

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

JEAN CARLOS WILL

APLICATIVO MOBILE PARA BUSCA DE RESTAURANTES

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2017**

JEAN CARLOS WILL

APLICATIVO MOBILE PARA BUSCA DE RESTAURANTES

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientadora: Profa. Beatriz Terezinha Borsoi

**PATO BRANCO
2017**



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Departamento Acadêmico de Informática
Curso de Tecnologia em Análise e Desenvolvimento de
Sistemas



TERMO DE APROVAÇÃO
TRABALHO DE CONCLUSÃO DE CURSO
APLICATIVO MOBILE PARA BUSCA DE RESTAURANTES

por

JEAN CARLOS WILL

Este trabalho de conclusão de curso foi apresentado no dia 23 de junho de 2017, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Universidade Tecnológica Federal do Paraná. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Banca examinadora:

Prof^ª Dr^ª Beatriz Terezinha Borsoi
Orientador

Prof^ª Me. Andreia Scariot Beulke

Prof. MSc. Vinicius Pegorini

Prof. Dr. Edilson Pontarolo
Coordenador do Curso de Tecnologia em
Análise e Desenvolvimento de Sistemas

Prof^ª Dr^ª Beatriz Terezinha Borsoi
Responsável pela Atividade de Trabalho de
Conclusão de Curso

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

RESUMO

WILL, Jean Carlos. Aplicativo mobile para busca de restaurantes. 2017. 42 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2017.

Considerando a expansão das cidades, realizar refeições fora de casa, especialmente no almoço, tem se tornado um hábito comum. E, muitas pessoas também realizam o jantar fora de casa, seja para alimentar-se efetivamente, seja com o propósito de reunir-se com amigos, colegas de trabalho, familiares e outros. As opções de locais que oferecem refeições, muitas vezes com cardápios específicos, também têm crescido. A existência de cozinhas especializadas e a multiplicidade de locais que oferecem refeições podem dificultar na escolha de um restaurante, especialmente nos dias atuais nos quais tem sido comum que as pessoas estão cada vez mais atarefadas e, conseqüentemente, com menos tempo para realizar todas as suas atividades. Considerando esse contexto e as possibilidades que as tecnologias oferecem, como os aplicativos móveis, neste trabalho é apresentado o desenvolvimento de um aplicativo mobile para busca de restaurantes. O aplicativo conta com um sistema web para a realização dos cadastros e a busca por restaurantes é realizada com o uso de um dispositivo Android. O aplicativo foi desenvolvido com a tecnologia Ruby on Rails.

Palavras-chave: Aplicativos móveis. Localização de restaurantes. Ruby on Rails.

ABSTRACT

WILL, Jean Carlos. Mobile application for searching restaurants. 2017. 42 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2017.

Considering the expansion of cities it has become common for people to conduct meals outside the home, especially at lunch. In addition, many people do dinner out of the house, either to feed themselves effectively, or for the purpose of meeting with friends, co-workers, family, and others. The dining options, often with specific menus, have also grown. The existence of specialized kitchens and the multitude of places that offer meals can make it difficult to choose a restaurant, especially in the current days where it has been common for people to be more and more busy and with less time. Considering this context and the possibilities that technologies like the ones for development for mobile applications offer, in this work the development of a mobile application for searching restaurants is presented. The application was developed with Ruby on Rails.

Palavras-chave: Mobile application. Searching restaurants. Ruby on Rails.

LISTA DE FIGURAS

Figura 1 – Infraestrutura de software do Android	16
Figura 2 – Foursquare	18
Figura 3 – Yelp	18
Figura 4 – Foodspotting	19
Figura 5 – Trip Advisor	20
Figura 6 – Digrama de casos de uso.....	27
Figura 8 – Diagrama de entidades e relacionamentos	29
Figura 9 – Tela inicial apresentada no sistema web.....	30
Figura 10 – Tela de estabelecimentos	30
Figura 11 – Manutenção de imagens de restaurantes	31
Figura 12 – Tela inicial apresentada no aplicativo mobile	32
Figura 13 – Tela com dados de um restaurante selecionado.....	33
Figura 14 – Tela de avaliações realizadas	34
Figura 15 – Tela para realizar avaliações	35

LISTA DE QUADROS

Quadro 1 – Ferramentas e tecnologias utilizadas	22
Quadro 2 – Requisitos funcionais.....	26
Quadro 3 – Requisitos não funcionais.....	26

LISTAGENS DE CÓDIGO

Listagem 1 – Classe gerada pelo Gem Devise	36
Listagem 2 – Adição do Bootstrap ao projeto.....	37
Listagem 3 – Gem Parperclip para inclusão de imagens	37
Listagem 4 – Anotação ImageUploader	37
Listagem 5 – Uso de GoogleMaps	38
Listagem 6 – Uso da API Retrofit	39

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
CSS	<i>Cascading Style Sheets</i>
DRY	<i>Don't Repeat Yourself</i>
GPS	<i>Global Positioning System</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
SDK	<i>Software Development Kit</i>
URL	Uniform Resource Locator
MVC	<i>Model-View-Controller</i>
MVP	<i>Minimum Viable Product</i>
RF	Requisito Funcional
RNF	Requisito não Funcional

SUMÁRIO

1 INTRODUÇÃO	10
1.2 OBJETIVOS	11
1.2.1 Objetivo Geral	11
1.2.2 Objetivos Específicos	11
1.3 JUSTIFICATIVA	11
1.4 ESTRUTURA DO TRABALHO	12
2 REFERENCIAL TEÓRICO	13
2.1 DISPOSITIVOS MÓVEIS	13
2.1.1 Plataformas	13
2.1.2 Desenvolvimento Mobile	14
2.1.3 Desenvolvimento para Android	15
2.2 TRABALHOS RELACIONADOS	17
2.3 APLICATIVOS PARA GEOLOCALIZAÇÃO	20
3 MATERIAIS E MÉTODOS	22
3.1 MATERIAIS	22
3.2 MÉTODO	23
4 RESULTADO	25
4.1 DESCRIÇÃO DO SISTEMA	25
4.2 MODELAGEM DO SISTEMA	25
4.5 IMPLEMENTAÇÃO DO SISTEMA	36
5 CONCLUSÃO	40
REFERÊNCIAS	41

1 INTRODUÇÃO

Este Capítulo apresenta as considerações iniciais do trabalho, a justificativa e os objetivos. No final são apresentados os capítulos subsequentes que compõem o texto.

1.1 CONSIDERAÇÕES INICIAIS

O dia a dia das pessoas tem se tornado cada vez mais atarefado e o tempo cada vez mais escasso para realizar tantas atividades, sendo que a busca por bons restaurantes, ou que atendam aos interesses e preferências pessoais, pode se tornar mais difícil, pela falta de tempo das pessoas e pelo aumento de opções. As cidades crescem e a população se diversifica, aumentando a diversidade culinária ofertada e o número de estabelecimentos. Além disso, existem pessoas que gostam de conhecer novos lugares, experimentar cozinhas (pratos) diferentes, seja em sua própria cidade ou mesmo em outras localizações.

Ao escolher um restaurante, a realização da combinação de preço com qualidade torna-se um desafio. Ainda que o restaurante possua a indicação de amigos ou conhecidos, é sempre bom verificar a opinião de outras pessoas. Contudo, isso não é fácil, pois nem sempre as pessoas mais próximas conhecem o restaurante indicado.

Existem aplicativos e sites disponíveis que podem auxiliar o usuário na busca de locais de seu interesse, como o Yelp (YELP, 2017), que auxilia na localização de postos de gasolina e o Foursquare (FOURSQUARE, 2017), que visa a busca por restaurantes, cafés, lojas, entre outros. Existem, também, aplicativos como o iFood (IFOOD, 2017), que se aplicam à realização do pedido de refeições, que é um foco diferente do aplicativo desenvolvido como resultado deste trabalho.

A utilização de *smartphones* tem se tornado cada vez mais comum, principalmente para acesso à Internet e busca de informações. Tecnologias para desenvolvimento de aplicações *mobile* também tem se popularizado entre os desenvolvedores, principalmente para o ambiente Android, considerando a grande popularização dessa plataforma entre os usuários.

Para o desenvolvimento *mobile*, Ambros (2013) especificou três tipos de aplicações: nativas, *web* e híbridas. Esse autor ressalta que a grande vantagem do desenvolvimento nativo é poder utilizar todas as funcionalidades presentes nos dispositivos, como: câmera, *Global Positioning System* (GPS), sistema de notificação nativo e funcionar sem conexão com a Internet.

Considerando esse contexto, este trabalho visa desenvolver uma aplicação *mobile* nativa que auxilie os usuários a encontrarem restaurantes tendo como base a sua localização, o tipo de prato, o preço e as avaliações realizadas por outros usuários. O termo restaurante é utilizado neste texto para referir-se de forma genérica aos estabelecimentos que servem refeições.

1.2 OBJETIVOS

A seguir são apresentados os objetivos deste trabalho.

1.2.1 Objetivo Geral

Implementar um aplicativo para dispositivos móveis Android para auxiliar na busca por restaurantes de acordo com critérios como localização do usuário e avaliação de outros usuários.

1.2.2 Objetivos Específicos

- Permitir ao usuário buscar restaurantes próximos da sua localização.
- Facilitar ao usuário encontrar restaurantes e pratos de sua preferência de acordo com a avaliação realizada pelos usuários.
- Permitir que o usuário faça a avaliação de restaurantes que tenha frequentado.

1.3 JUSTIFICATIVA

As pessoas estão cada vez mais ocupadas com seus trabalhos e entre a

diversidade de funções exercidas pelos indivíduos o tempo para atividades rotineiras, porém essenciais, como a busca por novos restaurantes e conhecer outros tipos de cozinhas (culinárias específicas como a de determinados países ou regiões) torna-se escasso. Além disso, informações sobre restaurantes existentes e que estejam de acordo com os seus critérios de preferência também é um fator importante na escolha de um restaurante. Com base nisso, este trabalho visa auxiliar na busca de restaurantes, de modo que o usuário possa conhecer e frequentar novos ambientes que sejam convenientes de acordo com suas preferências, como preço, qualidade, tipo de prato e localização.

Com relação às tecnologias utilizadas para o desenvolvimento, justifica-se por se tratarem de tecnologias atuais, como Android Studio, para a implementação do aplicativo, elas foram escolhidas por facilitar o desenvolvimento de aplicações *mobile* para a plataforma Android.

1.4 ESTRUTURA DO TRABALHO

Este texto está organizado em capítulos. O Capítulo 2 apresenta o referencial teórico que fundamenta a aplicação desenvolvida que é para dispositivos móveis Android. O Capítulo 3 apresenta os materiais e o método utilizados na modelagem e na implementação do aplicativo. No Capítulo 4 é apresentado o aplicativo desenvolvido por meio da modelagem, das telas que representam as funcionalidades implementadas e a forma de interação do usuário com essas funcionalidades. E, ainda, por meio de exemplos de códigos implementados. Em seguida é apresentada a conclusão do trabalho que é seguida pelas referências utilizadas no texto.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o embasamento teórico que fundamenta a proposta deste trabalho que tem como foco o desenvolvimento de um aplicativo móvel baseado no conceito de localização. Como o aplicativo será para dispositivos móveis, o referencial teórico tratará do funcionamento desse tipo de aplicativo, apresentando exemplos e aplicativos com funcionalidades semelhantes ao que foi desenvolvido.

2.1 DISPOSITIVOS MÓVEIS

Atualmente a utilização de *smartphones* no Brasil já se tornou algo rotineiro na vida das pessoas. De acordo com a Revista Exame, já no primeiro semestre de 2015 o número de *smartphones* conectados a Internet tinha superado o de computadores, do total de 306 milhões de dispositivos conectados, 154 milhões eram *smartphones* (REVISTA EXAME, 2015).

2.1.1 Plataformas

Existem diversas plataformas de sistemas operacionais em uso nos *smartphones* atuais, sendo essas o meio que permite que o usuário interaja com os aplicativos e itens do *hardware*, como, câmeras, GPS, microfone, alto-falante. Segundo a Top 5 (TERRA, 2013), as principais plataformas são:

- **Android:** sistema operacional desenvolvido pela Google, com utilização de 70,1% do mercado. O que justifica sua ampla utilização é que as marcas e os modelos que possuem esse sistema são os mais variados, de aparelhos com muitos recursos aos modelos mais básicos ou simples.
- **iOS:** plataforma introduzida no mercado em 2007 pela Apple e de utilização exclusiva em produtos da própria marca. Por ser de uso exclusivo da marca Apple, o iOS é desenvolvido para um modelo específico de *hardware*, permitindo assim, que suas potencialidades sejam

exploradas ao máximo. Desde sua primeira versão, foi criado para utilizar em aparelhos com telas sensíveis ao toque. No último trimestre de 2012, 21% dos *smartphones* vendidos utilizavam o sistema operacional da Apple.

- **BlackBerry:** desde 1999, o sistema BlackBerry serve de base para os celulares de mesma marca e ao longo do tempo seus recursos foram adaptados a diferentes formas de controle do aparelho. A empresa já chegou a dominar o mercado governamental e empresarial, devido à segurança aplicada no tráfego de dados dos seus aparelhos.
- **Windows Phone:** até o final de 2014, o sistema operacional da Microsoft, possuía somente 4% do mercado. É um sistema voltado na atualização constante de aplicativos conectados às redes sociais. Empresas como a Nokia, HTC e Samsung possuem aparelhos que executam o Windows Phone.
- **Symbian:** empresas como a Sony Ericson, Samsung, LG e Motorola lançaram celulares que executam o sistema Symbian. Ele chegou a ser utilizado por 65% do mercado em 2007. Porém, no final de 2012 era utilizado em menos de 1% dos dispositivos móveis existentes.

2.1.2 Desenvolvimento Mobile

O desenvolvimento de aplicações móveis se tornou uma nova oportunidade de negócio para os desenvolvedores e para as empresas, tendo em vista o crescimento exponencial do número de *smartphones* no mercado. Cresce na mesma proporção a busca por aplicações que satisfazem as necessidades dos usuários. Essas necessidades vão de entretenimento a aplicações para uso no trabalho e que são vinculadas aos sistemas corporativos, muitas vezes de grande porte.

Segundo Ambros (2013), existem três tipos diferentes de aplicações *mobile*: híbridas, nativas e *web*.

- **Mobile Web App:** são aplicações acessadas com o uso de navegadores e desenvolvidas, normalmente, utilizando HTML 5 e não requerem a instalação do aplicativo no *smartphone*. Seu acesso é realizado por meio

de um site comum, pela *Uniform Resource Locator* (URL). Porém, recursos como notificações do sistema operacional, acelerômetro e outros recursos de hardware não são acessíveis por essas aplicações. São, na maioria das vezes, desenvolvidos para possuírem um comportamento semelhante às aplicações nativas e utilizadas quando há necessidade de disponibilizar conteúdo ou funcionalidade em *smartphones*, mas que não é viável construir uma aplicação nativa.

- **Aplicativos Nativos:** são aplicativos instalados no dispositivo e adquiridos por meio de lojas, como Google Play, App Store e Amazon Underground. Seu desenvolvimento é específico para determinado operacional, podendo aproveitar todos os recursos de hardware e de software presentes no dispositivo. Além do acesso fácil aos recursos do dispositivo, se o aplicativo for bem concebido, ele respeitará os padrões de *design* de cada plataforma, atendo a padrões de usabilidade. Sua maior desvantagem é o fato de ser necessário desenvolver aplicativos específicos para cada sistema operacional.
- **Aplicações Híbridas:** são aplicativos parcialmente nativos e parcialmente Web Apps. Como aplicativos nativos, eles podem aproveitar todos os recursos do *smartphone* e como Web Apps, eles podem ser baseados em HTML 5 e exibidos em um navegador presente no aplicativo. Essas aplicações são populares porque permitem o desenvolvimento multiplataforma, reduzindo os custos de produção.

2.1.3 Desenvolvimento para Android

O Android foi criado em 2003 por Andy Rubin, Rico Miner e Nick Sears, que fundaram a empresa Android Inc. Android é um sistema operacional que tem base no *kernel* do Linux e seu objetivo inicial era a criação de dispositivos móveis mais inteligentes, um sistema para câmeras fotográficas, mas depois foi redirecionado para ser um sistema operacional *mobile*. Em 2005 a Android Inc. foi comprada pela Google, por 50 milhões (HAMMERSCHMIDT, 2015).

Para o desenvolvimento de aplicações, a Google disponibiliza um *kit* de ferramentas chamado *Android Software Development Kit* (SDK). Segundo um estudo realizado por Ogliari e Brito (2014), a infraestrutura do Android é composta por uma pilha que inclui um sistema operacional Linux, um conjunto de bibliotecas, uma *Application Programming Interface* (API) chamada Android Runtime, aplicações preexistentes no Android e aplicações diversas. Essa estrutura é apresentada na Figura 1.

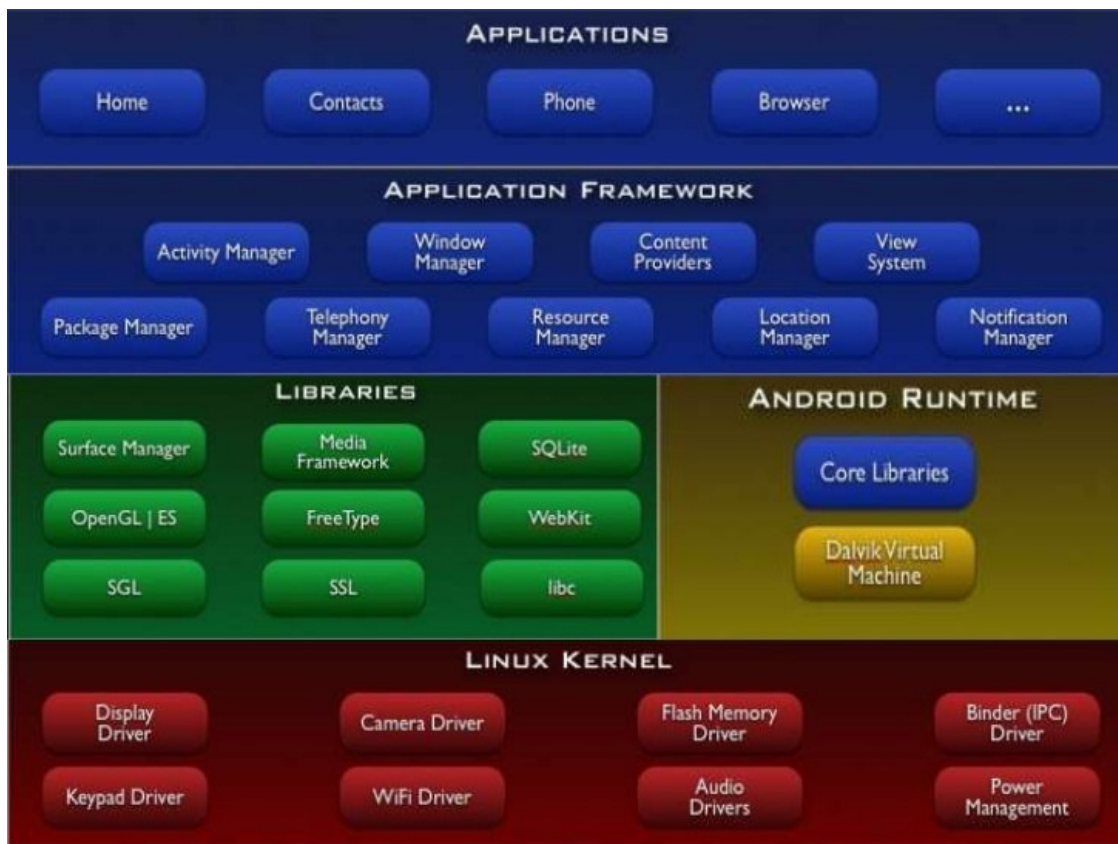


Figura 1 – Infraestrutura de software do Android
Fonte: Android Developer (2017, p..s.n.).

Comentando as partes que compõem a infraestrutura do Android, como apresentado na Figura 1:

- **Linux Kernel:** é responsável por tarefas como o gerenciamento de memória e de processos e acesso à rede. É responsabilidade do sistema operacional realizar a comunicação entre o hardware e o software desenvolvido.

- **Libraries e Android Runtime:** É um conjunto de bibliotecas C/C++ utilizadas por vários componentes do Android e seu acesso ocorre pelo Android Application Framework. Dentre as bibliotecas existentes, destacam-se: SQLite, Media Libraries e Surface Manager. O Android Runtime se resume às bibliotecas básicas do sistema operacional, que serão utilizadas pelas demais bibliotecas.
- **Application Framework:** contém bibliotecas de mais alto nível, responsáveis por funções específicas no aparelho, como, o gerenciamento de janelas e de recursos dos dispositivos móveis e notificações. Essas bibliotecas são acessadas pelos aplicativos desenvolvidos para interagir com o dispositivo móvel.
- **Applications:** são os aplicativos e as ferramentas disponibilizadas aos usuários.

2.2 TRABALHOS RELACIONADOS

Existem diversos aplicativos móveis para busca por localizações, incluindo restaurantes, como:

a) Foursquare: aplicativo de rede social focado na busca de localizações de acordo com as preferências do usuário (FOURSQUARE, 2017). Permite seguir amigos e pessoas formadoras de opinião e visualizar as dicas de lugares deixadas por eles. O leiaute da aplicação pode ser visualizado na Figura 2.



Figura 2 – Foursquare

b) **Yelp**: é um aplicativo de código aberto, licenciado sob a LGPL v2.1. Permite ao usuário buscar por eventos e conversar com outros usuários do aplicativo (YELP, 2017).. A Figura 4 ilustra duas telas desse aplicativo.

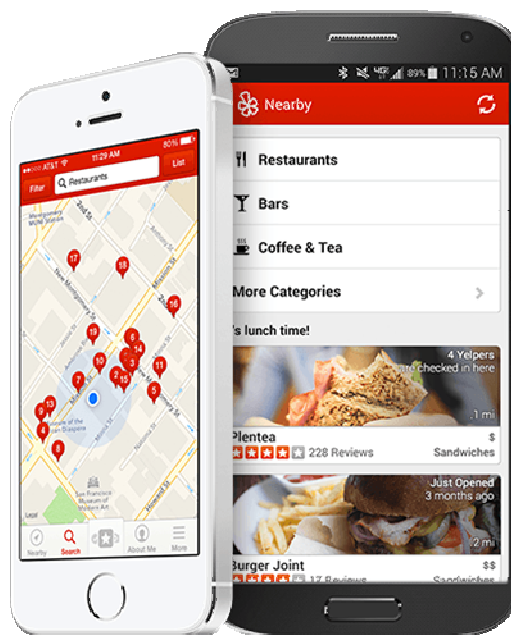


Figura 3 – Yelp

c) **Foodspotting**: diferente das outras aplicações é voltado na busca de pratos específicos e não de restaurantes (FOODSPOTTING, 2017). A Figura 4 ilustra a aplicação Foodspotting.

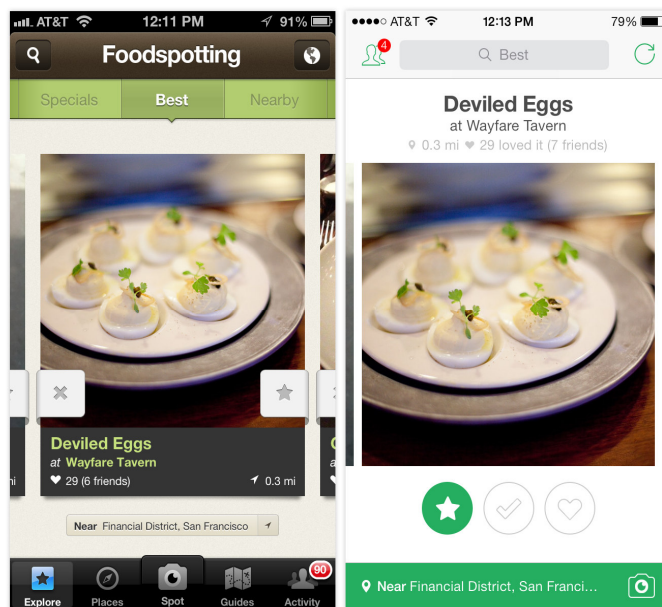


Figura 4 – Foodspotting

d) **Trip Advisor**: auxilia o usuário a planejar o roteiro de viagem e buscar e reservar hotéis e restaurantes (TRIP ADVISOR, 2017). A Figura 5 apresenta duas telas que fornecem uma ideia do funcionamento da aplicação.

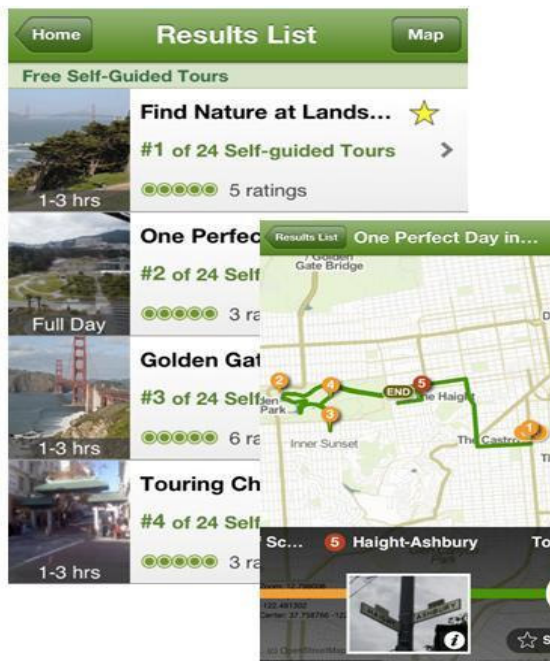


Figura 5 – Trip Advisor

2.3 APLICATIVOS PARA GEOLOCALIZAÇÃO

Geolocalização é o processo de encontrar a localização exata de um computador, dispositivo de rede ou equipamento com base em medições e coordenadas geográficas (NEKA, 2017). Atualmente, cada vez mais *smartphones* estão incluindo software e hardware de geolocalização.

Com a utilização dos dados de localização coletados pelo dispositivo é possível determinar, por exemplo, lojas, cafés, restaurantes e cinemas nas proximidades do usuário por meio das informações do GPS.

Os dados de localização são frequentemente utilizados por aplicações de mapas, como por exemplo:

- **Waze:** é um aplicativo baseado na colaboração popular para compor o melhor percurso para o usuário chegar ao local desejado. Assim que um usuário insere um destino e utiliza o aplicativo para visualizar o caminho, ele passa a contribuir passivamente com as informações do tráfego ou ativamente, informando a localização de acidentes, perigos e outros

eventos (WAZE, 2017).

- **HERE Maps:** esse aplicativo de mapas foi escolhido como “melhor aplicativo automotivo” de 2015 pela Revista Auto Build e possui o investimento das empresas fabricantes de automóveis AUDI AG, BMW Group e Daimler AG (HERE, 2017). Possibilita a comparação de rotas de carro, a pé e de transporte público, também possui informação em tempo real sobre o trânsito e horários de metrô. Oferece também a possibilidade da navegação *off-line* em aproximadamente 160 países, sendo possível encontrar o caminho sem ou com pouca cobertura de rede.
- **Google Maps:** o Google Maps permite monitorar o trânsito em tempo real, recalculando seu trajeto dinamicamente com base nos padrões de trânsito encontrados, ajudando a evitar congestionamentos (GOOGLE, 2016). Esse aplicativo oferece a possibilidade de visualizar comentários e fotos de outros usuários que moram ou que já estiveram no local de destino pretendido.

3 MATERIAIS E MÉTODOS

Este capítulo apresenta os materiais e o método utilizado para a realização deste trabalho. Os materiais estão relacionados às tecnologias e às ferramentas utilizadas e o método apresenta a sequência das principais atividades realizadas.

3.1 MATERIAIS

O Quadro 1 apresenta as ferramentas e as tecnologias utilizadas para a modelagem e a implementação do sistema.

Ferramenta /Tecnologia	Versão	Referência	Finalidade
Android Studio	2.3.2	http://developer.android.com/sdk/index.html	Ambiente de Desenvolvimento
Ruby	2.3.0	https://www.ruby-lang.org/pt/	Linguagem de Programação
Rails	5.0.0.1	http://rubyonrails.org/	Framework <i>web</i>
PostgreSql	9.3.14	http://www.postgresql.org/download/windows	Banco de Dados
Visual Paradigm Community	11.1	http://www.visual-paradigm.com	Documentação da modelagem baseada na UML.
Bootstrap	3.3.7	http://getbootstrap.com/	Estilização do sistema <i>web</i> .

Quadro 1 – Ferramentas e tecnologias utilizadas

A seguir é apresentado sobre o Ruby e Rails por serem as duas tecnologias de uso menos comum no contexto do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, entre as apresentadas no Quadro 1.

O Ruby é uma linguagem de programação interpretada multiparadigma, com tipagem dinâmica e forte e gerenciamento automático de memória. As principais características do Ruby são (RUBY, 2017):

- Todas as variáveis são objetos;
- Os “tipos primitivos” (*int*, *float*, *char*, etc.), são classes;
- Tipagem dinâmica, porém forte;
- Blocos de código podem ser passados como parâmetro para métodos;
- Utilizando as RubyGems é possível instalar e atualizar bibliotecas por

meio da linha de comando.

O Rails é um *framework* para o desenvolvimento de aplicações *web* escrito em Ruby utilizando o padrão de arquitetura *Model-View-Controller* (MVC), criado em 2003 por David Heinemeier Hansson enquanto trabalhava na base de código do Basecamp, tendo sua versão 1.0 lançada em 2005 (SCORZA, 2016).

O Rails segue dois conceitos principais (RAILS, 2017):

- *Don't Repeat Yourself* (DRY): não se repita: é um princípio de desenvolvimento que afirma que todo pedaço de código deve ter uma única representação dentro do sistema. Isso é no sentido que um mesmo código não deve ser reescrito, mas reusado, tornando o sistema mais sustentável, escalável e menos passível de defeitos.
- Convenção sobre a configuração: o Rails assume qual é a melhor maneira de se desenvolver uma aplicação e utiliza esse conjunto de convenções em vez de exigir que o desenvolvedor especifique cada um deles.

3.2 MÉTODO

O modelo de processo adotado para o desenvolvimento do aplicativo como resultado deste trabalho é o Processo Unificado, o qual se utiliza de ciclos iterativos e incrementais de modelagem e implementação. De acordo com Larman (2004), no Processo Unificado não há uma corrida para implementar o software e nem um longo e demorado projeto que tenta resolver todos os problemas antes da programação. Contudo, ao final de cada iteração tem-se uma versão executável, mas que ainda não está pronta para ser colocado em produção. No desenvolvimento deste trabalho, ao longo de cada iteração é adotada uma abordagem linear (PRESSMAN, 2006) para a realização das atividades que a compõe.

A seguir serão descritos os processos realizados e suas principais atividades.

a) Levantamento de requisitos

O levantamento de requisitos iniciou com estudo e observação das necessidades e dificuldades principais dos usuários alvo do aplicativo, bem como pesquisa acerca de aplicativos com finalidades semelhantes. Esses dados geraram uma base de conhecimento necessária para a proposta e o desenvolvimento do

aplicativo, a partir do qual foi definido o escopo do aplicativo, bem como os requisitos que deveriam estar presentes no aplicativo. Os requisitos foram divididos em funcionais e não funcionais.

b) Análise e Projeto

Na fase de modelagem ocorreu a definição dos casos de uso do sistema, a elaboração de um diagrama de sequência e de um diagrama entidade-relacionamento para o banco de dados.

c) Implementação

A implementação do aplicativo foi dividida em dois *Minimum Viable Product (MVP)*, que são utilizados para a divisão de versões de implementação do aplicativo. O *MVP 1* do aplicativo foi desenvolvido contendo somente a listagem dos estabelecimentos por avaliações realizadas pelos usuários, bem como a visualização dos detalhes do estabelecimento e a ferramenta para sua avaliação. Também foi implementado o aplicativo *web* para o cadastro dos estabelecimentos.

No *MVP 2* do aplicativo foi implementada a listagem de estabelecimentos por proximidade e a visualização das avaliações realizadas por outros usuários.

d) Testes

Os testes foram realizados de maneira informal e paralelo ao desenvolvimento. Os testes foram basicamente de unidade (verificação de código) e testes da interação com o aplicativo por meio da verificação se as funcionalidades definidas estavam sendo implementadas como planejado.

4 RESULTADO

Este capítulo apresenta o resultado deste trabalho que tratou da modelagem e da implementação de um aplicativo *mobile* para busca de restaurantes a partir da localização, preferências e avaliações de usuários.

4.1 DESCRIÇÃO DO SISTEMA

O aplicativo visa facilitar a busca de restaurantes de acordo com as necessidades e preferências do usuário.

Para usar o aplicativo e realizar buscas, não é necessário que o usuário realize cadastro prévio. Porém, para que seja possível o usuário realizar uma avaliação sobre um local visitado, deve ser feito o *login* na rede social Facebook.

O aplicativo também conta com uma página *web* para que seja possível o cadastro dos restaurantes e dos pratos por eles oferecidos.

4.2 MODELAGEM DO SISTEMA

O Quadro 2 apresenta os requisitos funcionais definidos para o aplicativo desenvolvido. Nesse quadro RF significa Requisito Funcional.

	Requisito	Descrição
RF01	Manter restaurante	Cadastro de restaurantes. Esse cadastro será utilizado para a realização de buscas. Manutenção se refere à inclusão, exclusão, consulta e alteração de dados. O cadastro permitirá a inclusão de imagens. A manutenção de restaurantes é realizada pelo sistema <i>web</i> .
RF02	Manter produto	Cadastro dos itens de cardápio que serão divulgados pelo restaurante. Esses produtos são utilizados na busca e são vinculados ao restaurante que os oferece. Manutenção se refere às operações realizadas no cadastro de produtos. A manutenção de restaurantes é realizada pelo sistema <i>web</i> .
RF03	Avaliar estabelecimentos	Avaliar restaurantes e outros estabelecimentos cadastrados no aplicativo, inserindo comentários.
RF04	Buscar estabelecimento	Realizar a busca de restaurantes, considerando a localização geográfica do usuário e cálculo de proximidade do

	s	estabelecimento em relação às coordenadas geográficas do usuário.
RF05	Inserir imagens	Inserir imagens relacionadas aos restaurantes cadastrados.
RF06	Buscar pratos e restaurantes	A busca de locais que oferecem refeições e de determinados tipos de pratos. O aplicativo oferecerá diversos filtros de busca.

Quadro 2 – Requisitos funcionais

No Quadro 3 estão os requisitos não funcionais levantados para o aplicativo. Nesse quadro RNF significa Requisito não Funcional.

	Requisito	Descrição
RNF01	Vincular usuário ao perfil do Facebook	Para realizar avaliações, o usuário deverá vincular o seu perfil ao Facebook
RNF02	Campos obrigatórios no cadastro de restaurante	No cadastro de restaurantes há campos de preenchimento obrigatório.
RNF03	Cadastro de produtos	Será obrigatória a inclusão de descrição e do valor do produto.
RNF04	Busca de restaurantes	A busca de restaurantes deve levar em conta a localização geográfica em que o usuário se encontra.
RNF05	Localização do usuário	A localização do usuário deverá ser obtida de acordo com os dados do GPS presente no <i>smartphone</i> .
RNF06	Cálculo da distância	A distância será calculada de acordo com a localização do usuário e o endereço do restaurante.
RNF07	Avaliação de restaurantes	Para efetuar a avaliação o aplicativo deve estar vinculado ao perfil do Facebook do usuário.
RNF08	Cadastro de restaurante e pratos	Para cadastrar restaurantes e pratos o usuário deve estar autenticado e utilizar o sistema <i>web</i> .

Quadro 3 – Requisitos não funcionais

Os requisitos funcionais foram organizados em casos de uso. A Figura 6 apresenta o diagrama de casos de uso. Foram identificados três atores para interação com as funcionalidades do aplicativo:

- Usuário *web* – terá como funcionalidades essenciais o cadastro de estabelecimentos, de pratos servidos pelos restaurantes e de imagens.
- Usuário sem *login* – acesso à consulta de estabelecimentos e visualização de avaliações realizadas.
- Usuário com *login* – herda as permissões do usuário sem *login* e, ainda, pode avaliar estabelecimentos. Além de manter o próprio cadastro.

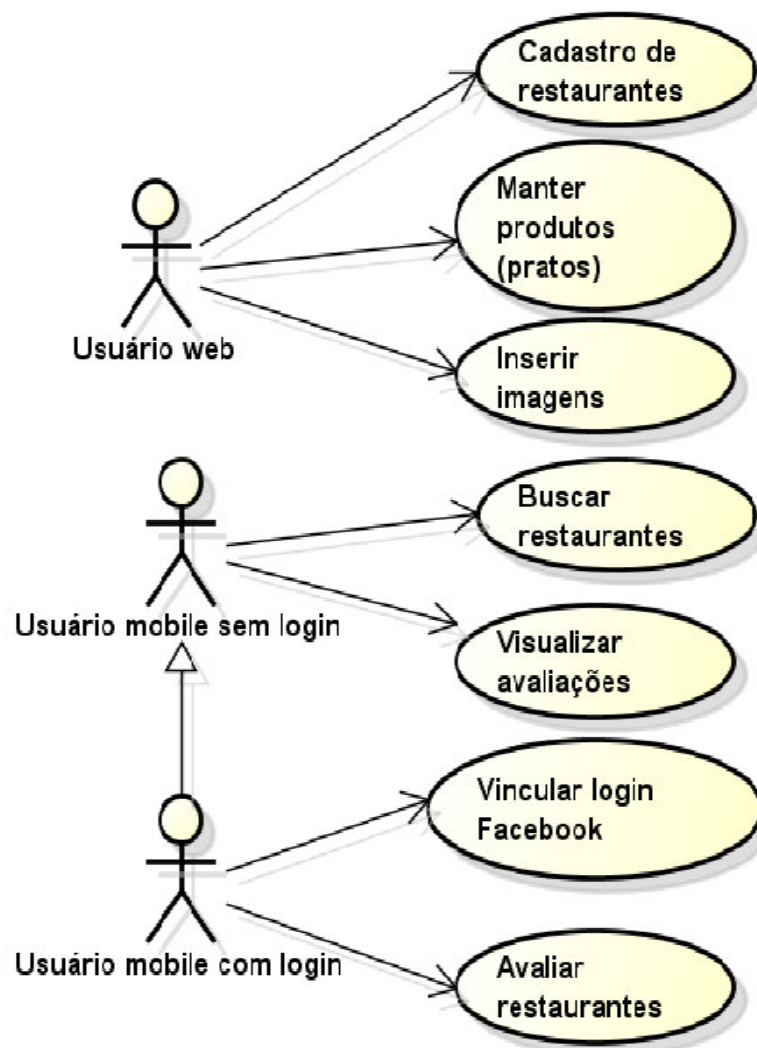


Figura 6 – Diagrama de casos de uso

A Figura 7 apresenta o diagrama de sequência que representa o fluxo de interação do usuário com o aplicativo. Esse fluxo apresenta as principais ações realizadas quando o usuário quer localizar restaurante indicando os filtros de busca. Selecionados os restaurantes, o usuário pode visualizar fotos (imagens) disponibilizadas pelo restaurante, comentários realizados pelos usuários e realizar avaliações.

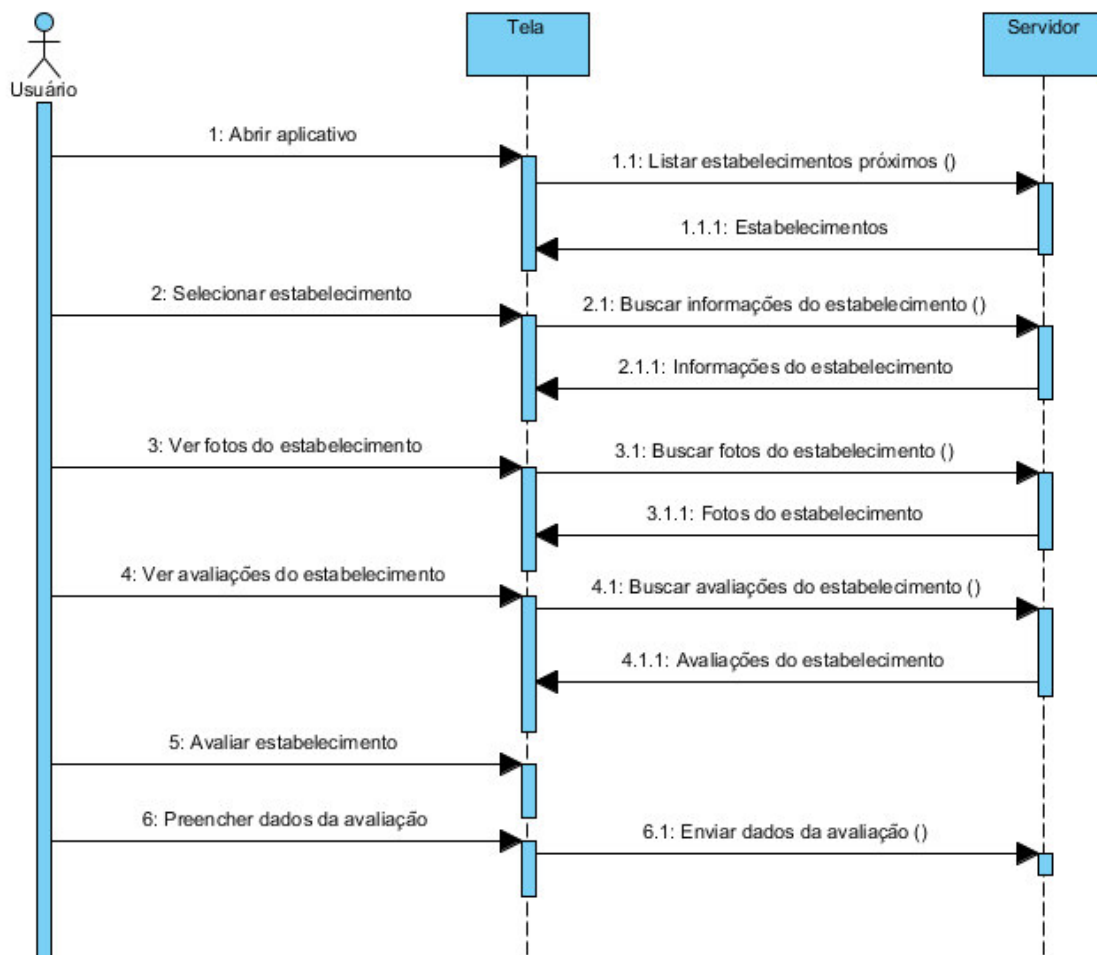


Figura 7 – Diagrama de atividades para buscar e avaliar restaurantes

Na Figura 8 está o diagrama de entidades e relacionamentos do banco de dados do aplicativo. A tabela principal é a de restaurantes (*restaurants*) que está vinculada com todas as outras tabelas, exceto a de Estados que está relacionada com a tabela cidades. As entidades e o banco de dados possuem seus nomes e atributos em inglês por padrão do *framework Rails*, que por meio de comandos no terminal gera as tabelas e as entidades, escrevendo, assim, os seus plurais.

O diagrama representa o banco de dados da versão *web* da aplicação, pois a aplicação *mobile* não possui persistência dos dados, os dados são apenas consultados.

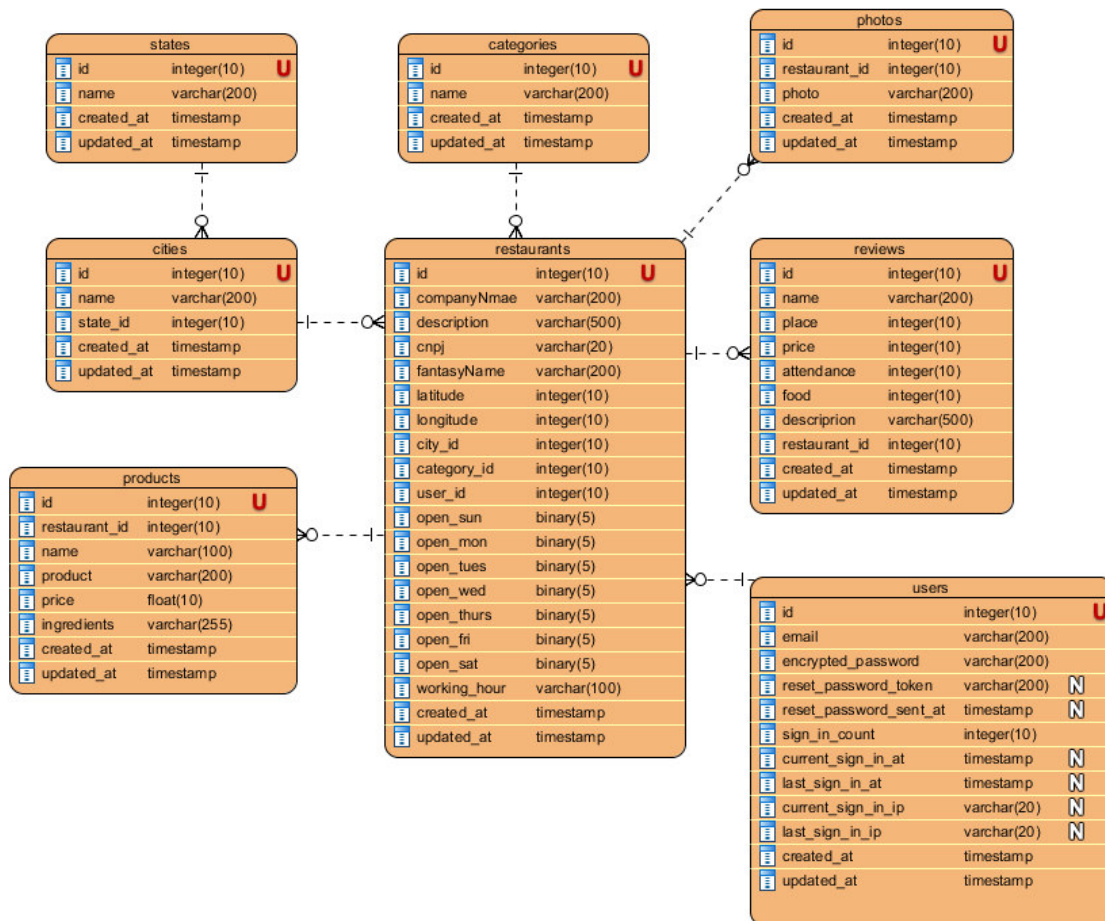


Figura 8 – Diagrama de entidades e relacionamentos

4.4 APRESENTAÇÃO DO SISTEMA

O leiaute da página de apresentação do sistema *web* é composto por uma imagem de fundo e dois botões alinhados no centro da tela, conforme apresentado na Figura 9. O primeiro botão é responsável por abrir a tela de cadastro. O segundo botão permite o acesso ao sistema por usuários cadastrados.

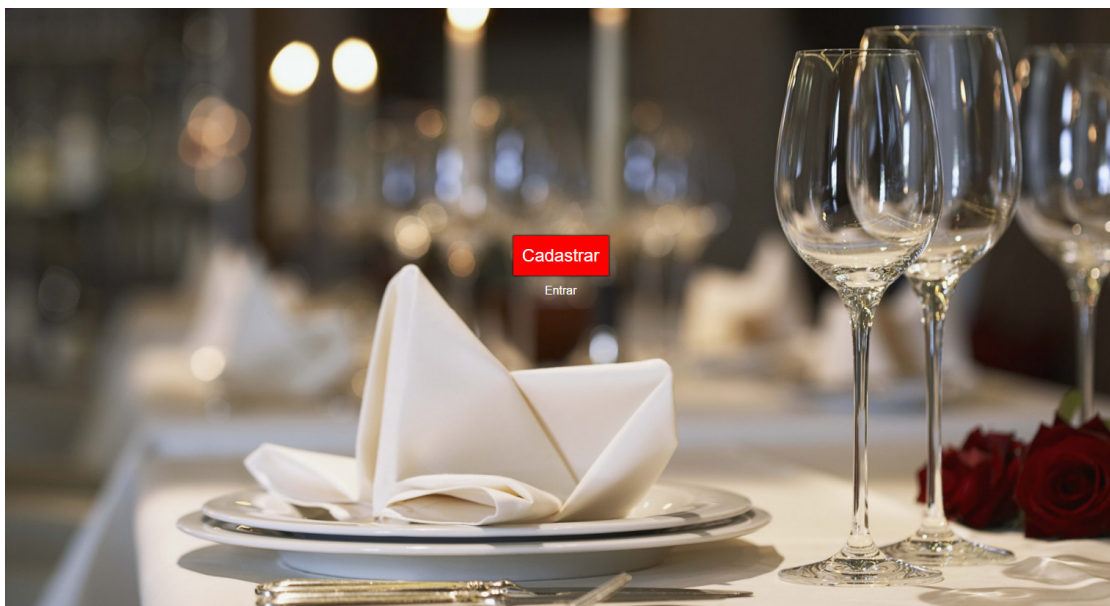


Figura 9 – Tela inicial apresentada no sistema web

Após o usuário estar autenticado é apresentada a listagem de todos os restaurantes cadastrados. Essa tela oferece opções para visualizar as fotos, os menus e as avaliações realizadas dos restaurantes, bem como editar ou excluir. Nessa tela também existe um botão para inclusão de um novo restaurante. A tela de estabelecimentos é apresentada na Figura 10.

Estabelecimentos

Estabelecimento	Descrição	Cnpj	Nome fantasia	Cidade	Categoria			
Restaurante João M.E.	O restaurante Tudo de Bom tem a melhor comida caseira da cidade, feitas com muito carinho e utilizando sempre os melhores ingredientes encontrados no mercado.	28.377.696/0001-99	Restaurante Tudo de Bom	Pato Branco	Restaurante	Fotos	Produtos	Avaliações
						Editar	Excluir	
Restaurante do Juca	Restaurante do Juca	79.627.586/0001-79	Restaurante do Juca	Pato Branco	Restaurante	Fotos	Produtos	Avaliações
						Editar	Excluir	

[Adicionar estabelecimento](#)

Figura 10 – Tela de estabelecimentos

Na Figura 11 é apresentada a listagem de fotos de um determinado

restaurante. Nessa tela é possível incluir e excluir fotos, voltar à tela de listagem e outro botão para adicionar uma nova foto.

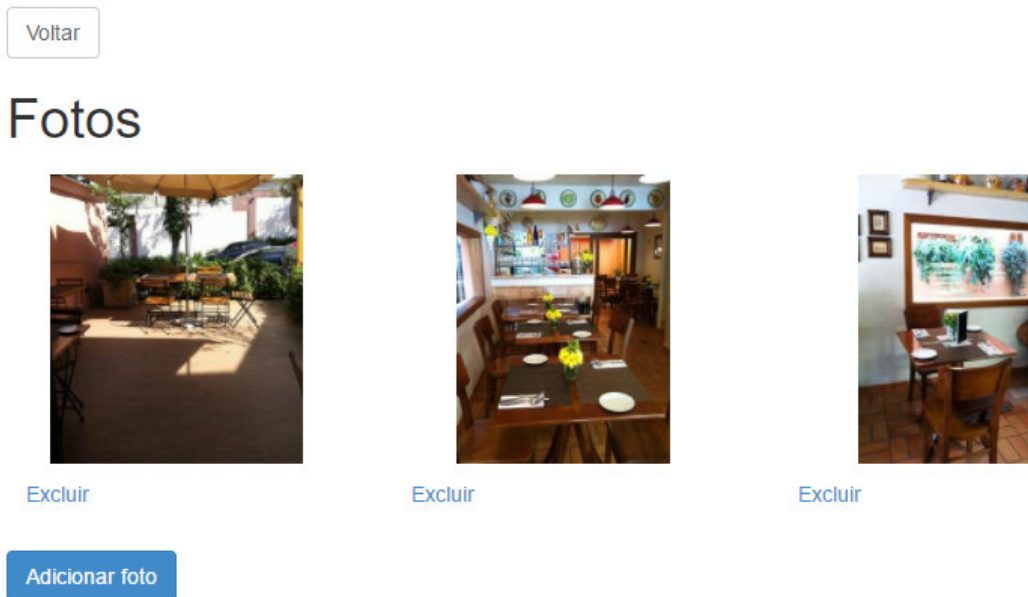


Figura 11 – Manutenção de imagens de restaurantes

A tela que o usuário visualiza ao abrir o aplicativo *mobile* é um mapa da localização em que ele se encontra. Nesse mapa são pontuados todos os estabelecimentos cadastrados que possuem localização próxima ao usuário, como visto na Figura 12.



Figura 12 – Tela inicial apresentada no aplicativo mobile

Após o usuário clicar sobre um dos pontos apresentados no mapa que representam restaurantes, ele é redirecionado para a tela que contém as informações do restaurante, como o nome do restaurante, descrição e dias de funcionamento. Também são apresentados os botões para que sejam visualizadas as fotos do local, o cardápio e as avaliações feitas por outros usuários. A Figura 13 apresenta essa tela.



Figura 13 – Tela com dados de um restaurante selecionado

A Figura 14 apresenta a listagem de avaliações. O usuário pode visualizar todas as avaliações realizadas por outros usuários e também pode adicionar uma nova avaliação do estabelecimento.



Figura 14 – Tela de avaliações realizadas

Na Figura 15 está a tela em que o usuário poderá fazer a avaliação do estabelecimento que ele visitou, informando uma descrição e as avaliações sobre o local, o preço, o atendimento e o prato. A avaliação é realizada pela indicação de “estrelas”. Para que a avaliação seja efetivada, o usuário deverá entrar com seu perfil da rede social Facebook no aplicativo.

Find Restaurant

Descrição:

Local

★ ★ ★ ★ ★

Preço


★ ★ ★ ★ ★

Atendimento

★ ★ ★ ★ ★

Comida

★ ★ ★ ★ ★

 Sair

ENVIAR AVALIAÇÃO

Figura 15 – Tela para realizar avaliações

4.5 IMPLEMENTAÇÃO DO SISTEMA

A implementação do sistema foi realizada com a linguagem de programação Ruby e seu *framework* para aplicativos *web* chamado Rails. Para implementar o *login* do usuário no sistema foi utilizada a RubyGem Devise, que faz todo o gerenciamento da inclusão e autenticação do usuário no sistema. Uma RubyGem ou a sua forma abreviada de referência como Gem é uma biblioteca (um conjunto de arquivos Ruby visando reuso). Essa biblioteca possui um nome que a identifica e uma versão e é disponibilizada por meio de um arquivo denominado *gemspec*.

A Listagem 1 apresenta a classe gerada pela Gem Devise.

```
<h2>Log in</h2>

<%= form_for(resource, as: resource_name, url:
session_path(resource_name)) do |f| %>
  <div class="field">
    <%= f.label :email %><br />
    <%= f.email_field :email, autofocus: true %>
  </div>

  <div class="field">
    <%= f.label :password %><br />
    <%= f.password_field :password, autocomplete: "off" %>
  </div>

  <% if devise_mapping.rememberable? -%>
    <div class="field">
      <%= f.check_box :remember_me %>
      <%= f.label :remember_me %>
    </div>
  <% end -%>

  <div class="actions">
    <%= f.submit "Log in" %>
  </div>
<% end %>

<%= render "devise/shared/links" %>
```

Listagem 1 – Classe gerada pelo Gem Devise

Para a estilização do sistema foi utilizado o *framework* Bootstrap, que fornece o *Cascading Style Sheets* (CSS) para os componentes *HyperText Markup Language* (HTML) utilizados. A Listagem 2 apresenta o código que é necessário para que o *framework* seja adicionado ao projeto.

```
@import "bootstrap-sprockets"
@import "bootstrap"
```

Listagem 2 – Adição do Bootstrap ao projeto

Para o cadastro de fotos e de cardápios dos estabelecimentos foi utilizada a Gem Paperclip, que permite gerenciar o envio e a visualização das imagens. Para o funcionamento do *upload* de imagem, foi adicionada a anotação '*f.file_field*' na página HTML, como mostrado na Listagem 3.

```
<%= form_for(photo) do |f| %>
  <% if photo.errors.any? %>
    <div id="error_explanation">
      <h2><%= pluralize(photo.errors.count, "error") %> prohibited
this photo from being saved:</h2>

      <ul>
        <% photo.errors.full_messages.each do |message| %>
          <li><%= message %></li>
        <% end %>
      </ul>
    </div>
  <% end %>

  <div class="field" style="display: none">
    <%= f.label :restaurant_id %>
    <%= f.text_field :restaurant_id %>
  </div>

  <div class="field">
    <%= f.label :photo, 'Foto' %>
    <%= f.file_field :photo %>
  </div>

  <div class="actions">
    <%= f.submit %>
  </div>
<% end %>
```

Listagem 3 – Gem Paperclip para inclusão de imagens

Também foi adicionada a anotação de '*ImageUploader*' no modelo do banco, conforme apresentado na Listagem 4.

```
class Photo < ApplicationRecord
  belongs_to :restaurant
  mount_uploader :photo, ImageUploader
end
```

Listagem 4 – Anotação ImageUploader

Na aplicação *mobile* foi utilizada a implementação nativa para dispositivos

Android. Conforme pode ser visto na Listagem 5, foi utilizado o mapa do Google para a identificação de locais.

```

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    setConfiguracoesMapa();

    constroiGoogleApiClient(this);

    this.mMap.setOnInfoWindowClickListener(new
    GoogleMap.OnInfoWindowClickListener() {
        @Override
        public void onInfoWindowClick(Marker marker) {
            Integer idRestaurant = hashMarkers.get(marker);
            Intent intent = new Intent(MainActivity.this,
RestaurantActivity.class);
            intent.putExtra("restaurant", idRestaurant);
            MainActivity.this.startActivity(intent);
        }
    });
}

private void setConfiguracoesMapa() {

    if (ActivityCompat.checkSelfPermission(MainActivity.this,
android.Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED &&

    ActivityCompat.checkSelfPermission(MainActivity.this,
android.Manifest.permission.ACCESS_COARSE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {

        mMap.setMyLocationEnabled(true);
        mMap.getUiSettings().setZoomControlsEnabled(true);
        mMap.getUiSettings().setCompassEnabled(false);
        mMap.getUiSettings().setMapToolbarEnabled(false);
        mMap.setMapType(1);
    }
}

private synchronized void constroiGoogleApiClient(Context context)
{
    this.mGoogleApiClient = new GoogleApiClient.Builder(context)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)
        .build();

    this.mGoogleApiClient.connect();
}

```

Listagem 5 – Uso de GoogleMaps

Para a comunicação dos dados entre o sistema *web* e a aplicação *mobile* foi utilizado o Retrofit que é uma API para comunicações *Hypertext Transfer Protocol* (http). Na Listagem 6 é possível verificar como é feita a configuração dessa API e sua utilização por meio do método de listagem de fotos.

```

public class RestHelper {
    private RestApi api;
    public RestHelper () {
        api = new Retrofit.Builder()
            .baseUrl("https://find-restaurant-
jeanwill.c9users.io/")
            .addConverterFactory(GsonConverterFactory.create())
            .build().create(RestApi.class);
    }

    public void listPhotos(Integer restaurantId, final
GenericResultReceiver receiver) {
        final Bundle b = new Bundle();
        receiver.send(GenericResultReceiver.RUNNING, Bundle.EMPTY);

        Call<JsonArray> call = api.listPhotos(restaurantId);
        call.enqueue(new Callback<JsonArray>() {

            public void onResponse(Call<JsonArray> call,
Response<JsonArray> response) {
                b.putSerializable("response",
response.body().toString());
                receiver.send(GenericResultReceiver.LIST_PHOTOS, b);
            }

            @Override
            public void onFailure(Call<JsonArray> call, Throwable t)
{
                Log.e(RestHelper.class.getSimpleName(), "Falha ao
comunicar com o servidor" + t.getMessage(), t);
                b.putSerializable("error", t);
                receiver.send(GenericResultReceiver.ERROR, b);
            }
        });
    }
}

```

Listagem 6 – Uso da API Retrofit

5 CONCLUSÃO

O objetivo deste trabalho foi implementar um sistema *web* com um aplicativo *mobile* vinculado para cadastro, consulta (busca) e avaliação de estabelecimentos que fornecem o serviços de refeições como restaurantes, pizzarias, bares e outros. Visando, assim, facilitar o acesso dos usuários que estão à procura de um estabelecimento bem avaliado para realizar suas refeições. Para esse desenvolvimento foram utilizadas as linguagens de programação Java para o aplicativo *mobile* e Ruby para o sistema *web*.

O aplicativo tem como sua principal funcionalidade a busca pelos estabelecimentos na região de localização do usuário. Para que esses estabelecimentos sejam visualizados pelos usuários é necessário que eles sejam previamente cadastrados por meio do sistema *web*.

A solução *web* foi desenvolvida utilizando como linguagem o Ruby em conjunto com um de seus principais *frameworks* para esse tipo de aplicações, o Rails. Para ser possível o acesso aos dados inseridos na aplicação pelo cliente *mobile*, ela foi hospedada na nuvem, possibilitando assim, que os dados sejam consumidos por usuários, independentemente de sua localização.

A tecnologia de aplicações em nuvem é bastante utilizada, pois permite às empresas não somente reduzir custos com a aquisição de servidores próprios, como também reduzir os custos com a manutenção e instalação dos mesmos, sendo assim uma opção muito viável para a utilização em aplicações de pequeno e médio porte.

O trabalho realizado atingiu seus objetivos, podendo ser utilizado por pessoas que estejam interessadas em encontrar e/ou conhecer novos restaurantes na sua cidade ou região.

REFERÊNCIAS

AMBROS, Luisa. **Diferença entre aplicativos nativos, híbridos e mobile web apps**. 2013. Disponível em: <<http://www.luisaambros.com/blog/diferenca-entre-aplicativos-nativos-hibridos-e-mobile-web-apps/>>. Acesso em: 20 abr. 2017.

ANDROID DEVELOPER. **Android, the world's most popular mobile platform**. 2012. Disponível em: <<https://developer.android.com/about/android.html>>. Acesso em: 08 mai. 2017.

FOODSPOTTING. Disponível em: <<http://www.foodspotting.com/find/in/The-World>>. Acesso em: 13 jun. 2017.

FOURSQUARE. Disponível em: <<https://pt.foursquare.com/>>. Acesso em: 13 jun. 2017.

GOOGLE. **Google Maps**. Disponível em: <<https://www.google.com/intl/pt-BR/maps/about/>>. Acesso em: 18 mar. 2017.

GOOGLE PLAY. **Top gratuitos em Apps Android**. Disponível em: <https://play.google.com/store/apps/collection/topselling_free>. Acesso em: 18 mar. 2017.

HAMMERSCHMIDT, Roberto. **Linha do tempo: por dentro da evolução do Android**. 2015. Disponível em: <<https://www.tecmundo.com.br/android/82344-linha-tempo-dentro-evolucao-do-sistema-android.htm>>. Acesso em: 13 jun. 2017.

HERE. **We are here**. Disponível em: <https://company.here.com/here/>. Acessado em: 18 mar. 2017.

IFOOD. Disponível em: <<https://ifood.delivery/br/sobre>>. Acesso em: 13 jun. 2017.

LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao Processo Unificado**. 2 ed. Porto Alegre: Bookman, 2004.

NEKA, Evelyn. **Geolocalização no desenvolvimento de aplicativos mobile é importante?**. Disponível em: <https://pt.yeeply.com/blog/geolocalizacao-no-desenvolvimento-de-aplicativos-mobile-e-importante/>. Acesso em: 20 mai. 2017.

OGLIARI, Ricardo da Silva; BRITO, Robison Cris. **Android - do básico ao avançado**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2014.

PRESSMAN, Roger S. **Engenharia de Software**, 6 ed. São Paulo: McGraw Hill,

2006.

RAILS. **Getting started with Rails**. Disponível em:
<http://guides.rubyonrails.org/getting_started.html>. Acesso em: 13 jun. 2017.

REVISTA EXAME. **Número de smartphones supera o número de computadores no Brasil**. 2015. Disponível em:
<<http://exame.abril.com.br/tecnologia/noticias/numero-de-smartphones-supera-o-de-computadores-no-brasil>>. Acesso em: 20 abr de 2017.

RUBY. **Sobre o Ruby**. Disponível em: < <https://www.ruby-lang.org/pt/about/>>.
Acesso em: 13 jun. 2017.

SCORZA, Leonardo. **A história do Rails**. 2016. disponível em:
<<http://onebitcode.com/a-historia-do-rails/>>. acesso em: 13 jun. 2017.

TERRA. **TOP 5 – Cinco sistemas operacionais para celular**. 2015. Disponível em:
<<http://noticias.terra.com.br/ciencia/top-5-cinco-sistemas-operacionais-para-celular,1e40d2b834bdd310VgnCLD200000dc6eb0aRCRD.html>> Acesso em: 18 mar. 2017.

TRIP ADVISOR. Disponível em:
<https://play.google.com/store/apps/details?id=com.tripadvisor.tripadvisor>. Acessado em: 18 mar. 2017.

WAZE. **Sobre nós**. Disponível em: <https://www.waze.com/pt-BR/about>. Acessado em: 18 mar. 2017.

YELP. **About us**. Disponível em: <http://www.yelp.com/about>. Acessado em: 18 mar. de 2017.