UNIVERSIDADE FEDERAL DO PIAUÍ
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS
CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO

# FINDPICOS: PROTÓTIPO DE APLICAÇÃO MÓVEL PARA BUSCA E AVALIAÇÃO DE ESTABELECIMENTOS

ANDRÉ LUIZ DA SILVA LIMA

PICOS – PIAUÍ
2016

ANDRÉ LUIZ DA SILVA LIMA

# FINDPICOS: PROTÓTIPO DE APLICAÇÃO MÓVEL PARA BUSCA E AVALIAÇÃO DE ESTABELECIMENTOS

Trabalho de Conclusão de Curso apresentado como requisito parcial para a Conclusão do Curso de Bacharelado em Sistemas de Informação da Universidade Federal do Piauí - Campus Picos.

Orientador: Prof. Esp. Thiago José Barbosa Lima

PICOS – PIAUÍ

2016

**FICHA CATALOGRÁFICA**

**Serviço de Processamento Técnico da Universidade Federal do Piauí**

**Biblioteca José Albano de Macêdo**

**L732f** Lima, André Luiz da Silva.

FINDIPICOS: protótipo de aplicação móvel para busca e avaliação de estabelecimentos / André Luiz da Silva Lima.– 2016.

CD-ROM : il.; 4 ¾ pol. (54 f.)

Monografia (Curso Bacharelado em Sistemas de Informação) – Universidade Federal do Piauí, Picos, 2016.

Orientador(A): Profª. Esp. Thiago José Barbosa Lima

1. *Android*. 2. Dispositivo Móvel. 3.Estabelecimentos-Busca . I. FindPicos.

**CDD 005.2**

FINDPICOS: PROTÓTIPO DE APLICAÇÃO MÓVEL PARA BUSCA E AVALIAÇÃO DE ESTABELECIMENTOS

ANDRÉ LUIZ DA SILVA LIMA

Monografia aprovada como exigência parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Data de Aprovação

Picos – PI, 11 de Fevereiro de 2016

Thiago José Barbosa Lima
Prof. Esp. Thiago José Barbosa Lima
Orientador

Patrícia Vieira da Silva Barros
Profª. Ma. Patrícia Vieira da Silva Barros
Membro

Allan J. R. Gonçalves
Prof. Esp. Allan Jheyson Ramos Gonçalves
Membro

## DEDICATÓRIA

Em primeiro lugar a Deus, por estar presente em todos os momentos...

A minha família que foram companheiros em todas as horas...

## AGRADECIMENTOS

Agradeço primeiramente a Deus por minha existência.

Aos meus pais, pela educação e motivação.

A minha esposa Ingrid, pessoa especial e que esteve presente em todos os momentos.

Ao meu irmão, que sempre me ajudou quando precisei.

Ao meu filho, por sempre me deixar feliz até em momentos difíceis.

A todos que, com boa intenção, colaboraram para a realização e finalização deste trabalho.

A todos os professores do curso, que foram tão importantes na minha vida acadêmica e no desenvolvimento desta monografia.

Ao professor Orientador de TCC, Thiago José Barbosa Lima, que sempre me incentivou a pesquisar mais para dar maior qualidade à minha monografia.

“Sonhos determinam o que você quer. Ação determina o que você conquista”.

Aldo Novak

## Resumo

Devido ao constante crescimento do número de usuários, os dispositivos móveis têm se tornado ferramentas essenciais na vida das pessoas, já que seus recursos são sem dúvidas de grande utilidade. O objetivo geral deste projeto foi a criação de um protótipo de aplicativo para dispositivos móveis na plataforma *Android,* com o intuito de auxiliar os usuários a encontrar de forma rápida pontos de interesse na cidade de Picos-PI, pelo fato de muitas vezes o usuário sentir a necessidade de conhecer um local novo, ficando com receio de perder seu tempo ou dinheiro. O FindPicos faz uso do conceito de *web service* para fornecer informações passadas pelos próprios donos dos estabelecimentos provendo assim uma forma de publicidade, além de deixar aos usuários todo o conteúdo que seu estabelecimento tem a oferece-los. As ferramentas utilizadas para a concretização desse projeto foram o Java, como principal linguagem de programação, o Eclipse, como plataforma de programação (IDE), MySQL, como banco de dados, e um Web Service construído na linguagem *PHP*.



**Palavras-chave:** *Android, Web Service*, Busca de estabelecimentos.

**Abstract**

Due to the constantly growing number of users, mobile devices have become essential tools in people's lives, as their resources are no doubt very useful. The aim of this project was to create a prototype application for mobile devices on the Android platform, in order to help users quickly find points of interest in the city of Picos-PI, because often you feel the need to know a new place, being afraid to waste your time or money. The FindPicos makes use of the web service concept to provide information passed by the owners of the establishments thus providing a form of advertising, as well as let users all the content that your property has to offer them. The tools used for the realization of this project were the Java as the main programming language, Eclipse, as a development platform (IDE), MySQL as the database, and a Web service built on PHP language.

**Keywords:** Android, Web Service, establishments Search.

LISTA DE FIGURAS

LISTA DE QUADROS

# LISTA DE ABREVIATURAS

| | |
|---|---|
| *ADT* | *Android Developer Tool* |
| *API* | *Application Programming Interface* |
| *CSS* | *Cascading Style Sheets* |
| *GPS* | *Global Positioning System* |
| *HTML* | *HyperText Markup Language* |
| *IDE* | *Integrated Development Environment* |
| *LBS* | *Location-based Service* |
| *OHA* | *Open Handset Alliance* |
| *PHP* | *Personal Home Page* |
| *SDK* | *Software Development Kit* |
| *SGBD* | *Sistema Gerenciador de Banco de Dados* |
| *SQL* | *Structured Query Language* |
| *VM* | *Virtual Machine* |

**SUMÁRIO**

# 1 INTRODUÇÃO

Com o grande aumento do uso de aplicações móveis que desempenham a função de auxilio de seus usuários, desenvolvedores procuram criar novos aplicativos. O aumento da popularidade dos smartphones combinado com a crescente disponibilidade de internet tem criado um ambiente ideal para uma maior demanda de aplicações para dispositivos móveis (ARIMA, 2009).

A Internet surgiu na década de 50 através de esforços militares e acadêmicos, foi desenvolvida e cresceu mundialmente ao longo dos anos. De acordo com uma pesquisa realizada pela Tic Domicílios, desenvolvida pelo Comitê Gestor da Internet no Brasil (CGI.br) e divulgada em 15 de setembro de 2015, 47% dos brasileiros com dez ou mais anos usaram o telefone móvel para navegar na *web.* Destes, 84%, afirmaram que o fazem todos os dias ou quase todos os dias. Isso mostra que a *internet* está cada vez mais presente na vida das pessoas. Além disso surgiram várias tecnologias que fazem o uso da *internet*.

A tecnologia de Serviços *Web* oferece recursos para a utilização, através da *Internet,* de classes e métodos remotos. Seu objetivo é permitir que aplicações de diferentes plataformas se comuniquem através de uma rede.

Fazendo o uso da tecnologia *web service* pela plataforma *Android,* o aplicativo FindPicos traz informações sobre diversos estabelecimentos existentes na cidade de Picos, de maneira clara e fácil para os usuários do aplicativo. A ideia surgiu a partir de duas necessidades:

- As pessoas precisam conhecer o que um estabelecimento tem a oferecer sem precisar se deslocar até o estabelecimento, evitando assim perda de tempo ou dinheiro ao se deslocar a determinado estabelecimento, ou até mesmo o fato de não souberem onde encontrar um local que ofereça um serviço de qualidade.
- Pequenas empresas/estabelecimentos não usam de publicidades a seu favor, pelo fato de até muitos desses estabelecimentos não terem condições financeiras para isso.

O sistema *web* além de prover informações para o aplicativo *Android,* disponibiliza uma área, onde o próprio dono do estabelecimento poderá gerenciar à sua maneira, podendo criar seus anúncios a vontade.

Com isso, o principal objetivo deste trabalho foi a criação de um protótipo de um aplicação desenvolvida para dispositivos móveis, mais precisamente para a plataforma *Android,* capaz de solucionar esse problema para os usuários, um aplicativo que facilita a busca de

estabelecimentos através de ferramentas simples e fácil utilização, o usuário pesquisa o seu ponto de interesse (algo que o usuário busca) e como resultado uma lista com os principais será listada, contendo várias informações a respeito do estabelecimento. Além disso, ainda servirá como uma ótima opção de propaganda para os estabelecimentos de menor porte.

## 1.1 Objetivos

### *1.1.1 Objetivo Geral*

O objetivo geral deste trabalho é a criação de um aplicativo para dispositivos móveis na plataforma *Android,* com o intuito de auxiliar os usuários a encontrar de forma rápida pontos de interesses na cidade de Picos - PI. O Aplicativo também servirá como forma de divulgação de empresas de todos os setores, onde, através do sistema *web* será possível o cadastramento dessas empresas. O sistema *web* do aplicativo FindPicos disponibiliza uma área para que essas empresas divulguem seus negócios e todas essas informações podem ser visualizadas a qualquer momento por todos os usuários do aplicativo móvel.

### *1.1.2 Objetivos Específicos*

- Estudo sobre a implementação e funcionamento da plataforma *Android*.
- Desenvolver um protótipo que possibilite a consulta de estabelecimentos por parte dos usuários.
- Desenvolver um sistema *web.*
- Analisar as tendências de mercado na cidade onde o protótipo será lançado.
- Elaborar uma *interface* amigável e chamativa.
- Realizar testes com usuários interagindo com o protótipo

## 1.2 Metodologia

A metodologia implementada durante a realização desse trabalho teve 4 etapas: Estudo sobre a viabilidade do aplicativo, onde é feita uma pesquisa com o intuito de verificar o quanto a criação desse aplicativo será útil. Levantamento bibliográfico, onde é realizado um estudo das áreas de *Web Service,* plataforma *Android* e linguagens de programação *web.* Desenvolvimento do projeto, onde, com base nos estudos realizados, é dado início ao desenvolvimento do protótipo incluindo a análise de requisitos, projeto, implementação e testes. E por fim, o

Avaliação, onde o protótipo desenvolvido é avaliado em relação a sua utilidade, tempo de resposta, usabilidade, interface gráfica, tipos de buscas e quantidade de informações disponibilizadas, coletados por meio de questionário

## 1.3 Organização do Trabalho

Após a introdução, onde foi apresentado os principais motivos do desenvolvimento deste trabalho, serão apresentados os próximos capítulos, organizados da seguinte maneira: no capitulo 2 é abordado toda a fundamentação teórica apresentando e explicando alguns conceitos utilizados durante o desenvolvimento deste trabalho. No capítulo 3 foi abordado sobre as várias etapas para o desenvolvimento deste trabalho, como e onde é utilizada cada tecnologia apresentada no capitulo 2, além da apresentação das funcionalidades tanto do sistema *web* como aplicação móvel. No capítulo 4 são mostrados os resultados de alguns testes realizados. E por fim, no capítulo 5 foi feita a conclusão a respeito de todo o trabalho, além da indicação de trabalhos futuros.

# 2 FUNDAMENTAÇÃO TEÓRICA

Aqui serão apresentadas algumas explicações mais detalhadas sobre as ferramentas utilizadas durante o desenvolvimento deste trabalho, entre elas temos a plataforma *Android* e suas tecnologias, a linguagem *PHP*, a *IDE* de desenvolvimento, o SGBD *MySQL e Web Service* e suas vantagens.

## 2.1 *Android*

O *Android* é um sistema operacional totalmente personalizável e fácil de usar e já conta com mais de um bilhão de dispositivos ao redor de todo o mundo, a maioria com smarthphones, mas também pode ser facilmente encontrado nos dias atuais em outros dispositivos que vão desde tablets a relógios, tvs, carros e outros mais.

O sistema operacional *Android* teve seu desenvolvimento iniciado em 2003 pela empresa *Android* Inc. Em 2005, a empresa foi adquirida pela *Google*, que hoje lidera o desenvolvimento do *Android* (DEITEL, 2013).

Este Sistema Operacional voltado para dispositivos móveis foi idealizado por uma aliança de grandes empresas da área de tecnologia e telecomunicações que estão tentando criar uma plataforma padrão e aberta para dispositivos móveis. Esta aliança foi batizada de OHA e é composta por empresas como: *Acer, Alcatel, Asus, Dell, HTC, Lenovo, LG, Motorola*, *Samsung*, *Sony Ericsson* e *Toshiba*, entre outras (LECHETA, 2010).

O *Android* não distingue aplicações internas das aplicações escritas pelo *SDK*, o que permite desenvolver aplicações robustas utilizando os recursos disponíveis pelo dispositivo (ABLESON, 2012). Vale lembrar que a plataforma *Android* não é uma plataforma de hardware, e sim um *software* para dispositivos móveis.

### 2.1.1 Arquitetura Android

A arquitetura da plataforma *Android* é formada por várias camadas, onde cada camada possui sua função específica. Na Figura 2.1 veremos a especificação de cada camada.


**Figura 2.1** – *Arquitetura Android (Google, 2014)*

1. ***Applications*:** Onde ficam localizados os aplicativos desenvolvidos em *Java* e que são executados pelo sistema *Android* tais como: *e-mail*, calendário, relógio entre outros. Nessa camada ocorre a interação entre os mais diferentes aplicativos, enquanto o restante só será acessível aos desenvolvedores e fabricantes do *software*.

2. ***Application Framework*:** Essa camada é responsável pelos programas que gerenciam as funções básicas do telefone, o desenvolvimento está voltado para o uso de *APIs* das aplicações chaves do *Android.* Também facilita o desenvolvimento e reuso, além de possibilitar o uso de aplicações mais simples como apoio para construção de aplicações mais complexas.

3. ***Libraries*:** Nessa camada são disponibilizadas várias bibliotecas que possibilitam o acesso aos componentes da plataforma *Android*, como por exemplo, banco de dados.

4. ***Android Runtime*:** Nessa camada encontra-se a máquina virtual *Dalvik*, que foi criada exclusivamente para o sistema *Android.* Na *Dalvik*, são executados processos de aplicação de uma forma otimizada para o baixo consumo de memória. A *VM Dalvik* age como uma camada intermediária e utiliza os recursos do *kernel* do *Linux* para lidar com funcionalidades de baixo nível, como a segurança, processos, *thread* e gerenciamento de memória. Estas tarefas fortemente ligadas ao *hardware* são tiradas do desenvolvedor,

que passa a se preocupar somente com as regras ligadas ao desenvolvimento do *software* (MEIER, 2010).

5. ***Kernel Linux*:** O Google usou a versão 2.6 do *Linux* para construir o *kernel* do *Android*, o que inclui os programas de gerenciamento de memória, as configurações de segurança, o *software* de gerenciamento de energia e vários *drivers* de *hardware* (STRICKLAND, 2009).

### *2.1.2 Android SDK*

O *Android SDK* é o *software* utilizado para desenvolver aplicações no *Android*, que tem um emulador para simular o celular, ferramentas utilitárias e uma *API* completa para a linguagem *Java*, com todas as classes necessárias para desenvolver as aplicações (LECHETA, 2013).

Juntamente com o *SDK*, será necessário o uso do *ADT*, que se trata de um *plug-in* que estende as funcionalidades da *SDK Android* para a *IDE Eclipse*, facilitando assim o desenvolvimento de aplicações para a plataforma *Android.*

### *2.1.3 Intents*

Uma *Intent* (intenção) é uma descrição abstrata de uma operação a ser executada. Ela pode ser utilizada para iniciar uma *Activity* e ao mesmo tempo enviar uma mensagem para uma aplicação que roda em outro processo. Uma *Intent* faz parte da arquitetura do *Android* e é um conceito básico que deve ser dominado por todos que desejam programar para *Android*. Na Figura 2.2, podemos ver a estrutura de uma *Intent*.

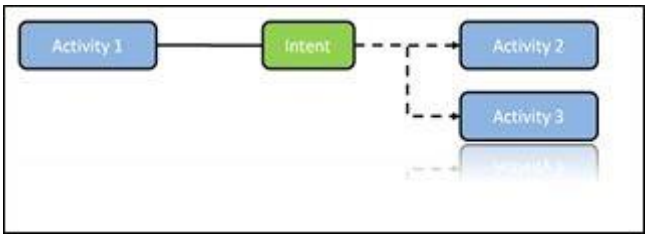



**Figura 2.2** – *Estrutura de uma Intent*

Fonte: http://www.programarandroid.com.br/2013/04/fundamental-trabalhando-com-intent-e.html.

### *2.1.4 Activities*

Uma *activity* é responsável por controlar apenas uma tela dentro de uma aplicação, quando uma aplicação é iniciada sobre um *activity* temos o método *OnCreate()* nesse método, ele passa o seu *layout* para ser inicializado junto com seus componentes.

Uma *activity* é o componente do *Android* responsável por representar uma *interface* visual, ou seja, com este componente é possível controlar o comportamento do *layout* da aplicação (LECHETA, 2013).

Uma *activity* é basicamente, o que vai gerenciar uma certa parte gráfica, a partir dela será possível controlar todas as ações do *layout* correspondente, ou seja, quando uma tela é chamada, na verdade sua *activity* que é lançada. Para que a classe responsável por exibir a *interface* gráfica para o usuário seja lançada com sucesso, não basta somente implementar os métodos que a classe *activity* exige. É necessário que toda e qualquer classe que estenda de *activity* esteja registrada no arquivo de configuração do *Android*, o *AndroidManifest.xml*.

Todas as aplicações *Android* precisam ter um arquivo *AndroidManifest.xml* e que esteja necessariamente com o mesmo nome e extensão. Ao iniciar uma aplicação esse arquivo é o responsável por fornecer informações essenciais sobre a aplicação para o sistema *Android.* Nesse arquivo ocorrem as seguintes tarefas, informar o nome do pacote *Java* da aplicação, declarar todos os componentes da aplicação como *Activities, Services, Providers*, entre outros, declarar quais permissões a aplicação deve ter, além de outras várias tarefas.

### 2.1.5 *Interface* do Usuário

A plataforma *Android* possui uma vasta quantidade de componentes de *interface* já pré-implementados, como *layouts* e entre outros que possibilitem ao desenvolvedor criar livremente a *interface* gráfica para sua aplicação. Os principais componentes de *interface* gráfica encontrados no *Android* são: os *layouts* (*LinearLayout, RelativeLayout, GridView* e *ListView*), os controles de entrada (*Buttons, EditTexts, RadioButton, CheckBoxes, Spinners* e *Pickers*), os componentes para exibir informações (*TextView e ImageView*), a *actionbar*, menus, componentes de configuração de aplicação, entre outros. Alguns desses componentes são descritos a seguir:

- **LinearLayout** - utilizado para organizar *views* verticalmente ou horizontalmente;

- **ListView** - componente utilizado para mostrar itens em uma lista vertical;

- **Button** - componente utilizado para servir como um botão, respondendo a cliques do usuário;
- **TextField** - também chamado de EditText, é o componente onde o usuário pode digitar textos, números, etc;

- **TextView** - componente utilizado para mostrar um texto na tela;

### *2.1.6 Java*

*Java* é a linguagem de programação orientada a objetos, desenvolvida pela *Sun Microsystems*, capaz de criar tanto aplicativos para *desktop*, aplicações comerciais, *softwares* robustos, completos e independentes, aplicativos para a *web* (SOBRAL, 2008).

A característica mais marcante dessa linguagem é que programas criados nela não são compilados em código nativo da plataforma. Programas em *Java* são compilados para um *bytecode*, que é executado por uma máquina virtual, o que permite aos desenvolvedores criarem um programa uma única vez e depois executar este em qualquer uma das plataformas suportadas pela tecnologia. Além de tudo a linguagem *Java* é de código-fonte aberto.

A linguagem *Java* já se firmou com uma alternativa madura e flexível para o desenvolvimento de vários tipos de sistemas e ao longo dos anos desde sua criação, foi incorporado a linguagem extensões, *APIs,* além de materiais de apoios em formas de livros e tutorias possibilitando o desenvolvimento em diversas áreas, inclusive na área *mobile.* Para a implementação do aplicativo foi utilizada a linguagem de programação *Java*, por ser a linguagem padrão de programação para *Android*.

## 2.2 PHP

O *PHP* é uma linguagem de *script open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento *web* e que pode ser embutida dentro do *HTML*.

O *PHP* também tem como uma das características mais importantes o suporte a um grande número de bancos de dados, como *Interbase, mySQL, Oracle, PostgreSQL*, e vários

outros. Além disso, *PHP* tem suporte a outros serviços através de protocolos como *IMAP*, *SNMP*, *NNTP*, *POP3* e, logicamente, *HTTP* (ROCHA, 2003).

É importante deixar claro que, apesar da grande reformulação dos recursos de orientação a objetos, o *PHP* continuará sendo uma linguagem procedural, já que ela é baseada no conceito de chamada de procedimentos. O objetivo que levou a equipe de desenvolvimento do *PHP* a realizar essas mudanças foi tornar a linguagem mais competitiva em relação às aplicações J2EE (como por exemplo, aquelas desenvolvidas com JSP) e .NET (NIEDERAURER, 2004). Pelas razões comentadas anteriormente, a linguagem *PHP* foi utilizada para a construção do sistema *web* implementado.

## 2.3 JavaScript

A linguagem *JavaScript* é bastante utilizada atualmente, já que ela é uma linguagem que promove uma interação entre o usuário e o navegador *web.*

> É uma linguagem de programação desenvolvida pelo *Netscape* em 1995, cujo principal objetivo era tornar os sistemas *Web* mais interativos. Com esta linguagem os programas podem, por exemplo, fazer a validação de valores inseridos pelo usuário no browser do cliente, sem a necessidade de efetuar uma requisição ao servidor. (GOODMAN, 2001)

Entre as principais características desta linguagem temos: não é necessário ter seus tipos declarados como variáveis. É uma linguagem orientada a eventos, ou seja, as funções são executadas a partir da execução de uma função. Durante a implementação do sistema *web*, a linguagem *JavaScript* foi bastante utilizada em funções de validações em tempo real, como por exemplo se o usuário digitou ou não um e-mail em formato válido para o sistema, ou ainda se o usuário preencheu todos os campos requeridos.

## 2.4 CSS

O *CSS* do inglês *Cascading Style Sheets*, é uma linguagem que serve para formatar documentos criados em *HTML*, ou seja, a linguagem possibilita montar *layouts* ao modo que desenvolvedor desejar, podendo mover imagens, textos e outros elementos proporcionando ao desenvolvedor um controle sobre a área de implementação.

A linguagem trabalha com alguns comandos e parâmetros que podem vir a alterar *layout* de um documento, como fontes, alinhamentos, margens, bordas. O *CSS* também é capaz de aplicar efeitos por passagem do ponteiro do mouse, ou seja, ao passar o mouse sobre algum elemento, algo ocorre. Durante a implementação do sistema *Web* o *CSS* foi responsável por toda a formatação do sistema *web*.

## 2.5 Eclipse IDE

O *Eclipse* é definido como uma IDE de desenvolvimento de programação, inicialmente desenvolvida pela *IBM*, que segundo notícias, gastou mais de 40 milhões de dólares no seu desenvolvimento antes de transformar essa ferramenta em *OpenSource* para um consórcio, chamado de *Eclipse.org*, que inicialmente incluiu a *Borland*, *IBM, Merant, QNX software Systems, Rational Software, Red Had, SuSE, TogetherSoft, e Webgai* (GONÇALVES, 2000).

É uma *IDE* de código aberto utilizado para desenvolvimento da linguagem *Java*, mas é também multilinguagem, ou seja, suporta outras linguagens de programação tais como C/C++ instalando-se os devidos *plug-ins* adicionais para cada linguagem. É um projeto livre de patentes por ser um *software* livre.

O *Eclipse* permite também a refatoração do código, que é uma forma organizada de reestruturar o código para ser melhorado. Um aspecto importante de uma refatoração é que ela melhora o *design* sem mudar a semântica do *design*, não adicionando nem removendo sua funcionalidade. Alguns exemplos de refatoração de código é renomear métodos, encapsular campos, extrair classes, introduzir afirmações e especializar os métodos (SANTOS e EDUARDO, 2008). A discutida *IDE* foi utilizada para implementação da aplicação, devido ser de fácil manuseio e apresentar um grande suporte aos programadores para realizar diferentes projetos de formas diversas.

## 2.6 SGBD MySQL

O *MySQL* é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem *SQL* (Linguagem de Consulta Estruturada, do inglês *Structured Query Language*) como *interface*. É atualmente um dos bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo.

Atualmente, o *MySQL* está entre os SGBDs mais populares do mundo, de acordo com um *ranking* de popularidade levantado pelo site *db-engines.com* o *MySQL* é o segundo SGBD mais popular perdendo apenas para *Oracle*, esse *ranking* leva em consideração o número de menções em sites, buscas no *google*, número de profissionais que mencionam o SGBD em seus *profiles*, número de vagas oferecidas onde o SGBD é mencionado, e discursões técnicas sobre o SGBD em alguns *sites* de fóruns. O SGBD *MySQL* foi utilizado para implementação do projeto, por apresentar um excelente desempenho e estabilidade.

## 2.7 Adobe DreamWeaver

É uma ferramenta grátis que permite a construção de websites profissionais e poderosos. Ele possui um ambiente no qual é possível trabalhar com várias linguagens como *PHP*, *JavaScript*, *Css*, *HTML*, *JSP*, *XML*, entre outros. A versão do *DreamWeaver* utilizada para a implementação do sistema *web* foi a CS6, uma versão mais recente do *software*, o *DreamWeaver* possui várias boas funcionalidades como, a programação em modo *design* onde é possível ao mesmo tempo observar como está ficando à disposição dos elementos, possui uma *interface* limpa e clara, fazendo com que os ícones sejam visualizados sem problemas. Além disso a ferramenta oferece alguns recursos que permitem executar algumas funções de modo mais simples, como por exemplo uma autenticação.

## 2.8 *Web Service*

*Web Service* é uma tecnologia que utiliza padrões para a integração entre sistemas implementados em diferentes plataformas. Simplificando, Web Service é uma maneira de expor funcionalidades para usuários *Web* através de protocolos padrão. Também podemos citar como uma aplicação identificada por uma URI (*Uniform Resource Identifier*), cujas interfaces podem ser descobertas e definidas através de artefatos XML, e que suporta interações diretas com outros softwares utilizando mensagens XML através de protocolos padrão da *Internet* (RECKZIEGEL, 2006).

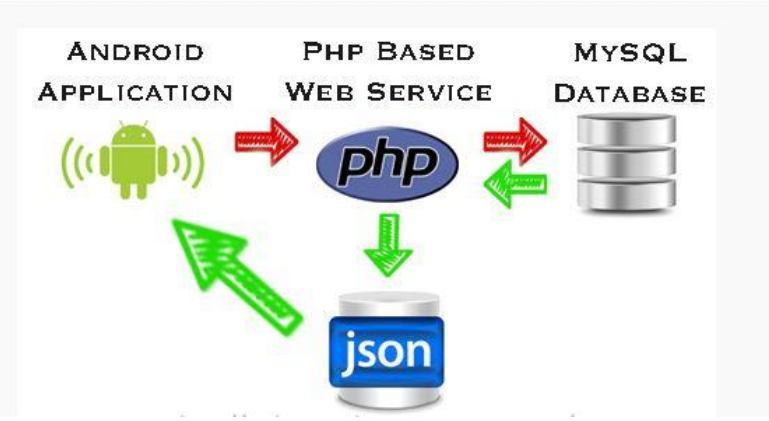



**Figura 2.3** – *Esquema do funcionamento da tecnologia Web Service*
Fonte: http://vslesson.com/vslesson/androids/how-to-connect-android-with-php-mysql.

Na Figura 2.3 podemos observar a arquitetura de uma aplicação *Android* se comunicando com uma *Web Service* desenvolvida em *PHP*. Uma aplicação *Android* irá solicitar a página *PHP* para que execute alguma operação desejada (*Create, Read, Update, Delete*, Em seguida a página *PHP* se conecta ao banco de dados para executar a operação. Assim, os dados fluem da aplicação *Android* para a página *PHP* e são armazenados no banco de dados.

O sistema que usa *web service* como meio de obtenção de dados possui várias vantagens como: O *Web Service* usa protocolos padrões como *HTTP, XML, e SOAP*, todas abertas, amplamente usadas, e com isso qualquer aplicativo utilizando os mesmos padrões poderá trocar informações, observando, assim, que a interoperabilidade é bastante forte em *Web Services*. Outra grande vantagem dos *Web Services* é que não se precisa saber nada sobre a plataforma que o está disponibilizando, ou qual a linguagem de programação que ele foi escrito.

## 2.9 *HttpClient*

Essa é a classe responsável por realizar a conexão entre o aplicativo *Android* e o sistema *web*. O *HTTP* é provavelmente o protocolo mais usado na internet. Dados, em um formato ou outro, são armazenados em um poderoso servidor centralizado, os clientes se conectam a aquele servidor usando protocolos designados como forma de recuperar dados e trabalhar com eles. Durante o desenvolvimento desta aplicação *Android* foi utilizada uma classe *Java* chamada *ConexaoHttpClient* que tem como objetivo gerenciar a troca de dados entre os clientes e o servidor. Com essa classe é possível recuperar ou enviar dados usando requisições *Http get* ou *Http post* para uma página *web*. Nesta classe teremos dois métodos, o primeiro é para termos o

retorno da conexão *HttpClient* e o segundo método será responsável por criar a conexão e passar os parâmetros para o servidor via *HttpPost*..

O método *executaHttpPost* que irá trabalhar com a *url* passada, além dos parâmetros que serão enviados ou recebidos. Uma vez com essa classe pronta já é possível utiliza-las através da aplicação para enviar ou receber dados de um servidor.

# 3 FINDPICOS: PROTÓTIPO DE APLICAÇÃO MÓVEL PARA BUSCA E AVALIAÇÃO DE ESTABELECIMENTOS

Nessa seção, será explicado como está sendo realizado o desenvolvimento do aplicativo FindPicos, bem como o seu funcionamento, em conjunto com o sistema *web* também implementado.

## 3.1 Sistema *WEB*

O sistema *web* foi desenvolvido visando além de servir como web service para o aplicativo *Android*, fazer com que algumas empresas de menor expressão possam anunciar seus serviços de forma mais barata. Para isso um cadastro deve ser realizado no sistema, informando alguns dados requeridos da sua empresa/negócio e a partir daí todos esses dados já estarão disponíveis tanto através dos dispositivos *Android*, como no próprio site do sistema *Web*. Com o uso da tecnologia *Web Service*, foi possível a recuperação das informações das empresas, que realizaram o cadastro no sistema *Web*, para serem visualizadas através da aplicação móvel.

### *3.1.1 Requisitos do Sistema Web*

Requisito de *software* é uma condição ou uma capacidade com o qual o sistema deve estar de acordo, expressando as necessidades do cliente (PRESSMAN, 2006). Os requisitos funcionais descrevem a funcionalidade ou os serviços que se espera que o sistema realize em benefício dos usuários (FILHO, 2000).

Os requisitos do sistema web foram definidos pensando no usuário final, sendo eles bem simples e diretos. O Quadro 1 mostra os requisitos funcionais, sua descrição e suas dependências.

| **Identificador** | **Descrição** | **Dependência** |
|---|---|---|
| RF01 | O usuário possuirá Login e Senha. | RNF01 |
| RF02 | O usuário poderá anunciar qualquer tipo de serviço/negócio. | |
| RF03 | O usuário poderá anunciar quantas vezes decidir. | |
| RF04 | O usuário poderá remover seu anúncio quando quiser. | |

| | | |
|---|---|---|
| RF05 | Todo novo anúncio, antes de ser ativado, passará por moderação | RNF02 |
| RF06 | O usuário deverá informar todos os campos necessários para o cadastro; | |

**Quadro 3.1** – *Requisitos Funcionais do Sistema Web*

De acordo com o Quadro 3.1, um usuário só deverá ter acesso ao sistema possuindo uma senha de acesso, como mostra o requisito funcional número 2. O usuário pode realizar tanto o cadastro de sua empresa física, como também de algum negócio que possua de maneira autônoma. O usuário pode anunciar quantas empresas ou produtos decidir. O usuário poderá remover seu anuncio a qualquer momento. Todo novo anúncio cadastrado no sistema passará por uma análise do moderador do sistema, por questões de segurança. O usuário tem que informar todos os dados que o sistema necessita.

Os requisitos não funcionais são aqueles que não dizem respeito diretamente às funcionalidades fornecidas pelo sistema. Podem estar relacionados a propriedades de sistemas emergentes, como confiabilidade, tempo de resposta, espaço em disco, desempenho e outros atributos de qualidade do produto (FILHO, 2000). No Quadro 3.2 encontra-se os requisitos não funcionais do sistema web.

| **Identificador** | **Descrição** | **Categoria** |
|---|---|---|
| RNF01 | Apenas usuários cadastrados devem ter acesso ao sistema. | Segurança |
| RNF02 | Todo novo anúncio deve ser moderado pelo administrador do sistema. | Segurança |
| RNF03 | A interface deve ser amigável e objetiva. | Usabilidade |
| RNF04 | O sistema deverá se mostrar ágil em seus processos | Eficiência |

**Quadro 3.2** – *Requisitos Não Funcionais do Sistema*

De acordo com o Quadro 3.2, por questões de segurança somente usuários cadastrados podem anunciar, assim como todo anuncio deve passar por moderação do administrador do sistema antes de ser liberado para consulta. O sistema deve apresentar uma interface fácil de usar para que o usuário não encontre nenhum tipo de dificuldade. O sistema deverá cumprir o que promete de forma eficiente

### *3.1.2 Funcionalidades*

O sistema *web* foi desenvolvido com a linguagem *PHP*, porém em alguns momentos a linguagem *JavaScript* e *CSS* se mostrou de grande utilidade. O programa utilizado para a implementação de todas as páginas foi o *Adobe Dreamweaver*. O sistema possui um método de autenticação muito simples, basta o usuário digitar um e-mail e sua senha que estejam previamente cadastrados no sistema.


**Figura 3.1** – *Login Sistema web*

Fonte: O Autor (2016)

Na Figura 3.1, o usuário deve informar o *e-mail* e senha cadastrados no sistema, caso ainda não possua deve clicar em ‘Cadastre-se’ para realizar o mesmo. Após o clique em ‘Cadastre-se’ a seguinte tela será mostrada (Figura 3.2).

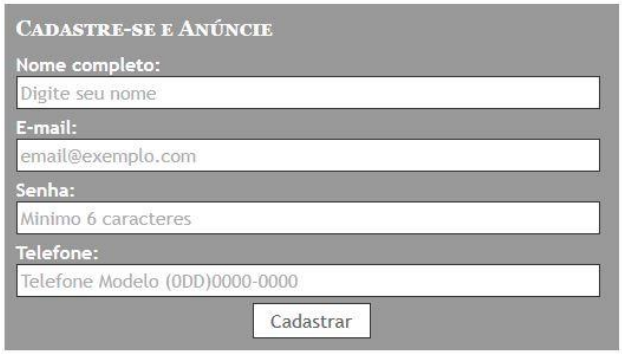



**Figura 3.2** – *Cadastro de Cliente*

Fonte: O Autor (2016)

Para que um novo usuário/empresa possa fazer seu primeiro anúncio será necessário realizar um cadastro, preenchendo nome, *e-mail*, senha e telefone para contato, como podemos observar na Figura 3.2. Após a realização deste cadastro, basta voltar a tela de *login* e informar seu *e-mail* e senha cadastrados, caso ocorra tudo bem, o menu da Figura 3.3 será mostrado.

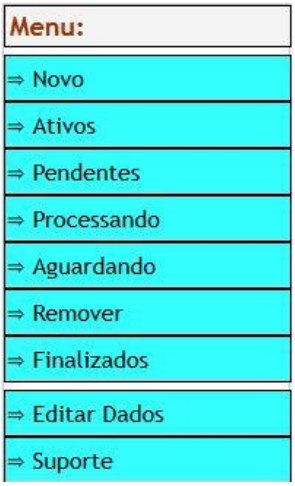



**Figura 3.3** – *Menu de Cliente*

Fonte: O Autor (2016)

Através do menu da Figura 3.3 é possível realizar várias funções, editar os dados pessoas, criar e remover anúncios, reativar anúncios finalizados entre outros. Clicando no meu 'Novo' é possível cadastrar um novo anúncio (Figura 3.3). Esse é o primeiro passo em relação a este cadastro.

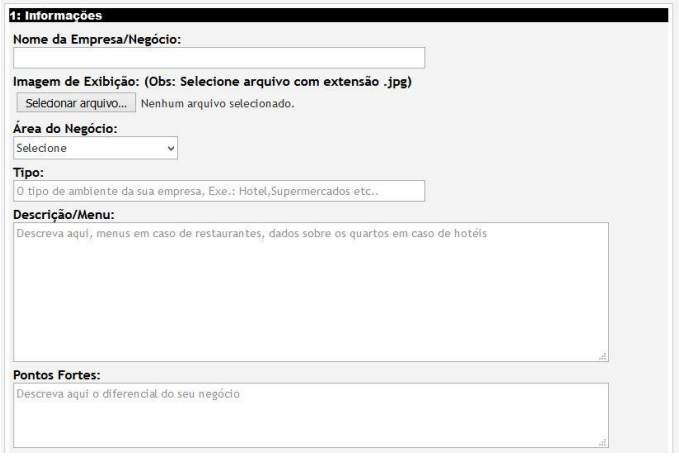



**Figura 3.4** – *Cadastrar Anúncio – Informações*

Fonte: O Autor (2016)

A Figura 3.4 mostra a primeira etapa, onde será inserido informações como, nome da empresa, área do negócio, pontos fortes, cardápios em alguns casos, entre outras informações

relativos ao ambiente, como, se possui ou não ar-condicionado, se possui ou não *wi-fi*, tipos de pagamentos aceitos pelo estabelecimento, horários de funcionamentos, e etc.. Segunda etapa são os endereços.

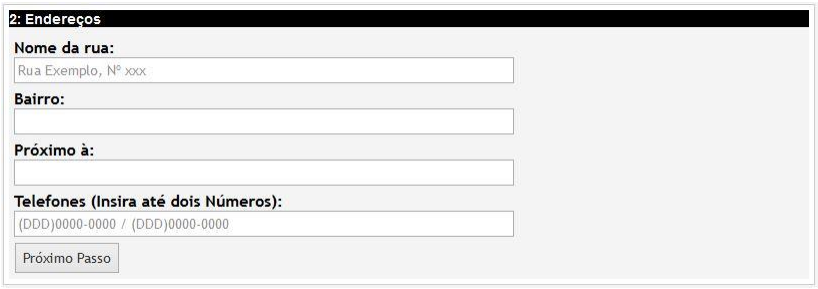



**Figura 3.5** – *Cadastrar Anúncio – Endereços*

Fonte: O Autor (2016)

A Figura 3.5 mostra a segunda etapa, onde será inserido o endereço do estabelecimento, rua, bairro, ponto de referência e contatos, caso algum cliente tenha interesse em entrar em contato diretamente com o estabelecimento. Esses dados são de muita importância, principalmente para os clientes que realizarem pesquisas pelo aplicativo.

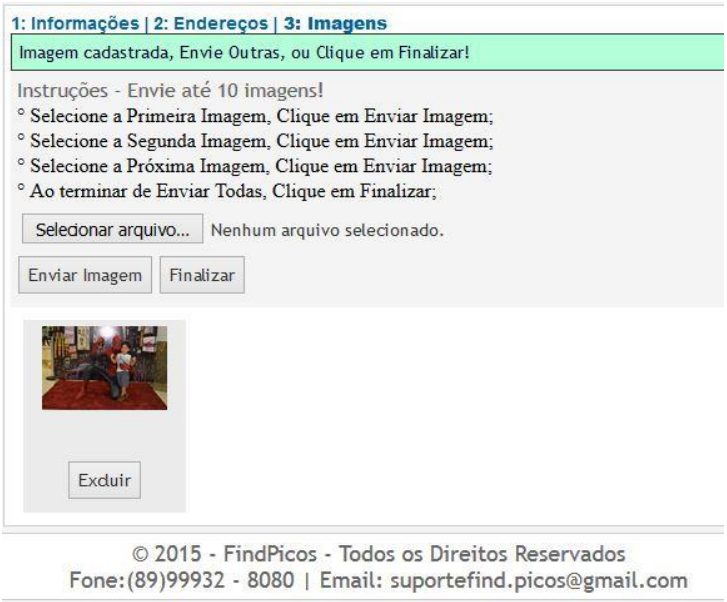
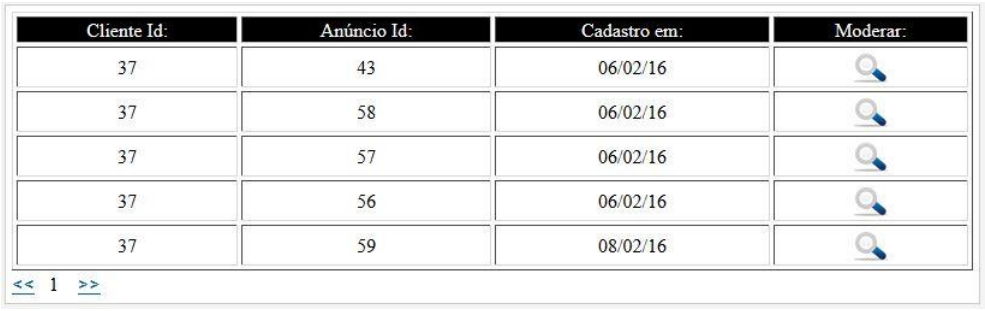



**Figura 3.6** – *Cadastrar Anúncio – Imagens*

Fonte: O Autor (2016)

A Figura 3.6, representa a parte final do cadastro de anúncio. Essas imagens não são obrigatórias. Após isso, esse anúncio cadastrado aparecerá como pendente para o administrador do sistema como mostra a Figura 3.7.



| Cliente Id: | Anúncio Id: | Cadastro em: | Moderar: |
|---|---|---|---|
| 37 | 43 | 06/02/16 | |
| 37 | 58 | 06/02/16 | |
| 37 | 57 | 06/02/16 | |
| 37 | 56 | 06/02/16 | |
| 37 | 59 | 08/02/16 | |

<< 1 >>

**Figura 3.7** – *Anúncios Pendentes*

Fonte: O Autor (2016)

A Figura 3.7 mostra todos anúncios pendentes por moderação, o administrador pode fazer isso clicando em no ícone abaixo da aba ‘Moderar’, no qual irá mostrar todos os dados cadastrados pelo cliente, onde o administrador irá decidir se aprova o anúncio ou não. Após a finalização do cadastro do anúncio, e o mesmo ter sido aprovado, todos os dados disponibilizados serão liberados para visualização tanto pelo site (Figura 3.8), tanto pelo aplicativo *Android.*


**Figura 3.8** – *Visualização site*

Fonte: O Autor (2016)

Através do menu de administração (Figura 3.9), é possível o administrador realizar várias funções como listar os anúncios pendentes para que passa estar moderando os anúncios, listar os clientes cadastrados no sistema *web*, entre outras funções.

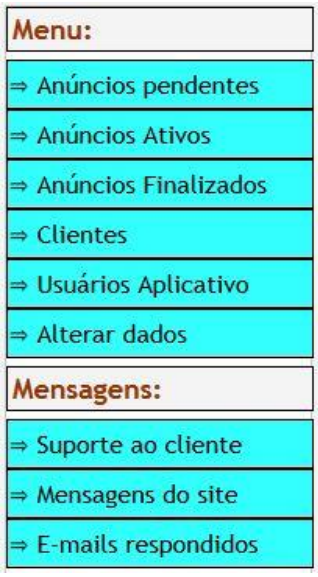



**Figura 3.9** – *Menu Administração*

Fonte: O Autor (2016)

Ainda através do menu de administrador (Figura 3.9), é possível saber quantos e quem são os usuários do aplicativo, através do item do menu 'Usuários Aplicativo' (Figura 3.10).

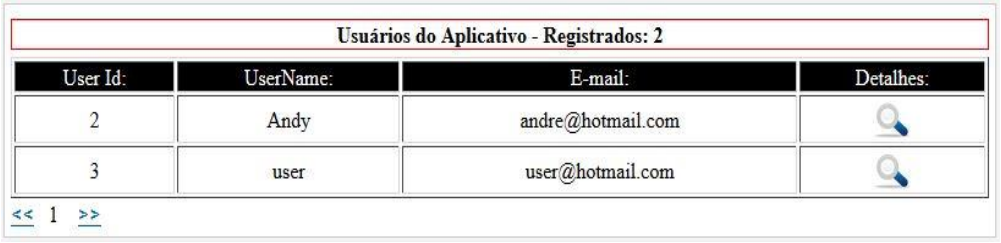



**Figura 3.10** – *Lista dos usuários cadastrados no aplicativo*

Fonte: O Autor (2016)

Na figura 3.10, podemos ver a quantidade de usuários que se cadastraram no aplicativo, além deste número, algumas informações como o *e-mail* e o nome do usuário também é disponibilizado.

## *3.2* Aplicativo

Nessa sessão será abordado como foi desenvolvimento do aplicativo na plataforma *Android*, desde seus requisitos funcionais e não funcionais, as telas existentes na aplicação, classes de conexão com o servidor, requisições de dados via *HTTP*, e o diagrama de classes de uso.

### *3.2.1 Requisitos do Aplicativo*

Nessa sessão serão discutidos os requisitos funcionais e não funcionais implementados no desenvolvimento da aplicação *Android.* Após algumas pesquisas, os requisitos funcionais e não funcionais foram definidos da forma como se pode ver nos quadros a seguir.

| **Identificador** | **Descrição** | **Dependência** |
|---|---|---|
| RF01 | O usuário possuirá Login e Senha. | RNF01 |
| RF02 | Qualquer tipo de usuário poderá se cadastrar. | |
| RF03 | A aplicação permitirá dois tipos de busca, simples e avançada. | |
| RF04 | O aplicativo deverá mostrar os estabelecimentos melhores avaliados. | |
| RF05 | O usuário do aplicativo poderá alterar ou excluir sua conta. | RNF01 |
| RF06 | O usuário terá acesso a detalhes de cada estabelecimento, previamente cadastrados. | |
| RF07 | O aplicativo trabalhará *online.* | |
| RF08 | O aplicativo não permitirá o cadastro de usernames duplicadas. | |

**Quadro 3.3** – *Requisitos Funcionais do Aplicativo*

No quadro 3.3 temos os requisitos funcionais do aplicativo, onde somente usuários cadastrados podem utilizar a aplicação, e qualquer pessoa pode se cadastrar, o aplicativo possui dois tipos de busca, simples e avançada. O aplicativo deverá mostrar uma lista contendo os estabelecimentos melhores avaliados. O usuário da aplicação, poderá excluir seu cadastro quando desejar. O usuário tem acesso aos detalhes de todos os estabelecimentos cadastrados, só é permitido o cadastro de um *username* por usuário.

| **Identificador** | **Descrição** | **Categoria** |
|---|---|---|
| RNF01 | Apenas usuários cadastrados devem ter acesso ao sistema. | Segurança |

| | | |
|---|---|---|
| RNF02 | A aplicação deve ser intuitiva para o usuário e facilitar a obtenção de dados | Eficiência |
| RNF03 | A interface deve ser amigável e objetiva. | Usabilidade |

**Quadro 3.4** – *Requisitos Não Funcionais da Aplicação*

### *3.2.2 Telas do aplicativo*

O aplicativo é composto por 6 telas principais que são: login, cadastro, principal, busca simples, busca personalizada e alterar dados. Cada tela possui sua funcionalidade específica, na Figura 3.11 é possível vermos o fluxo de chamada de telas dentro da aplicação.

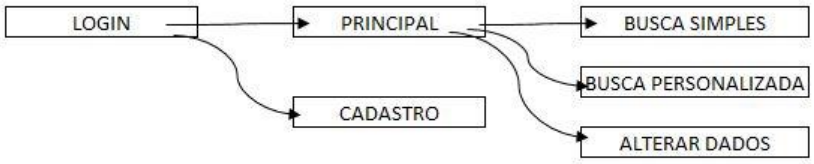



**Figura 3.11** – *Transição entre telas*
Fonte: O Autor (2016)

Como mostra a Figura 3.11, a tela de login é chamada logo quando a aplicação é iniciada. Através dela é possível ao usuário entrar no sistema, em caso de usuário já cadastrado, ou ir para a tela de cadastro, em caso de um novo usuário. A tela principal contém um menu central com várias opções de busca de estabelecimentos, como busca simples e personalizada, além da opção de listar os estabelecimentos melhores avaliados pelos clientes e a opção de alteração de dados. Algumas telas implementadas são usadas apenas como modelo para o uso de *ListViews* (listas).

A primeira tela que nos deparamos ao iniciar o aplicativo é a tela de *login* (Figura 3.12). Nela o usuário previamente cadastro poderá realizar sua autenticação e passar a usar a aplicação, já o usuário que não possui acesso poderá realizar seu cadastro clicando em cadastrar. No caso de os usuários não cadastrados tentarem acessar, uma mensagem será retornada informando que algo não está correto.


**Figura 3.12** – *Tela de Login*

Fonte: O Autor (2016)

A segunda tela do sistema é a principal, após o usuário realizar a sua autenticação é essa tela que será apresentada (Figura 3.13).


**Figura 3.13**– *Tela Principal*

Fonte: O Autor (2016)

Na tela principal, temos um menu que servirá como ponto de chamada para as outas telas da aplicação. Ao iniciada temos acima do menu as boas-vindas ao usuário que realizou o *login* na aplicação.

Na tela de cadastro (Figura 3.14), é possível criar novos usuários para terem acesso ao aplicativo, um nome, *e-mail*, *username* e senha são requisitos obrigatórios para obter sucesso durante o cadastro.

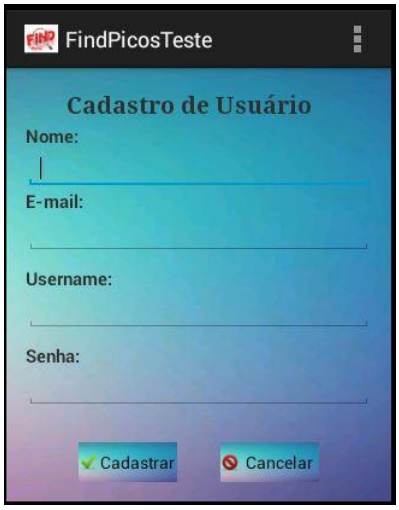



**Figura 3.14** – *Tela de Cadastro*

Fonte: O Autor (2016)

No menu principal, ao clicarmos em Busca/Simples será apresentado a tela de busca (Figura 3.15). Como o nome já exemplifica, nessa tela o usuário pode digitar o nome de um estabelecimento que ele já conhece pelo nome e poderá visualizar maiores detalhes desse local. É obrigatório a digitação de algo no espaço em branco, caso contrário o aplicativo avisará a falta do mesmo.


**Figura 3.15** – Tela *Busca Simples*

Fonte: O Autor (2016)

No menu principal, ao clicarmos em ‘Busca/Personalizada’ será apresentado a tela de busca personalizada (Figura 3.16). Aqui o usuário pode escolher entre procurar estabelecimentos que se encontram nos seguintes casos: tipo de pagamentos que aceitam, bairro

que se encontram, ou os serviços que oferecem. Não é obrigatório preencher todos os campos desta busca.

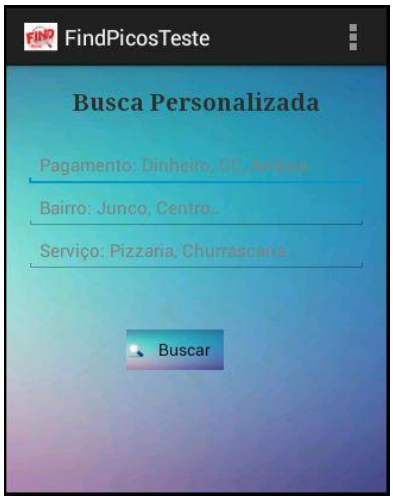



**Figura 3.16** –Tela *Busca Personalizada*

Fonte: O Autor (2016)

Para alterar a senha o usuário deve chegar a seguinte tela através do menu principal, tela de dados (Figura 3.17). Nessa tela o *username* já vem preenchido com relação ao *username* do usuário que executou *login* no aplicativo. Aqui é possível tanto alterar a senha do usuário quanto deletar o cadastro realizado na aplicação.


**Figura 3.17** – *Alterar Senha*

Fonte: O Autor (2016)

### 3.2.3 *Login via HttpPost*

Na *activity* de *Login*, é realizada da requisição de login realizado através da aplicação, que entra em comunicação com o sistema web, tendo o retorno positivo ou negativo de sua solicitação de login.

Nessa *activity*, contém o link para a página web que fará a análise dos dados enviados. Os parâmetros também serão declarados e enviados através dessa *Activity*, no caso foram enviados os dados escritos em dois *edit texts*, um nome de usuário e uma senha. O campo *respostaRetornada* receberá o resultado dessa solicitação. Na página web teremos um retorno 1 caso o usuário seja válido e 0, caso usuário inválido. Após isso, basta lermos esse retorno e analisarmos o resultado e se a resposta for igual a 1, o usuário é válido e a tela principal é chamada, caso contrário o usuário é inválido e uma mensagem de erro será mostrada.

Lembrando que após essa implementação é necessário que o arquivo *AndroidManifest.xml* tenha permissão total para acessar a internet, para isso bastar acessar o arquivo, clicar na aba *Permissions->add*, e em Name adicionar: *android.permission.INTERNET*.

### *3.2.4 Cadastro via HttpPost*

Na página *web* que foi solicitada pelo aplicativo para analisar o cadastro de novos usuários verifica três casos, onde no primeiro retorna 1 para quando o usuário for salvo com sucesso, 3 para quando um usuário tentar cadastrar um *username* já existente no banco de dados e 0 para quando algum outro erro ocorrer durante a comunicação.

Como ocorreu na classe do *login*, o mesmo ocorrerá aqui na *activity* de cadastro, com dois acréscimos de parâmetros. No caso do cadastro os parâmetros serão 4 (nome, *username*, *e-mail* e senha), e o retorno será validado como sucesso somente se vier como 1. Nessa classe também são realizadas algumas validações de campos de texto como verificação se estão vazios ou não, e verificação da digitação de um *e-mail* válido.

### *3.2.5 Recuperação de dados via HttpPost*

Para recuperar os dados requeridos do servidor é necessário enviar parâmetros para a página *web.* Na página de busca simples por exemplo, o que for digitado no *edittext* será enviado como parâmetro para a próxima *activity,* sendo que essa *activity* faz a leitura desses dados envia para a página *web* retornando os dados requeridos. Existe dois modos de enviar dados entre *activitys,* no caso foi enviado o conteúdo de um *edittext,* para isso devemos criar um *bundle.*

O *bundle* é responsável por transportar dados entre *activitis* diferentes. No caso há uma variável que será transportada. Na tela secundaria é possível recuperar esses dados e enviar como parâmetro para o servidor.

### *3.2.6 Diagrama de casos de uso*

Nessa sessão será mostrado e detalhado o diagrama de casos de uso da aplicação *Android* denominada *FindPicos*. O Caso de Uso descreve o comportamento do sistema sob diversas condições conforme o sistema responde a uma requisição de um dos stakeholders, chamado de *ator primário*" (COCKBURN, 2001).

Um Ator representa um usuário do sistema, podendo ser um humano ou outro sistema. A comunicação representa a troca que ocorre entre o ator e um caso de uso. O caso de uso tem como objetivo especificar as funcionalidades do sistema. Na Figura 3.18 temos o diagrama de casos de uso da aplicação *Android.*

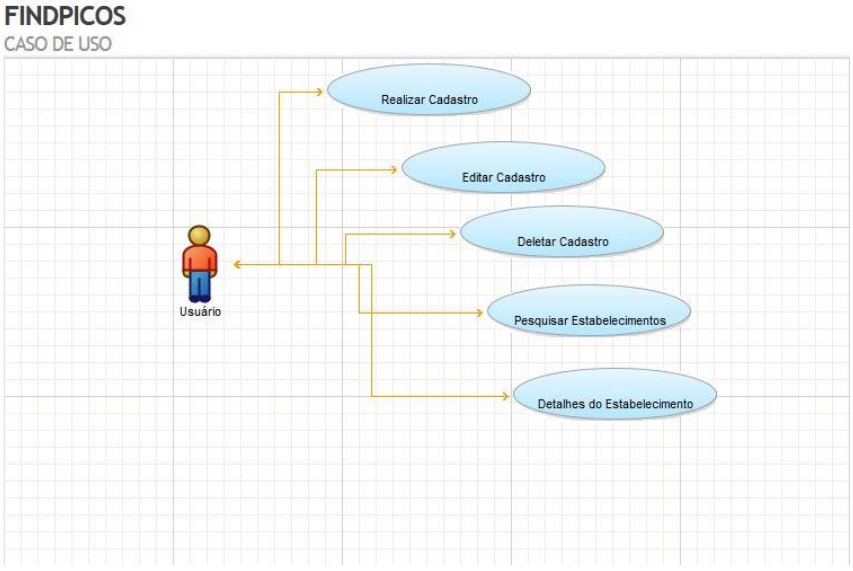



**Figura 3.18**– *Diagrama de Casos de Uso*

Fonte: O Autor (2016)

A seguir temos uma breve descrição sobre cada interação entre ator e casos ocorridos na Figura 3.18.

- Realizar Cadastro: Responsável pelo cadastro de novos usuários no servidor web. O usuário da aplicação preenche os campos, e as informações preenchidas são salvas diretamente no banco de dados do servidos.
- Editar Cadastro: Responsável pela edição/atualização do cadastro dos dados correspondentes a cada usuário. Esse caso ocorre quando o usuário deseja alterar sua senha de acesso.
- Deletar Cadastro: Responsável pela exclusão do cadastro realizado no aplicativo, caso o usuário queira.
- Pesquisar Estabelecimentos: Responsável por fazer com que o usuário realize buscas por estabelecimentos. Nesse caso existem dois tipos de busca implementadas, busca simples e busca avançada.
- Detalhes do Estabelecimento: Responsável por mostrar detalhes sobre qualquer estabelecimento cadastrado no servidor, o usuário poderá solicitar essas informações a qualquer momento.

# 4 TESTES

Após a conclusão da etapa de desenvolvimento do aplicativo, procurou-se verificar o desempenho e usabilidade do software, através de alguns testes. Os testes foram realizados na própria máquina via *ADT* enviando requisições localmente para o IP do link da página do servidor.

Para a realização de alguns testes vamos levar em consideração um usuário novo, ainda não cadastrado no sistema. Com isso após a iniciação do aplicativo, o usuário deve clicar em 'cadastrar', onde será possível a criação de um *username* e senha. Preenchendo os campos como mostra a Figura 4.1.


**Figura 4.1** – *Cadastrando novo usuário*
Fonte: O Autor (2016)

O usuário deverá informar nome, um *e-mail* válido, um *username* e uma senha. Algumas orientações devem ocorrer durante este processo de cadastramento, como, no caso de o usuário informar um *username* já existente no servidor a aplicação o orientará a digitar um novo. No caso de usuário esquecer algum campo em branco a aplicação o orientará a preenchê-la. Caso o usuário não esbarre em nenhuma regra de validação a mensagem de sucesso será mostrada (Figura 4.2).

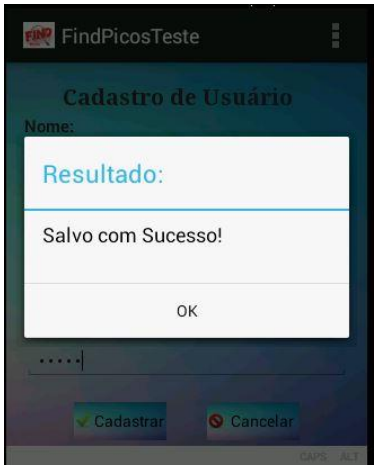



**Figura 4.2** – *Usuário Cadastrado com Sucesso*

Fonte: O Autor (2016)

Após o cadastro o usuário estará apto a realizar o login na aplicação e deve atentar-se para digitar o *username* e senha corretamente, caso ocorra um erro, o mesmo será notificado (Figura 4.3), e poderá realizar suas buscas através da tela principal. Na busca simples o usuário poderá digitar um nome de algum estabelecimento que ele já conheça pelo nome, mas que deseja saber o que eles têm a oferecer, ou, naquele caso onde o usuário veja algum estabelecimento, mas por alguma razão não quer entrar para tirar uma certa dúvida, o aplicativo pode o auxiliar nesse processo. Para isso, basta informar o nome no espaço em branco (Figura 4.4).

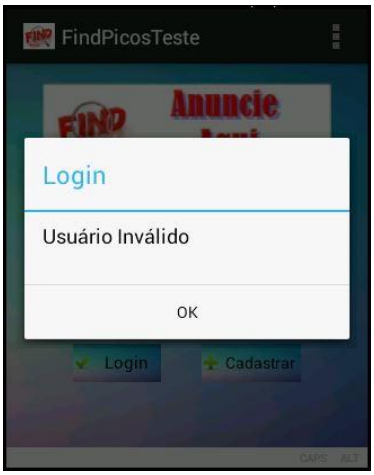



**Figura 4.3** – *Usuário Inválido*

Fonte: O Autor (2016)


**Figura 4.4** – *Buscando Estabelecimento pelo nome*

Fonte: O Autor (2016)

Após clicar no botão ‘buscar’, será mostrada as informações sobre o estabelecimento solicitado (Figura 4.5), tais como contato, endereço, que tipo de serviço, quantidade de gostei recebidos por outros usuários, tipo de pagamento que aceitam, horários de funcionamento, entre outras informações.


**Figura 4.5** – *Resultado de busca pelo nome*

Fonte: O Autor (2016)

Se por falta de conhecimento na cidade, o usuário optar por outro tipo de busca, é possível ir até a tela de busca avançada, onde o usuário pode optar em buscar por três diferentes campos. No teste realizado foi feita a busca levando em consideração o bairro que o estabelecimento está localizado (Figura 4.6).

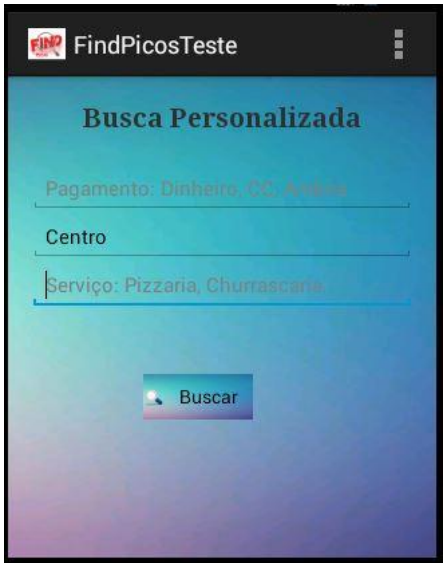



**Figura 4.6** – *Busca Avançada*

Fonte: O Autor (2016)

Esse tipo de busca é interessante, pelo fato de o usuário não querer sair do seu bairro para buscar algo que deseja. Ao clicar em 'buscar' será mostrada uma lista contendo os estabelecimentos que se encontram naquele bairro informado (Figura 4.7). Nesses resultados é possível clicar no nome de algum estabelecimento para que as informações sobre ele sejam mostradas.

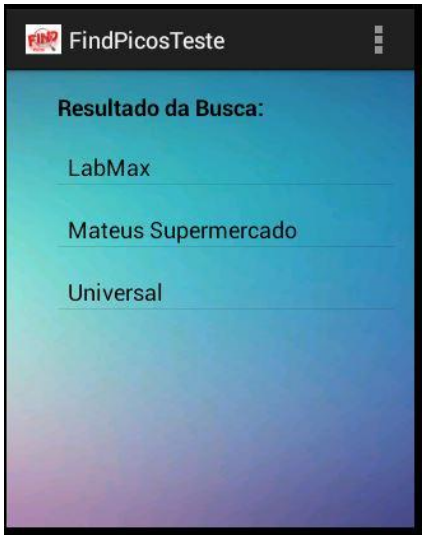



**Figura 4.7** – *Resultado Busca Avançada*

Fonte: O Autor (2016)

A partir da tela principal é possível o usuário verificar uma lista onde possui a relação dos estabelecimentos com melhores avaliações, essa avaliação é feita pela quantidade de gosteis

recebidos através do sistema *web*, a lista apresenta os estabelecimentos na ordem mais para menos gosteis (Figura 4.8).


**Figura 4.8** – *Lista de Melhores Avaliados*

Fonte: O Autor (2016)

Ao clicar em um estabelecimento, detalhes de informações também estarão disponíveis ao usuário exatamente como mostrado na Figura 4.3.

Através da tela de ‘alterar dados’ é possível ao usuário alterar sua senha de acesso e deletar o cadastro realizado. Para isso o usuário deve acessar a tela e ao ser iniciada a tela já indica o *username* do usuário, sendo que, basta digitar a nova senha e clicar em atualizar (Figura 4.9).

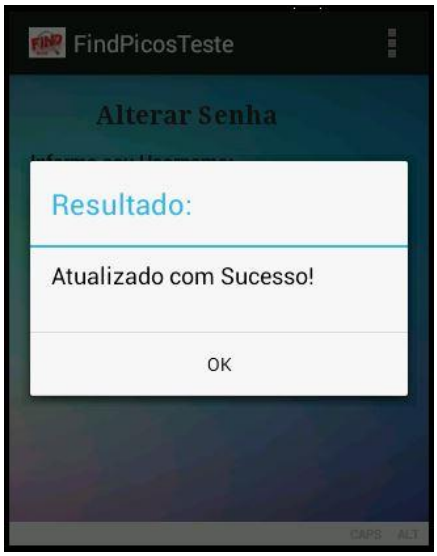



**Figura 4.9** – *Alterar senha*

Fonte: O Autor (2016)

Para realizar a exclusão do cadastro basta clicar em deletar, e uma mensagem de segurança será mostrada (Figura 4.10). Se o usuário confirmar a exclusão, o cadastro será excluído do servidor a aplicação o redirecionará para a tela de login, caso não confirmado a exclusão, nada ocorrerá.

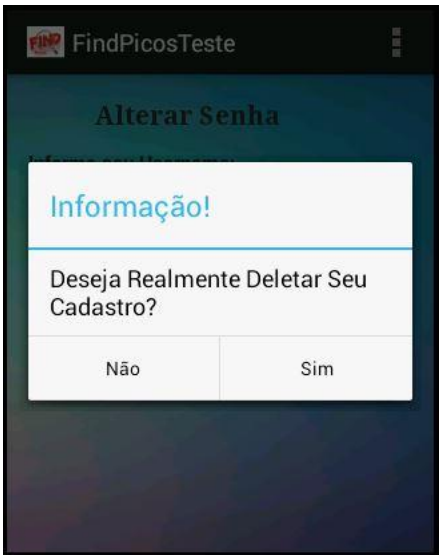



**Figura 4.10** – *Confirmação de Exclusão*

Fonte: O Autor (2016)

## 4.1 Avaliação dos Testes

O aplicativo FindPicos visa como objetivo, fazer com que o usuário pesquise e tenha detalhes sobre cada estabelecimento requerido, além de promover pequenas empresas/negócios através do sistema *web*.

Após a realização de alguns testes, pode-se observar que os objetivos buscados pelo aplicativo foram obtidos, pois após uma busca o usuário recebe seus resultados facilmente em seu aparelho, e as informações cadastradas no sistema pela empresa também foram mostradas para os usuários da aplicação de maneira prática e totalmente simples.

Para que o aplicativo pudesse ser avaliado de uma forma mais geral, foi elaborado um questionário contendo 6 perguntas relacionadas a usabilidade, interface e desempenho do aplicativo. O questionário encontra-se no Apêndice A deste trabalho. O questionário foi respondido por 10 pessoas de diferentes áreas de atuação. Na Figura 4.11 temos o gráfico de respostas relativos a utilidade que o aplicativo terá para a população da cidade.

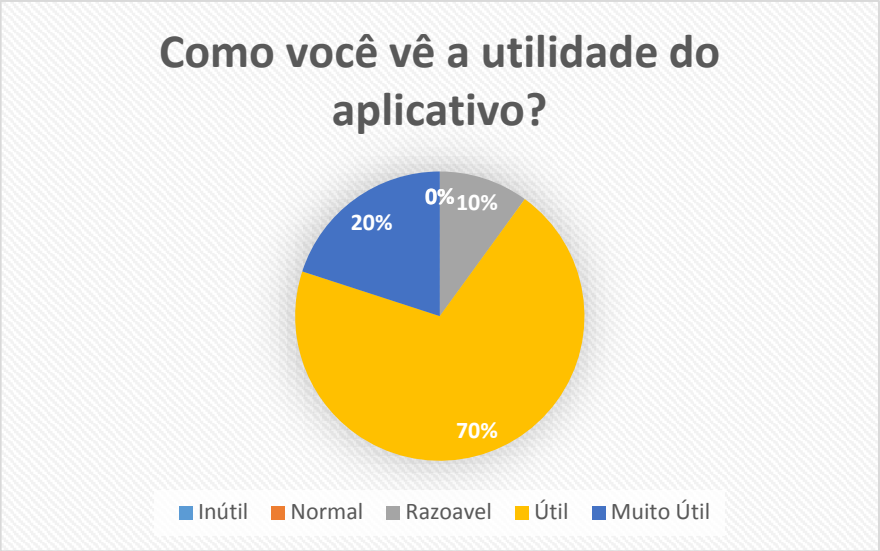



**Figura 4.11** – *Gráfico de Utilidade*

Fonte: O Autor (2016)

Na Figura 4.12, temos o gráfico de respostas relativos ao tempo de resposta para uma requisição realizada, um login, um cadastro, busca e etc.

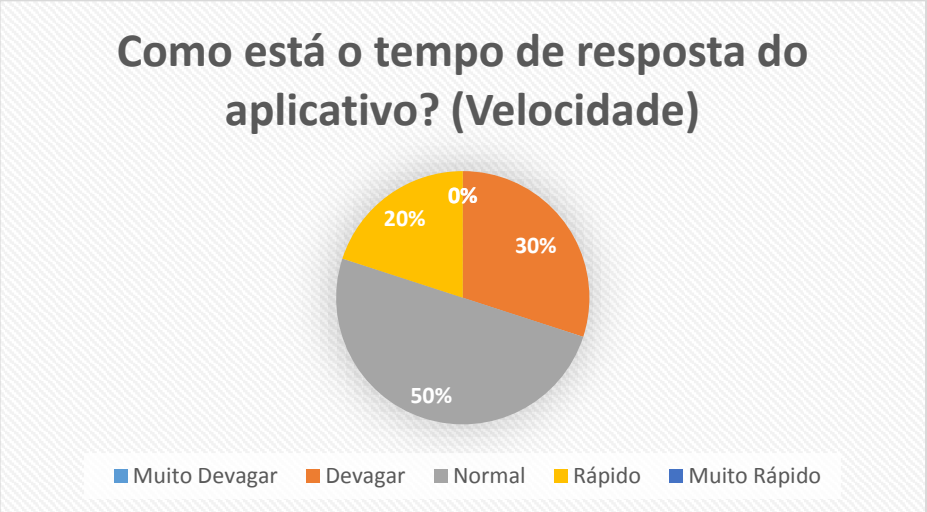



**Figura 4.12** – *Gráfico de Tempo de Resposta*

Fonte: O Autor (2016)

Na Figura 4.13, temos o gráfico de respostas relativos a usabilidade do aplicativo, se é fácil de usar, se é claro e objetivo e etc.

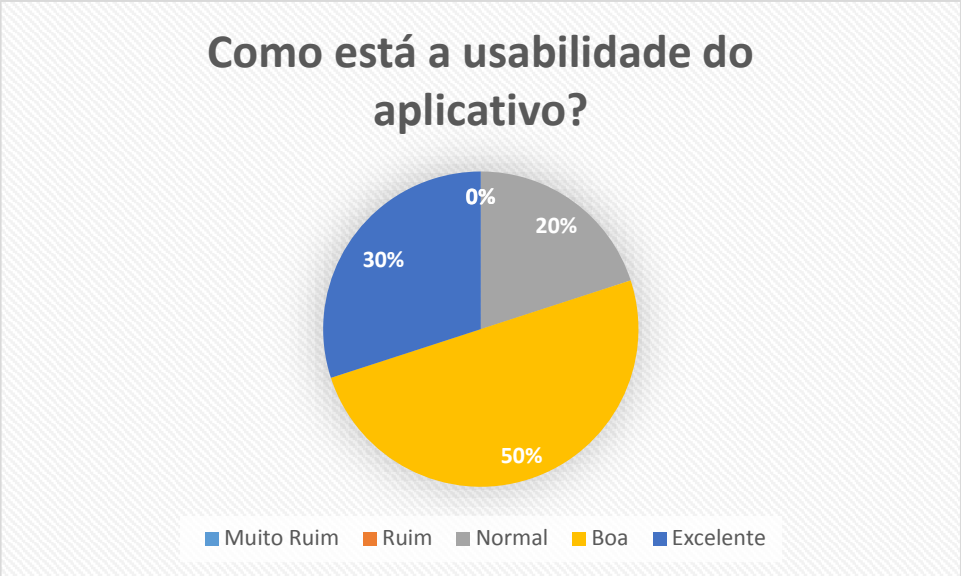



**Figura 4.13** – *Gráfico de Usabilidade*

Fonte: O Autor (2016)

Na Figura 4.14, temos o gráfico de respostas relativos a interface gráfica do aplicativo, o layout está apresentável?

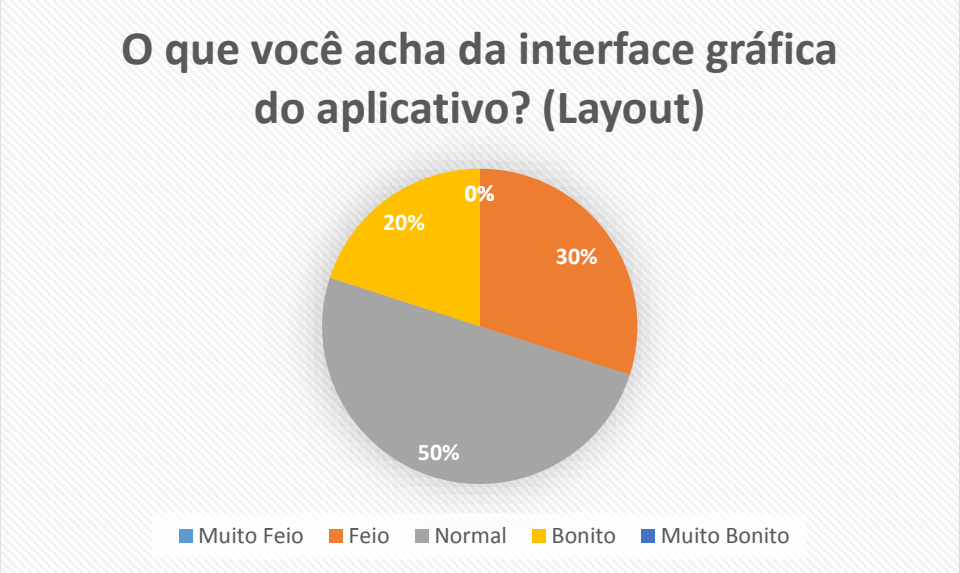



**Figura 4.14** – *Gráfico de Interface Gráfica*

Fonte: O Autor (2016)

Na Figura 4.15, temos o gráfico de respostas relativos aos tipos de buscas realizados pelo aplicativo, se são suficientes ou não.

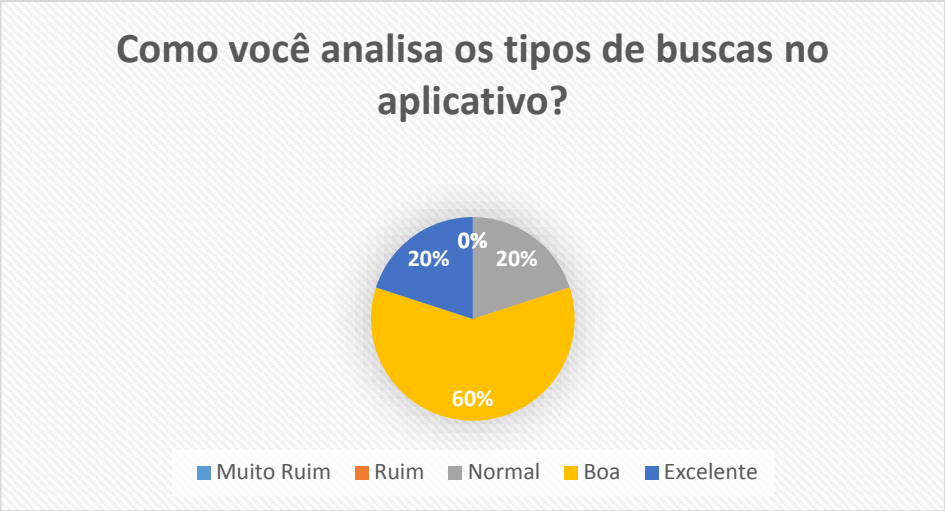



**Figura 4.15** – *Gráfico de Buscas Realizadas*

Fonte: O Autor (2016)

Na Figura 4.16, temos o gráfico de respostas relativos as informações disponíveis de cada estabelecimento cadastrado, se essas informações são insuficientes, se tem muitas informações desnecessárias e etc.

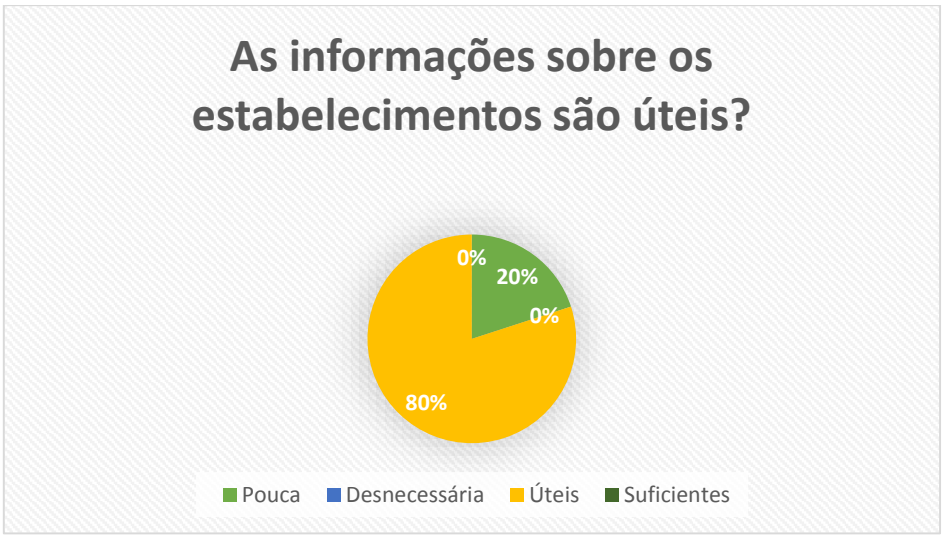



**Figura 4.16** – *Gráfico de Informações*

Fonte: O Autor (2016)

Ao final da realização dos testes e da análise do questionário, podemos observar que o aplicativo em termos gerais teve uma aceitação muito boa. Já em relação ao principal objetivo deste trabalho, podemos dizer que foi concluído, contando que são passados aos usuários as informações dos estabelecimentos com clareza e com um bom tempo de resposta.

# 5 CONCLUSÃO

O trabalho proposto foi desenvolvido com o propósito de apresentar um Protótipo de um Aplicativo Móvel para a busca de estabelecimentos na cidade de Picos-PI. Com o crescente uso de *smarthphones* no mundo, a quantidade de aplicativos na área *mobile* tem crescido cada vez mais, com isso, o programador moderno deve estar sempre atento a oportunidades e problemas existentes de modo que possa criar soluções.

O aplicativo FindPicos, surge como uma opção para usuários que necessitam conhecer o que um certo estabelecimento tem a oferecer, sem que tenha que ir ao local. Através do desenvolvimento desse projeto foi possível se aprofundar na prática na área de programação para dispositivos móveis.

Com relação a conexão com o servidor, o aplicativo também cumpriu com as propriedades pré-estabelecidas, conseguindo recuperar e enviar dados sem problemas, mas como trata-se de *web service,* surge a necessidade de o aplicativo está conectado a uma rede de internet, fator que pode vim a ser uma limitação da aplicação. Pode-se concluir que o aplicativo FindPicos, resolve o problema proposto, mas, por enquanto como trata-se de um protótipo, muitas funcionalidades ainda podem ser implementadas, porém seus objetivos por hora, são cumpridos.

Tendo em vista o objetivo principal deste trabalho, podemos afirmar que foi alcançado com êxito, um empreendedor pode livremente divulgar seu negócio e os usuários do aplicativo podem tranquilamente buscar esses estabelecimentos e visualizar detalhes de formar simples e rápida, sem precisar deslocar-se até o estabelecimento.

## 5.1 **Trabalhos Futuros**

Para trabalhos futuros sugere-se a implementação de uma interface gráfica mais apropriada para o usuário atual, além da implementação da tecnologia *LBS,* que se trata de um serviço baseado em localização, que funcione da a seguinte maneira: quando o usuário buscar por algum estabelecimento, o servidor retorne para a aplicação somente os estabelecimentos mais próximos ao usuário naquele exato momento.

**REFERÊNCIAS**

ABLESON, W. F. et al. **Android IN ACTION.** 3. ed. New York: Manning Publications Co., 2012.

ARIMA, K. **Qual será o seu próximo celular? Por que o Google, a Apple e a Palm têm cada vez mais chances de disputar essa resposta.** Info Exame, São Paulo, v. 1, 2009.

COCKBURN, A. **Escrevendo casos de uso eficazes**. São Paulo: Bookman, 2001.

DEITEL, P. et al. **Android para programadores:** uma abordagem baseada em aplicativos. Porto Alegre, Bookman, n. 1, p.481, 2013.

GOODMAN, D. **JavaScript Bible Gold**. Ed Gold. Hungry Minds. 2001.

GONÇALVES, E. **Dominando Eclipse**, Rio de Janeiro, Ciência moderna, 2000.

LECHETA, Ricardo R. **Google Android-3ª Edição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. 3**. ed. São Paulo: Novatec Editora, 2013.

LECHETA, Ricardo R. **Google Android – Aprenda a criar aplicações para dispositivos móveis com o Android SDK.** São Paulo: Novatec, 2010.

MEIER, Reto. **Professional Android 2 Application Development – Wrox professional guides.** 2. ed. USA: John Wiley & Sons, 2010.

NIEDERAUER, J. **PHP para quem conhece PHP**. São Paulo: Novatec, 2004.

FILHO, P. PÁDUA, W. **Engenharia de Software:** fundamentos, métodos e padrões. São Paulo: LTC Editora, 2000.

PRESSMAN, R.S. **Engenharia de Software**. 6ª Ed, McGraw-Hill, 2006.

RECKZIEGEL, M. **Entendendo os WebServices.** Disponível em <http://imasters.com.br/artigo/4245/web-services/entendendo-os-webservices/>. 23/06/2006. Acessado em 12 de Fevereiro de 2016.

Rocha, Cerli Antonio da – **Desenvolvendo web sites dinâmicos php, asp, jsp**- 2003 – 1ª Edição – Editora: Campus.

SANTOS, T. S., EDUARDO. C, **"Eclipse Tarde",** http://pesquompile.wikidot.com/eclipse- t, 2008.

SOBRAL, DANIELA B. C; MANGUEIRA. J. B. **Programação em Java**. Florianópolis, SC: Copyleft Pearson Education. 2008.

STRICKLAND, J. **Como funciona o Android (Google Phone)**. 16/09/2009. Disponível em <http://informatica.hsw.uol.com.br/google-phone2.htm>. Acessado em 12 de Fevereiro 2016.

# APÊNDICES

## APÊNDICE A – Questionário de avaliação do aplicativo

**1. Como você vê a utilidade do aplicativo?**
- inútil – 0 pessoas
- Normal – 0 pessoas
- Razoavel – 1 pessoas
- útil – 7 pessoas
- muito útil – 2 pessoas

**2. Como está o tempo de resposta do aplicativo? (Velocidade)**
- muito devagar - 0 pessoas
- devagar – 3 pessoas
- Normal – 5 pessoas
- Rápido – 2 pessoas
- muito rápido - 0 pessoas

**3. Como está a usabilidade do aplicativo?**
- muito ruim – 0 pessoas
- ruim – 0 pessoas
- Normal - 2 pessoas
- Boa – 5 pessoas
- excelente - 3 pessoas

**4. O que você acha da interface gráfica do aplicativo?** (Layout)
- muito feio – 0 pessoas
- feio – 3 pessoas
- Normal - 5 pessoas
- Bonito - 2 pessoas
- muito bonito - 0 pessoas

**5. Como você analisa os tipos de buscas no aplicativo?**
- muito ruim - 0 pessoas
- ruim - 0 pessoas
- Normal - 2 pessoas
- Boa - 6 pessoas
- muito boa - 2 pessoas

**6. As informações sobre os estabelecimentos são úteis?**
- pouca informação - 2 pessoas
- muita informação desnecessária - 0 pessoas
- muita informação útil - 8 pessoas
- as informações são suficientes - 0 pessoas

**TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA "JOSÉ ALBANO DE MACEDO"**

**Identificação do Tipo de Documento**

( ) Tese

( ) Dissertação

(X) Monografia

( ) Artigo

Eu, **André Luiz da Silva Lima**, autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de 02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação **FINDPICOS: PROTÓTIPO DE APLICAÇÃO MÓVEL PARA BUSCA E AVALIAÇÃO DE ESTABELECIMENTOS** de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título de divulgação da produção científica gerada pela Universidade.

Picos-PI, 4 de março de 2016.

André Luiz da S. Lima

Assinatura