ShopOn – Compras Inteligentes

Nuno Faião – 41566, a41566@alunos.isel.pt
Gonçalo Rações – 42225, a42225@alunos.isel.pt
José Cunha – 43526, a43526@alunos.isel.pt

Orientadores
Paula Graça, ISEL, mgraca@deetc.isel.pt
Rui Lopes, Do It Lean, rui.lopes@doitlean.com

Relatório beta realizado no âmbito de Projeto e Seminário,
do curso de licenciatura em Engenharia Informática e de Computadores
Semestre de Verão 2018/2019
29 Abril 2019

Índice

Índice iii

Lista de Figuras v

1. Introdução 1

1.1 Enquadramento 1

1.2 Objetivos 1

2. Formulação do Problema 3

2.1 Estado da arte 3

2.2 *Use Cases* 5

2.2.1 Cinema 7

2.2.2 Loja´ 9

2.2.3 Restaurante 12

2.3 Beacons 16

2.4 Plataforma da Outsystems 17

2.5 Metodologia de desenvolvimento (Scrum) 18

3. Solução Proposta 21

3.1 Arquitetura da solução 21

3.1.1 Descrição com diagrama figurativo 21

3.1.1 4 Layer Canvas 22

Referências 25

A.1 Modelos de dados 27

A.1.1 Cinema 27

A.1.2 Restaurante 28

A.1.3 Loja 29

A.1.4 Comum 30

Lista de Figuras

Figura 1- Use Case geral 5
Figura 2- Wireframe HomeScreen 6
Figura 3- Use Case cinema 7
Figura 4- Wireframe cinema 8
Figura 5-Use Case loja 9
Figura 6- Wireframe loja 11
Figura 7- Use Case restaurante 12
Figura 8- Wireframe restaurante 14
Figura 9- Wireframe produto de restaurante 15
Figura 10- Board Trello (22/04/2019) 18
Figura 11- Diagrama figurativo 21
Figura 12- 4 Layer Canvas 22

1. Introdução

1.1 Enquadramento

Atualmente, com os progressos tecnológicos, o comércio físico tem perdido espaço perante o comércio online [1]. Esta preferência é devido ao facto da facilidade com que se procuram os produtos desejados no conforto da nossa casa, de uma forma rápida e eficaz, tendo em conta as normais necessidades de uma presença física na loja. Contudo, especialmente com o desenvolvimento das aplicações *mobile*, encontra-se uma possibilidade de quebrar uma barreira com muitos anos nos estabelecimentos, ou seja, não ser necessário recorrer a um funcionário para obter qualquer tipo de informação comercial. Com o projeto Compras Inteligentes, propomos o desenvolvimento de uma solução IoT (*Internet of Things*), através da combinação de dispositivos móveis e *beacons* (dispositivos que emitem sinais através da tecnologia *bluetooth*), permitindo melhorar a experiência dos utilizadores nos espaços físicos comerciais, reutilizando e apresentando numa nova perspetiva, algumas das vantagens do comércio virtual.

1.2 Objetivos

A aplicação proposta, permitirá ajudar tanto os clientes nas suas compras, através das indicações dos produtos que possam pretender ou daqueles que os rodeiam, como aos comerciantes nas lojas físicas, possibilitando-os dar a conhecer os seus produtos de uma forma mais eficiente através do suporte tecnológico. Também se propõe neste projeto, estudar as utilidades da tecnologia BLE *Beacon* (*Bluetooth Low Energy Beacon*) no âmbito das superfícies comerciais, onde poderá ser inserido demonstrando o potencial dos *beacons* num meio em constantemente crescimento.

2. Formulação do Problema

2.1 Estado da arte

A interação com os *beacons* já é algo bastante utilizado em vários locais com diversos fins, para além do proposto no nosso projeto, dos quais como eventos, aeroportos, educação, escritórios, hospitais, entre outros[2]. Alguns desses exemplos são:

- A *Google* usa por volta de 2000 *beacons* em estações de comboio na Índia. Os viajantes que estejam perto dos *beacons* recebem notificações silenciosamente sobre como conectar à rede *wifi*.
- Outro exemplo de utilização é nos transportes públicos de Luxemburgo, estes têm *beacons* que notificam sobre a direção e próximas paragens do autocarro, bem como o tempo que demora até ao seu destino.
- No aeroporto *The Changi Airport* na Singapura é também utilizada a tecnologia dos *beacons*. Estão a apostar no método "*do it yourself*", que faz com que os utilizadores possam fazer "*self check-in*", identificar a bagagem e verificações de imigração. Isto faz com que o aeroporto se torne mais eficiente e os seus serviços não se comprometem com o aumento do número de passageiros.

No entanto no ano passado houve uma subida drástica de *beacons* "ruidosos", ou seja, *beacons* com informação irrelevante e/ou que faziam *spam* de notificações. Devido a isso, a *Google* terminou o seu serviço *Nearby Notifications*.
Nearby Notifications foi desenhado com o intuito de ajudar os utilizadores do serviço a receber notificações do *beacon*. Não existindo nenhum serviço com os mesmos benefícios, para puder continuar a beneficiar de notificações na aproximação de um *beacon* é preciso a existência de uma aplicação que tenha esse *beacon* registado. [3][4]

2.2 *Use Cases*

No âmbito deste projeto, existem 4 use cases: cinema, loja, restaurante e espaço comercial. Contudo, antes do utilizador conseguir aceder a qualquer das funcionalidades apresentadas, é necessário que realize os seguintes passos:

1) Abrir a aplicação e ligar o *bluetooth*, tal como *wifi* / dados móveis do seu *smartphone*.

2) Aproximar-se da zona do *beacon*.

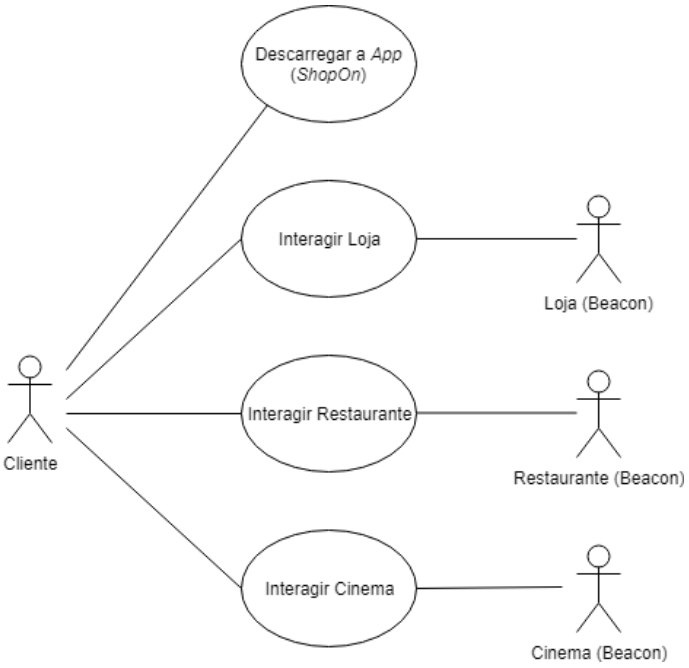


Figura 1- Use Case geral

No cenário do utilizador perder a ligação do *bluetooth* ao sair da zona que contém *beacons* e que por sua vez pertence a um estabelecimento, este continuará a ter acesso às funcionalidades adquiridas anteriormente. No entanto, se perder conexão *wifi* / dados móveis só é possível aceder às informações que já tinham sido permitidas.

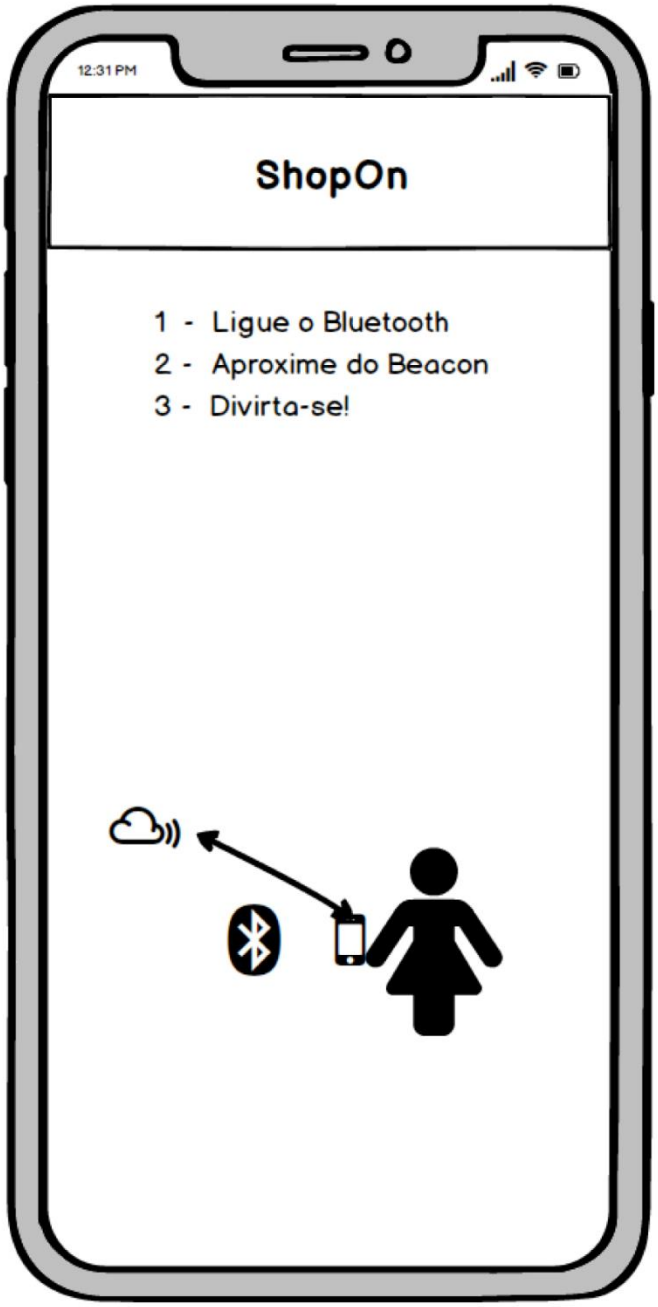


Figura 2- Wireframe HomeScreen

2.2.1 Cinema

O ecrã inicial mostrará a tabela de preços seguido por uma lista de filmes e seus resumos. Todos os itens desta lista irá conter um botão que nos levará a sessões do filme selecionado.

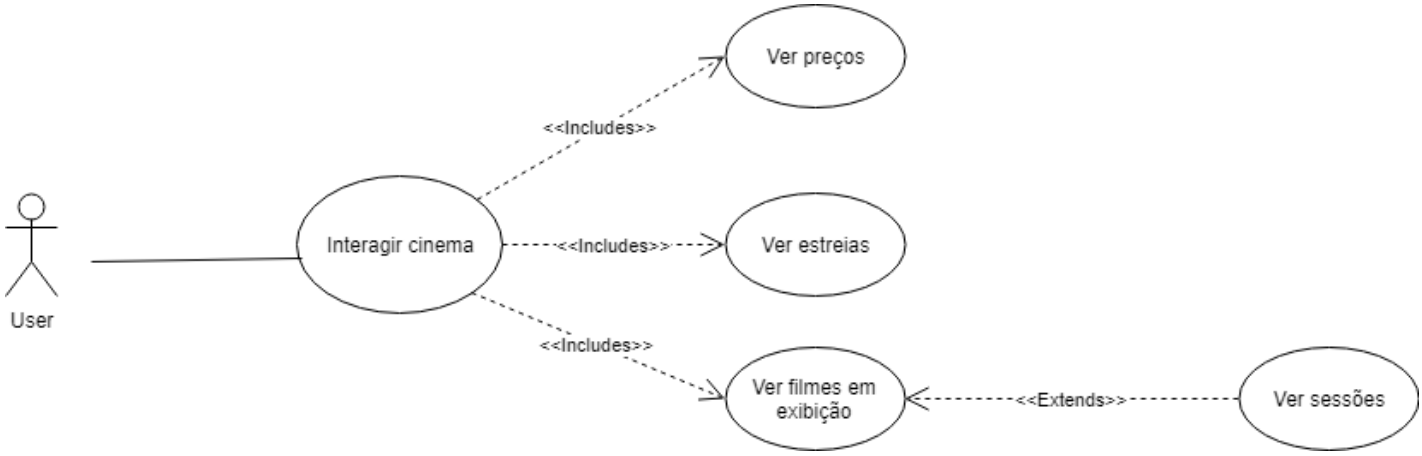


Figura 3- Use Case cinema

* Ver preço:
    - No ecrã inicial do cinema poderá ver o preço dos bilhetes tendo em conta os possíveis descontos ou extras (estudante, 3D, entre outros).

* Ver estreias:
    - No ecrã inicial de cinema serão também apresentadas as últimas estreias que estão em exibição.

* Ver filmes em exibição:
    - O ecrã inicial apresenta todos os filmes que estão em exibição no cinema. Ao selecionar um filme o utilizador tem a opção de ver as sessões desse filme.

* Ver sessões:
    - O utilizador terá um botão em cada item da lista dos filmes que navegará para um ecrã com o horário das próximas sessões.

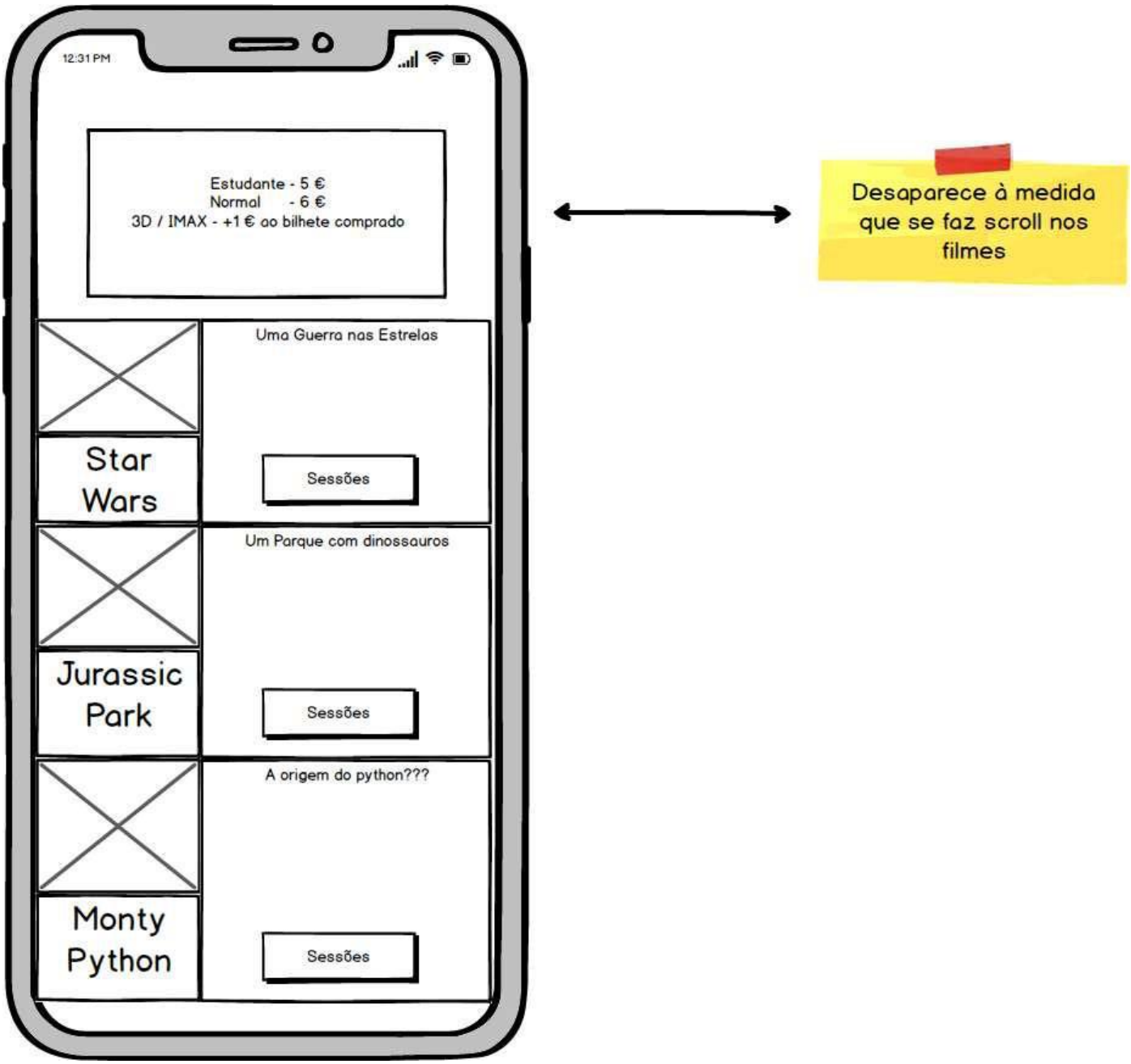


Figura 4- Wireframe cinema

2.2.2 Loja´

Ao interagir com o ecrã será apresentado em maior plano artigos da loja em causa e também as promoções correntes. Em cada artigo estará disponível mais informações, acedidas a partir de botões presentes. É oferecido a hipótese de filtrar por secção, ou seja, caso o utilizador queira visualizar somente artigos de mulher, pode realizar essa filtragem e no ecrã será visível apenas tais artigos.

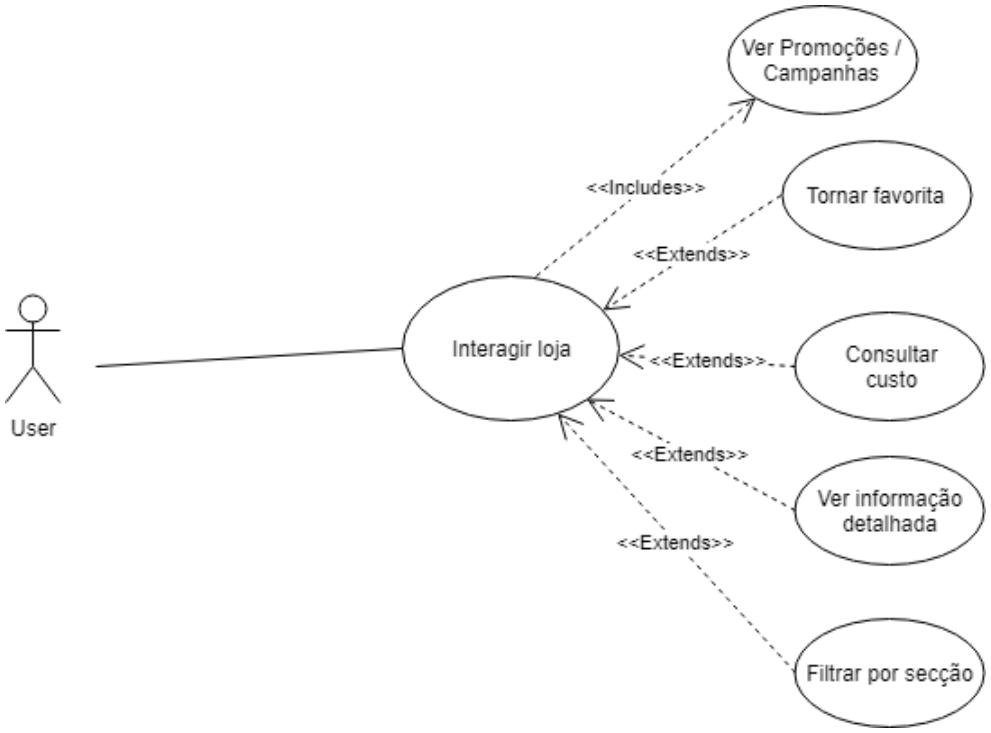


Figura 5-Use Case loja

* Ver Promoções / Campanhas:

- Esta secção do ecrã é dedicada a apresentar ao utilizador informação sobre as promoções/campanhas que existem na loja em questão.

* Tornar favorita:

- Nesta eventualidade, o utilizador poderá selecionar a loja como favorita e como resultado fica disponível a opção de receber notificações relacionadas com o estabelecimento mesmo que a aplicação não esteja em execução.

* Consultar custo:

- Neste ponto, o utilizador ao navegar pelos produtos, caso goste de um produto e faça intenções de comprar, pode adicionar ao "cesto". Desta forma, o utilizador tem acesso ao custo total a pagar.

* Ver informação detalhada:
    - Em cada loja, o utilizador terá a opção de ver detalhes sobre os produtos vendidos, tais como:
        * Descrição
        * Preço
        * Tamanho
        * Cor
        * Características técnicas
        * Categoria
        * *Stock*

* Filtrar por secção:
    - Opção de o utilizador ver produtos que pertencem a uma categoria específica. Por exemplo, numa loja de roupa seria roupa de mulher/homem/criança enquanto que numa loja de tecnologia poderia consultar apenas os produtos de áudio ou vídeo.

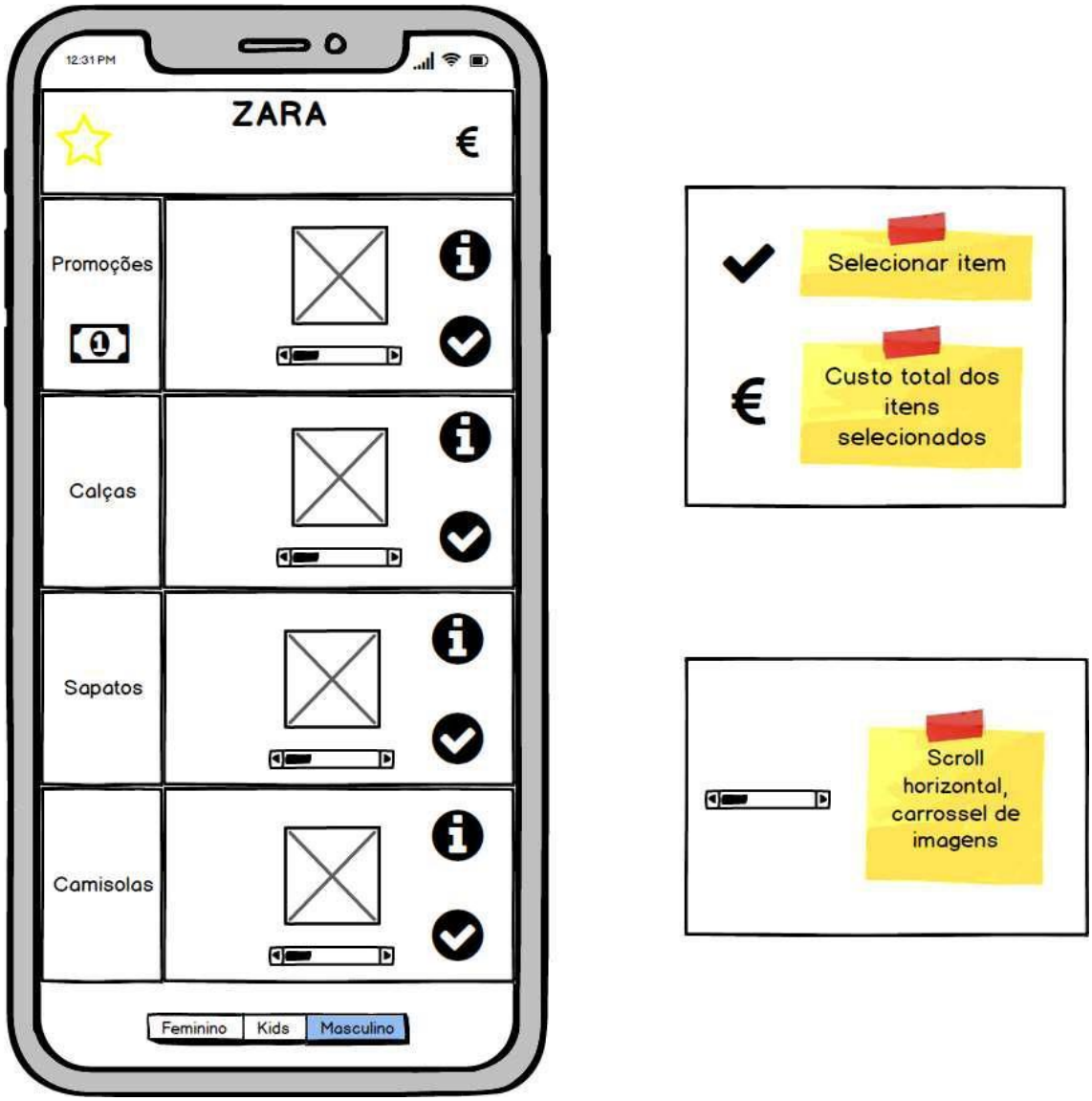


Figura 6- Wireframe loja

2.2.3 Restaurante

O ecrã inicial terá um retângulo com o nome do restaurante e o botão de “Tornar favorito” e o de “Consultar custo”, seguido por uma lista de produtos divididos em secções (prato, bebida, sobremesa e entrada). É permitido a opção de filtrar por intolerâncias.

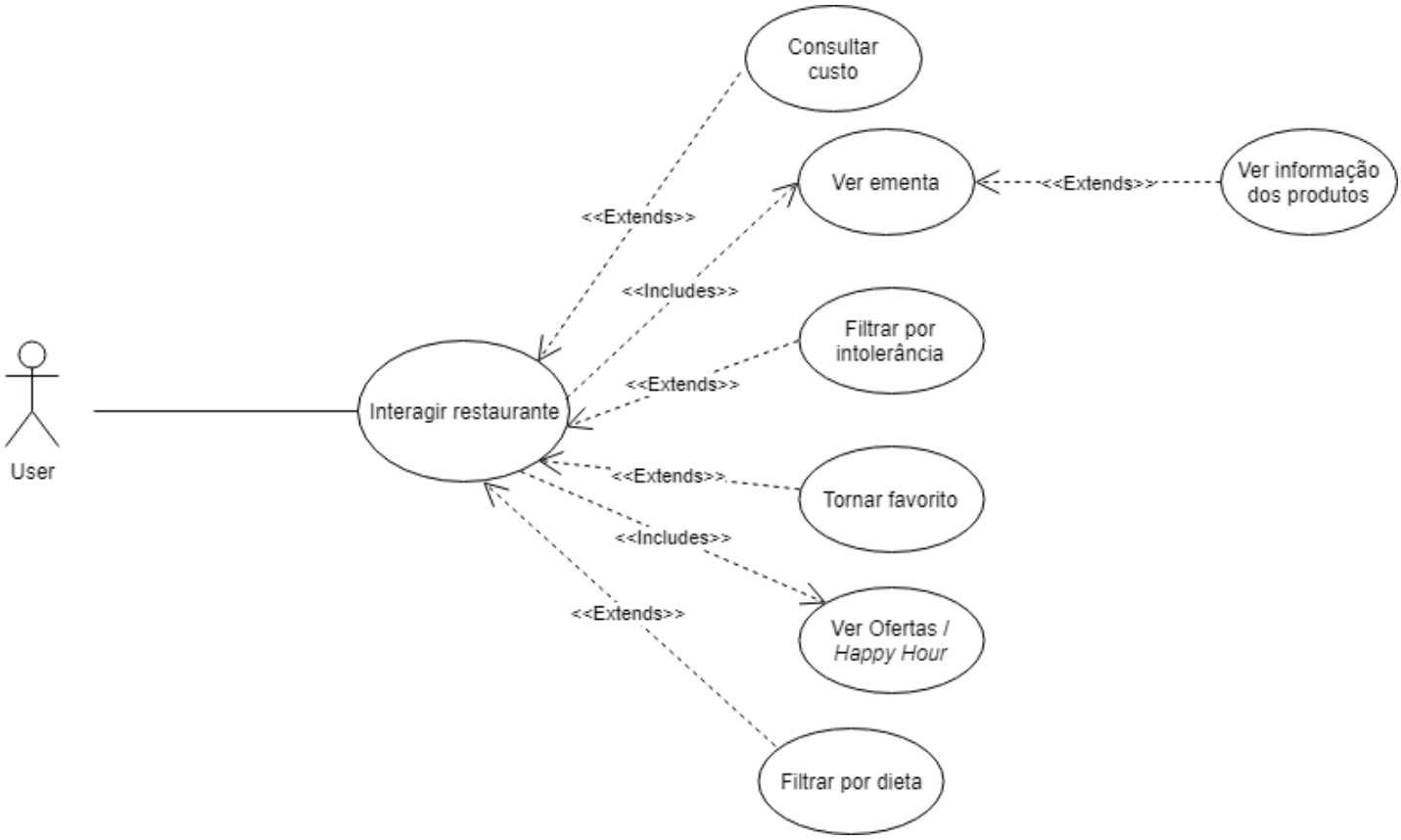


Figura 7- Use Case restaurante

* Consultar custo:

- O utilizador ao consultar produtos, caso goste de um produto e faça intenções de comprar, pode o adicionar ao "cesto". Desta forma, o utilizador tem acesso ao custo total a pagar.

* Ver ementa:

- O ecrã inicial de restaurante apresenta a sua ementa, ou seja, mostra todos os pratos que o restaurante serve divididos pela sua categoria.

* Ver informação dos produtos:

- Aqui o utilizador pode ver informação dos pratos selecionados tais como:

* Descrição
* Preço
* Informação nutricional
* Composição do prato

* Filtrar por intolerância:
    - Nesta situação, o utilizador conseguirá filtrar os pratos por categorias (exemplo: peixe, carne, vegan, entre outros) ou também por alergénicos (glúten, marisco, laticínios, entre outros).

* Ver Ofertas / *Happy Hour*:
    - Aqui temos um local de fácil acesso às promoções ou campanhas correntes da loja.

* Tornar favorito:
    - Utilizador, caso pretenda, selecionará o restaurante como favorito e por consequência fica possível a disponível receber notificações relacionadas com o restaurante mesmo que a aplicação não esteja em execução.

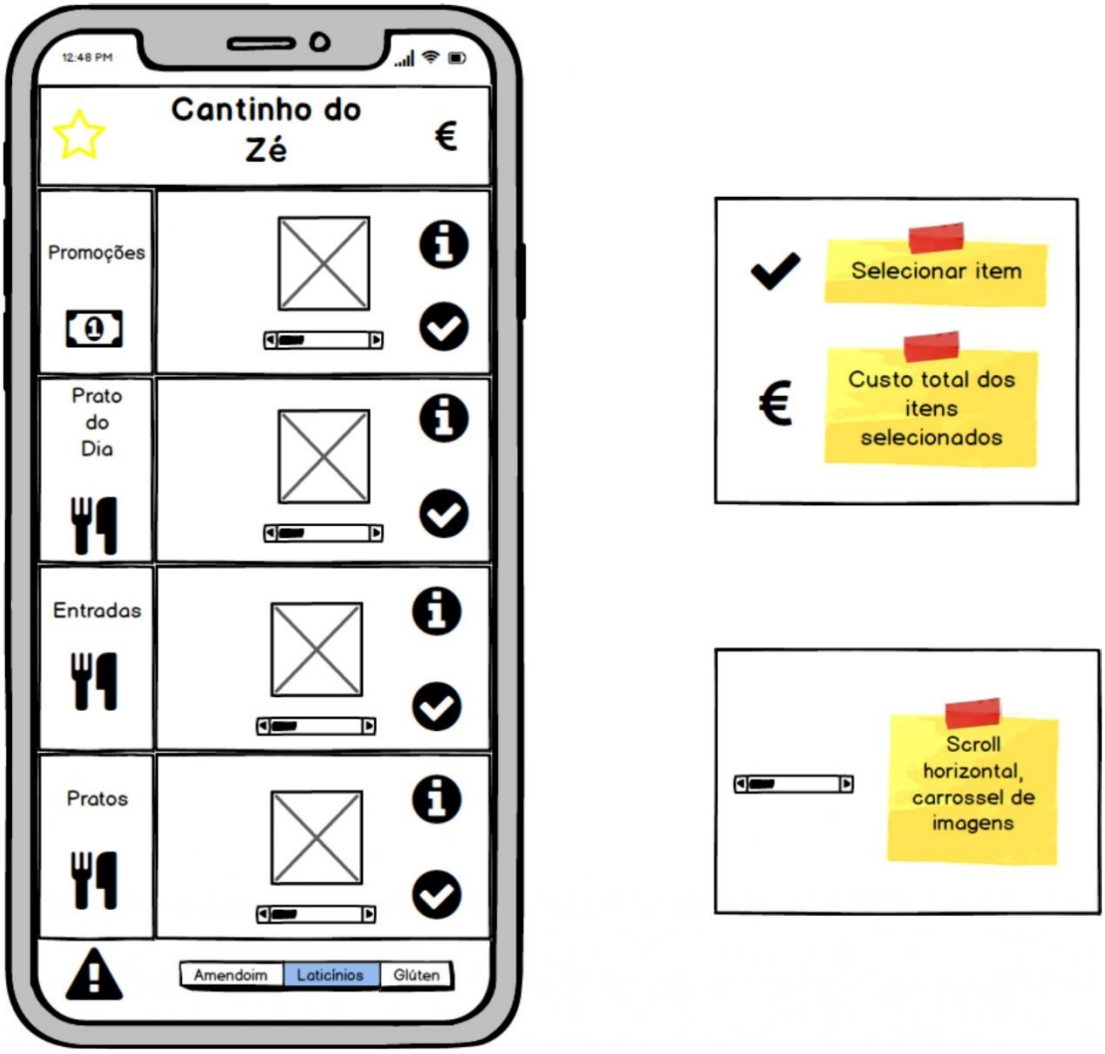


Figura 8- Wireframe restaurante

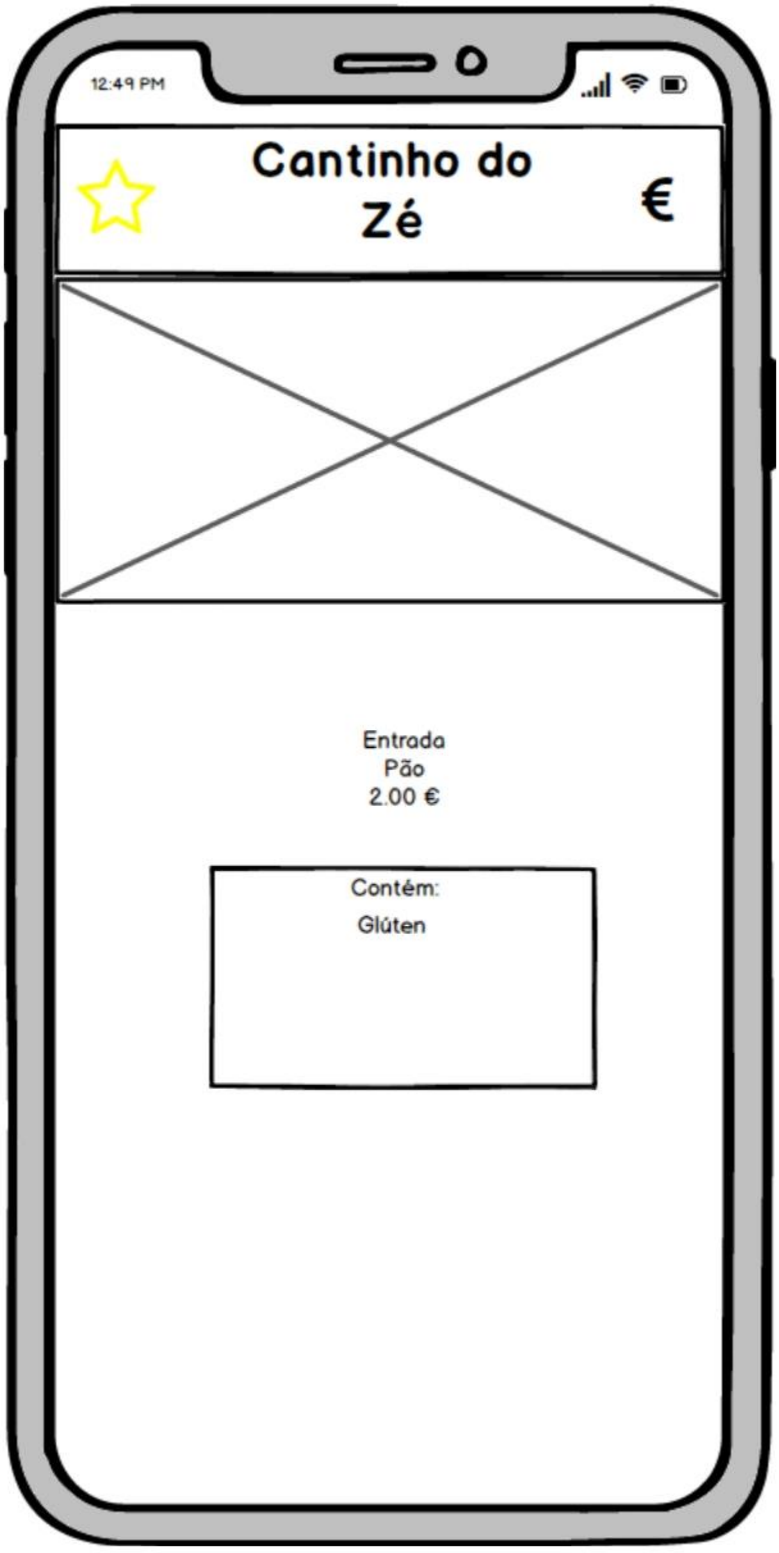


Figura 9- Wireframe produto de restaurante

2.3 Beacons

Os beacons são dispositivos de proximidade que emitem sinais através da tecnologia bluetooth, que os smartphones podem captar quando se encontram dentro da sua área de alcance [5].

O *beacon* que estamos a usar é o "*iBKS 105*" distribuído pela *Accent Systems*. Este *beacon* suporte tanto o protocolo *iBeacon* como o *Eddystone* (em simultâneo!). Para o configurar temos de o conectar com *SDK* "*iBKS Config Tool*", retirando a bateria e só depois é possível fazer a ligação. No painel de configuração temos depois a opção de ativar vários sinais de qualquer de um dos dois protocolos. No entanto detetámos um bug, no *iBeacon* tanto o *major* como o *minor* não conseguimos configurar apesar de no *SDK* dizer que é possível.

Escolhemos o *iBeacon* devido:

- O *workshop* exemplo usava o *iBeacon* https://www.outsystems.com/forge/component-overview/1695/beacons-workshop
- A estrutura "*Beacon*" criada pelo plugin que estamos a usar é de um *iBeacon* devido à presença de *major* e *minor*
- Apesar de ter sido desenvolvido pela *Apple* funciona também nos *smartphones Android*

Maiores dificuldades:

- A aplicação *iBKS* tem um bug nos valores do *minor* e *major* sendo obrigados a usar o "*Beacons Scanner*" para obter os valores reais de *minor* e *major*
- O *uuid* na *OutSystems* para ser detetado tem de estar separado pela seguinte estrutura "12345678-1234-1234-1234-123456781234"

Os dois protocolos mais comuns no uso dos *beacons* são *iBeacon* e *Eddystone*. O *iBeacon* é um protocolo desenvolvido pela Apple em 2013. O sinal emitido por este protocolo é constituído por 3 informações: *uuid* (*Unique Universal Identifier*), *major* e *minor* (sendo a *uuid* a específica do *beacon* enquanto o *major* e o *minor* normalmente é informação extra em relação à região do *beacon*). O *Eddystone* é um protocolo desenvolvido pela *Google*, no entanto, é *open source* tendo recebido colaborações de outras empresas. O protocolo pode enviar 4 pacotes de informação: *uid* (*Unique ID*, é

semelhante ao sinal do *iBeacon*, mas é dividido em *Namespace* e *Instance*), *url*, *tlm* (para propósitos de gerenciamento dos *beacons*) e *eid* (uma medida de segurança)

2.4 Plataforma da Outsystems

O nosso projeto será desenvolvido utilizando a plataforma *OutSystems* [6]. As aplicações móveis geradas pela *OutSystems* podem ser instaladas em multiplataforma (*Android* e *iOS*) devido a serem geradas com recurso à *framework Cordova*.

2.5 Metodologia de desenvolvimento (Scrum)

O *Scrum* é uma metodologia *agile* de desenvolvimento de projetos.

Escolhemos este método devido à facilidade de separar um grande problema em pequenas partes e por ser mais fácil paralelizar o trabalho.

Nesta metodologia o trabalho é dividido por sprints, um sprint é uma etapa de desenvolvimento num intervalo de tempo.

Usamos o *Trello* como *Backoffice* em que temos 4 estados (*ToDo*, *Doing*, *To Review* e *Done*) possíveis de uma tarefa.

O *ToDo* era as tarefas a fazer, o *Doing* é as tarefas a ser já desenvolvidas no presente *sprint*, o *To Review* é tarefas feitas, no entanto que ainda não teve aprovação total da maioria dos membros e o *Done* significa que já está feito e confirmado.

O nosso projeto está dividido em 3 sprint.

No início de cada *sprint* colocamos vários *ToDo* no *Doing*, no caso de ainda existir tarefas no *Doing* do *sprint* passado, tem de ser concluídas primeiramente antes de passar a desenvolver as tarefas do *sprint* seguinte.

A maioria das nossas tarefas correspondem a uma *use case* da nossa aplicação para no final de cada *sprint* termos uma versão testada e funcional que possamos apresentar.

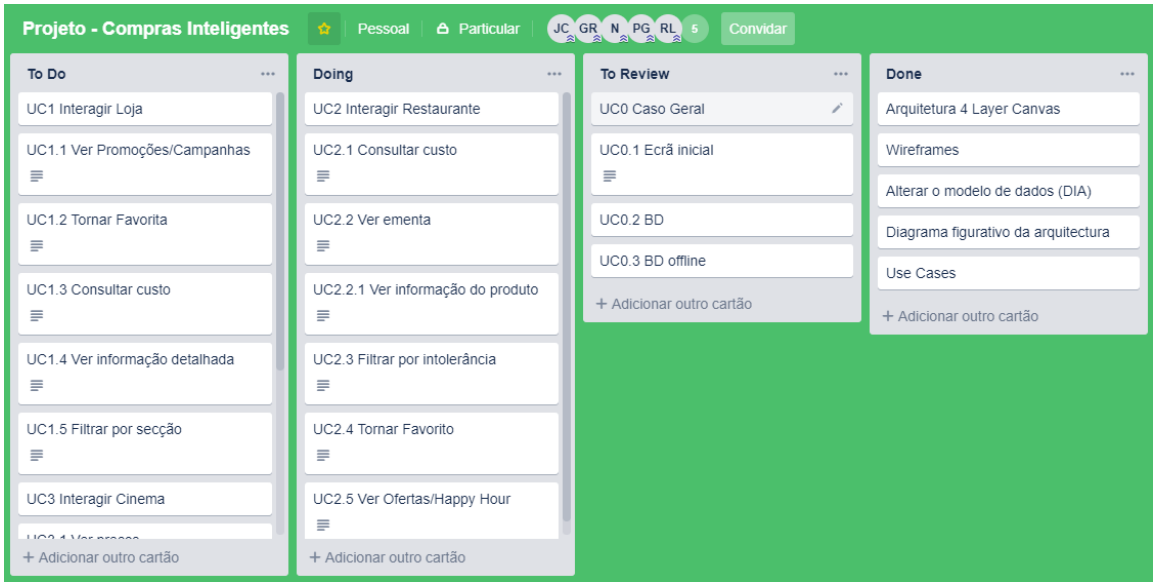


Figura 10- Board Trello (22/04/2019)

3. Solução Proposta

3.1 Arquitetura da solução

3.1.1 Descrição com diagrama figurativo

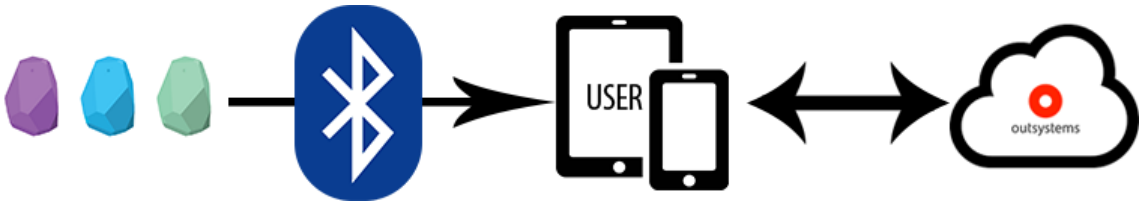


Figura 11- Diagrama figurativo

3.1.1 4 Layer Canvas

Para desenharmos a nossa arquitetura seguimos a metodologia sugerida pela plataforma *OutSystems*, o *4 layer canvas*.

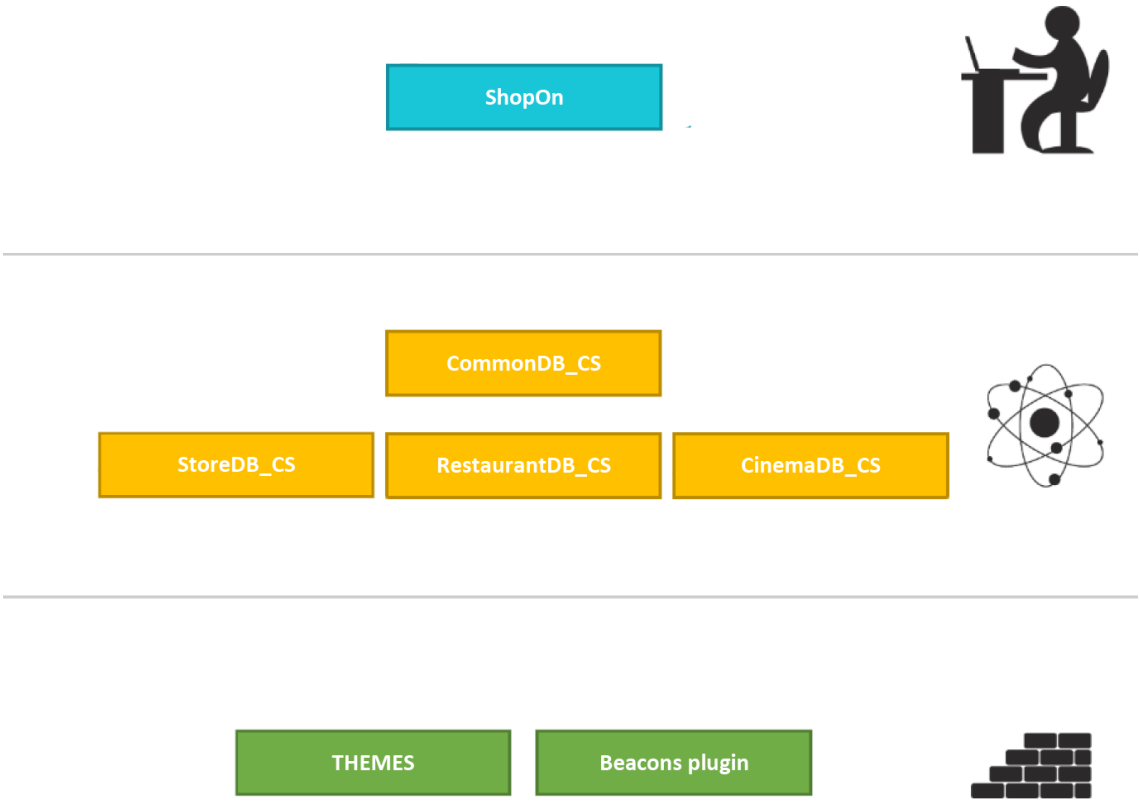


Figura 12- 4 Layer Canvas

Esta metodologia propõe que se estruture as várias funcionalidades da aplicação por quatro camadas, sendo estas:

* *Library Layer* - Aqui devem constar os módulos que são transversais ao domínio do problema, tais como: temas, bibliotecas, etc.

* *Core Layer* - Módulos referentes à lógica de negócio, modelo de dados.

* *End User Layer* - Nesta camada é tratada toda a parte de interface e experiência do utilizador fazendo uso das camadas anteriores.

* *Orchestration Layer* - Camada que coordena a comunicação entre várias aplicações.

De notar que apesar da metodologia apresentar quatro camadas, a nossa arquitetura apenas faz uso das primeiras três já que como o nosso projeto consiste em apenas uma aplicação móvel não necessidade de coordenar interações com outras aplicações.

Sendo assim a nossa solução assenta em sete módulos, representados pela figura 7.

Começando pela *Library Layer*, vemos que é constituída pelo módulo *THEME*, onde é definido o tema visual que suportam os ecrãs da aplicação.

Beacons plugin é a peça que suporta a lógica associada à interação entre o *beacon* e o dispositivo móvel.

De seguida temos o *Core Layer*, onde definimos as entidades de domínio e suas operações.

Por fim a *End User Laye*r onde são definidos os ecrãs e a lógica de cliente.

Referências

[1] “New Insights on Retail E-Commerce” [Online]. Available: https://www.commerce.gov/news/fact-sheets/2017/07/new-insights-retail-e-commerce/. [Accessed: 11-Mar-2019].

[2] "11 Incredible beacon use cases from 2017 every marketer should know in 2018"[Online]. Available:https://blog.beaconstac.com/2017/12/11-incredible-beacon-use-cases-from-2017-every-marketer-should-know-before-2018/. [Accessed: 25-Abr-2019].

[3] "The end of Nearby Notifications and its alternatives - Accent Systems"[Online]. https://accent-systems.com/blog/the-end-of-nearby-notifications-and-its-alternatives [Accessed: 25-Abr-2019].

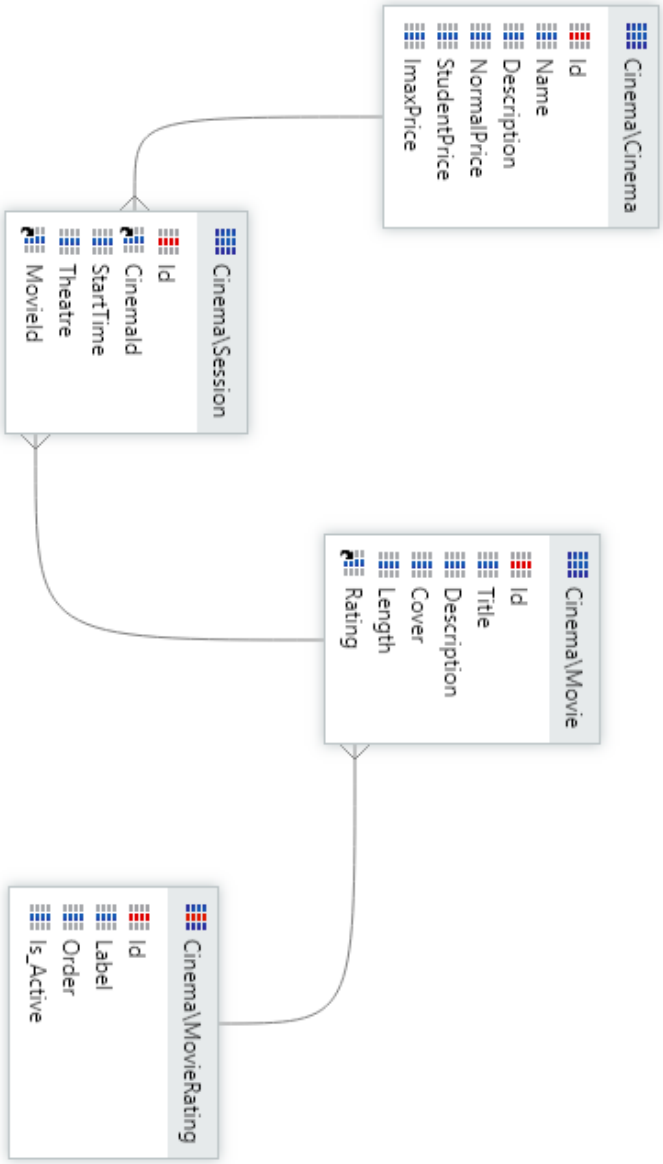
[4] "Android Developers Blog: Discontinuing support for Android Nearby Notifications"[Online].https://android-developers.googleblog.com/2018/10/discontinuing-support-for-android.html[Accessed: 25-Abr-2019].

[5] “Simplifying Enterprise IoT - kontakt.io” [Online]. Available: https://kontakt.io/. [Accessed: 11-Mar-2019].

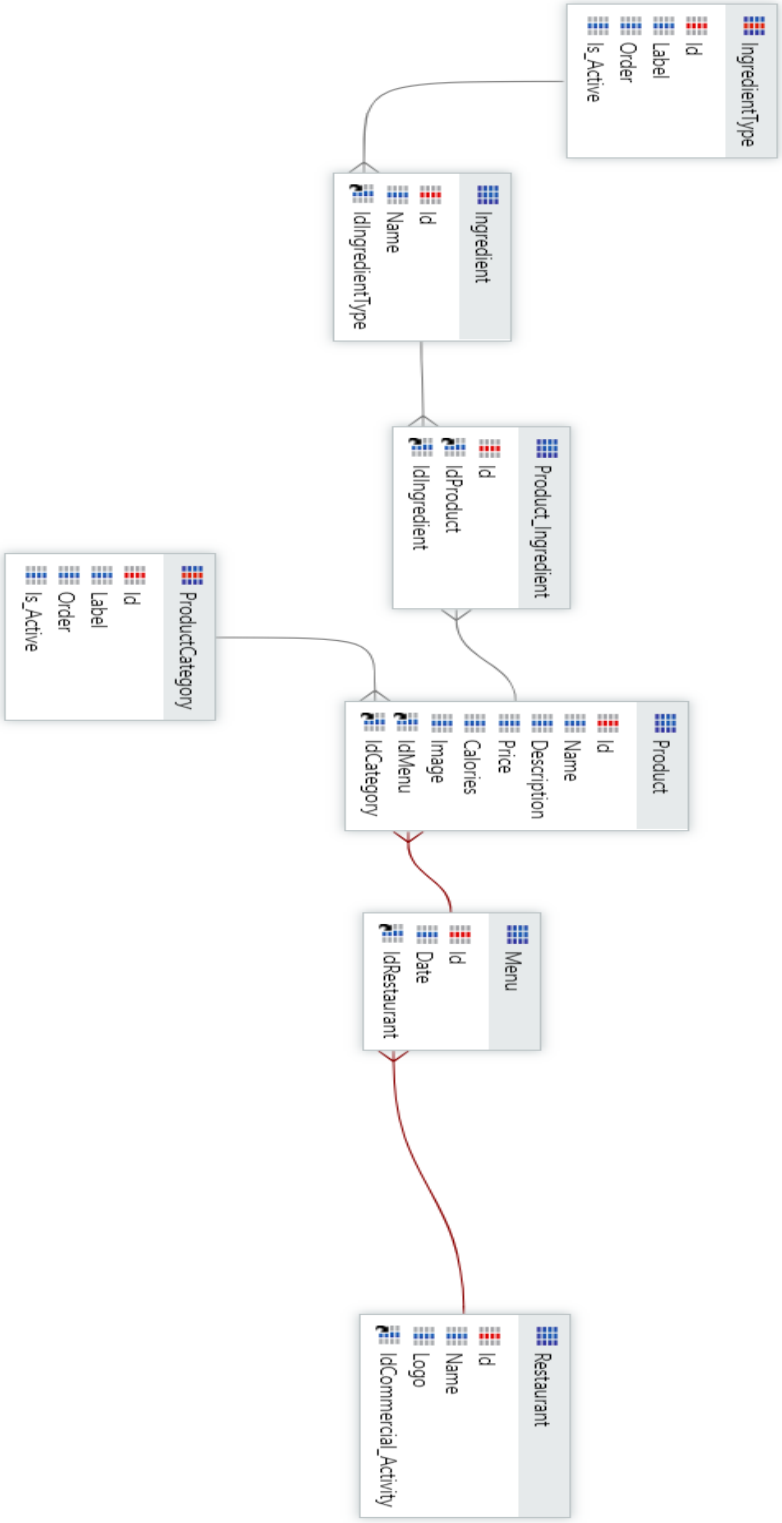
[6] “Low-Code Development Platform for Mobile and Web Apps | OutSystems.” [Online]. Available: https://www.outsystems.com/platform/. [Accessed: 11-Mar-2019].

A.1 Modelos de dados

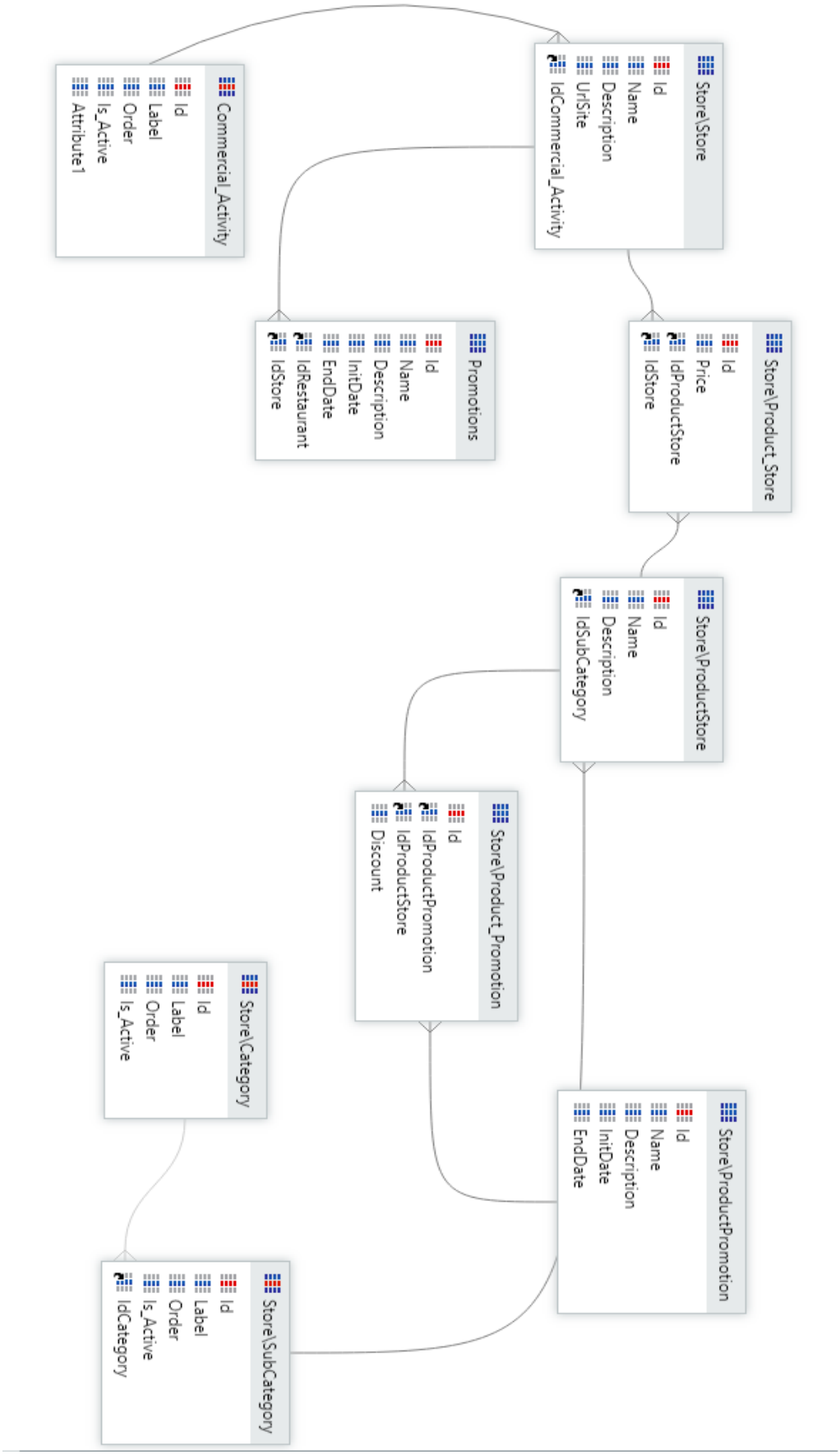
A.1.1 Cinema



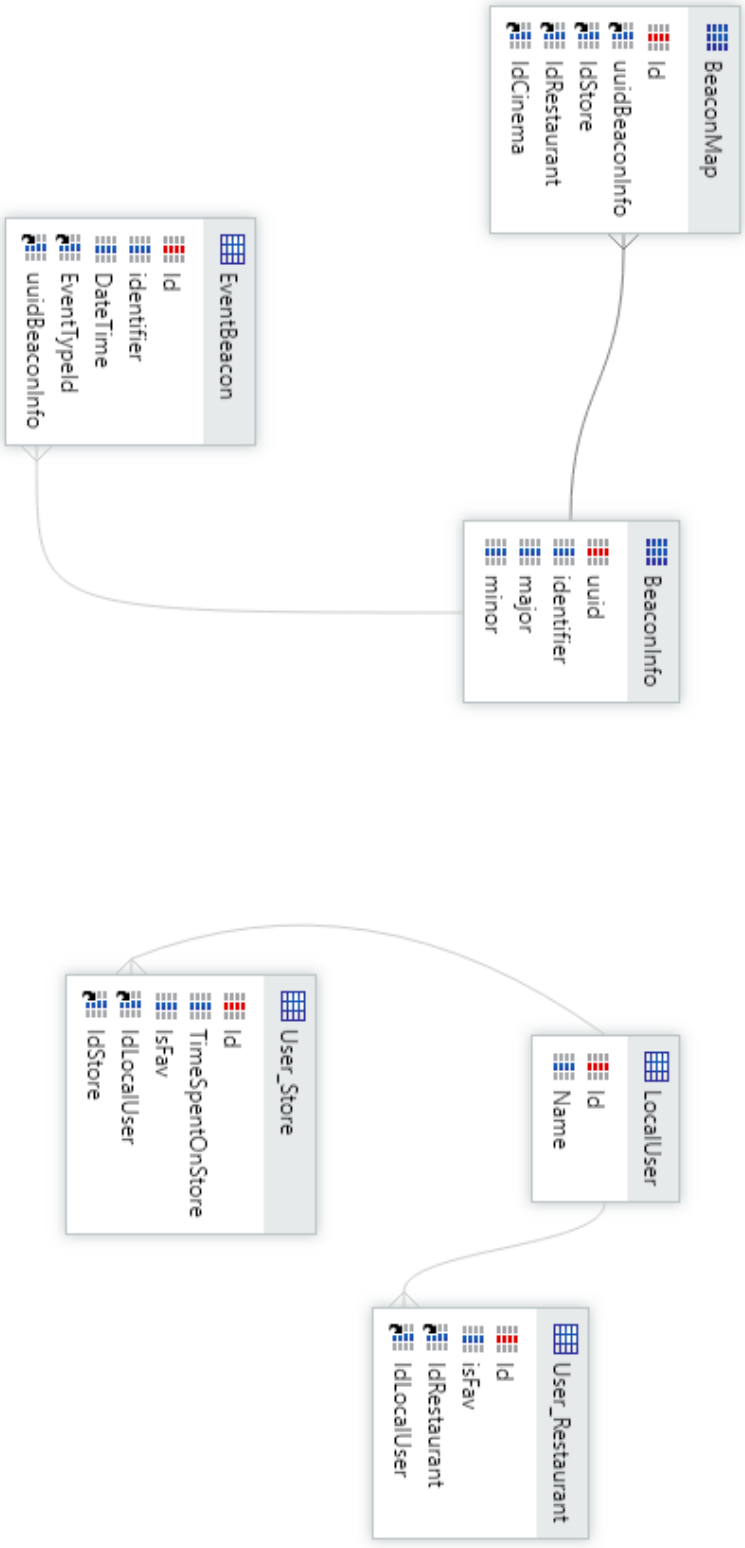
A.1.2 Restaurante



A.1.3 Loja


Store\Store
Id
Name
Description
UrlSite
IdCommercial_Activity
Commercial_Activity
Id
Label
Order
Is_Active
Attribute1
Promotions
Id
Name
Description
InitDate
EndDate
IdStore
IdRestaurant
Store\Product_Store
Id
Price
IdProductStore
IdStore
Store\ProductStore
Id
Name
Description
IdSubCategory
Store\Product_Promotion
Id
IdProductPromotion
IdProductStore
Discount
Store\ProductPromotion
Id
Name
Description
InitDate
EndDate
Store\Category
Id
Label
Order
Is_Active
Store\SubCategory
Id
Label
Order
Is_Active
IdCategory

A.1.4 Comum


BeaconMap
Id
uuidBeaconInfo
IdStore
IdRestaurant
IdCinema
EventBeacon
Id
identifier
DateTime
EventTypeId
uuidBeaconInfo
BeaconInfo
uuid
identifier
major
minor
User_Store
Id
TimeSpentOnStore
IsFav
IdLocalUser
IdStore
LocalUser
Id
Name
User_Restaurant
Id
isFav
IdRestaurant
IdLocalUser