

**UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**APLICATIVO DE MAPEAMENTO DE LANCHERIAS MÓVEIS E  
FOOD TRUCKS PARA GERAÇÃO DE DADOS QUE VIABILIZEM  
ANÁLISE DE PERFIL DE USUÁRIO**

**TRABALHO DE GRADUAÇÃO**

**Caroline Jardim Siqueira**

**Santa Maria, RS  
2017**

**Caroline Jardim Siqueira**

**APLICATIVO DE MAPEAMENTO DE LANCHERIAS MÓVEIS E FOOD TRUCKS PARA  
GERAÇÃO DE DADOS QUE VIABILIZEM ANÁLISE DE PERFIL DE USUÁRIO**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de **Bacharel em Ciência da Computação**.

Orientadora: Prof. Dra. Lisandra Manzoni Fontoura

432

Santa Maria, RS  
2017

**Caroline Jardim Siqueira**

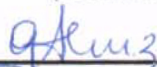
**APLICATIVO DE MAPEAMENTO DE LANCHERIAS MÓVEIS E FOOD TRUCKS PARA  
GERAÇÃO DE DADOS QUE VIABILIZEM ANÁLISE DE PERFIL DE USUÁRIO**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de **Bacharel em Ciência da Computação**.

**Aprovado em de Julho de 2017:**



**Lisandra Manzoni Fontoura, Prof. Dra. (UFSM)**  
(Presidente/Orientador)



**Giliane Bernardi, Prof. Dra. (UFSM)**



**Marcos Alexandre Rose Silva, Prof. Dr. (UFSM)**

## AGRADECIMENTOS

*Dedico meus agradecimentos a todos que contribuíram com minha formação durante o período de graduação, em especial:*

- à minha família, que prestou suporte e amor incondicional nessa trajetória;*
- à Professora Lisandra, que sempre acreditou nas ideias e projetos que busquei implementar no período de graduação e nunca desistiu de mim;*
- aos colegas e amigos que fiz, com quem compartilhei experiências e aprendizados in-críveis e sempre procuraram me ajudar no que fosse preciso;*
- à todos da CompAct Jr e demais empresas juniores da UFSM, que me proporcionaram incontáveis oportunidades de crescimento, inspiração e vontade de continuar evoluindo.*



*"It is our choices that show what we truly are, far more than our abilities."  
[São nossas escolhas que mostram quem realmente somos, muito mais do que nossas  
habilidades.]*

*J.K. Rowling*

## RESUMO

### APLICATIVO DE MAPEAMENTO DE LANCHERIAS MÓVEIS E FOOD TRUCKS PARA GERAÇÃO DE DADOS QUE VIABILIZEM ANÁLISE DE PERFIL DE USUÁRIO

AUTORA : Caroline Jardim Siqueira

ORIENTADORA : Lisandra Manzoni Fontoura Prof. Dra.

Com o intenso crescimento do uso de dispositivos móveis com acesso à *Internet*, surgiu a preocupação de proporcionar melhores experiências aos usuários. Estes, que carregam seus *smartphones* nos mais diversos ambientes e possuem diferentes comportamentos em relação ao contexto o qual estão inseridos, trazem à tona a necessidade de adaptação dos aplicativos aos seus respectivos contextos. Além disso, com a geração massiva de dados que produzem, é possível determinar eventuais oportunidades de impulsionamento de negócios, se considerarmos o entendimento do que esses usuários buscam nas e no respectivo contexto o qual estão inseridos. Este trabalho visou desenvolver um aplicativo de mapeamento e avaliação de lancherias móveis e *food trucks* que geram dados de comportamento dos usuários, afim de encontrar um modelo de análise de perfil de usuários, seguindo uma metodologia baseada na utilização de um *framework* e conceitos de análise de perfil. O resultado é uma simulação e estudo de caso com dados gerados por testes do aplicativo, e a geração de um perfil hipotético utilizando o modelo apresentado.

**Palavras-chave:** Análise de Perfil, Computação Móvel, Contexto, Dispositivos Móveis, *Framework*, Mobilidade.

## **ABSTRACT**

### **MAPPING AND EVALUATION OF FOOD TRUCKS APPLICATION FOR DATA GENERATION THAT ENABLES USER PROFILING ANALYSIS**

**AUTHOR :** Caroline Jardim Siqueira  
**ADVISOR:** Prof. Dra. Lisandra Manzoni Fontoura

With the intense increase of the use of mobile devices with access to the Internet, emerged the concern to provide better usability experiences to users. These users, who carry their smartphones in different environments and have different behaviors related to the context they are inserted, bring the concern to adapt applications to their respective contexts. In addition, with the massive generation of data they generate, it is possible to determine eventual business-boosting opportunities, if we consider the understanding of what these users are looking for in, and in the context of which they are inserted. This work aims the development of a mapping and evaluation of mobile snack bars and food trucks application, that generates user behavior data, in order to find a user profile analysis model, following a methodology base on a framework and profile analysis concepts. The result is a simulation and case study with data generated by application testing, and a generation of a hypothetical user profile.

**Keywords:** Context, Framework, Mobile devices, Mobile Computing, Mobility, Profile Analysis.

## LISTA DE FIGURAS

|           |   |    |
|-----------|---|----|
| FIGURA 1  | PROPOSTA DE <i>framework</i> PARA PROVISÃO DE SERVIÇO <i>Context Aware</i>  | 14 |
| FIGURA 2  | ÍNDICE DE DISTRIBUIÇÃO DE VERSÕES DO ANDROID . . . . .  | 17 |
| FIGURA 3  | CÓDIGO DE INSERÇÃO DO FACEBOOK NA APLICAÇÃO . . . . .   | 18 |
| FIGURA 4  | PAINEL DO <i>Facebook Analythics</i> . . . . .  | 19 |
| FIGURA 5  | CÓDIGO DE INSERÇÃO DO GOOGLE MAPS NA APLICAÇÃO . . . . .  | 19 |
| FIGURA 6  | CÓDIGO DE INSERÇÃO DO GOOGLE MAPS NA APLICAÇÃO . . . . .  | 20 |
| FIGURA 7  | CÓDIGO DE INSERÇÃO DO GOOGLE MAPS NA APLICAÇÃO . . . . .  | 20 |
| FIGURA 8  | DIAGRAMA DE RELACIONAMENTO DE CLASSES . . . . .   | 25 |
| FIGURA 9  | DIAGRAMA DE CASOS DE USO . . . . .  | 26 |
| FIGURA 10 | ADAPTAÇÃO DO <i>framework</i> PARA PROVISÃO DE SERVIÇO <i>Context Aware</i><br>. . . . .  | 27 |
| FIGURA 11 | MODELO DE CATEGORIZAÇÃO 5W2H . . . . .  | 29 |
| FIGURA 12 | DIAGRAMA DE SEQUÊNCIA DO SISTEMA . . . . .  | 31 |
| FIGURA 13 | DIAGRAMA DE CLASSES DO APLICATIVO . . . . .   | 32 |
| FIGURA 14 | DIAGRAMA DE ATIVIDADES . . . . .  | 33 |
| FIGURA 15 | TELA INICIAL: CADASTRO/ <i>Login</i> . . . . .  | 35 |
| FIGURA 16 | CÓDIGO DE REQUISIÇÃO DE PERMISSÃO . . . . .   | 35 |
| FIGURA 17 | TELA DE RESQUISIÇÃO DE PERMISSÃO: USUÁRIO DEVE CLICAR NO<br>BOTÃO PARA QUE A REQUISIÇÃO DE PERMISSÃO OCORRA . . . . .                           | 36 |
| FIGURA 18 | TELA DE RESQUISIÇÃO DE PERMISSÃO: O SISTEMA PEDE PERMISSÃO<br>DO USUÁRIO. SE FOR CONCEDIDO, O USUÁRIO PODERÁ ENTRAR NA TELA DO<br>MAPA. . . . . | 37 |
| FIGURA 19 | TELA DO MAPA . . . . .  | 38 |
| FIGURA 20 | TELA DO MAPA: NOVO MARCADOR INSERIDO . . . . .  | 39 |
| FIGURA 21 | TELA DE CADASTRO DE NOVO PONTO DE VENDA . . . . .   | 40 |
| FIGURA 22 | TELA DE PERFIL DO USUÁRIO . . . . .   | 41 |
| FIGURA 23 | TELA DE INFORMAÇÕES DO PONTO DE VENDA: <i>Check-in</i> . . . . .  | 42 |
| FIGURA 24 | TELA DE INFORMAÇÕES DO PONTO DE VENDA: ADICIONAR FOTO AO<br>ÁLBUM E BOTÃO DE EXCLUIR . . . . .  | 42 |
| FIGURA 25 | TELA DE INFORMAÇÕES DO PONTO DE VENDA: AVALIAÇÃO . . . . .  | 43 |
| FIGURA 26 | TELA DE INFORMAÇÕES DO PONTO DE VENDA: EDITAR DESCRIÇÃO . . . . .   | 44 |
| FIGURA 27 | EXEMPLO DE REGISTRO DE OBJETO DE USUÁRIO . . . . .  | 45 |
| FIGURA 28 | EXEMPLO DE REGISTRO DE OBJETO DO LOCAL . . . . .  | 46 |
| FIGURA 29 | PROCESSO DE EXECUÇÃO E GERAÇÃO DO PERFIL DO USUÁRIO . . . . .   | 47 |
| FIGURA 30 | EXEMPLO DE PERFIL DO USUÁRIO CONFORME ÚLTIMO <i>check-in</i> . . . . .  | 48 |

## **LISTA DE ABREVIATURAS E SIGLAS**

|      |                                    |
|------|------------------------------------|
| API  | Application Programming Interface  |
| IDE  | Integrated Development Environment |
| JSON | JavaScript Object Notation         |
| MVP  | Minimum Viable Product             |

## SUMÁRIO

|       |  |    |
|-------|--|----|
| 1     | INTRODUÇÃO   | 10 |
| 1.1   | OBJETIVO DO TRABALHO . . . . .                       | 11 |
| 1.2   | OBJETIVOS ESPECÍFICOS . . . . .                      | 11 |
| 1.3   | JUSTIFICATIVA . . . . .                              | 12 |
| 2     | FUNDAMENTAÇÃO TEÓRICA                                | 13 |
| 2.1   | REVISÃO DE CONCEITOS . . . . .                       | 13 |
| 2.1.1 | CONTEXTO E PERFIL DE USUÁRIO . . . . .               | 13 |
| 2.1.2 | ANÁLISE DE DADOS E CATEGORIZAÇÃO DE PERFIL . . . . . | 15 |
| 2.2   | TECNOLOGIAS . . . . .                                | 16 |
| 2.2.1 | ANDROID . . . . .                                    | 16 |
| 2.2.2 | FACEBOOK . . . . .                                   | 17 |
| 2.2.3 | GOOGLE MAPS . . . . .                                | 19 |
| 2.2.4 | FIREBASE . . . . .                                   | 20 |
| 3     | METODOLOGIA  | 22 |
| 3.1   | MVP DO APLICATIVO . . . . .                          | 22 |
| 3.2   | PROPOSTA DE MODELO PARA GERAÇÃO DE PERFIL . . . . .  | 23 |
| 4     | DESENVOLVIMENTO                                      | 25 |
| 4.1   | ENTIDADES E REQUISITOS DO APLICATIVO . . . . .       | 25 |
| 4.2   | ARQUITETURA DO SISTEMA . . . . .                     | 27 |
| 4.3   | IMPLEMENTAÇÃO . . . . .                              | 30 |
| 4.3.1 | TELA INICIAL: CADASTRO/ <i>Login</i> . . . . .       | 34 |
| 4.3.2 | TELA DE REQUISIÇÃO DE PERMISSÃO . . . . .            | 35 |
| 4.3.3 | TELA DO MAPA . . . . .                               | 37 |
| 4.3.4 | TELA DE CADASTRO DE NOVO PONTO DE VENDA . . . . .    | 39 |
| 4.3.5 | TELA DE PERFIL DO USUÁRIO . . . . .                  | 40 |
| 4.3.6 | TELA DE INFORMAÇÕES DO PONTO DE VENDA . . . . .      | 41 |
| 5     | ANÁLISE E ESTUDO DE CASO                             | 45 |
| 6     | CONCLUSÃO  | 50 |
|       | REFERÊNCIAS  | 52 |

## 1 INTRODUÇÃO

O estudo e desenvolvimento de tecnologias voltadas ao uso pessoal pretende entregar cada vez mais e melhores experiências ao usuário. Isso porque a utilização de artigos tecnológicos se intensificou de tal forma nas últimas décadas que seu impacto afeta diretamente em fatores socioeconômicos globais. No Brasil, possuir um smartphone pode representar possuir um status social maior, independente da renda do usuário (Bacha e Figueiredo Neto, 2013). Estar conectado à internet significa estar conectado à sociedade.

O estilo de vida atual preza pela velocidade, otimização de tempo e automatização de tarefas cotidianas simples. Portanto, a percepção de contexto o qual uma aplicação vai atender a um usuário é essencial para que ele tenha uma experiência satisfatória. A tendência é que aplicações se comuniquem com o usuário de forma natural, mais próxima de interações humanas, como recomendações de tarefas habituais, tais quais lembrar onde o carro está estacionado ou de pegar o guarda-chuva quando o tempo estiver ruim (Raghunathan e Jaswal, 2013).

Dessa forma, torna-se indispensável, para o modelo de negócios de aplicativos para dispositivos móveis, o entendimento do contexto ambiental no qual o usuário se encontra para que o desenvolvimento facilite a utilização e torne a experiência do cliente em determinada situação mais completa (Dey, 2001).

Ao considerar o ambiente urbano, onde as pessoas estão se movimentando constantemente, para ir ao trabalho ou para a escola, indo a supermercado ou saindo para se entreter - e em todas essas situações, elas podem portar seus smartphones. Além disso, é natural que em zonas urbanas nos deparamos com vendedores de lanches e *food trucks* em pontos de venda estratégicos. E esse tipo de negócio de venda de comida com mobilidade seguem estratégias comerciais pontuais, ou seja, estarão onde a conversão de *leads* (potenciais clientes) é maior. Como é o caso de festivais gastronômicos locais, eventos de jogos esportivos ou grandes festas, que são cenários oportunos para vendas de artigos alimentícios de rápido preparo.

Observa-se ainda a tendência em sistemas colaborativos, nos quais os usuários possuem a liberdade de compartilhar informações por meio da tecnologia, aumentando cada vez mais a conexão entre pessoas em um ambiente virtual. Em diversos casos, a qualidade de vida de quem utiliza desse tipo de sistema pode ser influenciada de forma positiva, o custo de trocas de serviços podem sair mais em conta do que serviços prestados pelos meios tradicionais, permitindo que os usuários tenham a possibilidade de experimentar novas vivências por meio da tecnologia colaborativa (R. E. Miles, G. Miles e Snow, 2005).

Esses aplicativos também influenciam na formação de opinião, já que usuários procuram contar seus pontos de vista sobre a experiência que tiveram no contexto em que estão inseridos. A grande questão desse tipo de aplicação, no entanto, é garantir segurança e privacidade de seus usuários. Uma vez que esses quesitos são assegurados e os serviços oferecidos sejam de fato vantajosos aos clientes, é natural que esses modelos de negócio tenham boa repercussão (Schafer, Konstan e Riedl, 2001).

Por fim, além de compreender o aspecto ambiental e social de um modelo de negócio, é necessário entender o perfil dos futuros usuários da aplicação. Afinal, é necessário conhecer o usuário final da aplicação para que seja possível encontrar métodos mais eficientes de lhe proporcionar uma experiência positiva. Portanto, é fundamental realizar análise do perfil básico de usuário, ou seja, conhecer quem são as pessoas que estão inseridas no contexto ambiental abrangido pelo sistema.

A análise de perfil de um sistema deve se dar pelo reconhecimento de quem são essas pessoas, de onde são, qual a faixa etária, grau de escolaridade, gênero e dados que possam informar o porquê esse usuário se identifica com a aplicação e se sua experiência é modificada e melhorada com o uso. Em vista disso, este trabalho propõe um modelo conceitual de análise de perfil do usuário, adaptado do *framework* de Panagiotakis (2012) e a implementação de uma aplicação de mapeamento e avaliação de lancherias móveis e *food trucks* considerando o contexto do usuário. Posteriormente, é realizada uma análise de como os dados gerados por testes do aplicativo, que podem ser usados para a geração de categorias de perfis de usuários utilizando a metodologia escolhida.

### 1.1 OBJETIVO DO TRABALHO

O objetivo deste trabalho é desenvolver um aplicativo para mapeamento e avaliação de lancherias móveis e *food trucks* que possibilite uma análise de perfil de usuários, seguindo uma metodologia baseada na utilização de um *framework* e conceitos de análise de perfil. Para isso, foi implementado um aplicativo que interage com outros aplicativos como Facebook e Google Maps e gera os dados necessários para posterior análise de perfil de usuários.

### 1.2 OBJETIVOS ESPECÍFICOS

- Modelagem e implementação de uma aplicação para dispositivos móveis Android para uso colaborativo de usuários, com acesso a um mapa de lancherias móveis e *food trucks*;
- Integrar o aplicativo desenvolvido com Facebook e Google Maps;
- Definir um modelo de análise dos dados para construção de perfis de usuário considerando a aplicação desenvolvida;
- Coletar e armazenar dados que, posteriormente, possam ser usadas para análise de perfis por outros usuários ou por empresas (lancherias móveis e *food trucks*).



### 1.3 JUSTIFICATIVA

Como o objetivo deste trabalho é desenvolver um aplicativo de mapeamento e avaliação de lancherias móveis e *food trucks* para coleta de dados e, posteriormente, analisar o comportamento dos usuários em relação aos pontos de venda, a discussão sobre meios para compreender perfis de clientes em ambientes adversos torna-se plausível. Nesse contexto, a produção de melhores ferramentas para estudo e análise de contexto e perfil de usuários promove oportunidades de aperfeiçoamento de negócios com o uso de tecnologias móveis, que podem agregar valor na prestação desses serviços. Neste caso, falamos sobre possíveis melhorias na experiência de consumidores de lancherias móveis e *food trucks*, tal como a possibilidade desses comerciantes conhecerem pontos estratégicos para impulsionar seu negócio.

O texto está organizado como segue. No capítulo 2 são apresentadas referenciais teóricos e revisões de conceitos relacionados a contexto, perfil de usuário e análise de dados e categorização de perfil, além de introduzir tecnologias utilizadas para desenvolvimento do aplicativo. No capítulo 3 são descritas as metodologias utilizadas na criação do aplicativo e do modelo proposto para análise de perfil do usuário. No capítulo 4 é apresentada a formulação de desenvolvimento do aplicativo e da proposta do modelo de análise de perfil. No capítulo 5 é realizada uma análise de dados gerados pela aplicação e um estudo de caso do funcionamento do modelo apresentado. O capítulo 7 descreve as considerações finais.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo introduz conceitos que são utilizados no trabalho desenvolvido para alcançar os objetivos deste projeto, e está organizado pela Seção Revisão de Conceitos (2.1), que introduz definições a respeito de Contexto e Perfil do Usuário (Subseção 2.1.1) e Análise de dados e Categorização de Perfil (Subseção 2.1.2). Na Seção Tecnologias (2.2) são apresentadas ferramentas utilizadas para desenvolvimento do aplicativo.

### 2.1 REVISÃO DE CONCEITOS

Nesta Seção são introduzidos conceitos de contexto, perfil de usuário e análise de dados para realização de categorização de perfis, abordando brevemente trabalhos que serviram de referência para o desenvolvimento deste projeto.

#### 2.1.1 Contexto e Perfil de Usuário

Este trabalho está inserido em um contexto social que propõe utilizar dados de usuários para compreender seu comportamento em determinado contexto (lugar). Para entender este raciocínio, é necessário conhecer o conceito de contexto e como isso influencia na tomada de decisão do indivíduo. Segundo Dey (2001), a definição de contexto é “qualquer informação que possa ser utilizada para caracterizar a situação de entidades (pessoa, lugar ou objeto) que sejam consideradas relevantes para interação entre um usuário e uma aplicação (incluindo o usuário e a aplicação)”.

Uma vez que a tecnologia é capaz de fazer uma leitura analítica de comportamentos individuais, é possível se utilizar desse conteúdo para criar e entender perfis de usuários e personalizar suas experiências, de forma que se tornem mais atrativas e as aplicações sejam complemento do cenário diário do indivíduo. Para esta finalidade, existem métodos para realizar a captura de informações relativas ao comportamento do usuário, que visam à análise do contexto de uso. Essas informações podem determinar o contexto situacional de um produto utilizando palavras-chave de questionamento, como quem usará, com qual finalidade usará, quando e onde usará (Amato e Straccia, 1999).

Amato e Straccia (1999) apresentam um *framework* capaz de compilar as informações coletadas do usuário e aprimorar a experiência desse usuário dentro do contexto inserido, utilizando métodos de perguntas simples e palavras-chave. O trabalho retrata um modelo abstrato de perfil do usuário no cenário de busca por conteúdo e define categorias de usuários. Após essa definição, o trabalho exhibe um esquema de perfil para usuários de bibliotecas digitais e finaliza propondo duas arquiteturas de sistema de bibliotecas digitais de forma que os tipos definidos de usuários possam se adequar.

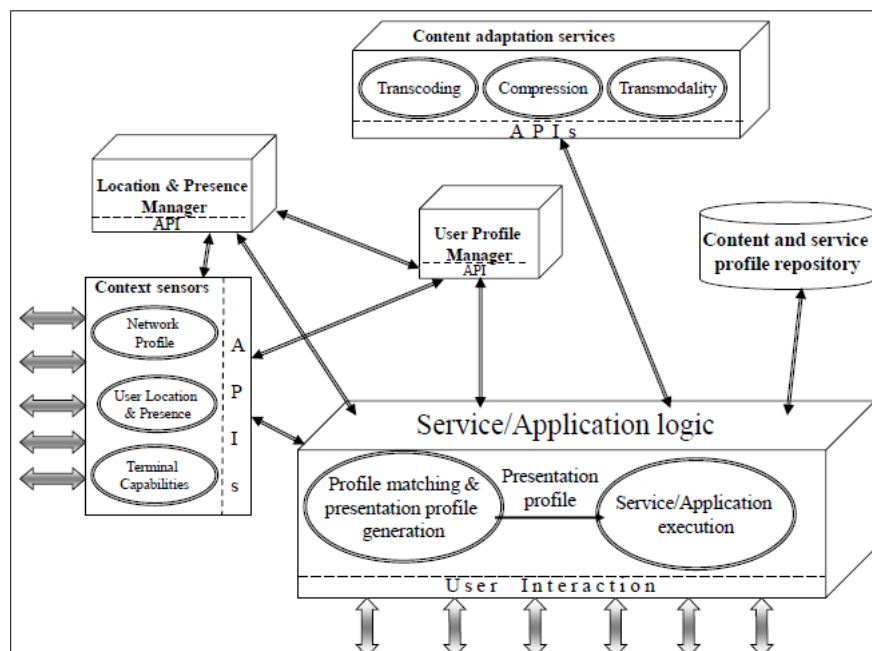
Havendo compreensão, portanto, do contexto o qual o usuário está inserido, o próximo passo é observar que tipo de informação o indivíduo busca na aplicação, o que se torna rele-

vante para então compor o perfil do usuário inserido em determinado contexto. Para tanto, é necessário entender o que é um perfil de usuário. Pode-se considerar que perfil do usuário não corresponde apenas às preferências de configuração do aplicativo, mas o perfil de suas decisões e até necessidades de uso quanto ao que é oferecido de informação no aplicativo (Suh e Woo, 2005).

Em vista disso, tem-se que a coleta de dados deve ser utilizada para análise de modo que o perfil é formado a partir do que o usuário busca no aplicativo, dado que o cenário de estudo contempla seu comportamento padrão quanto ao uso da aplicação em correlação com sua identidade. Esses estudos mostram a pertinência de entender o contexto e perfil do usuário, de forma que seja possível aprimorar o desenvolvimento de aplicações personalizadas às preferências do usuário.

Panagiotakis (2012) trabalha com entendimento de contexto da aplicação e propõe um modelo capaz de fazer reconhecimento do contexto para elaboração de User Profiling, por meio da captura de dados de sensores do dispositivo da aplicação e envia esses dados ao sistema, de forma que a plataforma identifique o contexto e realize a categorização de perfis. Este modelo contém módulos responsáveis por gerenciar localização e presença do usuário com insumos fornecidos pelos sensores e para gerir a formação de perfil do usuário. A Figura 1 ilustra sua proposta de *framework* para provisão de serviço *Context Aware*.

Figura 1 – Proposta de *framework* para Provisão de Serviço *Context Aware*



Fonte: (Context-Awareness via Ubiquitous User Profiling: An Implementation Paradigm)

A estrutura proposta é uma aglomeração de módulos constituídos por uma camada de *middleware* distribuída aberta, compartilhados entre si. O módulo da aplicação (*Service/Application logic*) é uma interface de comunicação com o usuário e é responsável por reconhecer

o perfil do usuário gerado pelo sistema, apresentar à aplicação e fazer atualização dos componentes de adaptação (*Content adaptation services*) e repositório de dados (*Content and service repository*). Ele se comunica com o módulo de sensor de reconhecimento de contexto (*Context sensors*), com o módulo de gerenciamento de dados em análise de contexto (*Locations and Presence Manager*) e de perfil do usuário (*User Profile Manager*).

Suh e Woo (2005) inferem que para ter ciência das atividades desempenhadas pelo usuário no seu cotidiano, é necessário que o sistema esteja de acordo com o contexto o qual está inserido. Este trabalho define um método de descrição de contexto, tais como os tipos de contexto que podem abranger, realizando mapeamento de atividades dos usuários por meio de sensores, gerando valores numéricos que representam os atributos dos contextos. Esses dados são treinados para gerar categorias adequadas do perfil do usuário, que permitem o aprendizado de comportamento e a atualização dinâmica de suas preferências.

Já o projeto de Wong (2015) faz uma análise do cenário escolhido e encontra espaço para aprimorar a experiência do usuário por intermédio da inserção digital nesse contexto. Neste trabalho, a autora desenvolve um aplicativo teste para *food trucks* e implementa o uso em Vancouver e conclui que a experiência do usuário torna-se mais completa quando obtém informações sobre o *food truck* antes de realizar sua compra - e que a implementação de funcionalidades como realizar pedido online e receber notificações de que o pedido está pronto tornaram o uso do aplicativo ainda mais eficiente.

### **2.1.2 Análise de dados e Categorização de Perfil**

No projeto de Adomavicius e Tuzhilin (2001), os autores argumentam que para determinação de regras, neste caso para aplicação, é necessário criar um modelo do perfil com as características do usuário, que incluem dados demográficos - tais como: nome, endereço, gênero, idade - que o desenvolvedor deseja recolher, além de dados comportamentais do usuário, os quais a aplicação pode fazer a coleta. A partir dessa definição, o estudo mostra que a personalização deve ocorrer individualmente para cada usuário para que as regras possam ser descobertas por meio de algoritmos de mineração de dados (Adomavicius e Tuzhilin, 2001).

Esse formato de coleta possui embasamento para seguir o estudo, conforme o trabalho de Suh e Woo, segundo o qual, é possível categorizar o perfil do usuário de duas formas, de acordo com a caracterização dos dados recolhidos: *static user-related*, que se referem a dados estáticos armazenados, tais como nome, idade, e outras informações que a aplicação permitir inserir; e *dynamic user-related*, que se referem a condições biológicas do usuário que podem ser coletadas por *wearables* (tecnologias “vestíveis”), tais como: estresse indicado pela pressão do usuário, nível de atenção que pode ser coletado por sensores do dispositivo.

De acordo com os autores, as informações estáticas do usuário são descritas usando técnicas do 5W1H para determinar o Contexto Condicional do Usuário, que são definidas por cinco perguntas essenciais a respeito das características do usuário em determinado contexto.

A técnica 5W1H ou 5W2H, dependendo do contexto, é também uma ferramenta que auxilia na definição das atividades ou atributos específicos a serem desenvolvidas para o cumprimento do objetivo de um projeto. Cada letra representa uma propriedade do projeto, sendo elas: *What* (O QUE), *When* (QUANDO), *Who* (QUEM), *Where* (ONDE), *Why* (POR QUÊ), *How* (COMO) e *How Much* (QUANTO). No caso do artigo em questão, os autores consideram QUEM como fator focal da questão para categorização do perfil e desconsideram a propriedade QUANTO.

No estudo de Amato e Straccia (1999), os dados do usuário são categorizados de forma semelhante, mas focando no objetivo do usuário em relação à proposta da aplicação, com os questionamentos do quê o usuário busca e como a informação deve ser entregue a esse usuário.

## 2.2 TECNOLOGIAS

Nesta Seção são apresentados conceitos sobre as tecnologias selecionadas para o desenvolvimento do aplicativo neste projeto, tais como: a plataforma de desenvolvimento Android na Seção 2.2.1, método de autenticação pelo Facebook na Seção 2.2.2, ferramenta de mapeamento dos pontos de venda por meio do Google Maps na Seção 2.2.3 e repositório de dados com o Firebase na Seção 2.2.4.

### 2.2.1 Android

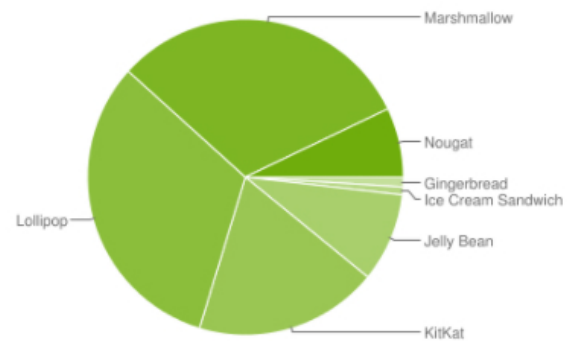
Android é uma pilha de *software open source* para dispositivos móveis, que inclui um sistema operacional, *middleware* e aplicações-chave (A. Developers, 2011).

O sistema mais utilizado mundialmente é o Android, superando o número de equipamentos que utilizam o sistema operacional *Windows*, segundo o jornal El País, matéria de Zurriarrain (2017).

A plataforma ainda é amplamente operada no Brasil. Segundo Oliveira (2017) cerca de 62% da população brasileira possui smartphones. Tendo em vista a proporção massiva de usuários que a plataforma agrega, é necessário observar que as versões disponibilizadas aos *smartphones* também podem impactar no desenvolvimento e uso de aplicativos.

Figura 2 – Índice de Distribuição de Versões do Android

| Version          | Codename              | API | Distribution |
|------------------|-----------------------|-----|--------------|
| 2.3.3 -<br>2.3.7 | Gingerbread           | 10  | 1.0%         |
| 4.0.3 -<br>4.0.4 | Ice Cream<br>Sandwich | 15  | 0.8%         |
| 4.1.x            | Jelly Bean            | 16  | 3.2%         |
| 4.2.x            |                       | 17  | 4.6%         |
| 4.3              |                       | 18  | 1.3%         |
| 4.4              | KitKat                | 19  | 18.8%        |
| 5.0              | Lollipop              | 21  | 8.7%         |
| 5.1              |                       | 22  | 23.3%        |
| 6.0              | Marshmallow           | 23  | 31.2%        |
| 7.0              | Nougat                | 24  | 6.6%         |
| 7.1              |                       | 25  | 0.5%         |



*Data collected during a 7-day period ending on May 2, 2017.  
Any versions with less than 0.1% distribution are not shown.*

Fonte: (Dashboards)

Como pode ser observado na Figura 2, é notório que as versões mais antigas (API 16 a 22) ainda representam boa parcela de usuários. E a cada lançamento de versão do Android são disponibilizados recursos incompatíveis com versões mais antigas. Portanto foi necessário escolher a API mínima que pudesse agregar o maior número de usuários possível, mas sem perder recursos dos sistemas essenciais à aplicação. Por isso, a API 17 (4.2.X *Jelly Bean*), que é compatível com 87.4% dos usuários e possui os mecanismos necessários para implementação, foi escolhida como SDK mínimo de compilação do aplicativo.

### 2.2.2 Facebook

Um estudo do Facebook de 2016 aponta que mais de 102 milhões de brasileiros estão registrados em um perfil desta rede social, sendo que 93 milhões destes fazem acesso ao aplicativo via dispositivos móveis. Esse dado indica que o brasileiro possui um perfil conectado às tendências sociais do universo digital (Facebook para Empresas, 2016).

O Facebook possui uma segmentação de seu negócio para desenvolvedores que queiram utilizar seus recursos para acesso, em outras aplicações, como estratégia de difundir ainda mais a marca. A ferramenta dispõe acesso ao *login* pela conta do Facebook do usuário, compartilhamento da aplicação no perfil do usuário, Facebook Analytics para análise de desempenho do

aplicativo integrado ao Facebook, formas de monetizar a aplicação e integração com o *Messenger*.

Um desenvolvedor pode vincular sua conta do Facebook ao Facebook Developer para utilizar desses recursos gratuitamente conectando-os às suas aplicações, permitindo que crie diversas aplicações integradas a múltiplas plataformas, utilizando um ID gerado para identificação da aplicação em questão. Então o desenvolvedor pode controlar a aplicação e suas estatísticas por meio de um painel da aplicação.

Figura 3 – Código de inserção do Facebook na aplicação

```
<meta-data
    android:name="com.facebook.sdk.ApplicationId"
    android:value="@string/facebook_app_id" />

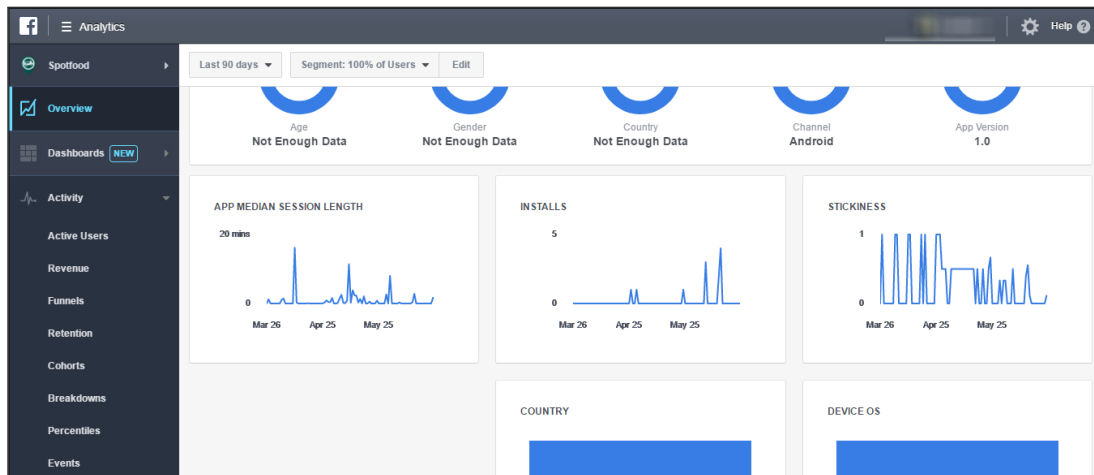
<activity
    android:name="com.facebook.FacebookActivity"
    android:configChanges="keyboard|keyboardHidden|screenLayout|screenSize|orientation"
    android:label="@string/app_name" />
<activity
    android:name="com.facebook.CustomTabActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />

        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />

        <data android:scheme="@string/fb_login_protocol_scheme" />
    </intent-filter>
</activity>
```

Fonte: Autora

Para integrar com a aplicação do Android (Figura 3), é necessário indicar o *package name*, nome da classe que fará a chamada da API e uma *key hash* gerada pela máquina na qual o projeto iniciou desenvolvimento no painel do desenvolvedor do Facebook (plataforma de acesso da ferramenta) e indicar na aplicação qual a chave de identificação da API na *manifest* (arquivo de definições do projeto Android).

Figura 4 – Painel do *Facebook Analytics*

Fonte: *Facebook Analytics*

A integração com a API do Facebook na aplicação permite que o desenvolvedor tenha acesso a um painel analítico, que reúne informações a respeito do uso da aplicação. Como pode ser visualizado na Figura 4, o painel do Facebook Analytics exibe gráficos com informações estatísticas gerais da aplicação que apontam a atividade dos usuários mensal, semanal e diariamente, tal como o número de usuários únicos, usuários ativos, número de instalações e quantas vezes a aplicação é aberta. Informa ainda dados demográficos dos usuários, tais como: idade média, gênero, país, plataforma de dispositivo utilizada e versão do aplicativo lançado.

### 2.2.3 Google Maps

Há algumas vantagens em relação a desenvolver em uma plataforma que já possui integração com o Google, como é o caso deste projeto em Android. O Google Maps API é um exemplo disso. É um pacote já oferecido por padrão para uso de mapa em aplicativos Android que faz renderização de um mapa que pode ser percorrido pelo usuário. Para utilizar a ferramenta, basta gerar uma chave de identificação pelo console do Google Maps API para que a aplicação seja reconhecida e tenha permissão de utilizar o recurso. Da mesma maneira que o Facebook, esta chave deve ser indicada no *manifest* do projeto (Figura 5).

Figura 5 – Código de inserção do Google Maps na aplicação

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="@string/google_maps_key" />
```

Fonte: Autora



E para reconhecer a localização do usuário, é preciso ter a permissão do acesso ao GPS do dispositivo e garantir que este se encontra ativo (Figura 6 e Figura 7). Caso isso não esteja garantido, a aplicação mostrará apenas o mapa em escala global.

Figura 6 – Código de inserção do Google Maps na aplicação

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Fonte: Autora

Figura 7 – Código de inserção do Google Maps na aplicação

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    mMap.setOnMarkerDragListener(this);

    //Initialize Google Play Services
    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (ContextCompat.checkSelfPermission(this,
            Manifest.permission.ACCESS_FINE_LOCATION)
            == PackageManager.PERMISSION_GRANTED) {
            mMap.setMyLocationEnabled(true);
        }
    }
    else {
        mMap.setMyLocationEnabled(true);
    }

    updateLocations();
}
```

Fonte: Autora

## 2.2.4 Firebase

Tendo em vista que o Firebase é um serviço de armazenamento *online*, portanto é possível assumir que a aplicação não funcionará sem conexão à Internet, o que pode ser um limitante ao usuário. Mas segundo Villela (2017), a forma principal de acesso à internet dos brasileiros é justamente via *smartphone*. Visto isso, podemos assumir que apesar da limitação, o uso da ferramenta pode ser explorado por vias melhores do que realizando armazenamento local no dispositivo.

O Firebase possui diversos serviços para aplicativos em dispositivos móveis em tempo real, tais como: ferramenta analítica, autenticação, banco de dados, armazenamento, mensagens

na nuvem, envio de notificações e outras ferramentas para aprimoramento de funcionalidades de aplicações. Segundo Sevilleja (2016), o Firebase é uma combinação dos melhores serviços da *Google* para desenvolvimento.

É possível utilizar cada um dos serviços separadamente durante o desenvolvimento de um aplicativo, o que causa maior ganho de desempenho do que utilizar um pacote fechado com funcionalidades que não são utilizadas. Para este projeto, utilizaremos a função de autenticação (integrada ao Facebook), banco de dados e armazenamento de arquivos. Para integrar a ferramenta ao aplicativo, não foi necessário gerar uma chave de acesso e inseri-la ao *manifest*, como foi o caso das APIs do Facebook e do Google Maps. Bastou inserir o pacote do projeto nas configurações e, tendo acesso a conta Google referente ao projeto, a IDE Android Studio faz a conexão entre as plataformas automaticamente. Com isso realizado, as bibliotecas do Firebase são importadas ao projeto e podem ser utilizadas.

Primeiramente, integramos a autenticação do Facebook à ferramenta de autenticação do Firebase. Basicamente, ela faz controle de todos os usuários que estão utilizando a aplicação, sem que isso interfira no método de *login*. No caso deste projeto, o único modo de registro é por meio do Facebook, então não haverá exemplos de como o controle acontece quando o registro é realizado via e-mail ou outros métodos.

Os primeiros testes realizados em relação a guardar e resgatar localizações específicas no banco de dados do Firebase. É possível visualizar todos os dados armazenados por meio do console do Firebase, e ainda, inserir os dados diretamente pelo console ou fazer importação de um arquivo JSON válido, também pelo console. Como teste, criamos alguns pontos de venda fictícios pela inserção direta no console. O mesmo esquema de inserção vale para o armazenamento de arquivos. No caso, utilizamos essa ferramenta para armazenamento de fotos enviadas pelos usuários para complementar informações a respeito dos pontos de venda.

Neste capítulo foram discutidos conceitos importantes para a compreensão do trabalho e foi explicado o funcionamento e motivo pelos quais as ferramentas utilizadas no desenvolvimento do aplicativo foram selecionadas. No próximo capítulo, é descrita a metodologia utilizada no trabalho, para concepção do aplicativo e fomento à proposta de modelo de análise.

### 3 METODOLOGIA

Este capítulo descreve a metodologia de desenvolvimento deste trabalho. Na Seção 3.1 explicamos a metodologia de criação e implementação do aplicativo de mapeamento de lancherias móveis e *food trucks* e suas tecnologias. Na Seção 3.2 explicamos de que forma a arquitetura para análise de perfil é proposta, baseada na aplicação e os dados que são gerados por esta.

#### 3.1 MVP DO APLICATIVO

O trabalho de Münch et al. (2013) introduz o conceito de Mínimo Produto Viável (*Minimum Viable Product* — MVP), onde a cada ciclo de implementação são considerados os mínimos requisitos para realizar uma venda que dê sustentabilidade ao empreendimento — para estudo de caso em ambiente colaborativo de indústria e academia. O trabalho apresenta uma visão geral de benefícios e limitações dessa metodologia para pesquisas acadêmicas com usuários utilizando esquemas de organização de desenvolvimento, entregas de prototipação rápidas para testes e recebimento de *feedbacks* de valor relacionado ao produto dos usuários, gerando insumos para melhorias do produto.

O desenvolvimento deste projeto deu-se pelo desdobramento de um planejamento prévio para estruturação da aplicação a ser implementada, utilizando a metodologia de MVP para criação do produto — aplicativo capaz de gerar dados sobre seus usuários. Este MVP é composto pela conceituação da aplicação segundo uma visão de negócio, e pelas ferramentas necessárias para construção de seu protótipo.

O projeto então recebeu o nome de *Spotfood* para ser comercialmente reconhecido. O conceito do nome deve-se a uma alusão ao fato de que o usuário busca pontos de venda para lanches, logo “*spot*” como referência ao ponto e “*food*” como referência à comida vendida. A visão de negócio é, portanto: “Para os famintos da noite que precisam encontrar um ponto de venda de lanches, o Spotfood é um aplicativo mobile colaborativo que facilita encontrar o ponto de venda mais próximo”.

Conciliando o conceito de visão de negócio com objetivos acadêmicos para esta pesquisa, foi necessário estipular o sistema operacional mais propício e que, portanto, tivesse maior número de usuários possíveis para a realização dos testes. Por isso, foi decidido que a primeira aplicação seria desenvolvida para o sistema operacional Android para dispositivos móveis.

A partir disso, seguimos ao próximo objetivo essencial desta pesquisa: coletar dados de usuários para análise. A forma mais eficiente de coletar dados do usuário é fazê-lo de forma que isso não demande tempo deste, fazendo com que o desinteresse de preencher formulários o impeçam de colaborar com a pesquisa. Ou seja, ao invés de solicitar ao usuário que preencha campos de pesquisa, o sistema faz isso automaticamente por meio de ferramentas disponham práticas de fazê-lo, coletando dados que já foram previamente preenchidos, em um perfil de rede social. Em razão disso, selecionamos o SDK do Facebook para realizar a coleta de dados

referentes ao usuário.

Definida a plataforma de desenvolvimento e a ferramenta para coleta de dados, nos encaminhamos ao terceiro objetivo deste projeto: mostrar localização de pontos de vendas de comida próximos ao usuário. Empiricamente é possível afirmar que a melhor forma de apontar uma localização em âmbito municipal, ao menos, é por meio de um mapa. Por esse motivo, escolhemos a API do Google Maps para fazer a relação entre usuário e pontos de venda.

Por fim, para que o projeto cumprisse todos os objetivos contidos, foi necessário encontrar uma forma de distribuir todas as informações geradas a respeito dos pontos de venda aos usuários em tempo real. Um servidor local poderia ser uma solução, porém o custo de obter um servidor que ficasse disponível para o serviço durante semanas estava fora de cogitação para este projeto. Com recursos financeiros limitados, a alternativa seria um serviço externo que o fizesse. O Firebase é uma plataforma online de banco de dados e armazenamento de mídias para aplicações, que recebe e envia dados em tempo real. Para o propósito de manter a aplicação colaborativa entre os usuários, o Firebase mostra-se a ferramenta ideal para o serviço.

Com todas as ferramentas necessárias selecionadas, foi realizada uma lista de objetivos com a ordem de execução das *features* que constam no aplicativo. Para tanto, foi seguida a lógica de implementação das ferramentas citadas: criar o projeto para Android, utilizando a IDE *Android Studio*; fazer método de registro do usuário utilizando a API do Facebook, integrar o mapa do Google Maps e inserir o Firebase como banco de dados e armazenamento de arquivos — tudo isso seguido da implementação de todas as telas e suas funções.

Com o MPV fundamentado, o próximo passo foi definir meios de realizar análise dos dados gerados pela aplicação e armazenados em banco, de forma que o resultado gerado pelo sistema, ou seja, informações de usuários e de pontos de venda, possam ser analisados e categorizados posteriormente.

### 3.2 PROPOSTA DE MODELO PARA GERAÇÃO DE PERFIL

Este trabalho realiza a implementação de um aplicativo protótipo que visa gerar dados de usuários e de pontos de venda que proporcionem oportunidades de estudo e aproveitamento de informações captadas com a finalidade de melhorar tanto a experiência de usuários quanto identificar o perfil de possíveis clientes de lancherias móveis e *food trucks*. Para isso, foi necessário elaborar entidades de usuários e locais (*food trucks* e lancherias móveis) dentro da aplicação que pudessem retornar insumos para uma possível análise.

Com a definição das entidades compostas, foi feito um estudo do conteúdo bibliográfico para propor uma arquitetura que, em hipótese (Rodrigues *et al.*, 2007) é capaz de realizar a geração de perfil do usuário. Em seguida, simulamos em um estudo de caso como essa arquitetura deve se comportar em relação ao recebimento de dados de usuário e contexto para entregar um perfil definido, de acordo com as propriedades definidas. Com isso, é possível compreender como o perfil pode acarretar em melhorias na aplicação tanto em relação à experiência do

usuário quanto às vendas dos pontos de venda.

Sabemos, portanto, que a aplicação realiza coleta de dados dos pontos de venda por meio da conexão com a plataforma Firebase, que faz registro da localização quando o usuário insere um novo ponto de venda no sistema, então temos a entidade *local*, que contém os dados: *id*, latitude, longitude, *name*, *rate* (avaliação), *status* (estado do ponto de venda — aberto ou fechado), *type* (descrição do local), *visitors* (quantidade de *check-ins* que o local já recebeu) e *photos*, que recebe uma lista de fotos. A base de dados, portanto, contém as chaves que acessam os locais que foram adicionados pelos usuários e que são reconhecidos pela aplicação.

Da mesma forma, a entidade *user* também é registrada no banco do Firebase, contendo os atributos *id*, *name*, *e-mail*, *gender* e *facebook-link* (*link* para a página do Facebook do usuário cadastrado). Os demais dados demográficos do usuário são analisados genericamente por meio da ferramenta *Analythics* do Facebook. Apesar disso, salvamos o *link* para página pessoal do usuário para o caso de necessitarmos realizar pesquisas personalizadas de amostragem.

A partir destes objetos (*user* e *local*), é possível realizar adaptações do trabalho de Panagiotakis (2012), descrito na Seção 2.1.1 (Figura 1), tais como em serviço e aplicação, conexão e registro no banco de dados das entidades do projeto. Portanto, a partir do contexto obtido com o uso da aplicação, será possível compreender o comportamento do usuário.

Neste capítulo foram descritas metodologias utilizadas na criação do aplicativo e da proposta de modelo de análise de perfil do usuário. No próximo capítulo, introduzimos as entidades, requisitos do sistema e desenvolvimento do aplicativo, além da adaptação do modelo proposto.

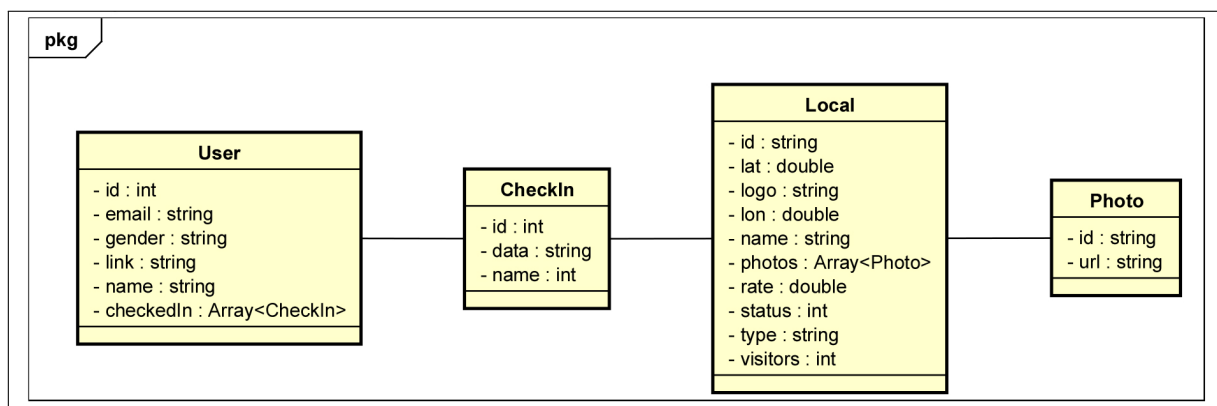
## 4 DESENVOLVIMENTO

Este capítulo descreve o desenvolvimento do projeto. A seção 4.1 descreve os requisitos do sistema e entidades a serem implementados pelo aplicativo. A seção 4.2 explica como foi elaborada a arquitetura de análise de perfil do sistema. A seção 4.3 descreve a implementação do aplicativo com base nas sequências e atividades que o usuário pode realizar, tal como suas classes e explicitação das telas do aplicativo em subseções específicas.

### 4.1 ENTIDADES E REQUISITOS DO APLICATIVO

As entidades principais do aplicativo são: *User*, com os atributos *name*, *email*, *gender*, *link* (para o perfil público do Facebook) e uma lista de *check-ins* dos pontos de venda visitados; *Local* (da lancheria móvel ou *food truck*), com os atributos inseridos pelos usuários: *name*, latitude e longitude atuais, *status* (aberto ou fechado), *type* (descrição do ponto de venda), *rate* (avaliação do local), *logo* (foto principal do ponto de venda) e uma lista de fotos (*Photo*) exibidas no local.

Figura 8 – Diagrama de Relacionamento de Classes



Fonte: Autora

Na Figura 8 é mostrado o relacionamento entre classes das entidades. As entidades *User* e *Local* se relacionam por meio do *check-in*, que possui um *User* e um *Local*, enquanto cada *User* e *Local* podem possuir diversos *check-ins*. Além disso, um *Local* pode estar relacionado a diversas *Photos*. Essas entidades representam conceitos explorados nos requisitos da aplicação.

A análise de requisitos da aplicação é um dos primeiros passos para o desenvolvimento de um *software* pois ajuda a clarificar quais são as funções que o sistema deve ou não realizar. Os requisitos funcionais descrevem as funcionalidades do *software*. Já os requisitos não funcionais definem propriedades e restrições do sistema (Pressman, 2016).

Os requisitos não funcionais deste projeto são:

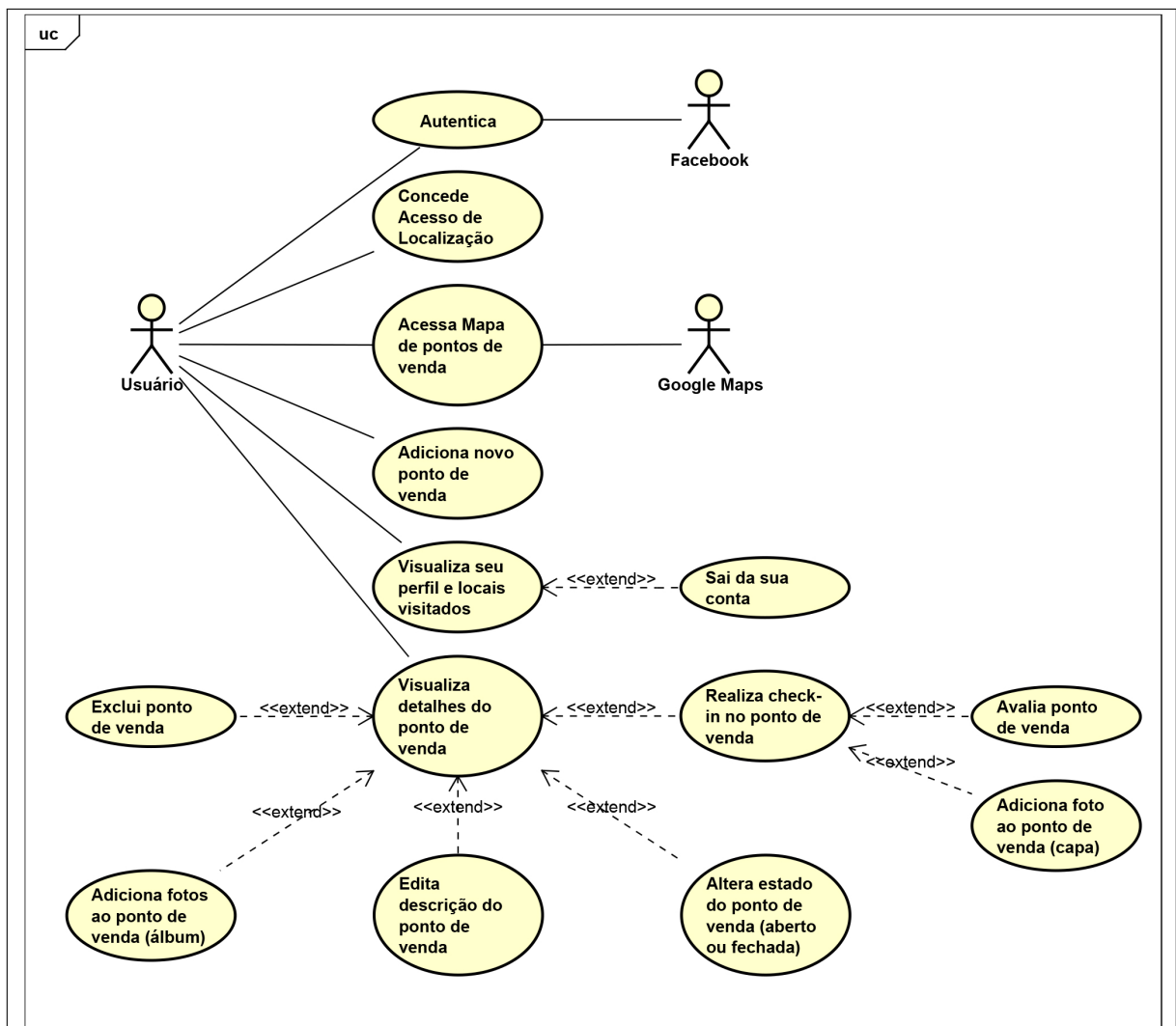
- Evitar erros por falta de conectividade;
- Evitar erros caso o usuário não tenha concedido a localização ou não a tenha habilitado;

- Evitar sobrecarga de dados no aplicativo;
- Armazenar dados gerados pelo usuário.

A aplicação necessita de conectividade para realizar autenticação do usuário e identificar pontos de venda no mapa, portanto, deve evitar falhas por falta de conexão com a Internet. No caso do usuário não permitir ou habilitar a localização, a aplicação não poderá identificar sua posição ou adicionar novos pontos de venda e por isso é necessário tratar esse tipo de situação. A sobrecarga de dados pode causar lentidão e travamento na aplicação, logo, deve ser evitada. E é necessário para o propósito deste trabalho realizar armazenamento dos dados gerados pelo usuário de forma que seja viável realizar análise posterior desses dados. Ao longo do capítulo, os requisitos não funcionais são melhor elucidados.

A Figura 9 denota o diagrama de casos de uso *Spotfood*, explicitando o que o usuário pode realizar pela aplicação, enquanto requisitos funcionais da aplicação.

Figura 9 – Diagrama de Casos de Uso

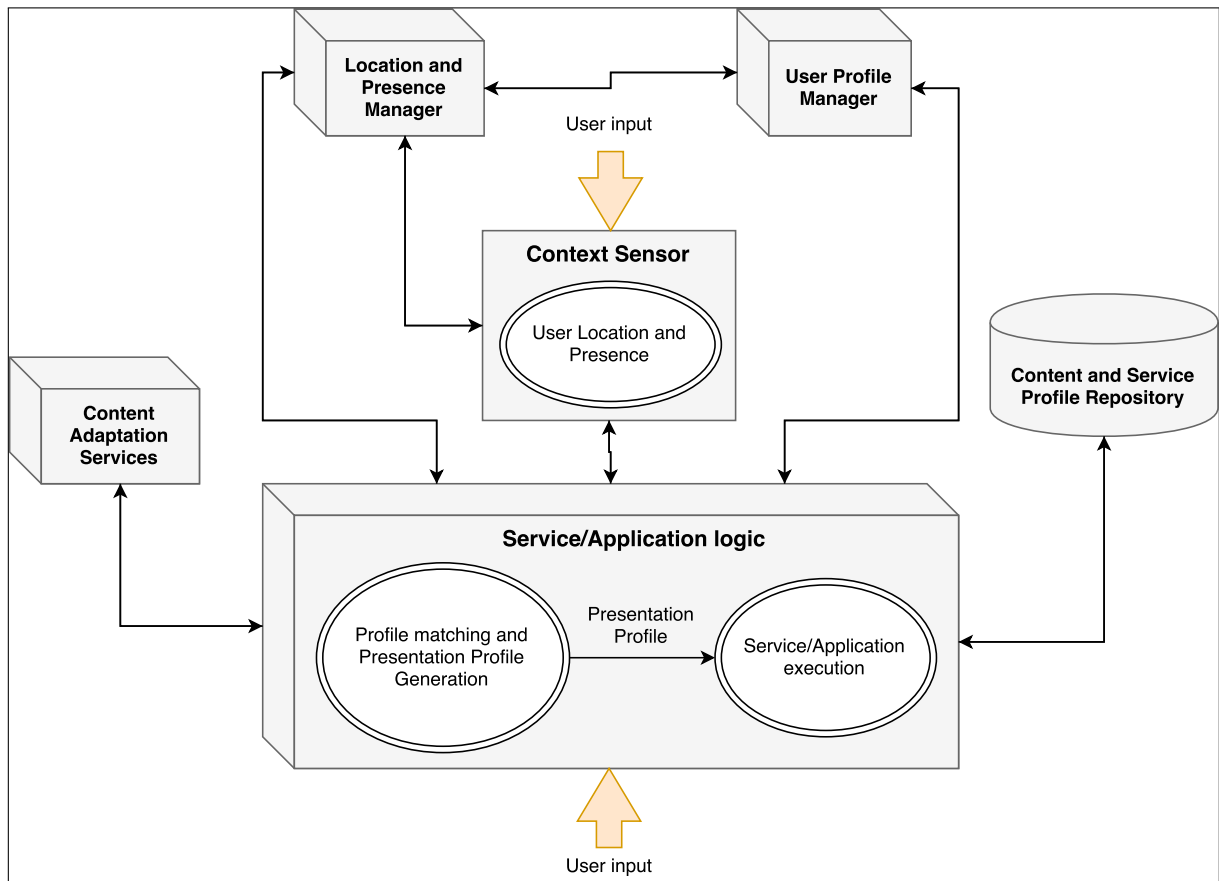


Fonte: Autora

## 4.2 ARQUITETURA DO SISTEMA

A Figura 10 representa a adaptação do *framework* para provisão de serviço *context aware* citado na Seção 2.1, contendo módulos o qual a aplicação foi implementada, tais como serviços a parte propostos para qualificar o modelo completo de análise de contexto e perfil do usuário. O modelo é composto, então, pelo módulo da aplicação, gerenciador de localização do usuário e gerenciador do perfil do usuário, conteúdo de adaptação da aplicação, repositório do conteúdo (banco de dados) e sensor de contexto – localização do usuário. Neste trabalho, os serviços foram modelados de forma que a implementação dos módulos seja abstrata (Chwif e Medina, 2006).

Figura 10 – Adaptação do *framework* para Provisão de Serviço *Context Aware*



Fonte: Autora

O módulo da lógica de aplicação (*Service/Application logic*) é responsável por realizar a execução da aplicação, recebendo portanto as entradas do usuário que está fazendo uso do serviço, por fazer reconhecimento do perfil do usuário e gerar a apresentação adaptada deste usuário durante a execução do aplicativo. Ele compartilha informações com todos os outros módulos para o cumprimento de suas funções.



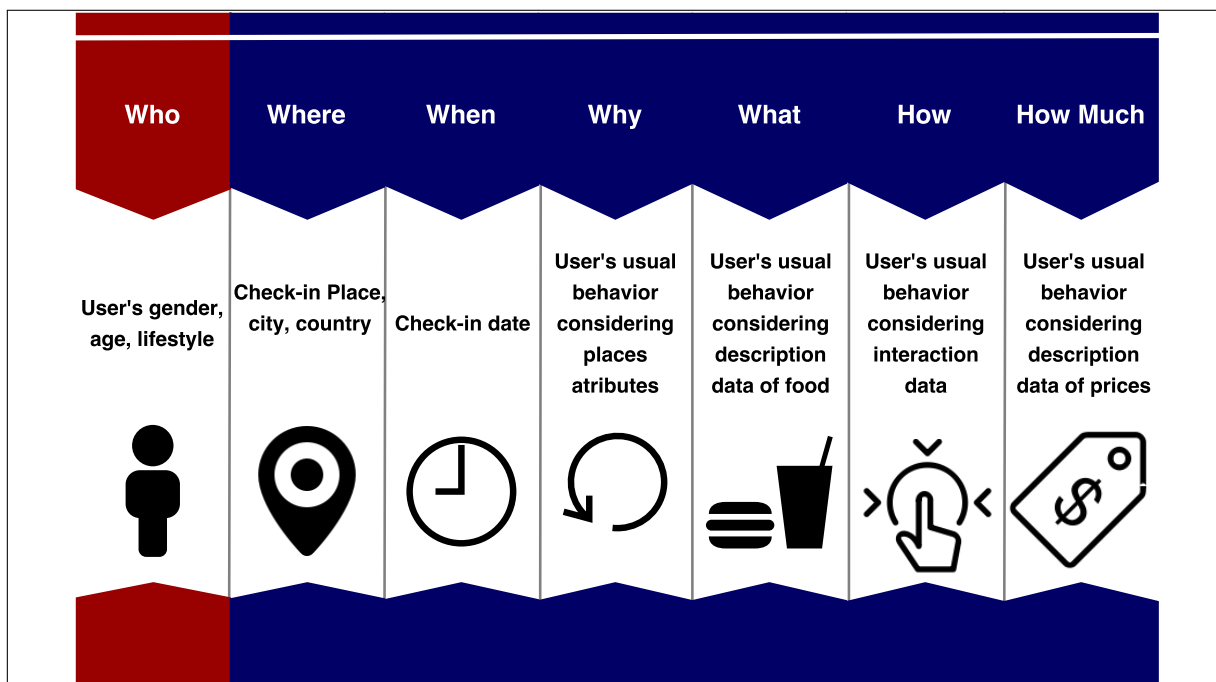
O módulo Context Sensor é responsável por reconhecer a localização e presença do usuário, de forma que o sistema possa desempenhar a geração do contexto atual o qual se encontra. O módulo *Content Adaptation Services* atualiza o sistema conforme as entradas entregues pelos usuários, que geram novos conteúdos a serem gerenciados pela aplicação. *Content and Service Profile Repository* é, portanto, o banco de dados onde os conteúdos gerados são armazenados, permitindo que a aplicação faça acesso do conteúdo salvo.

O módulo Location and Presence Manager é responsável por identificar os pontos de venda da aplicação e seus respectivos *check-ins* para gerenciamento desses dados. Ele recebe insumos de Context Sensor e percebe onde o usuário está no momento do *check-in*, tal como os demais *check-ins* já realizados pelo usuário. Nesta proposta, este módulo é responsável por encontrar padrões entre os *check-ins* realizados pelo usuário, afim de descobrir comportamentos padrões do usuário. Propõe-se que este módulo realize mineração de dados para geração e retorno dos dados.

Nesse sentido, o módulo Location and Presence Manager deverá dispor de algoritmos capazes de reconhecer padrões e associações entre os locais registrados, como avaliação ou palavras contidas na descrição e número de visitantes, se o número de fotos no álbum pode influenciar o aumento de visitas. Além disso, cada usuário pode apresentar um comportamento diferente de acordo com os atributos do local. Por exemplo, o comportamento do usuário pode indicar que ele só visita locais com avaliação acima de 4. Essas associações entre as variáveis da entidade podem determinar parte da categorização de usuários, logo a conexão com módulo User Profile Manager deve permitir que este realize suas associações utilizando os dados dos locais.

O módulo User Profile Manager necessita, portanto, de conexão com o módulo Location and Presence Manager, considerando que sua função é justamente realizar gerenciamento do perfil do usuário, considerando portanto informações demográficas e de personalidade do usuário. Logo, o módulo deve ser capaz de definir uma categorização do usuário baseado em padrões reconhecidos em comportamentos reconhecidos previamente e retornar à aplicação a categoria o qual o usuário ativo pertence.

Figura 11 – Modelo de Categorização 5W2H



Fonte: Autora

O módulo User Profile Manager deverá seguir o modelo descrito na Figura 11 a fim de reunir categorizações de usuários após recolhimento dos dados para análise de perfil do usuário. O foco da categorização é identificar a pessoa utilizando a metodologia 5W2H, sendo QUEM o fator focal da ordem. QUEM, portanto, reúne diz a respeito da personalidade do usuário, como idade, gênero e aspectos do estilo de vida – qual profissão, onde estudou, onde mora, dentro outros. É importante que o módulo esteja delimitado por parâmetros que possam identificar grupos de usuários que possuem o fator QUEM em comum com uma medida razoável, para que as categorias identificadas não sejam muito genéricas, tampouco exclusivas.

A pessoa identificada pelo QUEM, portanto, pode apresentar diversos comportamentos durante a utilização do aplicativo. Esses comportamentos poderão ser definidos pelos fatores ONDE, QUANDO, POR QUÊ, O QUE, COMO e QUANTO. Então perfis individuais dos usuários são aqueles formados pela pessoa dele e seu comportamento de acordo com esses fatores, enquanto as categorias de perfil são conjuntos dessas pessoas que possuem atributos e comportamentos semelhantes.

Portanto, os fatores de categorização possuem regras de definição diferentes.

- O fator ONDE deve representar os locais de *check-in* do usuário, bem como a cidade, país, dentre outros.
- O fator QUANDO deve representar as datas em que foram realizados os *check-ins*.
- O fator POR QUÊ deve considerar informações semelhantes de atributos

que levam o usuário a realizar *check-in* nesses locais, como por exemplo, o usuário sempre fazer *check-in* em locais que tem avaliação acima de 4.

- O fator O QUE deve identificar comportamentos semelhantes do usuário em relação a busca de alimentos, como por exemplo, o usuário sempre faz *check-in* em locais que possuem a palavra “*hot dog*” na descrição.
- O fator COMO deve identificar padrões no comportamento de uso do usuário, como por exemplo, a frequência com que ele usa o aplicativo.
- O fator QUANTO deve identificar padrões de preço em *check-ins* do usuário, como por exemplo, identificar que o usuário costuma fazer *check-ins* em locais que apresentem os valores de R\$10 a R\$20 na descrição.

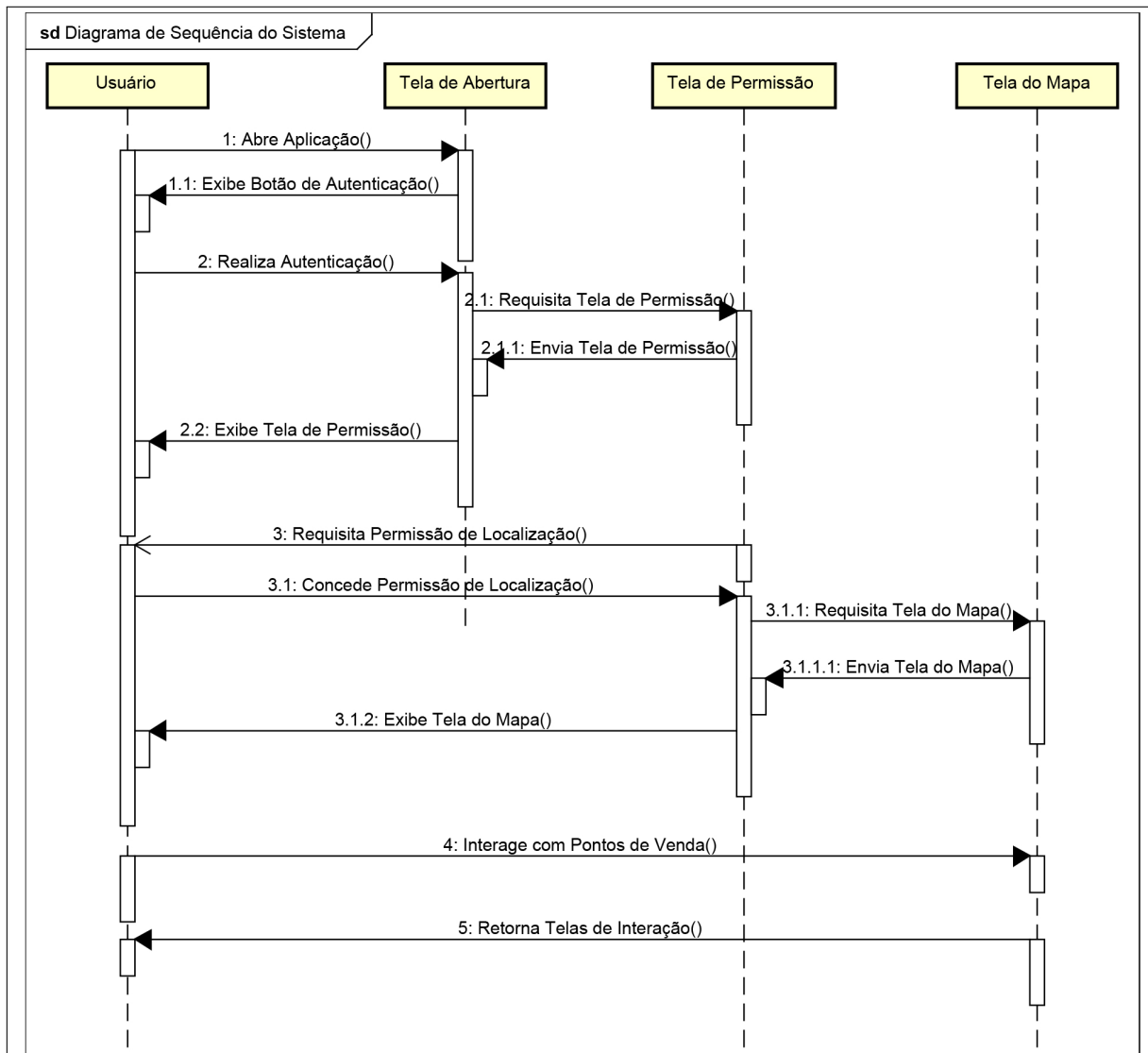
Todos os fatores poderiam ser identificados em condições ideais de uso da aplicação, onde os usuários se dispõem a preencher especificamente os atributos determinados por esses fatores. A exemplo do que seria o fator QUANTO, onde o usuário inseriu os preços dos itens vendidos no local. Já no fator O QUE, o módulo necessitará receber insumos de identificar que as palavras selecionadas se tratam de alimentos no algoritmo a ser utilizado. Mas considerando um cenário pessimista, na qual os usuários não preenchem adequadamente os atributos dos pontos de venda conforme prevê o modelo de análise, alguns dos fatores serão inutilizados.

Este modelo pode ser incorporado neste projeto pelo aplicativo desenvolvido para tal, que realiza o mapeamento de lancherias de rua e *food trucks* com entrada de usuários autenticados na aplicação, onde os mesmos poderão interagir com os locais indicados, de forma a gerar os dados necessários para uma possível experimentação e análise.

### 4.3 IMPLEMENTAÇÃO

Esta seção descreve as definições do aplicativo, a sequência de interações que o usuário deve executar para interagir com os pontos de venda, as classes de implementação do aplicativo, as atividades que o usuário pode realizar interagindo com o aplicativo e subseções que descrevem o funcionamento de cada tela.

Figura 12 – Diagrama de Sequência do Sistema



Fonte: Autora

O Diagrama de Sequências (Figura 12) representa processos de um sistema computacional, retratando as possíveis interações entre entidades, objetos e cenários realizadas por meio de ações, operações ou métodos. A Figura 12 detalha a interação com o aplicativo desde sua abertura, apontando de que forma o sistema deve se comportar conforme as ações do usuário, até a tela de mapas, onde a sequência de passos do usuário torna-se dinâmica, fora de um processo delimitado.

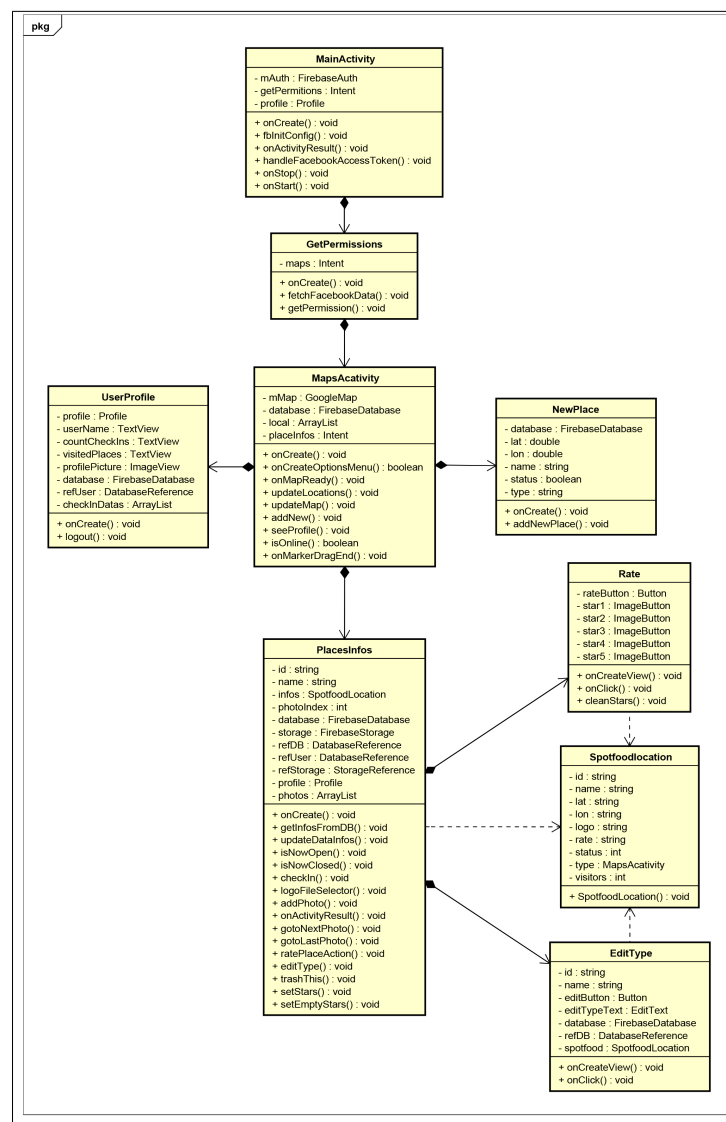
A Implementação do aplicativo é separado por partes prioritárias, iniciando-se pela criação desenvolvimento das *Activities* da aplicação, seguido da integração com APIs e com o servidor. Neste processo, serão levados em conta a implementação requisitos funcionais da aplicação e posteriormente o *front-end do projeto*.

O Diagrama de Classes exibido na Figura 13 mostra o fluxo de classes das telas da

aplicação, tais como seus atributos e métodos principais. A sequência portanto segue da *MainActivity*, que implementa o *login* do usuário, seguindo para a tela de permissão da localização do usuário, *GetPermissions*, que libera o acesso para a tela do mapa de pontos de venda. É necessário que o usuário ceda a sua localização, porque para criar um novo ponto de venda, a aplicação utiliza a localização atual do usuário.

Da tela do mapa, *MapActivity*, a aplicação permite que o usuário acesse seu perfil (*User-Profile*) e adicione um ponto de venda (*NewPlace*) por meio do menu. Ao interagir com o marcador, a *MapsActivity* ainda permite que se reconhecer interação sobre o marcador de um ponto de venda, a aplicação siga para activity *PlaceInfos*, onde o usuário pode visualizar todas as informações a respeito do ponto de venda selecionado. A classe *SpotfoodLocation* agrega todos os atributos referentes ao ponto de venda.

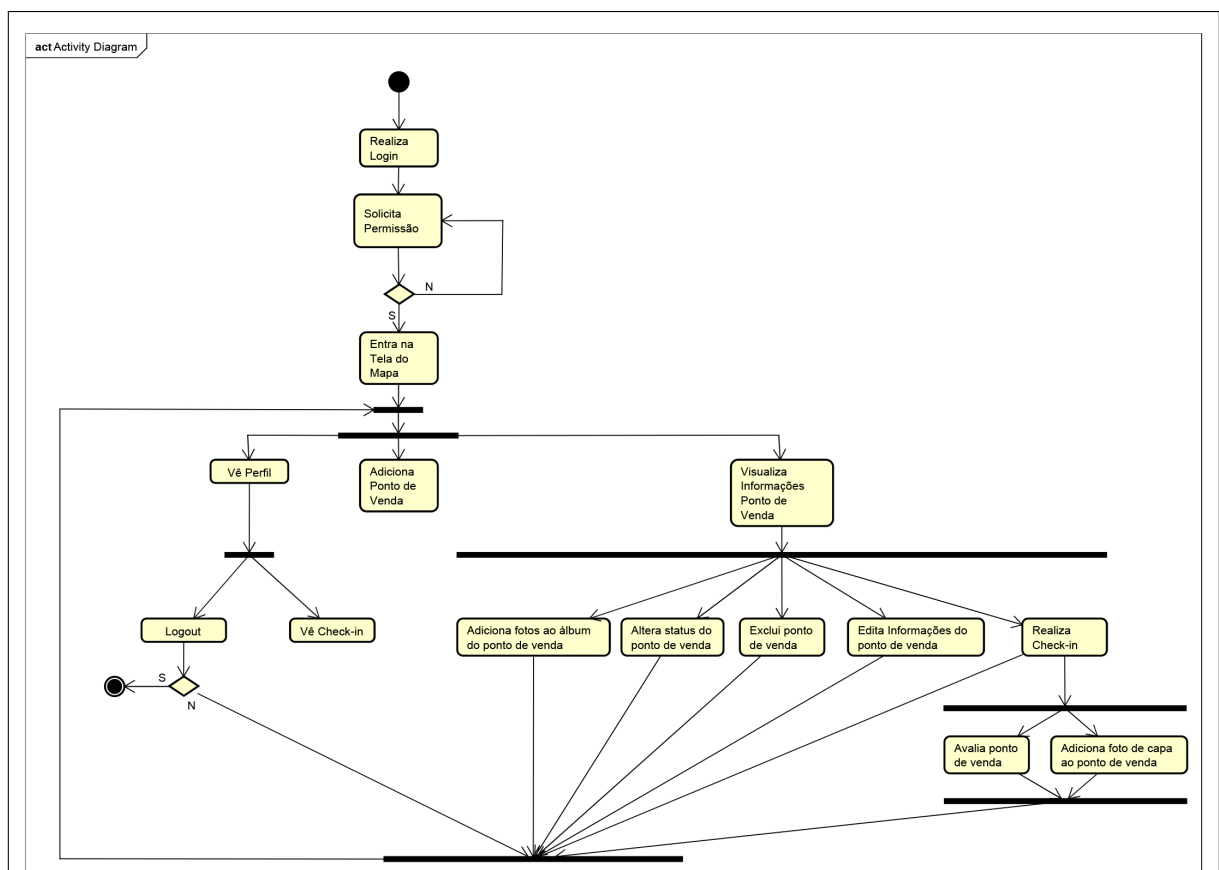
Figura 13 – Diagrama de Classes do Aplicativo



A classe *PlaceInfos* reúne todos os atributos referentes ao ponto de venda selecionado e funções de interação que o usuário pode realizar na tela. As classes *EditType* e *Rate* são modais que se conectam com o ponto de venda em questão, permitindo interação sem que a aplicação saia do contexto atual, ou seja, sempre relacionado ao ponto de venda selecionado pelo usuário. A classe *EditType*, portanto, permite que o usuário faça alterações do tipo e a classe *Rate* permite que o usuário envie sua avaliação.

O Diagrama de Atividades da Figura 14 explicita o fluxo de operações que o usuário pode percorrer na aplicação, a partir de cada tela ou interação. Seguindo este fluxo, portanto, primeiramente é implementada uma tela de *login* do usuário, que poderá se conectar ao aplicativo cadastrando-se por meio das redes sociais. É utilizada a API de desenvolvimento do Facebook e a ferramenta de autenticação do Firebase.

Figura 14 – Diagrama de Atividades



Fonte: Autora

A tela seguinte é de aplicação da API do Maps (Google), na qual é possível visualizar as lancherias já cadastradas. O usuário pode interagir com essa tela, podendo visualizar detalhes dos pontos de venda, entrar em uma das lancherias para mais informações, criar uma nova lancheria ou mover os pontos de lugar no mapa. Os dados relacionados à lancheria estão disponíveis no *Database* do Firebase. A partir desta tela, é possível interagir com outras duas opções

de tela e atualizar o mapa.

Ao selecionar um ponto de venda, o usuário pode visualizar informações como fotos e detalhes como foto de cardápio no álbum e comentários de outros usuários na descrição. Também pode adicionar fotos ao álbum, avaliar o local e realizar *check-in* nessa tela - que constarão em estatísticas sobre o local.

A próxima tela seria a de criação de um novo local, ou seja, permite que o usuário insira um novo local no mapa, contendo informações como nome, cardápio ou até fotos do local. Todas essas informações serão salvas no *Database* do Firebase e serão visíveis para os demais usuários. A última tela desenvolvida no aplicativo foi a de informações do perfil do usuário, que deverá conter informações como histórico dos últimos locais visitados e o número de *check-ins* realizados.

Implementado o aplicativo e realizados os devidos testes, o próximo passo deste projeto é distribuir a aplicação para que possíveis usuários façam reconhecimento e uso do aplicativo. Dessa forma, os dados de comportamento de diferentes usuários já serão automaticamente coletados para análise.

A análise deve ocorrer no decorrer da coleta de dados. Os dados coletados são referentes a duas categorias de informação: pontos de vendas e usuários. Segundo o modelo de análise proposto no Capítulo 3, ambos devem ser gerenciados por seus respectivos módulos de gerenciamento. No caso dos pontos de venda (lancherias de rua, *food trucks*) podem ser analisadas a quantidade de locais cadastrados, a interação de usuários com os locais, segundo o número de *check-ins*, avaliações e fotos registradas por estabelecimento - fatores a serem averiguados pelo módulo *Location and Presente Manager* na fase de experimentação.

Já no caso de usuários, podem ser coletadas informações que dizem respeito ao perfil geral, considerando data de nascimento, gênero, cidade que habita atualmente, histórico de educação e histórico de trabalho - coletados pela conta do Facebook do usuário, e *check-ins* dos locais onde este usuário interagiu. Para a criação do perfil geral do usuário, consideraremos portanto as médias gerais de comportamento dos usuários mais frequentes, considerando idade, gênero, educação e trabalho e relacionando aos tipos de pontos de vendas que eles buscam frequentar. Todos esses fatores deverão ser analisados pelo módulo *User Profile Manager* durante a experimentação.

#### **4.3.1 Tela Inicial: Cadastro/Login**

A tela inicial (Figura 15) possui a função de realizar cadastro/login do usuário no aplicativo. Essa função não terá diferença entre cadastro ou login pois a única forma do usuário acessar as próximas telas é autorizando sua entrada com o Facebook. Isso porque de um dos objetivos centrais desta pesquisa é justamente coletar os dados do usuário para realizar análise do perfil.

Figura 15 – Tela Inicial: Cadastro/Login



Fonte: Autora

#### 4.3.2 Tela de Requisição de Permissão

Esta tela (Figura 17) indica ao usuário que ele deverá ceder permissão de acesso à localização (Figura 18) para que o aplicativo reconheça onde o usuário está naquele momento. Caso o usuário não conceda a permissão, o aplicativo não permitirá que ele siga para a próxima tela. A Figura 16 demonstra o código de requisição de permissão do aplicativo.

Figura 16 – Código de requisição de permissão

```
public void getPermission(View v){
    Button permission = (Button) findViewById(R.id.getPermission);
    if (v.getId() == R.id.getPermission) {

        if (ContextCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_FINE_LOCATION)
            == PackageManager.PERMISSION_GRANTED) {
            startActivity(maps);
        }
    }
}
```

Fonte: Autora



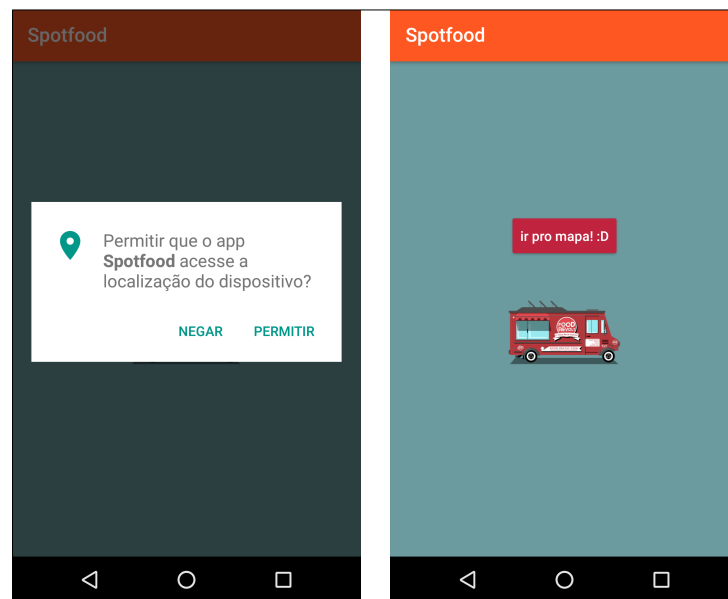
Figura 17 – Tela de Resquisição de Permissão: Usuário deve clicar no botão para que a requisição de permissão ocorra



Fonte: Autora

Além das permissões, esta tela é responsável por resgatar os dados do Facebook do usuário e enviar para o banco de dados do Firebase. Com uma variável de acesso ao banco de dados, esta função pega o *Token* de acesso do Facebook do perfil do usuário e abre uma requisição para acessar os dados que o usuário permitiu o uso. Então ele passa esses dados à variável do banco do Firebase.

Figura 18 – Tela de Resquisição de Permissão: O sistema pede permissão do usuário. Se for concedido, o usuário poderá entrar na tela do mapa.

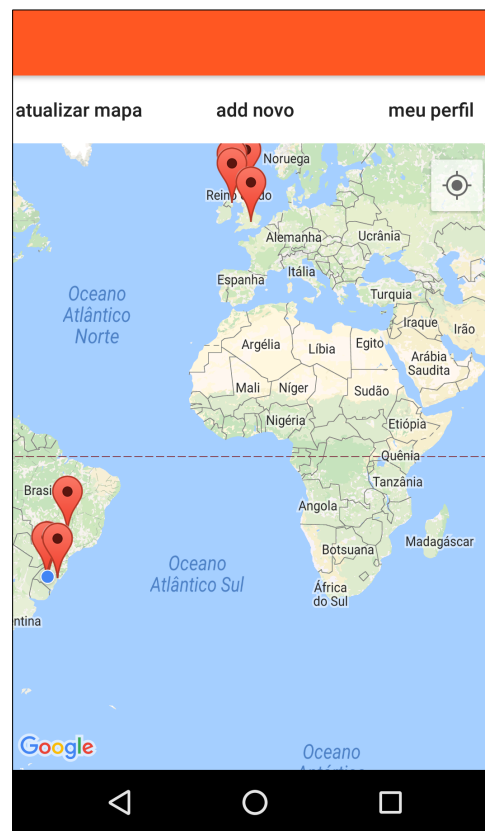


Fonte: Autora

### 4.3.3 Tela do Mapa

Com as permissões de localização e acesso aos dados do usuário garantidos, a tela seguinte carrega um mapa e todos os marcadores que estão salvos no banco de dados do Firebase (Figura 19). Cada marcador representa um ponto de venda, sendo ele lancheria de rua ou *food truck*, de forma que o usuário possa localizá-los no mapa. A tela ainda possui um menu com três botões: um para atualizar o mapa, caso outros usuários tenham cadastrado novos pontos de venda (atualizar mapa), um para cadastrar um novo ponto de venda (adicionar novo) e um para o usuário visualizar seu perfil ou fazer *logoff* (meu perfil).

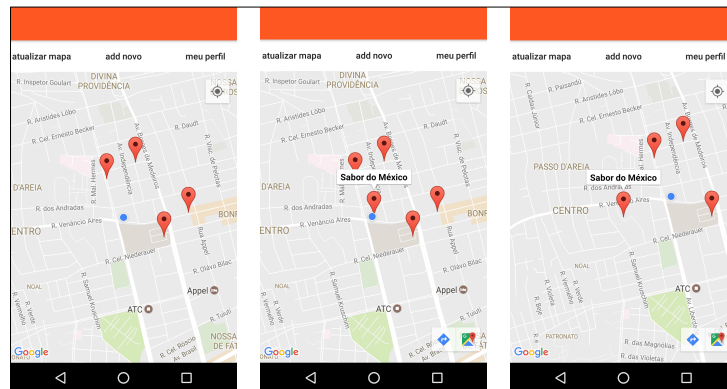
Figura 19 – Tela do Mapa



Fonte: Autora

É possível adicionar um novo marcador ao mapa e ele é atualizado em tempo real para os demais usuários do aplicativo. Quando se adiciona um novo marcador, ele aparece exatamente na localização em que o usuário está posicionado no momento (Figura 20). É possível ainda mover os marcadores do mapa ao arrastá-los pelo mapa, indicando que o ponto de venda está se movendo ou já está em outra localização. Isso também é atualizado automaticamente no banco de dados do Firebase.

Figura 20 – Tela do Mapa: Novo Marcador Inserido

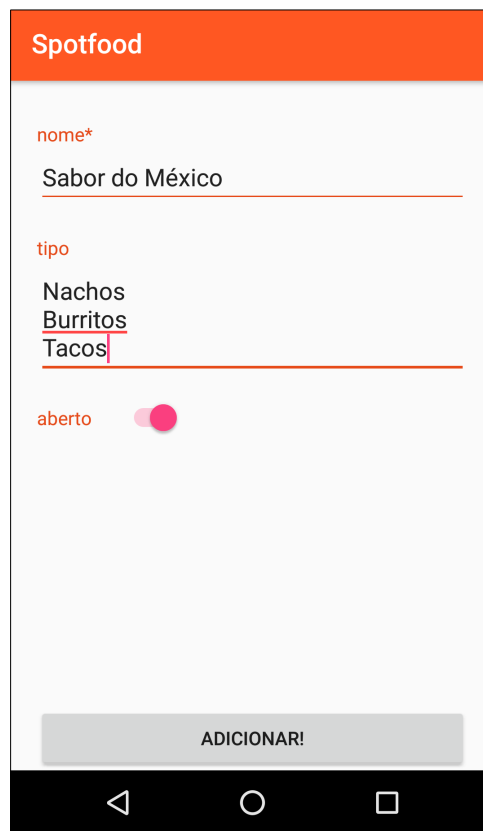


Fonte: Autora

#### 4.3.4 Tela de cadastro de novo Ponto de Venda

A tela de cadastro para novo ponto de venda (Figura 21) conterà apenas 3 itens a serem preenchidos. Isto porque a experiência do usuário é prejudicada quando ele é forçado a preencher longos formulários para uma aplicação em um dispositivo móvel. São esses itens: nome, tipo de produtos vendidos e se está aberto ou fechado no momento, sendo apenas o primeiro obrigatório.

Figura 21 – Tela de cadastro de novo Ponto de Venda



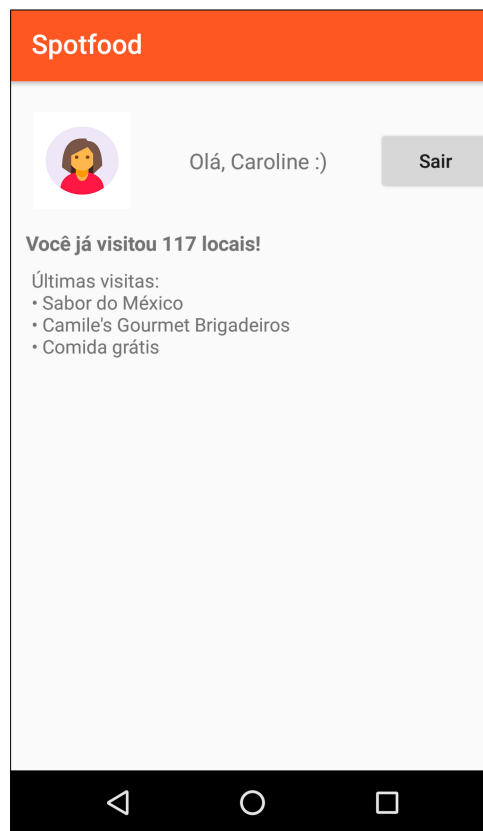
The screenshot shows the registration form for a new point of sale in the Spotfood app. The form is titled "Spotfood" and contains three fields: "nome\*" (name) with the value "Sabor do México", "tipo" (type) with a list of options including "Nachos", "Burritos", and "Tacos" (which is selected), and "aberto" (open) with a toggle switch that is turned on. A button labeled "ADICIONAR!" is located at the bottom of the form. The app's Android navigation bar is visible at the bottom.

Fonte: Autora

#### 4.3.5 Tela de perfil do usuário

A tela de perfil do usuário (Figura 22) exibe a foto de perfil dele do Facebook e o nome, indicando quem está logado no momento; o número de locais que o usuário já visitou até o momento e os últimos *check-ins* do usuário. Além disso, é nessa tela que o usuário pode realizar *logoff* da conta no aplicativo. Caso isso ocorra, ele volta para a tela inicial.

Figura 22 – Tela de perfil do usuário

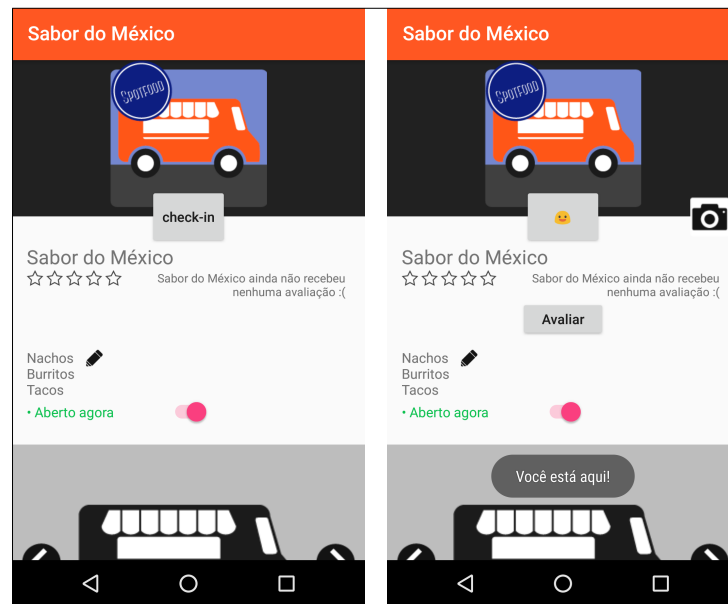


Fonte: Autora

#### 4.3.6 Tela de Informações do Ponto de Venda

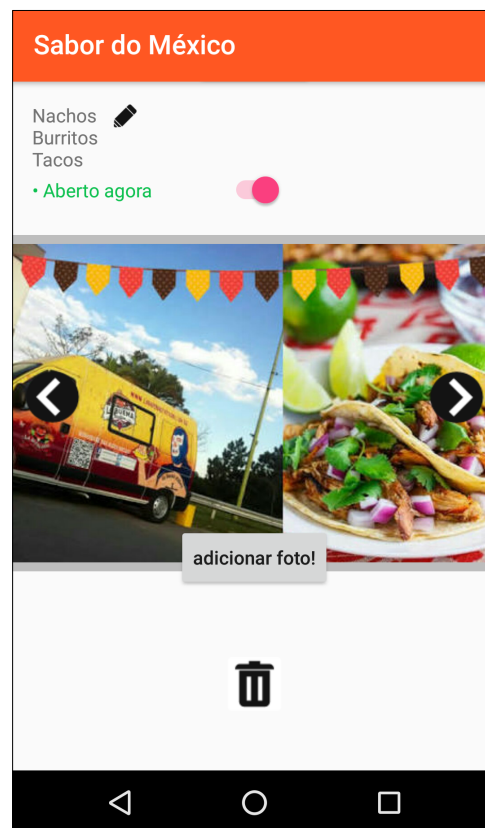
A tela de informações (Figura 23) do ponto de venda é a mais complexa em questão de informações contidas e interação do usuário com o aplicativo. Ela demonstra detalhadamente tudo sobre o ponto de venda selecionado, com os seguintes itens: nome, avaliação, tipo de comida vendida, se está aberto ou fechado, foto de capa do local, fotos do álbum no local, um botão para o *check-in* do usuário e um botão para excluir o local do mapa (caso não exista mais) (Figura 24).

Figura 23 – Tela de Informações do Ponto de Venda: *Check-in*



Fonte: Autora

Figura 24 – Tela de Informações do Ponto de Venda: Adicionar foto ao álbum e botão de excluir

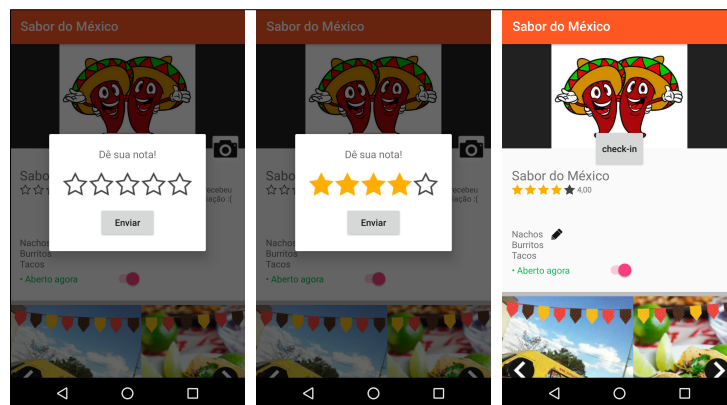


Fonte: Autora

A tela abre com alguns botões desativados, permitindo que o usuário só possa interagir com a tela depois que todos os itens tiverem sido carregados da nuvem. O botão de *check-in* vai registrar a visita do usuário ao local e liberar acesso aos botões de alterar foto de capa e avaliar o local. Mas o usuário só poderá fazer *check-in* no ponto de venda caso ele esteja aberto. Para abrir ou fechar o estabelecimento, o usuário deve alterar o status do switch (interruptor).

A avaliação do ponto de venda (Figura 25) é feita num *fragment* em forma de *dialog*, sendo cada estrela um botão que indica a nota do usuário de 1 a 5. Se o usuário não selecionar nenhuma estrela e enviar a nota, o sistema fará registro da nota como 0. Então, automaticamente é gerada uma média com o número de visitantes que já fizeram *check-in* no local.

Figura 25 – Tela de Informações do Ponto de Venda: Avaliação

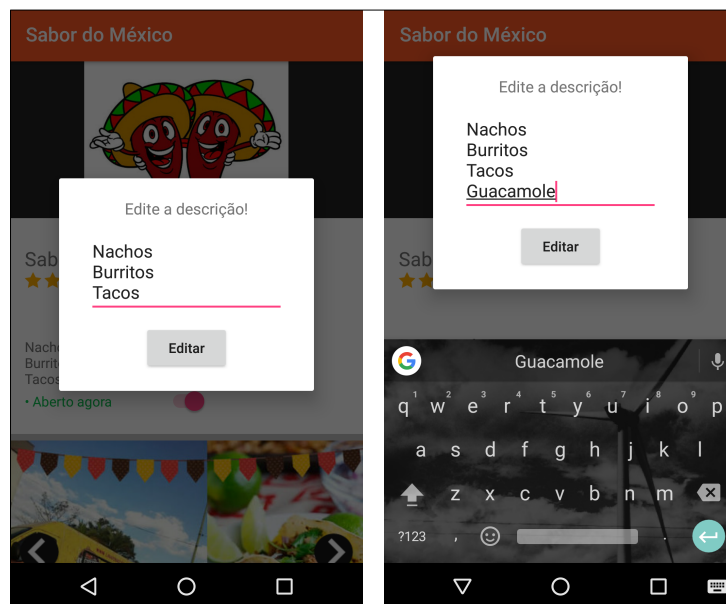


Fonte: Autora

O botão de editar também abre um *dialog fragment* com um *EditText* que permite que o usuário faça alterações no tipo de comida vendida pelo ponto de venda (Figura 26). O *fragment* mostra qual a descrição atual dos lanches oferecidos pelo local e permite que o usuário apague ou escreva informações adicionais.



Figura 26 – Tela de Informações do Ponto de Venda: Editar descrição



Fonte: Autora

O botão da câmera aparece próximo ao botão do *check-in* após o usuário registrar sua visita ao local. Ao ser pressionado, ele abre a galeria de fotos de preferência do usuário e permita que este escolha uma imagem para enviar ao servidor da aplicação. O *storage* do Firebase recebe automaticamente a imagem selecionada e a aplicação avisa o usuário de que sua imagem foi enviada com sucesso caso o *upload* tenha ocorrido e a foto selecionada fica no *display* de capa - que pode representar uma foto do local ou mesmo a marca do ponto de venda. O mesmo ocorre com as imagens que o usuário envia ao álbum de fotos do local.

Todas essas imagens possuem um endereço que são registrados no banco de dados do Firebase para que possam ser carregadas pelos demais usuários da aplicação quando estes acessarem a tela de informações.

Neste capítulo descrevemos o desenvolvimento do projeto. Explicamos tanto as entidades e requisitos do sistema, de que forma o modelo de análise de perfil deve ser composto e a implementação do aplicativo. No próximo capítulo, fazemos uma análise de dados gerados pelo aplicativo e um estudo de caso de como o modelo proposto para análise de perfil deve se comportar com os dados recebidos.

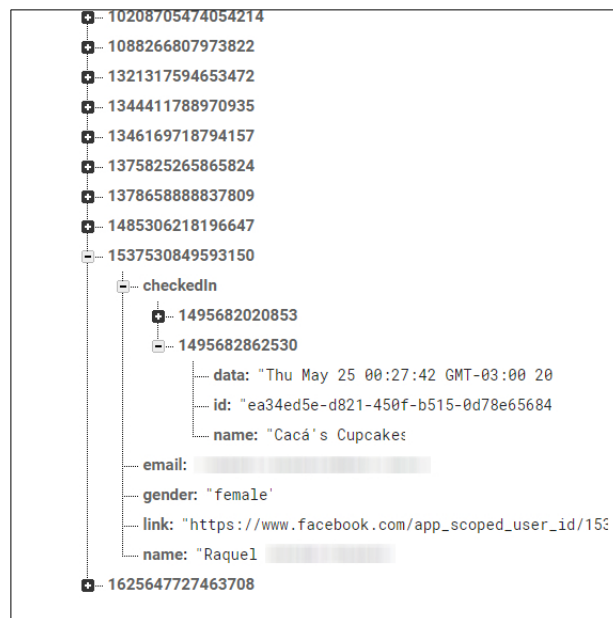
## 5 ANÁLISE E ESTUDO DE CASO

Para a primeira parte deste estudo, foi feita uma distribuição *beta* para testes com usuários, com a finalidade de gerar dados que fornecessem insumos à análise deste projeto. Em seguida, foram selecionados objetos do resultado dos testes, no caso, um objeto de usuário e um local para a primeira parte do experimento, cujo nível de interação correspondesse ao que experimento necessita. O sobrenome e e-mail da usuária foram ocultados para sua preservação.

Para este projeto não foi possível captar automaticamente informações mais precisas do usuário por questões burocráticas exigidas pela plataforma do Facebook, tais como cidade onde reside, data de nascimento para encontrar a faixa etária, histórico acadêmico ou de trabalho; mas idealmente, caso a aplicação venha a resgatar essa informação, o módulo User Profile Manager poderá gerar categorias mais precisas dos usuários. Informações que podem constar na plataforma como histórico de trabalho e de ensino do usuário também poderiam agregar mais conhecimento em prol das categorizações.

Os objetos da entidade do usuário são registrados conforme mostrado na Figura 27. As chaves do objeto são o número de identificação do usuário registrado no *Facebook*. Os atributos registrados são os mesmos apresentados na Figura 8, capítulo 4 da entidade *User*.

Figura 27 – Exemplo de Registro de Objeto de Usuário

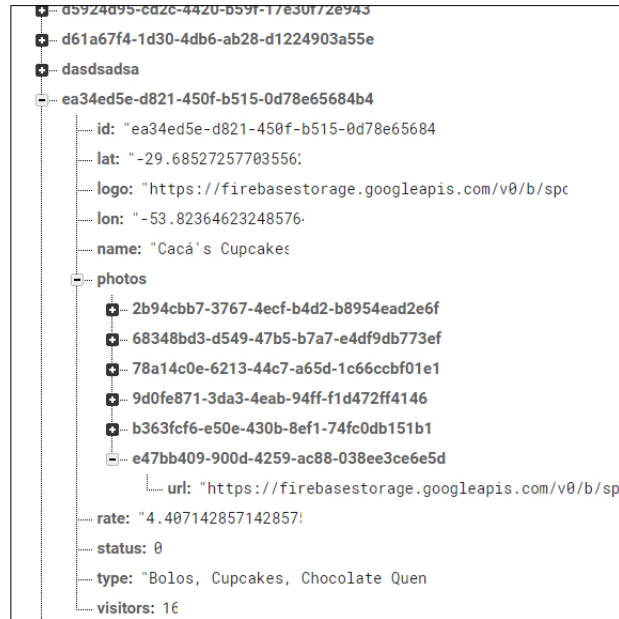


Fonte: Autora

A Figura 28 denota o objeto do local referenciado na Figura 27. Conforme o modelo proposto pela Figura 10, o módulo Locations and Presence Manager deve ficar encarregado de situar que a usuária “Raquel” fez seu último *check-in* no local “*Cacá's Cupcakes*”, que por sua vez possui diversos atributos que podem representar determinados fatores na decisão de Raquel de visitar este local, como por exemplo, estar na cidade que Raquel está ou reside, possuir

algumas fotos no seu álbum que possam ter chamado atenção de Raquel, possuir uma avaliação positiva acima de 4 e servir alimentos como Bolos, *Cupcakes* e Chocolate Quente, que apeteçam o apetite de Raquel.

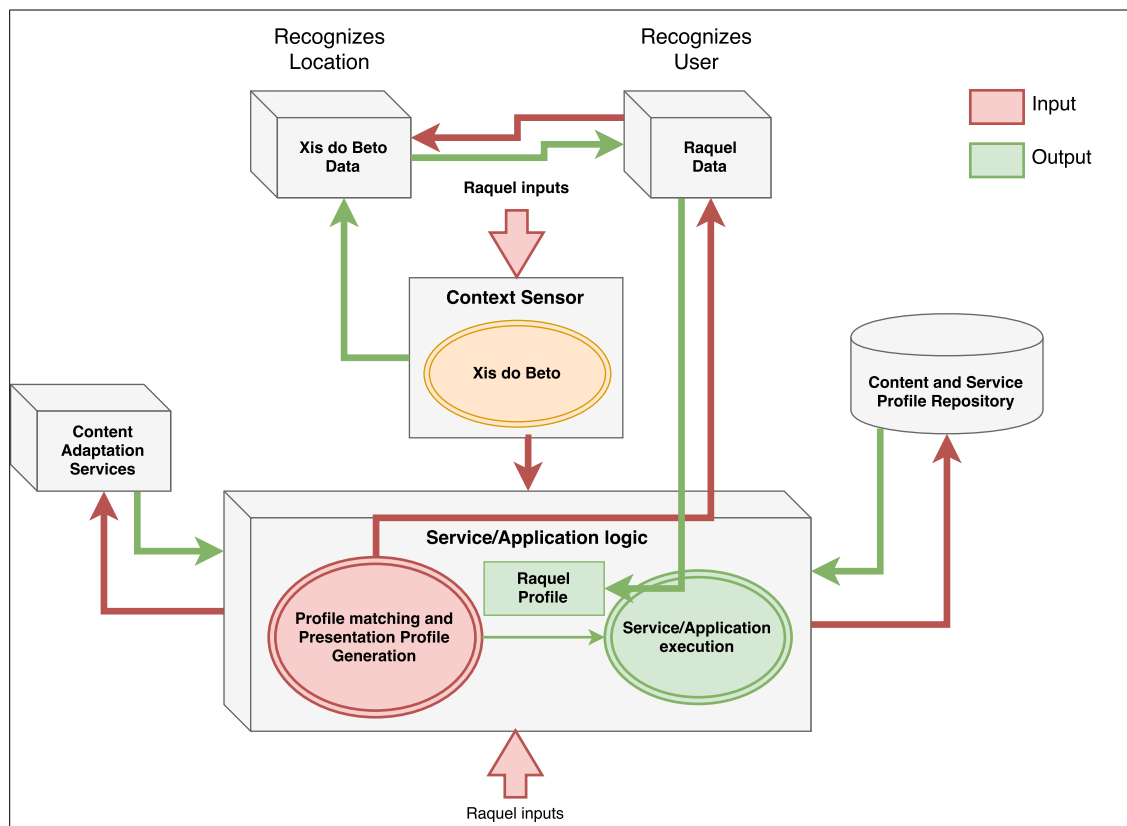
Figura 28 – Exemplo de Registro de Objeto do Local



Fonte: Autora

Para a definição desse perfil, segundo o modelo de geração de perfil do usuário, a aplicação deve seguir o fluxo mostrado na Figura 29, de forma que os algoritmos dos módulos Location and Presence Manager e User Profile Manager retornem um perfil baseado na metodologia 5W2H. A Figura 30 esboça um exemplo hipotético de como um perfil pode ser determinado, considerando portanto o histórico do usuário bem como o descobrimento de seus padrões comportamentais, além de conteúdo da pessoa que considera nome, gênero, faixa etária, entre outros.

Figura 29 – Processo de Execução e Geração do Perfil do Usuário

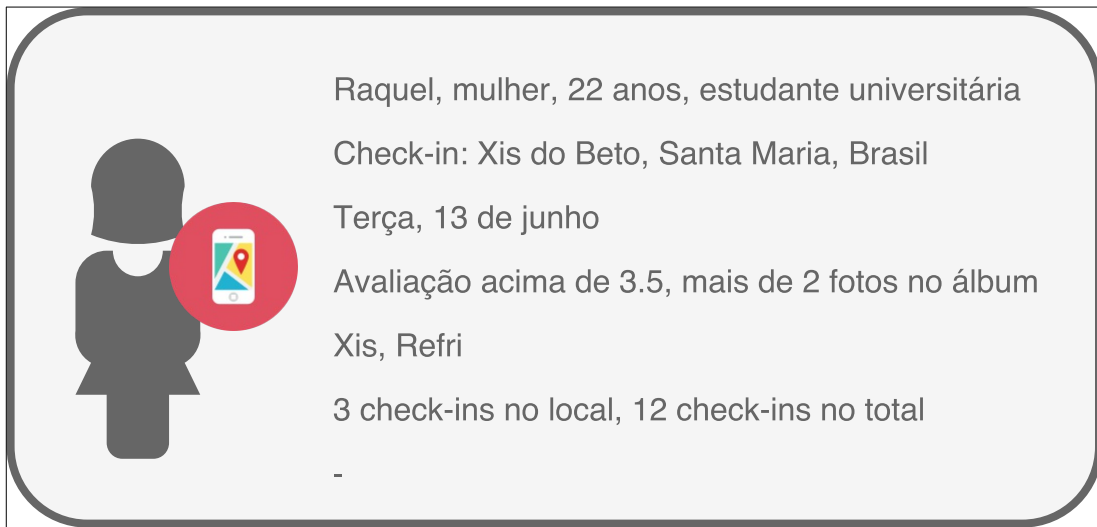


Fonte: Autora

O fluxo de análise (Figura 29) funciona, portanto, com a entrada de dados do usuário no módulo da aplicação (Service/Application logic), este por sua vez envia uma chamada ao módulo *User Manager Profile*, que requisita os dados do local em questão ao módulo Location and Presence Manager. Este recebe os dados do sensor (localização atual do usuário), gera insumos sobre o local e retorna ao User Profile Manager, que elabora o perfil conforme o 5W2H, e envia a informação de volta ao módulo da aplicação.

Dessa forma o sistema é capaz de reconhecer o perfil de seu usuário no dispositivo, que pode sofrer alterações a cada interação, tal como o desenvolvedor utilizar os dados gerados pelo módulo do User Profile a fim de encontrar novas técnicas de aprimoramento da aplicação de forma a se adequar ao costume dos usuários. De maneira abstrata, pode-se dizer que neste exemplo o sistema identificou padrões no comportamento da usuária Raquel e que seu perfil, portanto, é discriminado pela quantidade de *check-ins* realizados no último local visitado. A partir desse momento pode ser o contexto e comportamento da usuária com a finalidade de aprimorar sua experiência de uso, tal como proporcionar oportunidades aos donos dos pontos de venda.

Figura 30 – Exemplo de Perfil do Usuário conforme último *check-in*



Fonte: Autora

A Figura 29 explicita um exemplo de como o *framework* gera o perfil do usuário, baseado nos dados que o sistema já possui deste usuário e com a entrada de novos dados na aplicação. O exemplo de perfil individual da Figura 30 é a sugestão de uma possível entrada de dados de um usuário, cuja persona é Raquel, mulher, 22 anos e estudante universitária. A aplicação deve identificar que este usuário já possui interações prévias com a aplicação e isso permite um resgate do histórico dessa usuária, tal como identificação de alguns padrões.

O exemplo mostra o usuário que já realizou 12 *check-ins* no total, o que permite aos módulos Location and Presence Manager encontrar padrões nos locais selecionados, como os locais em que o usuário costuma checar possuem avaliação acima de 3.5, mais de duas fotos no álbum e que normalmente as palavras encontradas nas descrições são “xis” e “refri”. O módulo retorna essas informações ao User Profile Manager, junto com dados de localização e data, que permitem que este módulo faça a configuração dos dados de acordo com a metodologia usada para categorização do perfil (descrita na Seção 4.2, Figura 11).

Por exemplo, numa análise hipotética do perfil gerado, Raquel, 22 anos, estudante universitária, esteve em 12 pontos de venda enquanto usou o aplicativo, sendo 3 dessas visitas no Xis do Beto, em Santa Maria, Brasil. Os fatores QUEM e QUANDO identificados podem propagar insumos a respeito de seu comportamento, de acordo com os últimos locais que Raquel visitou. A frequência no número de *check-ins* neste local pode apontar que Raquel pode ser uma cliente fidelizada do Xis do Beto. Mas se esses *check-ins* foram realizados entre longos períodos de tempo, a informação anterior pode ser invalidada. Neste caso, se o dono do Xis do Beto tomar ciência do comportamento de Raquel, ele pode buscar práticas para tornar suas visitas mais frequentes.

Pode-se dizer também que Raquel costuma frequentar locais que vendem Xis, já que está entre as palavras identificadas com mais recorrências em suas visitas. Além disso, é possível

assumir que Raquel prioriza visitar locais que tenham uma avaliação positiva acima de 3.5 e com mais de duas fotos no álbum por serem atributos que demonstram credibilidade do local. Dessa forma, ao analisar perfis semelhantes ao de Raquel, os pontos de venda teriam a oportunidade de aprimorar sua página na aplicação, buscando melhores avaliações e adicionar mais fotos para que o fator credibilidade aumente e clientes como Raquel passem a frequentar seu ponto.

O mesmo esquema pode servir para gerar categorizações genéricas de perfis de usuário. Visto que o módulo User Profile Manager seja capaz de reconhecer padrões dentre os perfis individuais já gerados, é possível produzir categorias genéricas das personas e seus respectivos comportamentos. Dessa forma, é possível estipular públicos-alvo para determinados tipos de pontos de venda ou definir sistemas de sugestões para os usuários em versões posteriores do aplicativo.

Neste capítulo fizemos uma análise dos dados gerados pela aplicação e simulamos como esses dados devem ser processados pelo modelo de análise de perfil proposto. Em seguida, chegamos a considerações que podem servir como melhorias para a aplicação e para o negócio de lancherias móveis ou *food trucks*. No próximo capítulo fazemos as considerações finais deste projeto.

## 6 CONCLUSÃO

Neste trabalho foram apresentados conceitos que levaram à criação de um modelo conceitual para geração de perfis de usuário do *Spotfood*, um aplicativo desenvolvido neste projeto que possui o intuito de informar quais pontos de venda de refeição móveis, tais como *food trucks* ou lancherias de rua, mais próximos do usuário, bem como apontar detalhes sobre este estabelecimento, de maneira colaborativa e em tempo real.

O modelo foi trabalhado em cima dos possíveis dados gerados pela aplicação para geração de perfis de usuário, podendo qualificar portanto seu uso e comportamento de forma que haja sustentabilidade no desenvolvimento contínuo e atualização da aplicação conforme descobertas de perfis e públicos-alvo, tanto do próprio sistema, quanto dos pontos de venda. Este modelo teve como base metodologias que utilizam estudo do contexto e reconhecimento de perfil com propósitos definidos.

Para o desenvolvimento da aplicação foram utilizados conceitos e metodologias baseados em um possível modelo de negócios e a respectiva visão do produto. A partir dessa definição, o projeto se encaminhou para a formulação dos objetivos da aplicação, sendo estes receber usuários que pudessem compartilhar informações sobre pontos de venda móveis. Determinou-se então que isso seria feito através de mapas com localizadores que representassem os pontos de venda e que, destes localizadores, o usuário tivesse acesso a detalhes destes pontos de venda.

Estipulados os conceitos e objetivos e requisitos do projeto, foram selecionadas as ferramentas que pudessem realizar todas as funções necessárias para o cumprimento dessas metas. Pela razão de possuir maior distribuição no mundial, a plataforma móvel escolhida foi *Android*. Para que fosse possível coletar os dados do usuário, optou-se pela plataforma de *login* e analítica do *Facebook*. Para interação do usuário com pontos de venda, foi adotada a API do *Google Maps*. Finalmente, a plataforma *Firebase* foi preferida para realizar o armazenamento e atualização de dados da aplicação, tanto de usuário quanto de estabelecimentos.

A implementação deu-se na próxima fase do desenvolvimento deste projeto, seguindo um fluxo definido de modo que as classes e telas da aplicação induzissem o usuário a realizar *login* e fornecer autorização de sua localização para acessar o mapa com as marcações dos pontos de venda. A partir dessa tela, a aplicação permite que o usuário interaja com a adição de novos localizadores ao mapa, visualizar a tela de seu perfil – que indica quais locais já visitou, e a tela de detalhes do produto. Nesta parte, o sistema permite que o usuário realize *check-in* no local, confirmando sua visita e presença; altere o estado do local para fechado ou aberto, faça avaliação, adicione fotos, altere sua descrição e exclua o local, caso ele não esteja mais funcionando. Com a formulação do modelo proposto, este trabalho apresentou um exemplo do funcionamento dos recursos elaborados.

Para este projeto não foi possível captar automaticamente informações mais precisas do usuário para execução de testes por questões burocráticas exigidas pela plataforma do *Facebook*. Idealmente, caso a aplicação venha a resgatar essas informações, o módulo *User Profile*

*Manager* poderá gerar perfis mais precisos dos usuários. Informações que podem constar na plataforma como histórico de trabalho e de ensino do usuário, por exemplo, poderiam agregar insumos para aprendizagem dos algoritmos em prol das categorizações. Estas melhorias ficam como sugestões para trabalhos futuros.

Além disso, deixamos como sugestão para trabalhos futuros a implementação dos módulos apresentados no modelo de análise de perfil proposto, de forma a adequar os algoritmos para análise e mineração de dados a fim de entregar o resultado esperado. Propomos também a utilização do modelo de análise para criação de categorias genéricas de usuários que possuem comportamento semelhante, ou seja, identificar perfis de usuários que frequentam os pontos de venda e que representam o perfil de cliente desses estabelecimentos, visto que este tipo de resultado pode agregar valor na formulação de novas estratégias comerciais para *food trucks* e lancherias de rua.



## REFERÊNCIAS

- Adomavicius, G. e A. Tuzhilin. **Expert-driven validation of rule-based user models in personalization applications**. Em: *Data Mining and Knowledge Discovery* 5.1/2, pp. 33–58. 2001.
- Amato, G. e U. Straccia. **User profile modeling and applications to digital libraries**. Em: *Research and Advanced Technology for Digital Libraries*, pp. 670–670. 1999.
- Bacha, M. d. L. e C. Figueiredo Neto. **Smartphone: de objeto de desejo a ferramenta para afirmação social nas classes CD**. Em: 2013.
- Chwif, L. e A. C. Medina. **Modelagem e simulação de eventos discretos**. Afonso C. Medina. 2006.
- Developers. **Dashboards**. URL: <https://developer.android.com/about/dashboards>. 2017.
- Developers, A. **What is android**. 2011.
- Dey, A. K. **Understanding and using context**. Em: *Personal and ubiquitous computing* 5.1, pp. 4–7. 2001.
- Empresas, F. para. **102 milhões de brasileiros compartilham seus momentos no Facebook todos os meses**. URL: <https://www.facebook.com/business/news/102-milhes-de-brasileiros-compartilham-seus-momentos-no-facebook-todos-os-meses>. 2016.
- Miles, R. E., G. Miles e C. C. Snow. **Collaborative entrepreneurship: How communities of networked firms use continuous innovation to create economic wealth**. Stanford University Press. 2005.
- Münch, J., F. Fagerholm, P. Johnson, J. Pirttilahti, J. Torkkel e J. Jäärvinen. **Creating minimum viable products in industry-academia collaborations**. Em: *Lean Enterprise Software and Systems*. Springer, pp. 137–151. 2013.
- Oliveira, F. **Número de usuários de smartphones no e-commerce cresce 170% no Brasil**. URL: <http://www1.folha.uol.com.br/tec/2017/02/1862362-smartphones-estao-nas-maos-de-62-dos-brasileiros-diz-google.shtml>. 2017.
- Panagiotakis, S. **Context-Awareness via Ubiquitous User Profiling: An Implementation Paradigm**. Em: *Software Engineering* 2.3, pp. 77–90. 2012.
- Pressman, R. S. **Engenharia de Software - Uma Abordagem Profissional**. Vol. 8. Amgh Editora. 2016.

Raghunathan, S. K. e A. Jaswal. **Methods and systems for providing personalized and context-aware suggestions**. US Patent App. 13/747,509. 2013.

Rodrigues, W. C. et al. **Metodologia científica**. Em: *Faetec/IST. Paracambi*, pp. 2–20. 2007.

Schafer, J. B., J. A. Konstan e J. Riedl. **E-commerce recommendation applications**. Em: *Applications of data mining to electronic commerce*. Springer, pp. 115–153. 2001.

Sevilleja, C. **A Look at the New Firebase: A Powerful Google Platform**. URL: <https://scotch.io/bar-talk/a-look-at-the-new-firebase-a-powerful-google-platform>. 2016.

Suh, Y. e W. Woo. **Context-based user profile management for personalized services**. Em: *ubiComp workshop (ubiPCMM)*, pp. 64–73. 2005.

Villela, F. **Celular é principal meio de acesso à internet no Brasil, mostra IBGE**. URL: <http://agenciabrasil.ebc.com.br/economia/noticia/2016-04/celular-e-principal-meio-de-acesso-internet-na-maioria-dos-lares>. 2017.

Wong, C. **MERGING THE PHYSICAL AND DIGITAL: ENHANCING THE FOOD TRUCK EXPERIENCE THROUGH A MOBILE APPLICATION**. Em: 2015.

Zuriarrain, J. M. **Android já é o sistema operacional mais usado do mundo**. URL: [http://brasil.elpais.com/brasil/2017/04/04/tecnologia/1491296467\\_396232.html](http://brasil.elpais.com/brasil/2017/04/04/tecnologia/1491296467_396232.html). 2017.