



Universidade do Minho  
Escola de Engenharia  
Licenciatura em Engenharia informática

## Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2021/2022

### Piti Guru

André Gonçalves Pinto 93173  
Augusto César Oliveira Campos 93320  
Rui Pedro Chaves Silva Lousada Alves 93252  
Tiago Pinheiro da Silva 93285

11/2021

Data de Recepção	
Responsável	
Avaliação	
Observações	

## Piti Guru

André Gonçalves Pinto 93173  
 Augusto César Oliveira Campos 93320  
 Rui Pedro Chaves Silva Lousada Alves 93252  
 Tiago Pinheiro da Silva 93285

11/2021

# Resumo

Este relatório é referente ao trabalho que nos foi apresentado no âmbito da unidade curricular de Laboratórios de informática IV, que tem como propósito a realização de um projeto de software referente ao tema “Guia para locais de interesse”, neste caso um guia para restaurantes próximos do utilizador.

Nesta primeira fase de entrega (Fundamentação e especificação) deste projeto efetuamos um levantamento e especificação de requisitos funcionais e não funcionais que serão futuramente essenciais ao desenvolvimento da aplicação.

No desenvolvimento desta mesma fase criamos requisitos que a aplicação deve apresentar aquando da entrega da segunda fase do trabalho a ser executada por um grupo de colegas da unidade curricular. A criação destes requisitos foi feita com vista de que a especificação tivesse funcionalidade similar à esperada por aplicações que já existem no espaço de tecnologia associado.

Em suma, a aplicação será capaz de apresentar restaurantes ordenados por “rating” e preço, bem como apenas expor os restaurantes abertos, restaurantes pouco conhecidos ou então apenas aqueles associados a uma “keyword”.

Uma parte adicional da aplicação é o uso de questionários para melhor moldar o comportamento da mesma ao utilizador. Isto será feito através de “filters” que mudam de forma geral a apresentação dos restaurantes.

**Área de Aplicação:** Restauração.

**Palavras-Chave:** Piti-Guru, Restaurante

# Índice

1	Introdução .....	7
1.1	Contextualização .....	7
1.2	Motivação e objetivos .....	7
1.3	Justificação e utilidade do sistema .....	7
1.4	Estabelecimento da identidade do projeto. ....	7
1.5	Identificação dos recursos necessários. ....	8
1.6	Maqueta do sistema. ....	8
1.7	Definição de um conjunto de medidas de sucesso.....	9
1.8	Plano de desenvolvimento (diagrama GANTT). ....	9
2	Levantamento e Análise de Requisitos.....	10
2.1	Apresentação da estratégia e método. ....	10
2.2	Descrição geral dos requisitos (funcionais e não funcionais) levantados.....	10
	Requisitos Funcionais: .....	10
2.2.1	Mostrar Restaurantes disponíveis num raio <i>default</i> .....	10
2.2.2	Guardar estabelecimento como favorito .....	11
2.2.3	Inquérito ao utilizador .....	11
2.2.4	Mostrar todos os restaurantes associados a uma “keyword”.....	12
2.2.5	Mostrar restaurantes ordenados por “rating”.....	12
2.2.6	Mostrar restaurantes ordenados por preço .....	12
2.2.7	Mostrar apenas restaurantes abertos .....	12
2.2.8	Mostrar restaurantes pouco conhecidos .....	13
2.2.9	Mostrar Informação de um restaurante .....	13
2.2.10	Mostrar Restaurantes disponíveis em um determinado raio.....	13
2.2.11	Pesquisa de restaurante por nome .....	14
2.2.12	Aplicação funcionar offline .....	14
2.2.13	Aceder à localização .....	14
2.2.14	Guardar em Memória as informações da aplicação .....	14
2.2.15	Calcular tempo de viagem entre duas coordenadas geográficas .....	15
2.3	Validação dos Requerimentos estabelecidos. ....	15
3	Especificação e Modelação do Software .....	16

3.1	Diagrama de Domínio .....	16
3.2	Diagrama de Use Case .....	16
3.3	Use cases.....	17
3.3.1	Mostrar Restaurantes disponíveis num raio <i>default</i> .....	17
3.3.2	Guardar estabelecimento como favorito .....	17
3.3.3	Inquérito ao utilizador .....	17
3.3.4	Mostrar todos os restaurantes associados a uma “keyword” .....	18
3.3.5	Use Case: Mostrar restaurantes ordenados por rating .....	18
3.3.6	Mostrar restaurantes ordenados por preço .....	18
3.3.7	Mostrar apenas restaurantes abertos .....	18
3.3.8	Mostrar restaurantes pouco conhecidos .....	19
3.3.9	Mostrar Informação de um restaurante .....	19
3.3.10	Mostrar Restaurantes disponíveis em um determinado raio.....	19
3.3.11	Pesquisa por nome do Restaurante .....	19
4	Conceção do Sistema de Dados .....	20
4.1	Base de Dados .....	20
4.2	Modelo Lógico .....	21
4.3	Derivação e Relacionamentos .....	21
5	Esboço dos Interfaces do Sistema. ....	22
5.1	Página Questionário .....	22
5.2	Página 'Página Inicial' .....	23
5.3	Página 'Restaurantes Favoritos' .....	24
5.4	Página 'Procura Restaurantes' .....	25
5.5	Menu 'Definições de Procura' .....	26
5.6	Página 'Informações Gerais' .....	27
6	Conclusões e Trabalho Futuro.....	28
6.1	Conclusões.....	28
6.2	Trabalho Futuro .....	28
7	Referências Bibliográficas.....	29

# Índice de Imagens

<b>Figura 1 – Maqueta do Sistema.....</b>	<b>8</b>
<b>Figura 2 - Diagrama GANTT .....</b>	<b>9</b>
<b>Figura 3 - Diagrama de Domínio .....</b>	<b>16</b>
<b>Figura 4 - Diagrama de Use Case .....</b>	<b>16</b>
<b>Figura 5 - Modelo Logico da Base de Dados.....</b>	<b>21</b>
<b>Figura 6 - Página de Questionário .....</b>	<b>22</b>
<b>Figura 7 - Página Inicial .....</b>	<b>23</b>
<b>Figura 8 - Página Restaurantes Favoritos .....</b>	<b>24</b>
<b>Figura 9 - Página Procura Restaurantes.....</b>	<b>25</b>
<b>Figura 10 - Menu Definições de Procura.....</b>	<b>26</b>
<b>Figura 11 - Página Informações Gerais.....</b>	<b>27</b>
<b>Figura 12 - Diagrama GANTT (Trabalho Futuro) .....</b>	<b>29</b>

# 1 Introdução

## 1.1 Contextualização

A gastronomia é, em geral, o conjunto das práticas culturais relacionadas com a alimentação que representam um importante papel na sociedade.

Cada vez mais, na atual sociedade, os restaurantes são preferidos à alimentação em casa, quer devido à elevada carga horária da classe trabalhadora e à variada oferta na alimentação, como também devido à rapidez e qualidade dos serviços prestados.

Por este motivo e tendo em conta a elevada procura pela restauração, decidimos implementar uma aplicação que, além de guiar o utilizador até aos seus restaurantes favoritos e obter os respetivos rankings, irá facilitar a escolha do mesmo através das preferências do utilizador.

## 1.2 Motivação e objetivos.

A Piti Guru é uma empresa de software que tem a sua atenção dirigida para a criação da aplicação de software de mesmo nome, que é na sua essência um guia para restaurantes. Sendo constituída pelos seus 4 membros fundadores e estudantes de engenharia informática, que decidiram aproveitar esta oportunidade que na visão deles existe no espaço de software relacionado à restauração.

## 1.3 Justificação e utilidade do sistema

No mercado há uma deficiência de aplicações em que, na nossa opinião, a “User Experience” é o centro da mesma, com esta aplicação pretendemos revolucionar isso e criar um “diário de bordo”, um ajudante que não só recomenda quais são os restaurantes que te possam interessar, mas que também te ajuda nessa mesma escolha.

As atuais aplicações que têm finalidades semelhantes limitam-se apenas a indicar qual ou quais os melhores restaurantes de interesse para o público em geral, na nossa visão uma aplicação que vá além disto e que melhor se adapta ao “user” individual é uma necessidade do espaço de software no qual esta aplicação se enquadra.

## 1.4 Estabelecimento da identidade do projeto.

**Nome:** *Piti Guru*

**Categoria :** Restauração

**Faixa Etária:** Maiores de 16

**Breve descrição:** A aplicação possibilita que o utilizador possa, não só encontrar perto da sua localização, restaurantes em Braga, como também obter recomendações com base nos tipos de comida favoritos do utilizador. Tudo isto funciona sem o acesso à Internet. A aplicação será desenvolvida inicialmente para *Android*, podendo no futuro ser expandida para iOS.

**Criadores:** André Pinto, Augusto Campos, Rui Alves e Tiago Silva.

## 1.5 Identificação dos recursos necessários.

O planeamento dos recursos necessários é uma tarefa que tem de ser bem planeada, caso contrário pode facilmente atrasar bastante o desenvolvimento do projeto. Desta forma é essencial usar os recursos disponíveis da melhor forma possível.

Será necessária uma equipa de pelo menos 4 engenheiros de software.

Requer-se que a equipa tenha alguém experiente em desenvolver uma boa interface gráfica, alguém capaz de manusear e desenvolver uma robusta base de dados, mas principalmente que tenham todos uma boa comunicação entre equipa.

O “backend” da aplicação deve ser desenvolvido para guardar toda a informação possível sobre todos os restaurantes de Braga, sendo que os desenvolvedores podem usar qualquer método para atingir este objetivo.

Para o desenvolvimento da API e do “front-end” da aplicação não será imposta nenhuma restrição pois os desenvolvedores devem usar as ferramentas e as línguas de programação que mais se adequam ao trabalho.

## 1.6 Maqueta do sistema.

Planeamos uma maqueta do sistema bastante simples como é possível visualizar na imagem.

O usuário pode visualizar quais são os restaurantes perto de si em uma determinada área.



Figura 1 – Maqueta do Sistema



## 1.7 Definição de um conjunto de medidas de sucesso.

De forma a avaliar o sucesso da nossa aplicação foram estabelecidas algumas medidas de sucesso.

Para alcançar um grande e variado número de utilizadores, a Piti Guru irá, inicialmente, fazer um pequeno investimento publicitário, esperando alcançar 500 utilizadores no seu primeiro mês de vida e aspirando os 100.000 downloads no fim do seu primeiro ano.

Acompanhando o sucesso da aplicação, pretendemos atualizar e melhorar o produto através de uma equipa técnica organizada e capaz, trabalhando sempre em prol da evolução da aplicação, assegurando sempre velocidade e fluidez em todas as interações com o utilizador.

Esta aplicação, tendo fins lucrativos, irá permitir anúncios publicitários dos mais variados produtos e empresas, e no futuro para a monetização da aplicação criará uma plataforma para desenvolver possíveis parcerias com empresas relativas à restauração e/ou restaurantes, por exemplo a integração de serviços como “Uber Eats” ou “Glovo”.

## 1.8 Plano de desenvolvimento (diagrama GANTT).

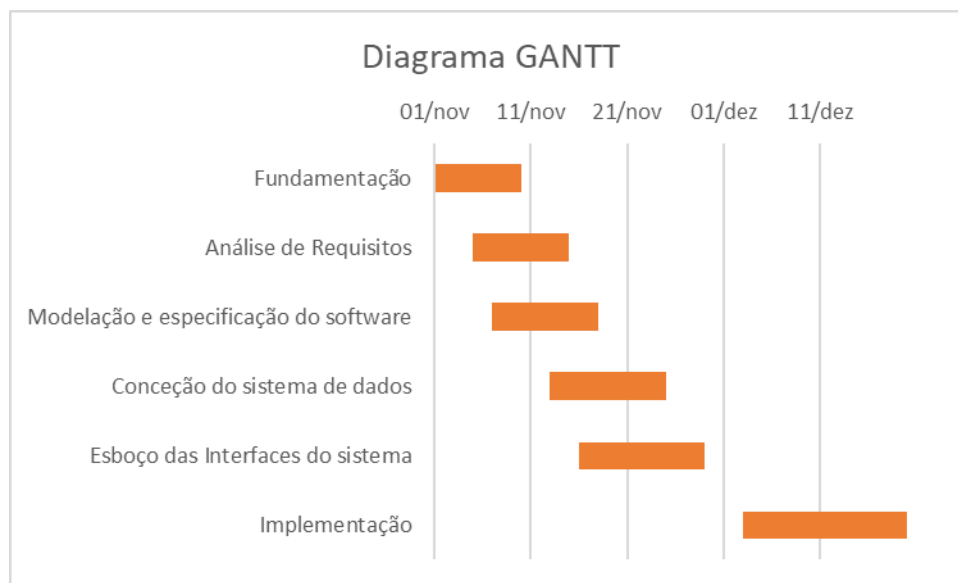


Figura 2 - Diagrama GANTT

## **2 Levantamento e Análise de Requisitos**

### **2.1 Apresentação da estratégia e método.**

O levantamento de requisitos é uma fase extremamente importante para, futuramente, facilitar a implementação da aplicação, garantir o seu bom funcionamento e, também, garantir que se comporta da forma que foi visionado pelos criadores da aplicação.

Para este processo de levantamento de requisitos foram efetuadas pesquisas sobre aplicações com funcionalidades similares procurando perceber de que forma é apresentada a informação ao utilizador, prevenindo assim falhas e erros na aplicação.

De maneira a obter informação e opiniões sobre a implementação de certas funcionalidades no sistema assim como novas funcionalidades que futuros utilizadores gostariam de ver implementadas, a nossa equipa, presente no evento “Portugal RestauranteWeek”, realizou vários inquéritos a possíveis utilizadores, mas também a possíveis patrocinadores desta aplicação.

A partir da análise dos dados obtidos de todos os inquéritos realizados, recolhemos informação crucial para estabelecer a lista de requisitos da aplicação e desta forma melhor adequar a mesma à realidade do espaço de software em que esta será futuramente inserida.

### **2.2 Descrição geral dos requisitos (funcionais e não funcionais) levantados.**

#### **Requisitos Funcionais:**

#### **2.2.1 Mostrar Restaurantes disponíveis num raio *default***

Esta funcionalidade será o requisito mais abrangente no que se trata da procura de restaurantes de interesse, as pesquisas posteriores deverão ser sempre feitas com o “output” desta funcionalidade de forma que o espaço de procura dos restaurantes seja o mínimo possível, mas que mantenha a relevância para o utilizador, isto ajudará a manter a velocidade e leveza da aplicação. A aplicação terá um raio por defeito mas que poderá ser futuramente modificado.

#### **Requisitos do utilizador:**

- O utilizador pretende ver que restaurantes estão perto de si em uma determinada área, cujo raio deve ser um valor “default”.
- A localização do utilizador tem de já estar partilhada com a aplicação.

#### **Requisitos da aplicação:**

- A aplicação mostra graficamente onde se situam esses restaurantes dependendo da posição do utilizador.

### 2.2.2 Guardar estabelecimento como favorito

Para o melhoramento da “User Experience” nós idealizamos uma funcionalidade da aplicação que guardaria estabelecimentos favoritos que funcionaria como “waypoints” (Um “waypoint” é um ponto de referência que pode ser usado para localização e navegação), de modo que as suas posições geográficas possam ser usados na aplicação.

#### Requisitos do utilizador:

- O utilizador pretende guardar um estabelecimento como favorito,

#### Requisitos da aplicação:

- O sistema consegue suportar a funcionalidade de o utilizador clicar num estabelecimento/restaurante e guardá-lo como favorito.
- A aplicação guarda quais as preferências do utilizador.

### 2.2.3 Inquérito ao utilizador

Este requisito disponibiliza uma série de questões/opções que ajudam a aplicação a moldar-se ao usuário, este inquérito deverá ser devidamente preenchido pelo utilizador caso este não o faça o sistema não irá aplicar qualquer restrição sobre os estabelecimentos apresentados.

Este inquérito será apresentado ao utilizador na primeira vez que a aplicação seja iniciada e posteriormente numa outra página caso o utilizador queira responder a algo que não tenha respondido.

Idealmente isto seria um algoritmo de “machine learning” que modificaria o comportamento da aplicação, mas para simplificar e diminuir o tempo de desenvolvimento da mesma esperamos algo que se comporte como “filters”, alguns absolutos (se um utilizador é vegan a aplicação não deve mostrar restaurantes sem opções vegan) e outros parciais (se um utilizador prefere ementas picantes os restaurantes com opções de ementas picantes devem ter prioridade sobre os de ementas não picantes no entanto, estes, podem ainda aparecer nas recomendações ).

#### Requisitos do utilizador:

- O utilizador tem de inserir o seu nome, o qual será guardado para o identificar.
- O utilizador pretende ajudar a aplicação a melhor entender que tipo de restaurantes devem ser recomendados.

#### Requisitos da aplicação:

- Apresenta um questionário sobre preferências tais como se prefere restaurantes veganos, churrascarias, fast-food ou tascos, que nível monetário o utilizador pretende (Caro, Médio, Barato), obtém possíveis restrições alérgicas do usuário.

**Sugestão:** *Os desenvolvedores podem ajustar o inquérito como acharem melhor. Os parâmetros do inquérito apresentados podem ser alterados.*

## **2.2.4 Mostrar todos os restaurantes associados a uma “keyword”**

Esta funcionalidade deverá modificar a apresentação da procura de restaurantes exibindo apenas aqueles que têm a “keyword” escolhida associada.

(As “keywords” associadas aos estabelecimentos são baseadas nas disponibilizadas pela API do Google, no entanto podem ser abordadas de uma maneira diferente).

### **Requisitos do utilizador:**

- O utilizador tem uma preferência que quer procurar em todos os restaurantes disponíveis.

### **Requisitos da aplicação:**

- A aplicação faz uma procura tendo em conta todos os restaurantes possíveis em um certo raio associados a uma “keyword”.
- Se a procura não retornar nenhum restaurante é pedido ao utilizador que reescreva outra “keyword” na procura.

## **2.2.5 Mostrar restaurantes ordenados por “rating”**

Esta funcionalidade deverá modificar a apresentação da procura de restaurantes ordenando-os por “rating”.

### **Requisitos do utilizador:**

- O utilizador pretende ver os restaurantes ordenados pelo seu “rating”.

### **Requisitos da aplicação:**

- Após efetuar a procura são apresentados os restaurantes ordenados pela sua “rating”.
- A aplicação deverá representar por ordem crescente ou decrescente.

## **2.2.6 Mostrar restaurantes ordenados por preço**

Esta funcionalidade deverá modificar a apresentação da procura de restaurantes ordenando-os por preço.

### **Requisitos do utilizador:**

- O utilizador pretende ver os restaurantes ordenados pelo seu preço.

### **Requisitos da aplicação:**

- Após efetuar a procura são apresentados restaurantes ordenados pelo seu nível de preço.

## **2.2.7 Mostrar apenas restaurantes abertos**

Esta funcionalidade deverá modificar a apresentação da procura de restaurantes exibindo apenas aqueles que se encontram abertos.

Esta funcionalidade terá que utilizar o horário de funcionamento do estabelecimento uma vez que a aplicação também irá funcionar offline.

**Requisitos do utilizador:**

- O utilizador pretende ver os restaurantes abertos.

**Requisitos da aplicação:**

- Tendo a procura feita apresenta apenas restaurantes abertos.

## **2.2.8 Mostrar restaurantes pouco conhecidos**

Esta funcionalidade deverá modificar a apresentação da procura de restaurantes exibindo aqueles que têm um “rating” alto, mas com poucas avaliações.

**Requisitos do utilizador:**

- O utilizador pretende visualizar os restaurantes com poucas avaliações.

**Requisitos da aplicação:**

- A aplicação organiza os resultados da pesquisa pelos restaurantes com poucas, mas altas avaliações primeiro.

## **2.2.9 Mostrar Informação de um restaurante**

Uma das funcionalidades que prevemos na aplicação é a capacidade de apresentar as informações gerais de um restaurante. Esta informação deve ser apresentada quando se clica num dos restaurantes na página de procura.

**Requisitos do utilizador:**

- O utilizador pretende ver informação adicional sobre um restaurante.

**Requisitos da aplicação:**

- Ser capaz de mostrar as informações gerais de um restaurante a partir do nome, como o seu nível de preço, a sua classificação entre outras características.
- Mostrar o tempo de viagem do utilizador até ao respetivo restaurante.

## **2.2.10 Mostrar Restaurantes disponíveis em um determinado raio**

Esta funcionalidade permite ao usuário alterar o raio de pesquisa de restaurantes caso este não queira utilizar o raio “default” de procura.

**Requisitos do utilizador:**

- O utilizador pretende alterar o raio de procura da pesquisa de restaurantes.

**Requisitos da aplicação:**

- A aplicação consegue suportar uma alteração do raio da pesquisa na procura de restaurantes.

### **2.2.11 Pesquisa de restaurante por nome**

**Requisitos do utilizador:**

- O utilizador pretende procurar um restaurante específico dando o seu nome.

**Requisitos da aplicação:**

- A aplicação faz uma procura tendo em conta todos os restaurantes presentes na base de dados.

## **Requisitos Não Funcionais:**

### **2.2.12 Aplicação funcionar offline**

A aplicação tem de funcionar sem acesso à internet, sendo que a informação necessária para o seu funcionamento existirá numa base de dados pré-populada, assim sendo não vai ser possível obter informação sobre todos restaurantes existentes, neste caso serão apenas restaurantes da cidade de Braga.

### **2.2.13 Aceder à localização**

A aplicação necessita de saber a localização atual do utilizador. Todas as pesquisas sobre restaurantes devem ser feitas com o ponto inicial sendo a posição do utilizador.

Para tal, caso o utilizador recuse permitir acesso à localização a aplicação deve encerrar devidamente.

### **2.2.14 Guardar em Memória as informações da aplicação**

Após o término da aplicação esta deve guardar informações como locais favoritos, preferências do utilizador e outras informações que se tornem relevantes, e deve também carregar novamente esta mesma informação no arranque do sistema.

Se esta informação não existir o programa deve iniciar com dados “default” que não devem influenciar a forma como são procurados os restaurantes de interesse.

**Requisitos da aplicação:**

- A aplicação deve guardar a informação existente temporariamente durante o “run time” da mesma para que, posteriormente possa ser carregada quando for iniciada novamente.

### **2.2.15 Calcular tempo de viagem entre duas coordenadas geográficas**

A aplicação tem de ser capaz de obter a distância entre dois pontos (coordenadas geográficas) de forma que sempre que for necessária essa informação ela possa ser obtida, por exemplo na apresentação das informações gerais do restaurante ou quando este é apresentado no mapa.

## **2.3 Validação dos Requerimentos estabelecidos.**

Estes requisitos todos sofreram bastantes alterações ao longo do desenvolvimento da especificação.

A equipa rapidamente notou que das especificações iniciais algumas eram demasiado complexas, tal como a implementação de “text-to-speech” e converter uma morada em coordenadas geográficas.

Foi removido também o requisito e use case de efetuar login e criação de conta pelo utilizador.

Desta forma os requisitos aqui propostos são apenas os essenciais para o desenvolvimento eficiente da aplicação.

## 3 Especificação e Modelação do Software

### 3.1 Diagrama de Domínio

Para representar o funcionamento do sistema desenvolvemos um diagrama de Domínio. Este contém todas as entidades presentes e necessárias ao funcionamento da aplicação assim como os seus respetivos atributos.

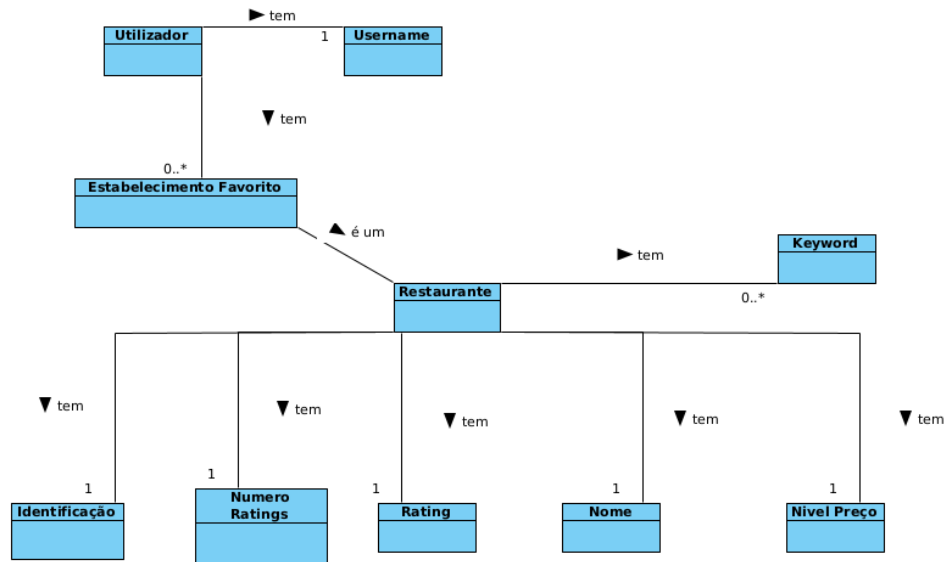


Figura 3 - Diagrama de Domínio

### 3.2 Diagrama de Use Case

Para representar o funcionamento do sistema desenvolvemos um diagrama de Use Case. Este reproduz a interação do utilizador com a aplicação.

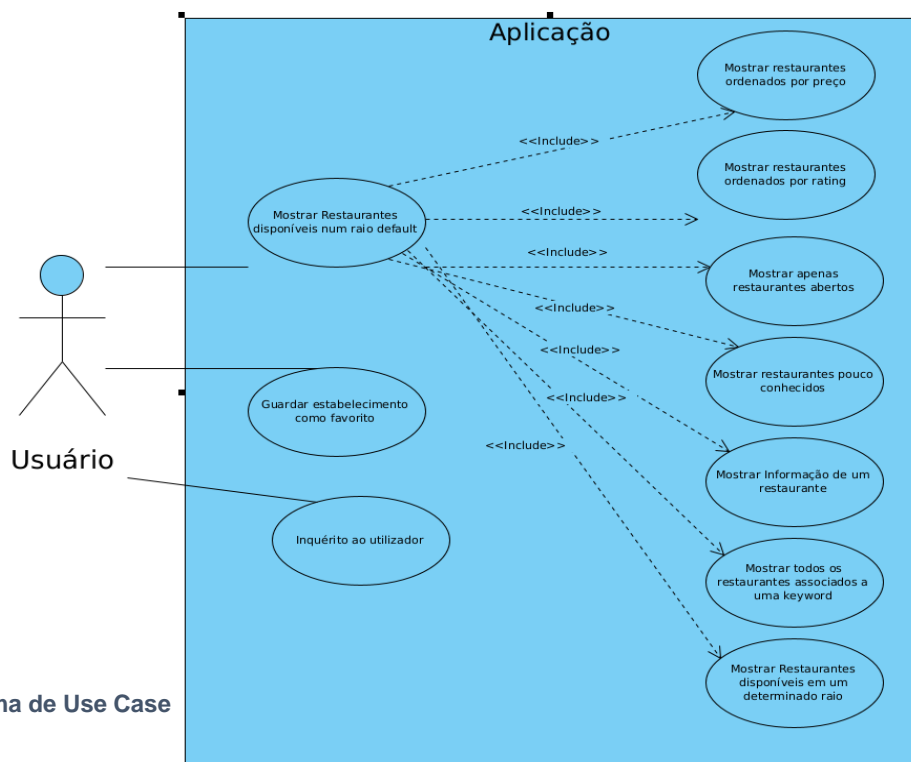


Figura 4 - Diagrama de Use Case



### 3.3 Use cases

#### 3.3.1 Mostrar Restaurantes disponíveis num raio *default*

Descrição: A aplicação mostra restaurantes disponíveis num determinado raio.

Pré-Condição: GPS está ativado

Pós-Condição: A Aplicação apresenta os respetivos restaurantes

Fluxo Normal:

- 1) Sistema procura na base de dados todos os restaurantes dentro do determinado raio
- 2) Sistema mostra graficamente a localização dos restaurantes
- 3) Sistema mostra a distância em KM a pé do utilizador até cada um dos restaurantes
- 4) <<include>> **3.10 Mostrar Restaurantes disponíveis em um determinado raio**
- 4) <<include>> **3.4 Mostrar todos os restaurantes associados a uma keyword**
- 5) <<include>> **3.5 Mostrar restaurantes ordenados por rating**
- 6) <<include>> **3.6 Mostrar restaurantes ordenados por preço**
- 7) <<include>> **3.7 Mostrar apenas restaurantes abertos**
- 8) <<include>> **3.8 Mostrar restaurantes pouco conhecidos**

Fluxo de Exceção(1) - Não existem restaurantes nas proximidades (Passo 1)

- 1.1) Sistema não consegue mostrar nenhum restaurante nas proximidades
- 1.2) Sistema encerra

#### 3.3.2 Guardar estabelecimento como favorito

Descrição : Sistema guarda um estabelecimento preferido do utilizador

Pré-Condição: True

Pós-Condição: A Aplicação guarda o respetivo local

Fluxo Normal:

- 1) O utilizador pretende guardar um estabelecimento como favorito
- 2) Sistema guarda o local como favorito

#### 3.3.3 Inquérito ao utilizador

Descrição : Sistema apresenta inquérito sobre preferências alimentares ao utilizador

Pré-Condição: True

Pós-Condição: Utilizador responde ao inquérito

Fluxo Normal:

- 1) Sistema apresenta inquérito ao utilizador
- 2) Utilizador responde
- 3) Sistema guarda respostas

Fluxo Alternativo(1) – utilizador não responde ao inquérito (Passo 2)

- 2.1) Utilizador envia inquérito incompleto
- 2.2) Aplicação preenche campos por defeito
- 2.3) Regresso ao passo 3

### **3.3.4 Mostrar todos os restaurantes associados a uma “keyword”**

Descrição : Sistema mostra restaurantes associados a uma keyword

Pré-Condição: True

Pós-Condição: Aplicação mostra os respetivos restaurantes

Fluxo Normal:

- 1) Sistema apresenta "Textbox" para inserir “keyword” ao utilizador
- 2) Utilizador insere “keyword”
- 3) Sistema apresenta restaurantes

Fluxo de Exceção (1) - Nenhum restaurante encontrado (Passo 3)

- 3.1) Sistema não encontra nenhum restaurante ao qual corresponda a “keyword”
- 3.2) Nada acontece na visualização dos restaurantes

### **3.3.5 Use Case: Mostrar restaurantes ordenados por rating**

Descrição: Sistema mostra restaurantes ordenados pela seu “rating”

Pré-Condição: Utilizador pretende visualizar os respetivos restaurantes

Pós-Condição: Aplicação mostra os respetivos restaurantes

Fluxo Normal:

- 1) Utilizador seleciona visualizar restaurantes ordenados pela seu rating
- 2) Aplicação procura na base de dados todos os restaurantes
- 3) Ordena os restaurantes encontrados pelo preço

### **3.3.6 Mostrar restaurantes ordenados por preço**

Descrição: Sistema mostra restaurantes ordenados por preço

Pré-Condição: Utilizador pretende visualizar os respetivos restaurantes

Pós-Condição: Aplicação mostra os respetivos restaurantes

Fluxo Normal:

- 1) Utilizador seleciona visualizar restaurantes ordenados por preço
- 2) Aplicação procura na base de dados todos os restaurantes
- 3) Ordena os restaurantes encontrados pelo preço

### **3.3.7 Mostrar apenas restaurantes abertos**

Descrição: Sistema mostra restaurantes abertos

Pré-Condição: Utilizador pretende visualizar os respetivos restaurantes

Pós-Condição: Aplicação mostra os respetivos restaurantes

Fluxo Normal:

- 1) Utilizador seleciona apresentar apenas restaurantes abertos
- 2) Aplicação procura na base de dados todos os restaurantes
- 3) Sistema apresenta apenas os restaurantes abertos atualmente

### **3.3.8 Mostrar restaurantes pouco conhecidos**

Descrição : Sistema apresenta restaurantes com poucas avaliações

Pré-Condição: Utilizador pretende visualizar os respetivos restaurantes

Pós-Condição: Aplicação mostra os respetivos restaurantes

Fluxo Normal:

- 1) Utilizador seleciona apresentar apenas restaurantes abertos
- 2) Aplicação procura na base de dados todos os restaurantes
- 3) Sistema apresenta apenas os restaurantes abertos atualmente

### **3.3.9 Mostrar Informação de um restaurante**

Descrição : Sistema apresenta informação adicional sobre um restaurante

Pré-Condição: Utilizador pretende visualizar a informação

Pós-Condição: Aplicação mostra a respetiva informação

Fluxo Normal:

- 1) Utilizador seleciona o restaurante
- 2) Sistema procura na base de dados informação sobre o restaurante
- 3) Aplicação apresenta informação

### **3.3.10 Mostrar Restaurantes disponíveis em um determinado raio**

Descrição: A aplicação mostra restaurante disponíveis num raio inserido pelo utilizador

Pré-Condição: GPS está ativado

Pós-Condição: A Aplicação apresenta os respetivos restaurantes

Fluxo Normal:

- 1) Utilizador insere raio
- 2) Sistema procura na base de dados todos os restaurantes
- 3) Sistema mostra graficamente a localização dos restaurantes

Fluxo Alternativo (1) - Utilizador insere raio menor que 0 ou valor invalido (Passo 1)

- 1.1) Utilizador insere raio menor que 0
- 1.2) Regresso ao passo 1

Fluxo de Exceção (1) - Não existem restaurantes nas proximidades (Passo 2)

- 2.1) Sistema não consegue mostrar nenhum restaurante nas proximidades
- 2.2) Nada acontece na visualização dos restaurantes

### **3.3.11 Pesquisa por nome do Restaurante**

Descrição: Utilizador pesquisa restaurantes por nome

Pré-Condição: Utilizador pretende visualizar a informação

Pós-Condição: Aplicação mostra todos os restaurantes associados a esse nome

Fluxo Normal:

- 1) Utilizador insere um nome para procurar

- 2) Sistema procura na base de dados todos os restaurantes associados a esse nome
- 3) Aplicação mostra todos os restaurantes disponíveis

Fluxo de Exceção (1) - Não existem restaurantes associados a esse nome (Passo 3)

- 3.1) Sistema não encontra nenhum restaurante associado a esse nome
- 3.2) Sistema informa o utilizador que não existem resultados
- 3.3) Nada acontece na visualização dos restaurantes

## 4 Conceção do Sistema de Dados

### 4.1 Base de Dados

Com a implementação desta base de dados pretendemos armazenar toda a informação necessária à utilização da aplicação, permitindo assim funcionar sem acesso à internet. Todos os atributos presentes foram baseados na API (“application programming interface”) do google e nos dados que esta nos disponibiliza.

- **Utilizador:** Sendo necessário associar um *user* a um local/estabelecimento favorito, achamos essencial guardar algum identificador do utilizador, neste caso utilizamos o seu nome, o qual é requisitado pela aplicação.
- **Estabelecimentos:** Cada estabelecimento terá de possuir um nome, e apresentar informação como nível de preço, o seu rating, número total de ratings e também “keywords” associadas ao mesmo.

## 4.2 Modelo Lógico

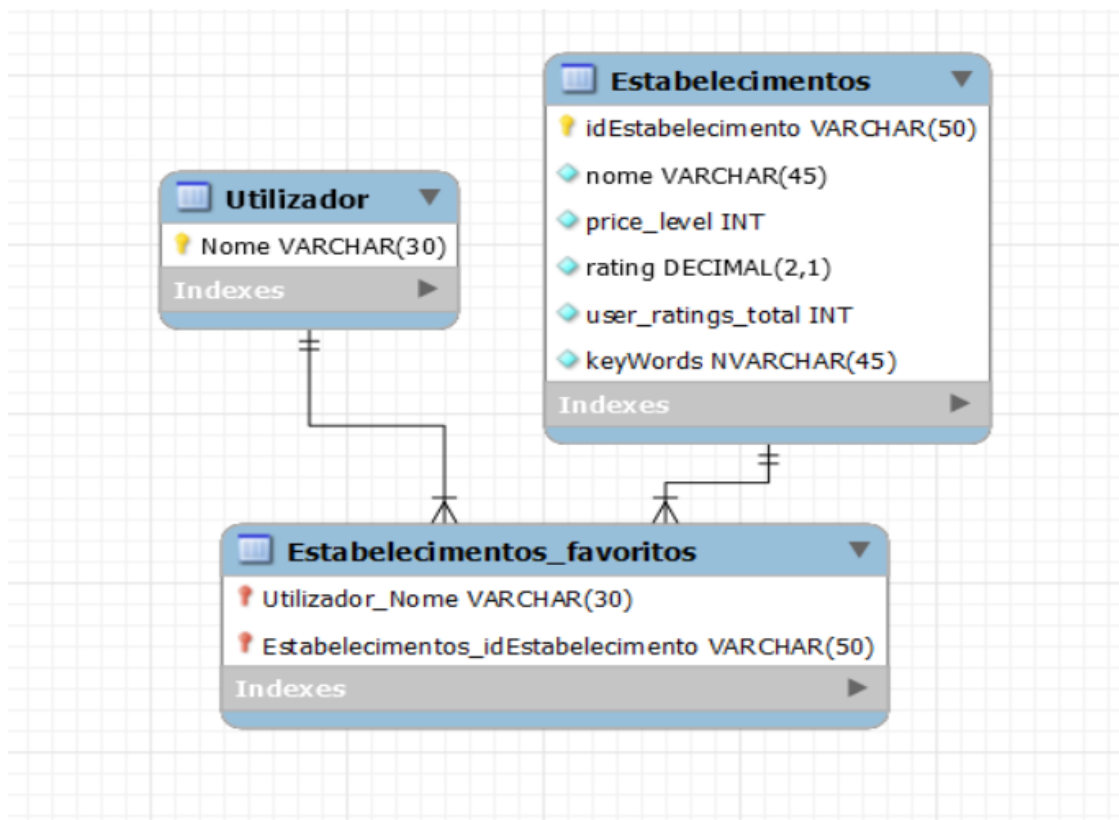


Figura 5 - Modelo Logico da Base de Dados

## 4.3 Derivação e Relacionamentos

- **Entidades Fortes**

Entidade que possui uma chave primária, e não depende de nenhuma outra.

- Utilizador (primary key: 'Nome')
- Estabelecimento (primary key: 'idEstabelecimento')

- **Entidades Fracas**

Entidade que não possui uma chave primária, e depende de entidades fortes.

- Não temos nenhuma entidade fraca presente na nossa base de dados.

- **Relacionamento N:N**

Os relacionamentos de tipo N:N ocorrem quando vários registros de uma tabela se relacionam a vários registros de outra, ou seja, em nenhum dos lados existe exclusividade.

- Utilizador: Estabelecimento

Em qualquer relacionamento N:N é criada uma terceira tabela para relacionar ambas as entidades, neste caso adicionamos a tabela "Estabelecimentos\_favoritos", na qual um utilizador pode ter vários estabelecimentos adicionados como favoritos, assim como vários estabelecimentos podem ser classificados como favoritos.

## 5 Esboço dos Interfaces do Sistema.

Os “mockups” apresentados servem de guias para o desenvolvimento da “GUI” da aplicação e não precisam ser seguidos rigorosamente. Os desenvolvedores devem fazer as alterações que acharem necessárias.

Na primeira utilização da aplicação esta irá abrir na página dos questionários para que se possa usar as respostas aos questionários durante o "runtime" da mesma. Posteriormente a aplicação deverá iniciar na página 'Página Inicial'.

### 5.1 Página Questionário



The mockup shows a smartphone screen with a status bar at the top displaying the time 9:31 and signal icons. The app interface has two tabs: 'Página Inicial' (highlighted in red) and 'Questionário'. The 'Questionário' page contains a list of questions, each followed by a text input field:

- Gosta de comida Picante ?
- Tempo de serviço mínimo?
- Descobrir Sítios Pouco conhecidos?
- Prefere Peixe ou Carne?
- Prefere Quente ou Frio?

Below these questions is a section titled 'Restrições' (Restrictions) with three items, each with a checkbox:

- Vegetariano ☐
- Glúten ☒
- Amendoins ☐

At the bottom of the page is a red button labeled 'Mudar Nome' (Change Name).

Será feito um questionário ao utilizador numa fase inicial para descobrir as suas preferências e/ou restrições alimentares que serão tidas em conta para gerar uma sugestão de restaurante a visitar.

Nesta mesma página será possível mudar o nome do utilizador que deverá ser em princípio um nome criado por defeito.

Figura 6 - Página de Questionário

## 5.2 Página 'Página Inicial'

Após a apresentação dos questionários e a configuração geral da aplicação, é apresentada aquela que consideramos ser a página principal e, após a primeira abertura da aplicação a inicial.

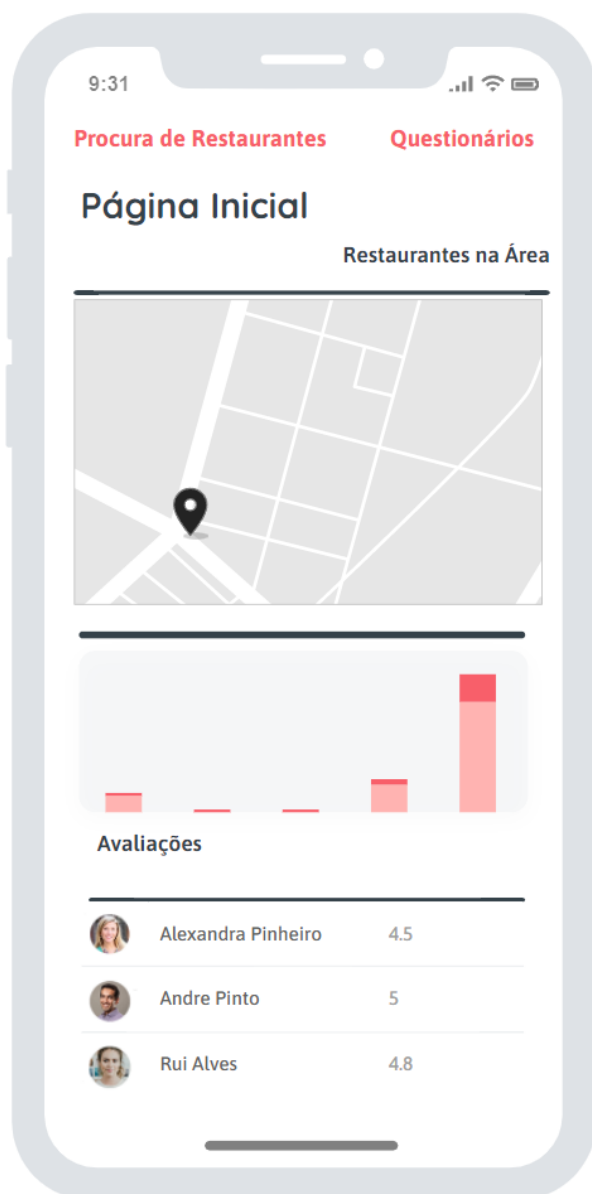


Figura 7 - Página Inicial

Nesta página é possível acessar a 'Procura de Restaurantes' bem como a página 'Questionários' clicando no topo esquerdo e direito respetivamente.

Será também apresentado um mapa da área próxima ao utilizador e possíveis restaurantes de interesse, bem como uma breve apresentação de um desses restaurantes que tiver em 'foco'.

Possivelmente poder-se-á aceder à página 'Informações Gerais' deste restaurante em 'foco'.

Ao fazer "Swipe Right" deve aparecer a página 'Restaurantes Favoritos' onde se encontrará uma lista de restaurantes que foram marcados como favoritos.

## 5.3 Página 'Restaurantes Favoritos'

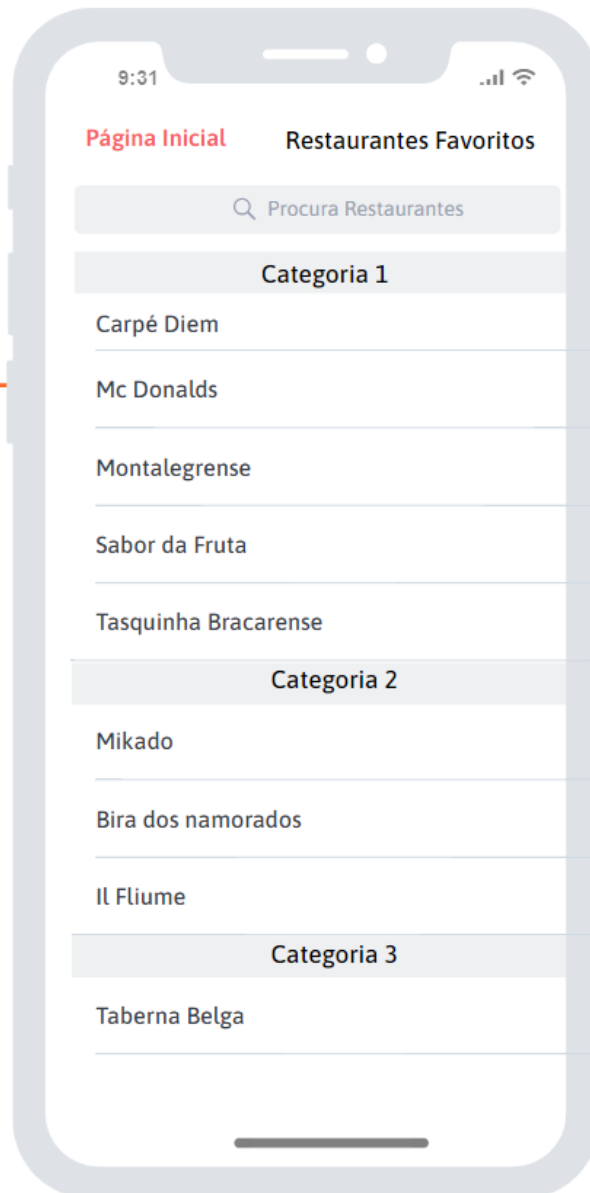


Figura 8 - Página Restaurantes Favoritos

Nesta página será possível aceder à página 'Página Inicial' clicando no topo esquerdo, ou então fazendo "Swipe Left".

Ao clicar num dos restaurantes apresentados esse deve-se tornar o restaurante em 'foco' e a aplicação deve retornar a página 'Página Inicial'.

Ao inserir o nome de um restaurante na barra de procura deve ser feita uma busca pelo mesmo usando o requisito de procura por nome.



## 5.4 Página 'Procura Restaurantes'

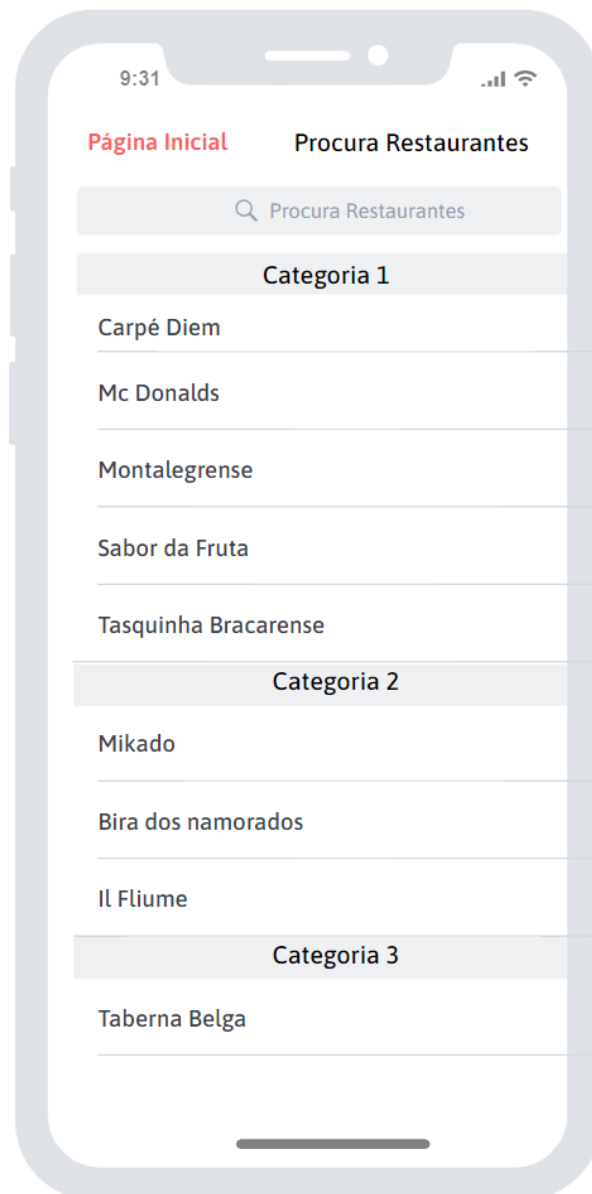


Figura 9 - Página Procura Restaurantes

Nesta página será possível aceder à página 'Página Inicial' clicando no topo esquerdo.

Ao fazer "Swipe Left" deve aparecer o submenu de 'Definições de Procura'.

Ao clicar num dos restaurantes apresentados a aplicação deve abrir a página 'Informações gerais' deste restaurante.

Ao inserir o nome de um restaurante na barra de procura deve ser feita uma busca pelo mesmo usando o requisito de procura por um nome.

## 5.5 Menu 'Definições de Procura'

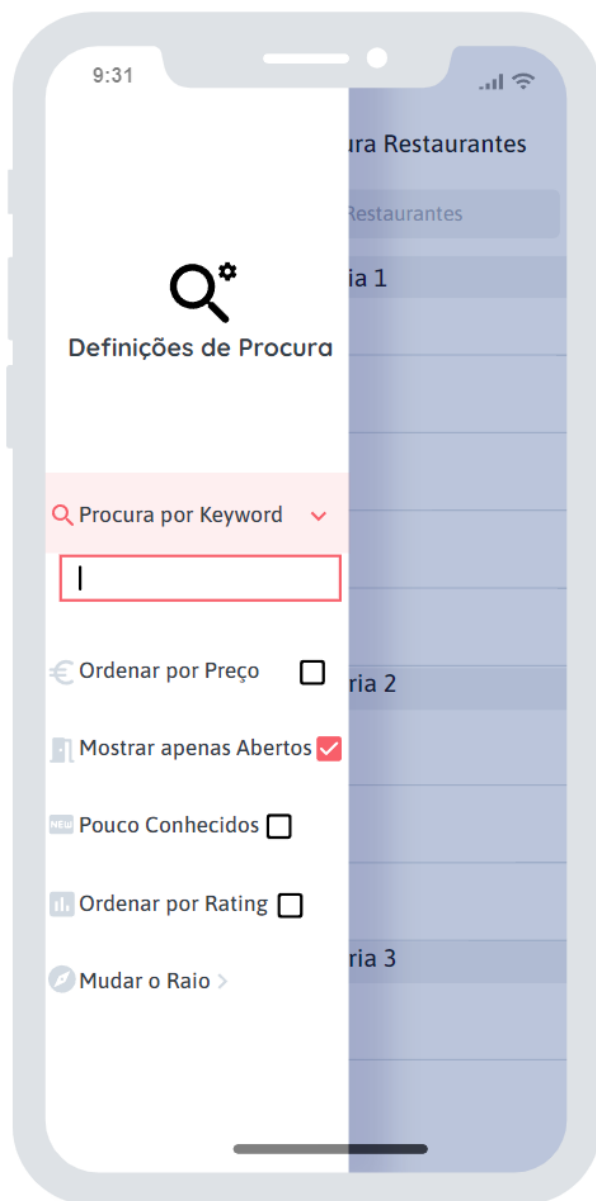


Figura 10 - Menu Definições de Procura

Esta será uma subpágina que mudará a apresentação da página 'Procura Restaurantes'.

Ao fazer "Swipe Right" deve voltar à página 'Procura Restaurantes'.

Aqui serão utilizados a maioria dos requisitos da aplicação que darão funcionalidade à procura de restaurantes.

No caso das opções 'Procura por Keyword' e 'Mudar Raio' deve ser apresentado um "DropDown Menu" que mudará a apresentação da página 'Procura Restaurantes'.

No resto das opções deverão ser definições de "True or False" se a opção estiver seleccionada a mesma é aplicada e muda a apresentação da página 'Procura Restaurantes' se não estiver nada acontece.

## 5.6 Página 'Informações Gerais'

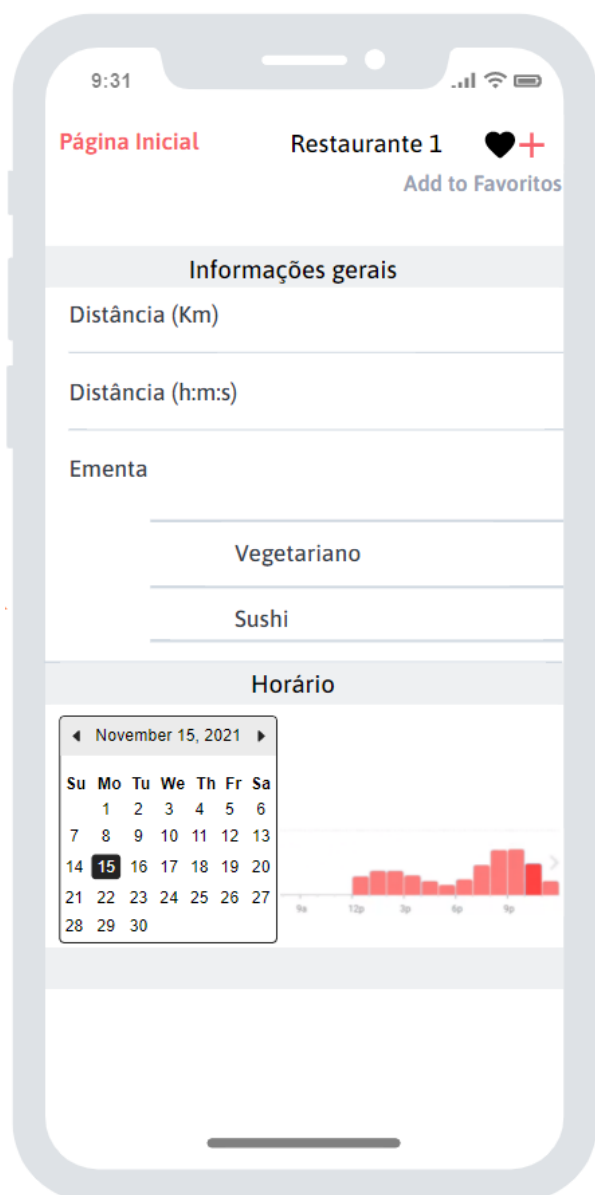


Figura 11 - Página Informações Gerais

Nesta página é possível acessar a página 'Página Inicial' clicando no topo esquerdo e adicionar o restaurante que está a ser visualizado aos favoritos clicando no símbolo 'Mais' no topo direito.

Deve ser apresentado no corpo da página toda a informação que for relevante sobre o restaurante em questão, esta página deve também conter mais informação que a apresentação simples feita na página 'Página Inicial' do restaurante em 'foco'.

## 6 Conclusões e Trabalho Futuro

### 6.1 Conclusões

No desenvolvimento deste trabalho, deparámo-nos com a complexidade que a elaboração de um projeto desta natureza tem. Para garantir que o produto final seja equivalente à aplicação que idealizamos, é necessário converter todas as ideias em conceitos bem definidos utilizando diagramas e a descrição do comportamento das funcionalidades da aplicação tornando essas mesmas ideias em algo claro e de fácil compreensão para os “developers” conseguirem ir de melhor forma ao encontro do que é pretendido por quem idealiza a aplicação. Foi de facto um desafio interessante tentar não esquecer nenhum detalhe ou pormenor, já que foi necessário especificar tudo e garantir que nenhuma ideia fique perdida, isto é ingenuamente subentendida ou ambígua.

### 6.2 Trabalho Futuro

Na implementação da especificação que nós realizamos não esperamos que tudo seja seguido à risca, ou seja, que apenas se faça o que está descrito no trabalho ou que se execute exatamente como foi descrito no mesmo, no entanto achamos crucial que a identidade da aplicação se mantenha, coisas como o ‘tema’ da aplicação, funcionalidades de “Swipe” ou os “DropDown menus”, tudo em conjunto cria uma identidade da aplicação como leve e intuitiva que achamos que deve ser seguida, prevemos também que mais passos possam ser tomados nessa direção para melhor frisar e marcar essa mesma identidade.

Muitos dos requisitos que descrevemos estão diretamente relacionados com a apresentação dos restaurantes pois essa será a maior utilidade da aplicação, no entanto não deverá ser a mais dispendiosa em termos de desenvolvimento, a integração e implementação dos “filters” que são criados com os inquéritos é, na nossa perspetiva, a parte da aplicação que abre mais espaço para criatividade mas também, a que pode possivelmente dar mais trabalho, diferenciar entre os “filters” absolutos e parciais, definir como os parciais devem ser aplicados e de que maneira devem ser codificados será o desafio mais interessante e menos “straightforward”, na nossa conceção da aplicação isto seria feito a partir de pesos, algo do género de: se um “user” tem uma preferência por pratos picantes então serão mostrados pratos picantes em 70% mais do que os não picantes. Em último caso e se o tempo de desenvolvimento se apertar todos os “filters” podem ser absolutos simplificando o desenvolvimento da aplicação que consideraríamos na mesma completa, mas sem o adicional que previmos na fase de especificação.

Apresentamos em seguida um diagrama GANTT que serve como orientação para o desenvolvimento do trabalho que irá ser realizado na fase seguinte.

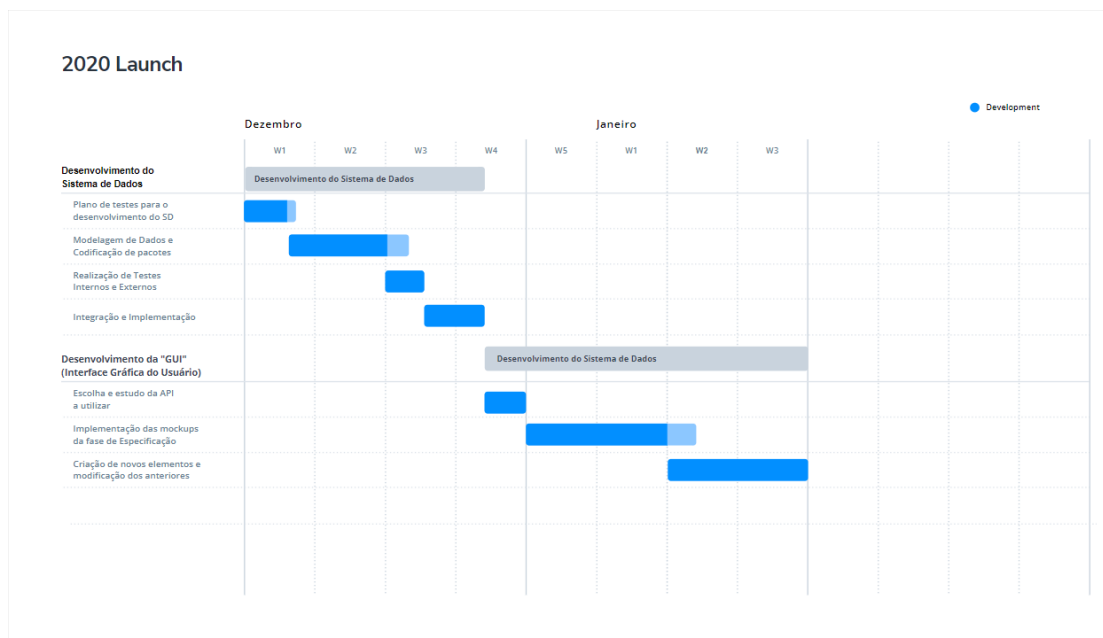


Figura 12 - Diagrama GANTT (Trabalho Futuro)

## 7 Referências Bibliográficas

[Visual Paradigm](#): Utilizado para conceber ambos os diagramas de Domínio e Use Case;

[Moqups](#): Utilizado para conceber um protótipo da interface da aplicação;

[mySQL workbench](#): Utilizado para elaborar o diagrama lógico da base de dados;

<https://www.devmedia.com.br/exemplo-pratico-regras-e-tipos-de-relacionamentos-sql-server-2008-parte-3/18409>: Tipos de relacionamentos (base de dados);

<https://fluxoconsultoria.poli.ufrj.br/blog/desenvolvimento-banco-de-dados/> :Etapas para a criação de uma base de dados;

<https://pt.fondoperlaterra.org/comdifference-between-strong-and-weak-entity-18>: Diferença entre entidade fraca e forte (base de dados);

<https://medium.muz.li/stages-of-interface-development-e3cf59e2ca6b> :Etapas para o desenvolvimento de uma interface gráfica para o utilizador (GUI);

<https://www.sitepoint.com/5-ways-to-improve-user-experience-with-machine-learning/> : Maneiras de usar “machine learning” para melhorar o “user experience”;