



Escola Superior de Tecnologia e Gestão

# **Click2Eat**

## **v1.0**

Fábio Pires  
8160127

Luís Marques  
8170485

Licenciatura em Engenharia Informática

Computação Móvel e Ubíqua  
3º Ano, 1º Semestre

7 de setembro de 2020

[Página propositadamente deixada em branco.]

# **Click2Eat**

**v1.0**

Fábio Pires  
Luís Marques

Licenciatura em Engenharia Informática

[Página propositadamente deixada em branco.]

# Histórico de Versões

Nome	Data	Razão da(s) mudança(s)	Versão
Luís Marques	2020-08-22	Start the report	1.0
Luís Marques	2020-08-20	Finish the first version	1.0

[Página propositadamente deixada em branco.]

# Conteúdo

<b>Glossário</b>	<b>vii</b>
<b>Abreviaturas</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Âmbito . . . . .	1
1.2 Definição do Problema . . . . .	1
1.3 Solução . . . . .	2
1.4 Objetivos . . . . .	2
1.5 Repositório . . . . .	4
<b>2 Contextualização</b>	<b>5</b>
2.1 Restaurante . . . . .	5
2.2 Aplicação móvel . . . . .	5
2.3 Combinação . . . . .	6
2.4 API da Zomato . . . . .	6
<b>3 Construção</b>	<b>7</b>
3.1 Requisitos . . . . .	7
3.1.1 Obrigatórios . . . . .	7
3.1.2 Bonificação . . . . .	8
3.2 Implementação . . . . .	8
<b>4 Produto Final</b>	<b>9</b>
<b>Referências</b>	<b>13</b>

[Página propositadamente deixada em branco.]



# Lista de Figuras

4.1	Login . . . . .	9
4.2	Register . . . . .	10
4.3	Live restaurants . . . . .	10
4.4	Search restaurants . . . . .	11
4.5	Settings . . . . .	11

[Página propositadamente deixada em branco.]

# Glossário

**Android** Sistema operacional baseado no núcleo Linux, desenvolvido por um consórcio de desenvolvedores conhecido como Open Handset Alliance, sendo o principal colaborador o Google. 4

**Interface de Programação de Aplicações** Conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços. ix

**Kernel** Componente central do sistema operativo da maioria dos computadores; ele serve de ponte entre aplicações e o processamento real de dados feito a nível de hardware. vii

**Linux** Termo popularmente empregado para se referir a sistemas operativos que utilizam o Kernel Linux. vii

[Página propositadamente deixada em branco.]

# Abreviaturas

**API** *Application Programming Interface* (Interface de Programação de Aplicações). 6

**CMU** Computação Móvel e Ubíqua. 1

**ESTG** Escola Superior de Tecnologia e Gestão. 1

**LEI** Licenciatura em Engenharia Informática. 1

[Página propositadamente deixada em branco.]

# Capítulo 1

## Introdução

---

1.1	Âmbito . . . . .	1
1.2	Definição do Problema . . . . .	1
1.3	Solução . . . . .	2
1.4	Objetivos . . . . .	2
1.5	Repositório . . . . .	4

---

### 1.1 Âmbito

Este projeto foi realizado na época especial de 2020, dentro do âmbito de Computação Móvel e Ubíqua da Licenciatura em Engenharia Informática na Escola Superior de Tecnologia e Gestão - P.Porto.

### 1.2 Definição do Problema

A atual situação económica e social devido à pandemia COVID-19 forçou a adoção de novas medidas e a adaptação de novas rotinas. A utilização de espaços de restauração ficou condicionada com novas restrições de limites de ocupação e horários de funcionamento.

Desta forma, torna-se útil pesquisar restaurantes e verificar o cardápio fornecido antes da deslocação ao estabelecimento, de modo a evitar que o utilizador se dirija ao local e a comida não seja do seu agrado, assim como descobrir novos restaurantes adaptados às necessidades de cada cliente.

### 1.3 Solução

De forma genérica pretende-se o desenvolvimento de uma aplicação para dispositivos móveis que melhore a experiência da visita a restaurantes e registre os considerados “preferidos” pelo utilizador, que estabelecimentos já frequentou e qual a pontuação fornecida de cumprimento de normas de saúde pública relacionadas com o COVID-19.

Deve ser permitido visualizar todos os restaurantes já visitados, os seus favoritos e todos em que se pretende efetuar uma refeição considerando a distância do utilizador ao estabelecimento, a classificação atribuída pelo utilizador (ou pela comunidade caso não exista uma classificação específica do utilizador) e a frequência de utilização (número de vezes que o utilizador realizou refeições nesse estabelecimento). A aplicação deve automaticamente detetar quando um utilizador se encontra perto de um restaurante favorito, e automaticamente lançar uma notificação sugerindo uma refeição nesse estabelecimento, mesmo quando a aplicação se encontrar em background.

Para aquisição de informação de contexto sobre restaurantes deve ser considerada, entre outras, a API Zomato. Outras APIs podem igualmente ser usadas em substituição (ex: google places) desde que consigam dar resposta ao problema. Informação proveniente de outras fontes de dados ou webservices podem também ser consideradas para enriquecimento da aplicação (ex: navegação, meteorologia, etc).

O histórico de refeições deve estar disponível para o utilizador e ser consultado pelo utilizador da aplicação. Cada histórico deve apresentar de forma breve informação sobre o estabelecimento, e outras informações relevantes como o tipo de refeição efetuada mais frequentemente, distância média percorrida para que o utilizador se desloque.

A pontuação fornecida (de 1 a 5) deve poder ser partilhada de forma anónima com todos os utilizadores da aplicação a desenvolver (considere o uso de uma base de dados como firebase para guardar as pontuações de cada restaurante). Um utilizador só pode pontuar um determinado restaurante quando se encontrar perto dele (considerar um raio ou tolerância entre as coordenadas do restaurante e a posição do utilizador).

### 1.4 Objetivos

Para este projeto foram estabelecidos alguns objetivos pelo professor:

- Especificar e coordenar um projeto em grupo de pequena dimensão;
- Compreender e dominar os conhecimentos teóricos e práticos sobre desenvolvimento de aplicações móveis na plataforma Android;
- Adquirir competências com vista à resolução de problemas, nomeadamente através da pesquisa e utilização autónoma de conteúdos e ferramentas externas;



- Estimular o trabalho em equipa como elemento essencial do processo de aprendizagem individual.

Para completar, nós também definimos alguns objetivos pessoais, para a realização deste trabalho da melhor forma e com a garantia de que obteríamos o máximo de conhecimentos possíveis sobre a plataforma Android:

- Construir uma aplicação totalmente amigável ao utilizador e fácil de compreender;
- Incorporar bastante responsividade na aplicação para que ela não quebre tão facilmente;
- Criar um design minimamente plausível e bonito da aplicação.

### **1.5 Repositório**

<https://github.com/LuisMarques99/ProjetoCMU2020>

## Capítulo 2

# Contextualização

---

2.1	Restaurante . . . . .	5
2.2	Aplicação móvel . . . . .	5
2.3	Combinação . . . . .	6
2.4	API da Zomato . . . . .	6

---

### 2.1 Restaurante

Um restaurante é um Estabelecimento comercial destinado a confeccionar e vender refeições, onde normalmente são também servidos todo o tipo de bebidas.

### 2.2 Aplicação móvel

Uma aplicação móvel é um software desenvolvido com o intuito de ser instalado num dispositivo móvel. Tem como propósito facilitar o quotidiano dos seus utilizadores.

Originalmente as aplicações móveis foram criadas e classificadas como ferramentas de apoio à produtividade e à recuperação de informação generalizada, incluindo email, calendário, contactos, mercado de ações, informações meteorológicas, etc. A crescente procura, a disponibilidade facilitada e a evolução, conduziu à rápida expansão para outras categorias, como jogos, GPS, serviços de informação meteorológica, serviços de acompanhamento de pedidos, compra de bilhetes, confirmações de presenças e redes sociais.

Nos dias de hoje, estão praticamente presentes em todo o lado incluindo nas áreas da saúde, desporto, banca e negócios.

### 2.3 Combinação

Atualmente, onde o tempo é um recurso cada vez mais escasso, é indispensável que até a mais simples tarefa de escolher o estabelecimento onde vamos realizar a nossa próxima refeição seja feita de forma rápida e eficiente.

Com a recente pandemia da Covid 19, torna-se também imprescindível que essa escolha seja feita da forma mais acertada possível de forma a evitar o constrangimento de nos deslocarmos para um local em vão, aumentando a probabilidade de sermos infectados.

Deste modo, e para que os utilizadores da nossa aplicação não só se possam manter o mais seguros possível, mas também desfrutar ao máximo da sua refeição, decidimos unir os dois conceitos acima relatados resultando numa aplicação móvel para servir de suporte aos utilizadores de restaurantes em Portugal continental.

### 2.4 API da Zomato

A *API* da Zomato permite obter informações relacionadas a mais de um milhão e meio de restaurantes, bares, cafés, pubs e nightlife situados em dez mil cidades em todo o mundo.

Dentro destas informações são incluídos geralmente o nome e tipo do estabelecimento, a sua localização, contactos, custo médio para duas pessoas, horários de funcionamento, contactos, tipos de “cozinha” e uma avaliação.

Além destas informações mais gerais, através da *API* também é possível saber se o estabelecimento aceita cartão de crédito, se tem take-away e entregas ao domicílio, se são servidas bebidas alcoólicas, se é provido de wi-fi, etc.

## Capítulo 3

# Construção

---

3.1	Requisitos . . . . .	7
3.1.1	Obrigatórios . . . . .	7
3.1.2	Bonificação . . . . .	8
3.2	Implementação . . . . .	8

---

### 3.1 Requisitos

#### 3.1.1 Obrigatórios

Para a realização deste projeto foram definidos alguns requisitos obrigatórios:

- Suporte para ecrãs de diferentes dimensões (Telemóvel e Tablet);
- Uso das bibliotecas de suporte (Android Support Library);
- Uso de listas (RecyclerView e Adapters);
- Uso de base de dados (Room);
- Uso de operações assíncronas (AsyncTask/Thread/IntentService);
- Uso de notificações;
- Uso de sensores de localização e disponibilização de informação em mapas;
- Uso das guidelines do material design;
- Integração com a API Zomato via pedidos REST (Retrofit).

### 3.1.2 Bonificação

Para bonificar o projeto foram estabelecidos os seguintes requisitos:

- Uso webservices adicionais via pedidos REST (Retrofit);
- Uso da biblioteca de detecção de atividades em Android para detecção do início ou fim de atividades;
- Exportar data de refeições realizadas para a API Google Calendar para registo de atividade no calendário pessoal;
- Interação com elementos do Android (Contactos, Mensagens, Dialer, etc.).

## 3.2 Implementação

Para a realização desta aplicação foram utilizadas as seguintes ferramentas:

- Java - linguagem de programação;
- AndroidStudio - IDE para desenvolvimento da aplicação;
- Visual Studio Code - editor de código para desenvolvimento dos documentos de suporte ao projeto;
- Gradle - controlo de dependências;
- Git - controlo de versões;
- GitHub - gestor do repositório do projeto.

## Capítulo 4

# Produto Final

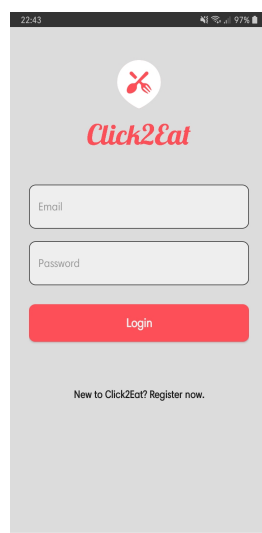


Figura 4.1: Login

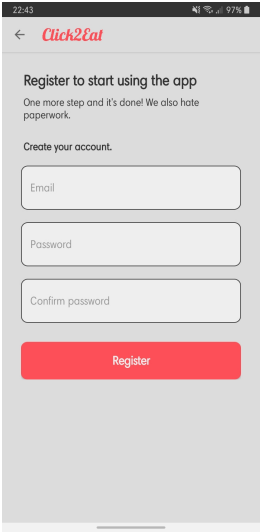


Figura 4.2: Register

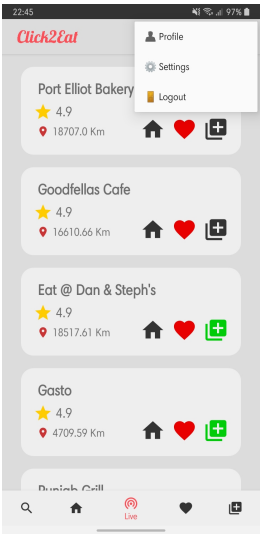


Figura 4.3: Live restaurants



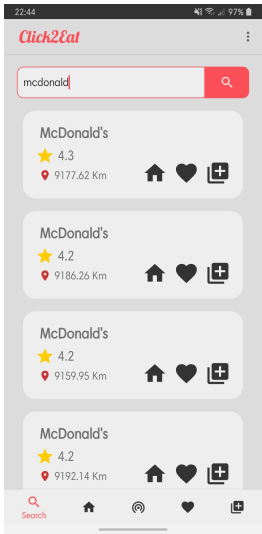


Figura 4.4: Search restaurants

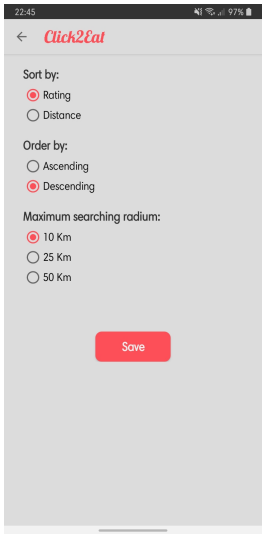


Figura 4.5: Settings

[Página propositadamente deixada em branco.]

# Referências

- [1] [en.wikipedia.org](https://en.wikipedia.org)
- [2] [www.newthinktank.com/2019/01/latex-tutorial](https://www.newthinktank.com/2019/01/latex-tutorial)
- [3] [pt.overleaf.com](https://pt.overleaf.com)
- [4] [www.dickimaw-books.com/gallery/glossaries-styles](https://www.dickimaw-books.com/gallery/glossaries-styles)
- [5] [developer.android.com/guide/topics/location/transitions](https://developer.android.com/guide/topics/location/transitions)
- [6] [developer.android.com/guide/topics/providers/calendar-provider](https://developer.android.com/guide/topics/providers/calendar-provider)
- [7] [developers.zomato.com/api](https://developers.zomato.com/api)
- [8] [stackoverflow.com](https://stackoverflow.com)
- [9] [github.com/pfn/qicr](https://github.com/pfn/qicr)
- [10] [github.com/firebase/quickstart-android](https://github.com/firebase/quickstart-android)
- [11] [github.com/android/location-samples](https://github.com/android/location-samples)
- [12] [www.jsonschema2pojo.org](https://www.jsonschema2pojo.org)
- [13] [pt.db-city.com](https://pt.db-city.com)
- [14] [pixplicity.com/dp-px-converter](https://pixplicity.com/dp-px-converter)
- [15] [www.youtube.com/watch?v=Nn4-Vn7qk9k](https://www.youtube.com/watch?v=Nn4-Vn7qk9k)