beacon* das boas-vindas recebendo assim uma mensagem personalizada.

No que diz respeito às notificações era para ser usado o *Firebase Mobile*. Este é um *plugin* externo usado para permitir a utilização de recursos da *Firebase*, como por exemplo, recursos de análise, notificações, avisos e logs de erros, monitorização da performance. Para tal foi necessário criar um projeto *mobile* na *Firebase* e ir buscar a *server key*. Esta foi usada para permissão de uso do *plugin* dentro da *Outsystems*. Depois disto foi necessário a configuração das *server actions*, fornecendo o *id* da aplicação, título da mensagem, corpo da mensagem e a *server key* como indicado antes. No entanto decidiu-se não utilizar, uma vez que, quando o utilizador se encontrava perto de um novo beacon este recebia uma notificação no telemóvel caso a aplicação estivesse a correr em background, no entanto, chegou-se à conclusão que, por causa dos problemas de sinal dos beacons iriam existir problemas com sobre-carga de notificações sempre que um novo beacon aparecesse. Outra razão para não se ter deixado esta implementação na versão final foi o facto de, mesmo depois da (suposta)correta configuração do *plugin*, mesmo seguindo variados vídeos e tutoriais para tal, a notificação não era recebida.

Anylogic

No que diz respeito ao *Anylogic* exigiu um pouco de aprendizagem geral bem como da biblioteca *Pedestrian*.

Após esta aprendizagem, foi necessário colocar o mapa onde a simulação deve ocorrer. De seguida foram definidas todas as paredes e estruturas do supermercado com o objetivo de delimitar quais as zonas onde os peões da simulação podem andar. Após serem definidas essas áreas foi necessário definir, através de *target lines*, onde estes peões devem começar e acabar a sua simulação, neste caso esse local é o mesmo.

Após estarem definidas as áreas em que o peão se pode mover e o local em que ele deve começar e acabar a simulação é necessário criar um fluxograma para simular o fluxo pedestre (Figura 4.14).

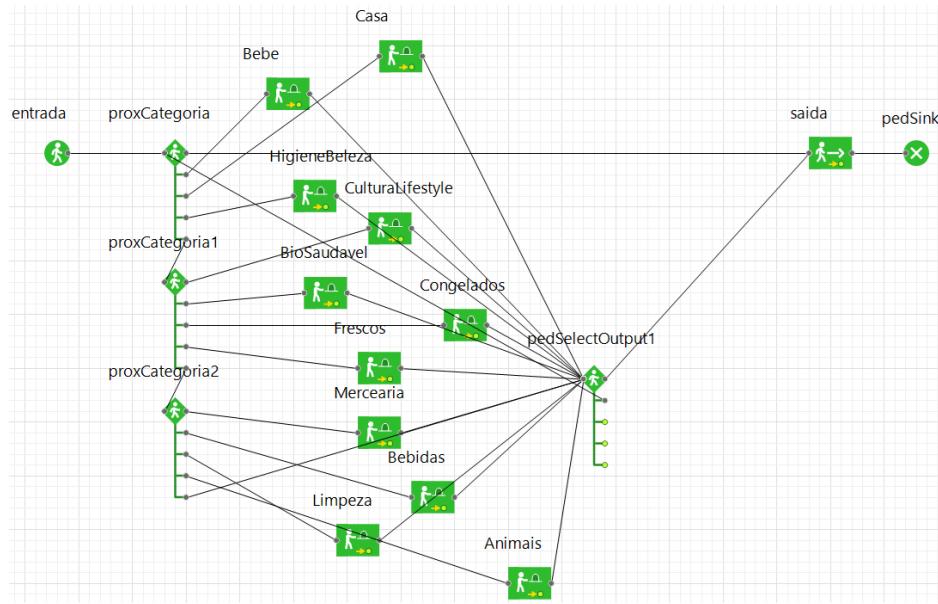


Figura 4.14: Fluxograma

Para isso foram usados os blocos que a biblioteca *Pedestrian* oferece. O bloco *Ped Source*, definido na (Figura 4.14) como ”entrada”, é onde o fluxo se inicia e ao qual teve que se associar a *target line* usada para definir o local onde os peões começam. O bloco *Ped Sink*, definido como ”saída” na (Figura 4.14), é onde o fluxo acaba e onde se associou a *target line* onde os peões acabam.

Após se ter definido qual o inicio e o fim do fluxo foi então definido o que acontece entre esses dois pontos.

Para este projeto foi tido em conta que cada uma das categorias do supermercado tem o mesmo interesse para o consumidor, para isso foi usado o bloco *Ped Select Output*, este bloco tem 5 saídas possíveis todas com probabilidades configuráveis. Como este projeto tem mais do que 5 categorias foram usados vários blocos destes. Desta forma foram configurados 3 blocos *Ped Select Output*, definidos na (Figura 4.14) como ”proxCategoria”, ”proxCategoria1”e ”proxCategoria2”, em que cada uma das saídas era uma categoria da loja e uma das outras saídas era para o cliente abandonar a loja.

Quando o consumidor entra na loja a probabilidade de a abandonar sem visitar nenhuma categoria foi considerada nula e, a cada iteração, ou seja, após visitar uma categoria, tem 20% de probabilidade de abandonar a loja.

Após o consumidor ir a uma categoria este tem então 80% de probabilidade de ir para outra categoria aleatória ou até de ficar na mesma, sendo que cada categoria tem a mesma probabilidade porque foi considerado que todas as categorias tinham o mesmo interesse.

Com esta simulação pretendeu-se observar quais as zonas mais frequentadas (Figura 4.15) para esta planta de supermercado com o objetivo de se perceber se esta colocação dos *Beacons* era apropriada.

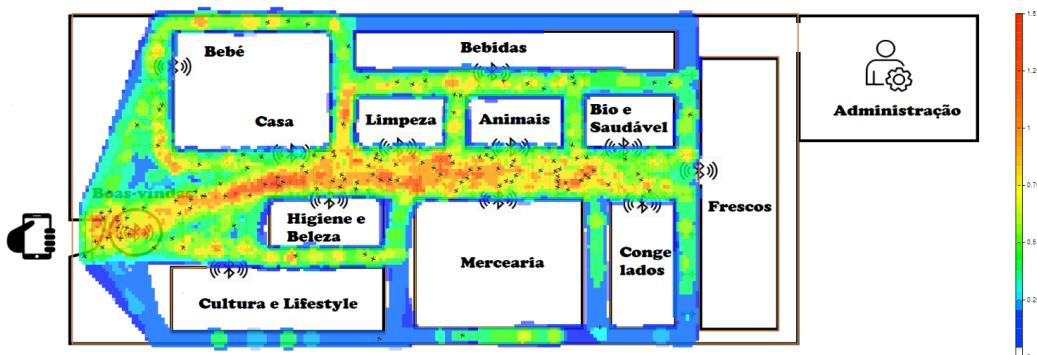


Figura 4.15: Mapa de densidade de pessoas

Como as probabilidades de cada zona são configuráveis é possível o interesse de cada zona ser alterado para efeitos de análise, por exemplo, quando uma zona tem promoções esta torna-se mais interessante.

Considera-se que esta simulação pode ser uma valia para os supermercados, podendo ajudar a otimizar a colocação dos beacons de forma a satisfazer tanto os consumidores como o supermercado.

Capítulo 5

Validação e Testes

5.0.1 Testes *back-office*

Nesta fase de testes vai-se começar por se testar as funcionalidades de back-office. Em primeiro lugar vai ser adicionado um produto à lista de produtos vazia (Figura 5.1).



Figura 5.1: Lista de produtos vazia

Para se adicionar um produto tem que ser preenchido um *form* (Figura 5.2) as características do mesmo.

Adicione um produto

Nome *

Preço *

Descrição

Categoria

Figura 5.2: Adicionar um produto

Após o *form* ser preenchido e guardado é possível ver-se (Figura 5.3) que o produto foi adicionado com sucesso.

Gerir os Produtos

Nome	Categoria	Preço	Promoção	Ações
teste	Cultura e Lifestyle	Preço original: €10.00		

Figura 5.3: Produto adicionado

De seguida o próximo teste foi alterar o nome do produto adicionado anteriormente sendo esta alteração realizada com sucesso. (Figura 5.4).

Gerir os Produtos

Nome	Categoria	Preço	Promoção	Ações
teste1	Cultura e Lifestyle	Preço original: €10.00		

Figura 5.4: Editar produto

Após se ter testado a adição de um produto e a alteração do seu conteúdo, o próximo passo foi adicionar uma promoção de 50% ao produto com a duração de apenas um dia (Figura 5.5). Estes testes foram realizados durante o dia 19 de julho portanto no dia 20 a promoção já não devia estar ativa.

Adicione uma promoção

Valor *

50

Data Início *

19/07/2020

Data Fim *

19/07/2020

Save

Figura 5.5: Adicionar uma promoção ao produto existente

Como se pode ver na Figura 5.6, a promoção foi adicionada com sucesso.

Gerir os Produtos					Adicionar produto
Nome	Categoria	Preço	Promoção	Remover	
teste1	Cultura e Lifestyle	Preço original: €10.00 (Preço com promoção: €5.00)	+ Promoção: 50% ✖		

Figura 5.6: Promoção adicionada ao produto

De seguida foi feito o teste de se adicionar ao mesmo produto outra promoção que começasse no mesmo dia, no entanto esta foi rejeitada uma vez que cada produto só pode ter uma promoção ativa (Figura 5.7).

Valor *

Data Inicio *

A promoção não pode ser adicionada pois sobrepõe a atual.

Data Fim *

Save

Figura 5.7: Tentativa de sobrepor uma promoção

Foi então testado adicionar uma promoção de 60% para começar dia 20 de julho em que não há nenhuma promoção ativa (Figura 5.8).

Valor *

Data Inicio *

Data Fim *

Save

Figura 5.8: Adicionar promoção para começar quando a promoção atual acabar

Na (Figura 5.9), já no dia 20, pode ver-se a nova promoção de 60% adicionada com sucesso ao produto.

Gerir os Produtos					Adicionar produto
Nome	Categoria	Preço	Promoção	Remover	
teste1	Cultura e Lifestyle	Preço original: €10.00 (Preço com promoção: €4.00)	+ Promoção: 60%		

Figura 5.9: Promoção de dia 20 de julho adicionada com sucesso

De seguida foi adicionado um *Beacon* ao *back-office*, este precisa de um *UUID* correto, senão dá erro como se pode ver na (Figura 5.10).

Uuid

00000000-0000-0000-000000001

Verify if Uuid is written correctly.

Figura 5.10: UUID errado

Foi então adicionado um *UUID* correto como se pode ver na (Figura 5.11).

Uuid

00000000-0000-0000-000000000001

Figura 5.11: UUID correto

Foi então adicionado um *Beacon* de teste como se pode ver na Figura 5.12. Neste caso já existiam alguns *Beacons* criados que são os *Beacons* usados no projeto.

Gerir os Beacons					Adicionar Beacon
Nome	Tipo de Beacon	Id Categoria	Minor	Remover	
Bio e Saudável	Product	Bio e Saudável	1		
<i>Bon-vende!!</i> Boas-vindas	Extra		2		
teste	Extra		3		

Figura 5.12: *Beacon* de teste adicionado

De seguida foi alterado o nome do *Beacon* com sucesso (Figura 5.13).

Gerir os Beacons					Adicionar Beacon
Nome	Tipo de Beacon	Id Categoria	Minor	Remover	
Bio e Saudável	Product	Bio e Saudável	1		
<i>Bon-vende!!</i> Boas-vindas	Extra		2		
teste1	Extra		3		

Figura 5.13: *Beacon* de teste alterado

5.0.2 Testes aplicação *mobile*

Na (Figura 5.14) podemos ver o ecrã geral dos produtos.

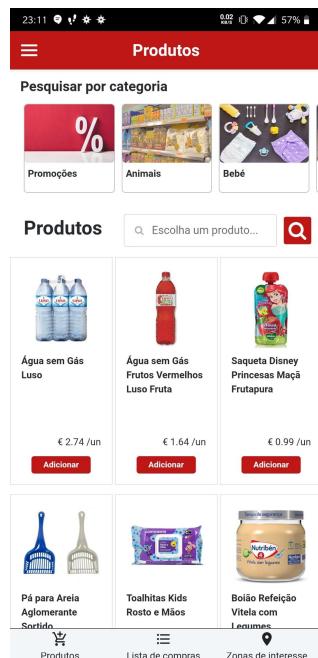


Figura 5.14: Ecrã base dos produtos

Quando aplicado o filtro por categoria, neste caso ”Animais” é possível ver que a aplicação faz a filtragem com sucesso (Figura 5.15).



Figura 5.15: Ecrã de produtos com filtro para a categoria ”Animal”

Na Figura 5.16 podemos ver a lista de compras com alguns produtos e com o valor total dos mesmos.

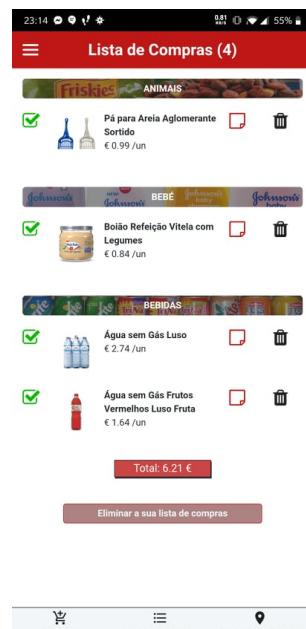


Figura 5.16: Lista de compras

É também possível adicionar e editar notas aos produtos da lista, como se pode ver na Figura 5.17



Figura 5.17: Adicionar uma nota

Após o consumidor adquirir um produto da lista ele pode selecionar a checkbox do produto e este aparecerá com o *background* a vermelho como forma de distinguir quais os produtos que já tem e os que ainda não tem (Figura 5.18).

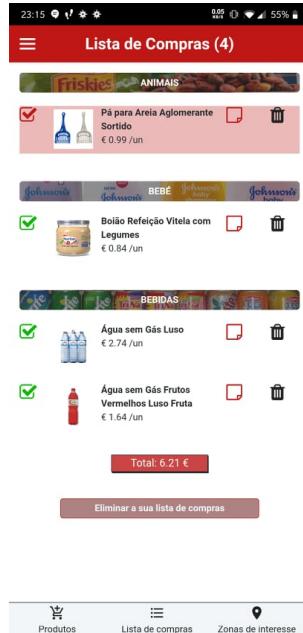


Figura 5.18: Produto adquirido

A aplicação conta com um gráfico que diz quais os produtos mais clicados da semana para cada utilizador. Pode ver-se o produto de teste a aumentar o seu número de clicks da Figura 5.19 para a Figura 5.20.



Figura 5.19: Produto teste com 1 click

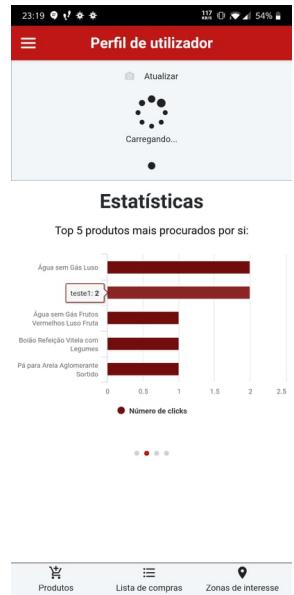


Figura 5.20: Produto teste com 2 clicks

A aplicação permite também a criação de uma conta para o utilizador. Na Figura 5.21 vemos que os dados que o utilizador introduziu estão incorretos ou não existem.

The figure shows a mobile login screen. At the top, there is a red error message box containing the text "Invalid username or password." Below this, there is a logo of a smartphone displaying a shopping cart icon. The login form has two fields: "Username *" with the value "Teste" and "Password *" with the value ".....". There are "Recuperar password" and "Login" buttons at the bottom. A small link "Não possui conta? Registe-se" is located at the very bottom.

Figura 5.21: Dados de inicio de sessão estão incorretos ou não existem

Foi então criada uma conta para o utilizador, como podemos ver na Figura 5.22.

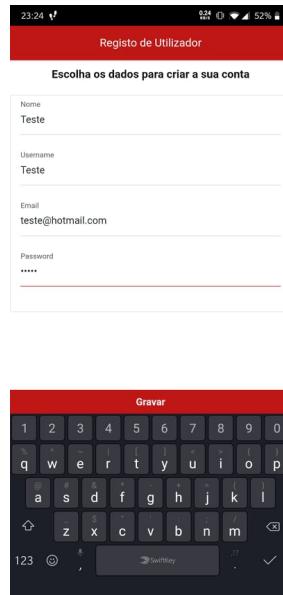


Figura 5.22: Criação de uma conta

Após criar a conta o utilizador tem então o seu próprio perfil (Figura 5.23).

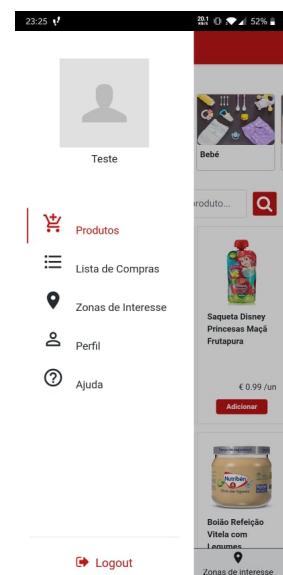


Figura 5.23: Perfil do utilizador

5.0.3 Testes Beacons

Para calcular as distâncias entre os *Beacons* utilizou-se a fórmula da Figura 5.24, esta foi proporcionada pela *IOT and Electronics* [Iotbymukund e says:, 2016] os parâmetros da fórmula são obtidos através da aplicação de configuração dos *Beacons*, falada anteriormente.

$$DISTANCE = 10^{\left(\frac{Measured\ Power - RSSI}{10^N} \right)}$$

Figura 5.24: Fórmula de cálculo das distâncias

Para se ter mais ou menos uma ideia da capacidade dos *Beacons* procurou-se algumas tabelas/estudos acerca da relação dos diferentes parâmetros, e a norma era mais ou menos parecida com a tabela da Figura 5.25.

Transmission Power	Distance in the app
0 dBm	50m
-50 dBm	1 m
-40 dBm	3 m
-20 dBm	12 m
-16 dBm	18 m
-12 dBm	25 m
-8 dBm	35 m
-4 dBm	40 m
0 dBm	50 m
+4 dBm	60 m
+5 dBm	80 m

Figura 5.25: Tabela de relação dos diferentes parâmetros

A usabilidade dos *Beacons* na aplicação passa pelo ecrã de "zonas de interesses", onde é demonstrada a lista dos *Beacons* que se encontram perto dos dispositivos. Como teste foram tiradas duas fotografias como "prova" do funcionamento, embora o seu conteúdo não seja muito visível é o necessário

para dar a entender o seu objetivo e funcionalidade. Para efeitos de teste foram utilizados dois telemóveis, um mais antigo com o *android* 8e um recente com o *android* 10.

Na Figura 5.26 dá para se ver um dos problemas que ocorrem quando se utiliza a tecnologia dos *Beacons* na *Outsystems* (pelo menos com o *iBeacons plugin*). O telemóvel da esquerda é o telemóvel de nível mais baixo, sendo assim, a sua qualidade de ligação *bluetooth* não é tão boa quanto o outro, no entanto pode observar-se que aparece a informação dos *Beacons*, o problema é que o sinal é relativamente fraco, fazendo com que haja perda da informação e que os *Beacons* apareçam e desapareçam ocasionalmente. O telemóvel da direita é um telemóvel relativamente bom, recente (saiu este ano) e por isso assume-se que a sua qualidade na ligação é boa, o problema é que este possui o *android* 10 e alguns *plugins* da *Outsystems* ainda não estão otimizados para este sistema. Neste telemóvel, quando apanha sinal, o sinal é ótimo e estável, podendo mesmo andar com a aplicação ligada e receber informação dos *Beacons* bastante coerente, o problema, é que demora até apanhar o sinal, acontece que às vezes tem de se deixar a aplicação aberta e só depois de navegar entre outras aplicações (a fazer tempo) para se conseguir ligar aos *Beacons*.

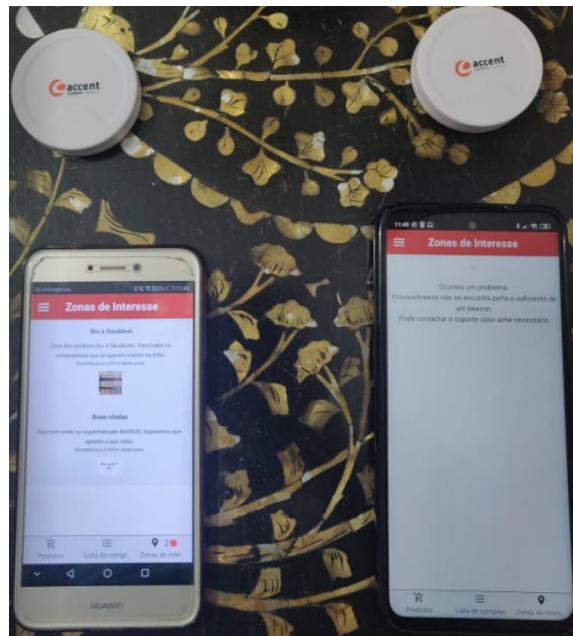


Figura 5.26: Teste em dois telemóveis em que um falha e outro não

Na Figura 5.27 já se pode observar o ”melhor” telemóvel a receber o sinal dos *Beacons* e, embora não possa ser demonstrado, o sinal foi bastante conclusivo, ao ponto de se afastar dos *Beacons* e a informação acerca da distância não falhar por muito. Quando o telemóvel se encontra a menos de 1.50m as falhas rondam em média os 15 cm’s, quando se começa a afastar as falhas vão então aumentando um bocado mas nada que comprometa o objetivo desta aplicação. Embora não se consiga ver muito bem na fotografia, as distâncias apontadas na aplicação, dos *Beacons* eram de, 0.156m e 0.141m enquanto que na realidade estavam próximos dos 0.10m, considera-se assim que, quando o sinal era estabelecido, este era relativamente bom.

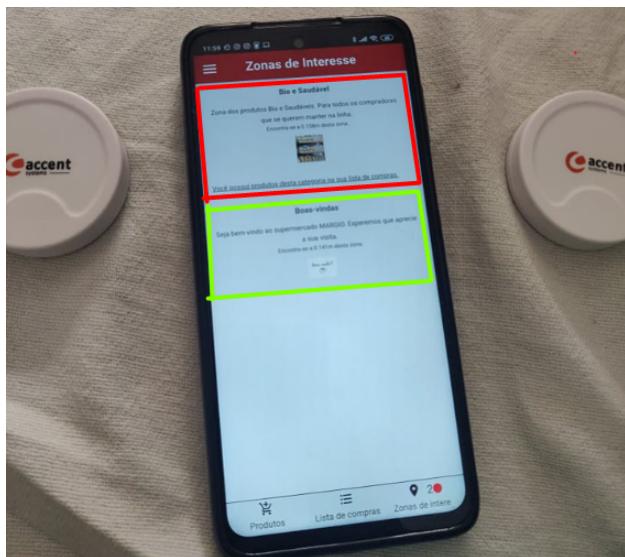


Figura 5.27: Teste de deteção de dois *Beacons*

Por fim, mais uma vez, com a ligação estabelecida ao telemóvel que seria mais capaz de utilizar todas as funcionalidades da aplicação (*Beacons*), colocaram-se os *Beacons* com uma distância de mais ou menos 1.50m e andou-se nessa zona só para se obter, verificar e confirmar que estaria tudo a funcionar (Figuras 5.28, 5.29, 5.30). Mais uma coisa, pode observar-se que no *Beacon* dos produtos bio e saudáveis, aparece a dizer que o utilizador possui um produto daquela categoria na sua lista de compras, isto tem o objetivo de chamar à atenção do comprador, que aquela zona (do *Beacon*) pode ser importante para o mesmo. Quando estas zonas possuem promoções, estas são também mostradas, mais uma vez, com o intuito de chamar à atenção

para aquela zona. A ideia inicial seria mostrar alguns produtos em promoção mas considerando as falhas com alguns dos telemóveis de teste, considerou-se que seria mais prejudicial se possuísse fotos, considerando que não iria dar para as ver se houvesse quebras na ligação aos *Beacons*.



Figura 5.28: Deteção *Beacon* Bio e Saudável a 0.531 metros e do *Beacon* de Boas-Vindas a 0.668 metros



Figura 5.29: Deteção *Beacon* Bio e Saudável a 1.189 metros e do *Beacon* de Boas-Vindas a 1.585 metros



Figura 5.30: Deteção *Beacon* Bio e Saudável a 0.668 metros e do *Beacon* de Boas-Vindas a 0.211 metros

5.0.4 Testes *Anylogic*

No que diz respeito ao *Anylogic* foram feitos alguns testes em que se aumentaram as probabilidades em categorias específicas para se simularem promoções nessa categoria e se perceber quais as zonas mais movimentadas de forma estratégica para que o supermercado possa colocar produtos específicos nessas zonas mais movimentadas.

Na Figura 5.31 vemos um mapa de densidade para uma simulação com 2000 pessoas por hora e em que o tempo simulado corresponde a uma hora (limite máximo de simulação do *Anylogic* para aprendizagem). Nesta simulação todas as categorias tinham a mesma probabilidade.

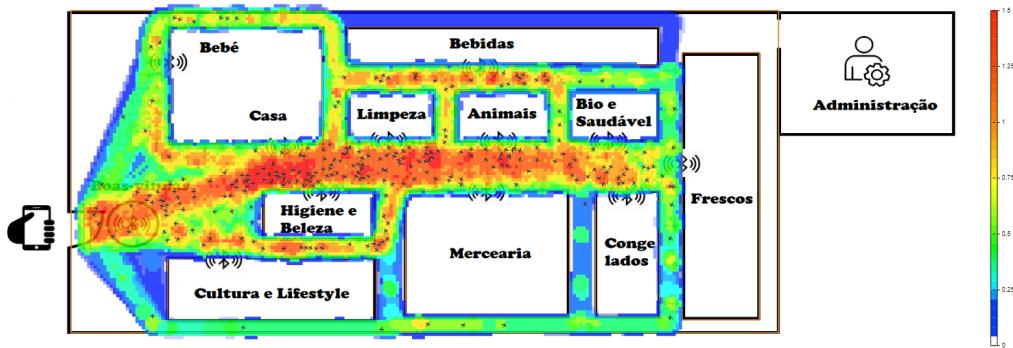


Figura 5.31: Mapa de densidade em que todas as categorias têm a mesma probabilidade

De seguida, decidiu-se aumentar a probabilidade na categoria ”Mercearia” e testar (Figura 5.32). Percebeu-se que a zona central do supermercado ficou com uma grande densidade de pessoas, sendo um bom ponto estratégico para colocar outros produtos nessa zona que os consumidores acabem por comprar.

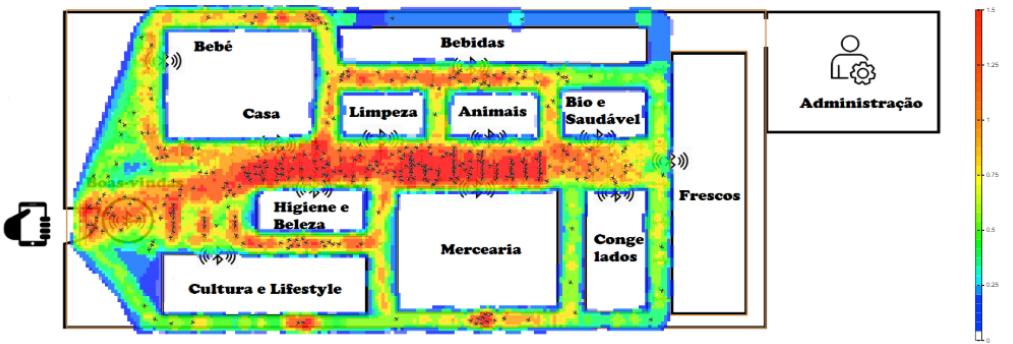


Figura 5.32: Mapa de densidade em que a categoria ”Mercearia” tem promoções

Estando a categoria ”Mercearia” situada numa zona central do supermercado decidiu-se testar também numa zona periférica. Foi escolhida a categoria ”Bebidas” para ser simulada a promoção. Como se pode ver na Figura 5.33 há quase um homogeneidade entre a zona central e a zona onde se encontra a promoção no que diz respeito à densidade.

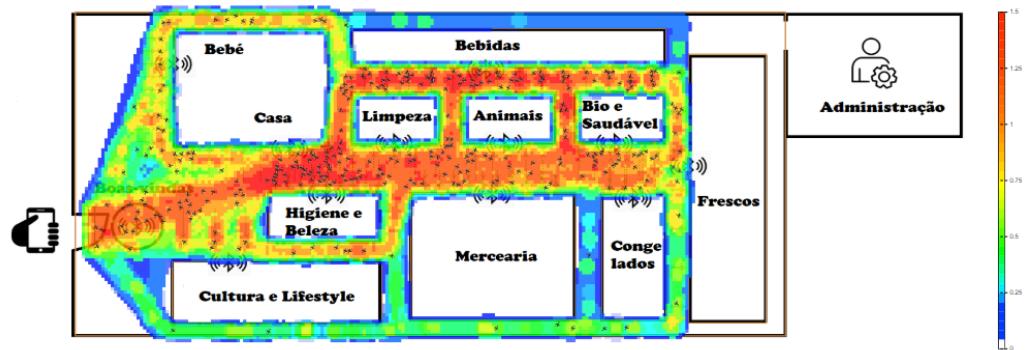


Figura 5.33: Mapa de densidade em que a categoria "Bebidas" tem promoções

Há muitos outros testes que podiam ser feitos, neste caso, foram feitos apenas testes que elucidassem o poder desta ferramenta e a possível capacidade de complemento relativo à nossa aplicação e para mostrar que pode ser uma mais valia para uma administração de um supermercado.

Capítulo 6

Conclusões e Trabalho Futuro

Este projeto tinha como objetivo o desenvolvimento de uma aplicação *mobile* que permitia criar uma lista de compras e que notificava o utilizador quando se encontrava perto de um produto da sua lista. O objetivo era melhorar a experiência do consumidor quando este realiza as suas compras. Para tal foi estudada uma plataforma *low-code*, a *Outsystems*. Esta plataforma permite a reutilização de código criada noutras sistemas, criar *webservices* sem ter que se recorrer a código e, caso seja preciso, desenvolver o próprio código. Tanto a aplicação *mobile* como a aplicação *web* que funcionava como *back-office* foram desenvolvidas nesta plataforma. No que diz respeito ao *back-office* foi criada uma aplicação *web* que permite adicionar e editar os produtos, as promoções e os *beacons* e associar as promoções aos produtos bem como associar os *Beacons* às categorias dos produtos.

Relativamente à aplicação *mobile* esta cumpre os requisitos propostos inicialmente, desde adicionar os produtos à lista de compras, detetar os *Beacons* ao seu redor, mostrar o valor total da lista e até enviar uma mensagem de boas-vindas quando o consumidor entra no supermercado.

Foi uma experiência nova a realização deste projeto, principalmente por ter sido realizado numa plataforma *low-code*.

Foi também utilizada a ferramenta *Anylogic*. Esta é uma ferramente de simulação bastante poderosa e que poderia ser explorada de variadas formas, no entanto, neste projeto apenas se queria simular o movimento dos clientes dentro de um supermercado com diferentes parâmetros de entrada e, posteriormente, criar um mapa de densidade para cada um desses parâmetros.

No que diz respeito à tecnologia dos *Beacons* esta tem alguns proble-

mas no que toca a transmissão de sinal quando este interceta algum tipo de obstáculo não sendo totalmente correta a receção do sinal, podem também haver problemas no caso de haver uma má ligação *Bluetooth* por parte do dispositivo ou até problemas de rede o que influencia a ligação aos dados móveis, no entanto, funcionaram minimamente bem para a realização e testes deste projeto.

Percebeu-se que a utilização da *framework Scrum* foi uma mais valia no que se refere à organização das etapas do projeto e das tarefas que deviam ser realizadas. Algumas melhorias que podiam ser feitas neste projeto são, por exemplo, associar a cada produto um mapa com a sua localização e que seria mostrado ao consumidor caso esse produto estivesse na sua lista e detetasse o *Beacon* associado a esse produto. Outra melhoria que podia ser feita era na aplicação de *back-office* no que diz respeito à performance de carregar o site quando tem muitos produtos criados. Em suma, foram cumpridos todos os objetivos propostos para este trabalho apesar de alguns obstáculos que foram aparecendo ao longo do seu desenvolvimento.

Apêndice A

Um Detalhe Adicional

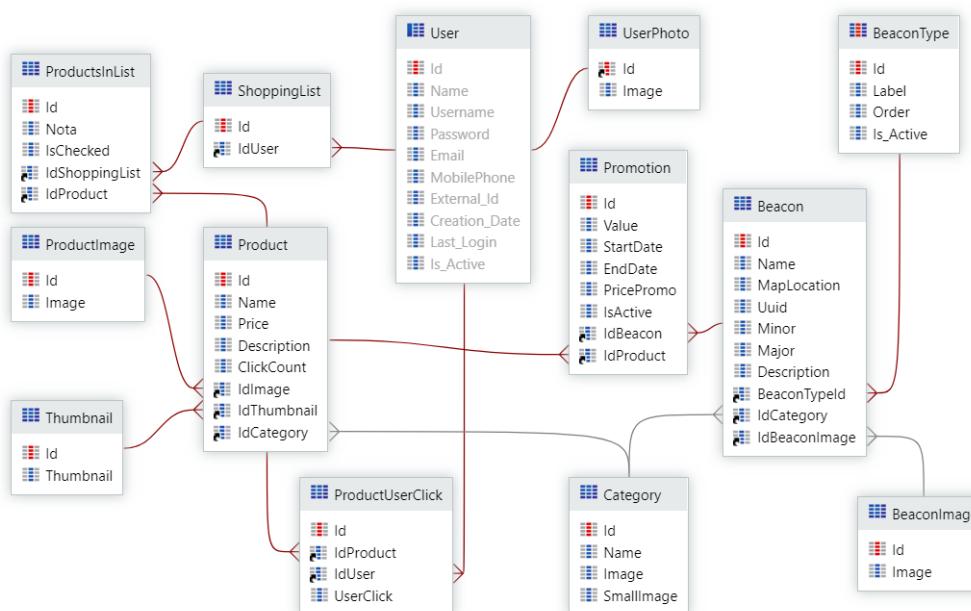


Figura A.1: Modelo de dados geral do projeto

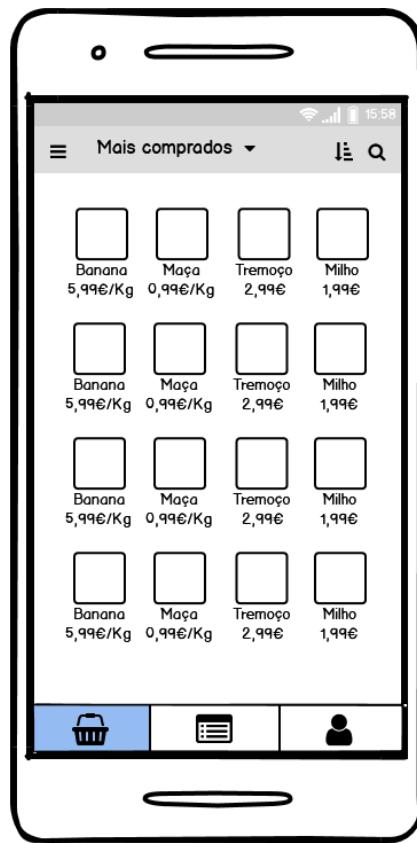


Figura A.2: Ecrã de produtos

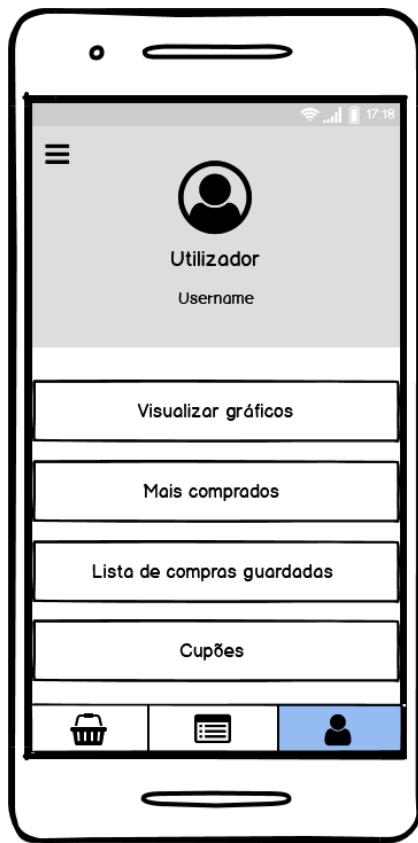


Figura A.3: Ecrã de perfil

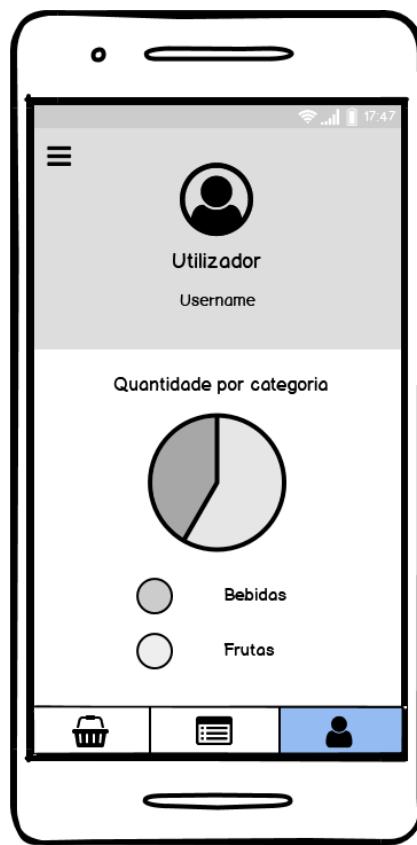


Figura A.4: Ecrã de perfil com gráficos

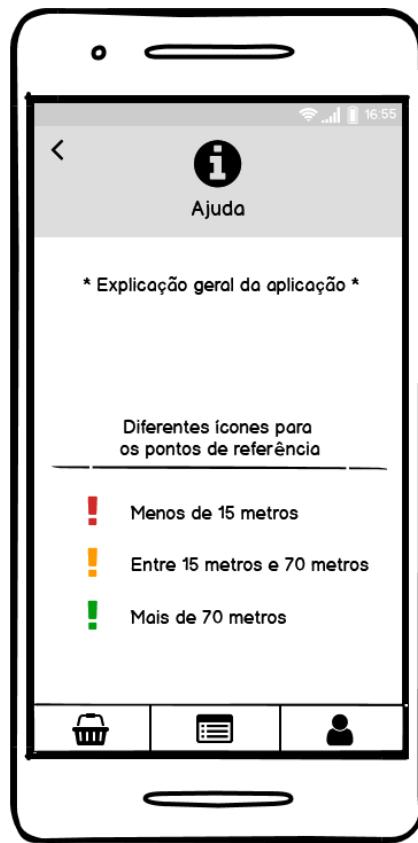


Figura A.5: Ecrã de ajuda



Figura A.6: Ecrã Lateral

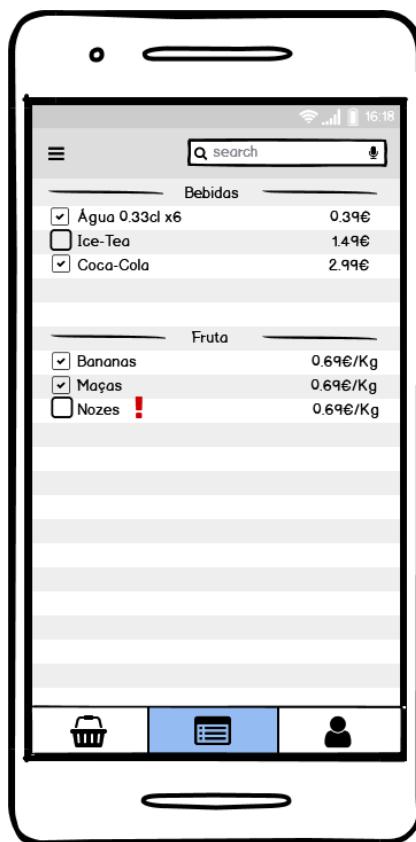


Figura A.7: Ecrã de lista de compras



Figura A.8: Ecrã de lista de compras com mapa

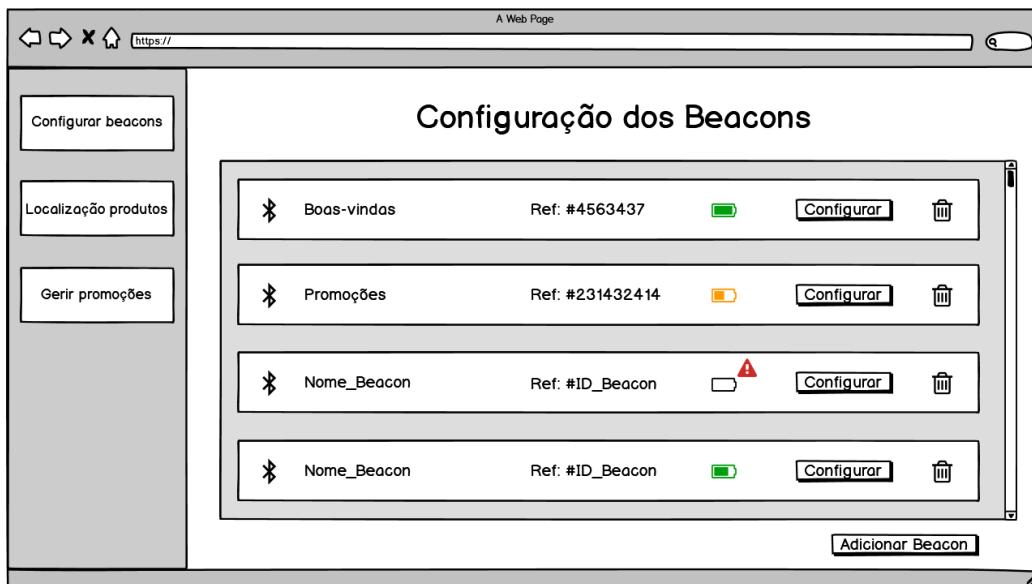


Figura A.9: Ecrã de configuração dos *Beacons*

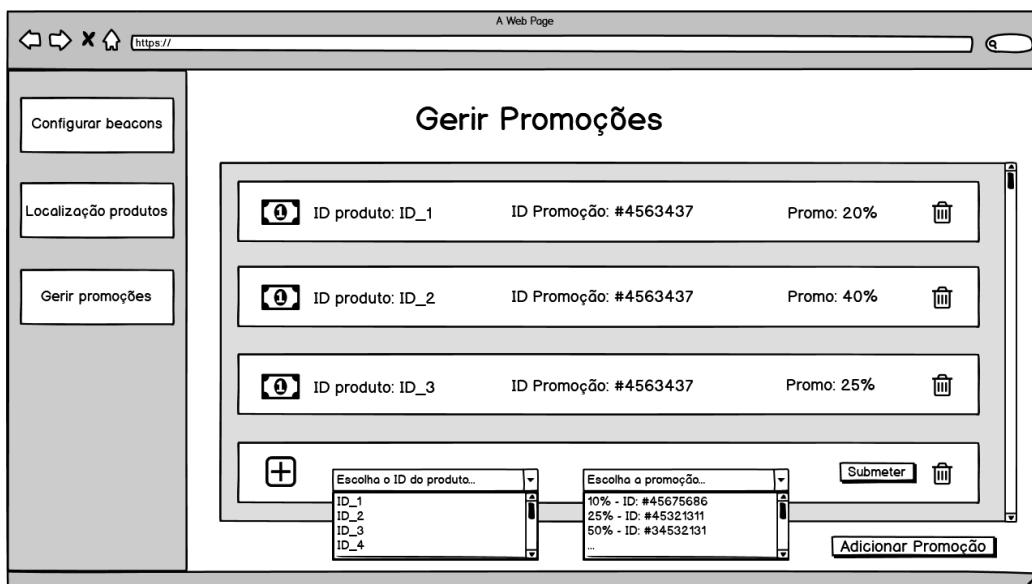


Figura A.10: Ecrã de gerir as promoções



Figura A.11: Ecrã de localização dos produtos

Bibliografia

[out,] The 1 low-code platform for digital transformation.

[bri,] Bring! shopping list.

[fir,] Firebase mobile - overview.

[for,] Forge.

[scr,] Home.

[iBe,] ibeacons plugin - overview.

[ibk,] ibks config tool user manual - android · accent systems.

[lac,] La comer.

[mlb,] Mlb ballpark.

[any, a] Pedestrian library.

[pro,] Proximitask twitter account.

[any, b] Simulation modeling software tools & solutions for business.

[sma,] Smart consulting.

[lis,] The smart shopping list app for android, ios, online.

[ksu,] Supermarket: Tavallista parempi ruokakauppa.

[4la, 2020] (2020). The 4 layer canvas.

[Inc.,] Inc., A. ibeacon.

[Insight,] Insight, I. Outsystems nomeada líder em plataformas low-code.

[Iotbymukund e says:, 2016] Iotbymukund e says:, L. H. (2016). How to calculate distance from the rssi value of the ble beacon.