

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

DANIEL DA SILVA

**DESENVOLVIMENTO DE UMA REDE SOCIAL NA ÁREA DE
REPAROS RESIDENCIAIS**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2016

DANIEL DA SILVA

**DESENVOLVIMENTO DE UMA REDE SOCIAL NA ÁREA DE
REPAROS RESIDENCIAIS**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. MSc. Luiz Rafael Schmitke

PONTA GROSSA

2016



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa
Diretoria de Graduação e Educação Profissional
Departamento Acadêmico de Informática
Bacharelado em Ciência da Computação



TERMO DE APROVAÇÃO

DESENVOLVIMENTO DE UMA REDE SOCIAL NA ÁREA DE REPAROS RESIDENCIAIS

por

DANIEL DA SILVA

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 31 de outubro de 2016 como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. MSc. Luiz Rafael Schmitke
Orientador

Prof. MSc. Vinicius Camargo Andrade
Membro titular

Prof. MSc. Rafael dos Passos Canteri
Membro titular

Prof. Dr. Augusto Foronda
Responsável pelo Trabalho de Conclusão
de Curso

Prof. Dr. Erikson Freitas de Moraes
Coordenador do curso

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

AGRADECIMENTOS

Agradeço primeiramente a minha mãe pelo apoio, confiança, dedicação e por acreditar em mim ao longo desses anos.

Agradeço também o Prof. Luiz Rafael Schmitke que me orientou com profissionalismo e tornou esse trabalho possível.

Meus professores e amigos da Universidade Tecnológica Federal do Paraná, Campus Ponta Grossa, que contribuíram com minha formação pessoal e profissional.

RESUMO

SILVA, Daniel da. **Desenvolvimento de uma Rede Social na Área de Reparos Residenciais**. 2016. 70 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2016.

Este Trabalho de Conclusão de Curso apresenta o desenvolvimento de uma rede social de avaliação na área de reparos residenciais. Ela permite que prestadores de serviços de uma determinada região possam cadastrar um anúncio de oferta de serviços nessa área. Usuários desse sistema podem avaliar anúncios com notas e comentários que posteriormente ficarão visíveis a outros usuários no perfil do anunciante. O gerenciamento do projeto foi feito por meio do método *Kanban* e o gerenciamento de desenvolvimento foi feito utilizando a metodologia de desenvolvimento ágil *Extreme Programming* (XP).

Palavras-chave: Redes Sociais. Rede Social de Avaliação. Reparos Residenciais. Site de Prestadores de Serviços.

ABSTRACT

SILVA, Daniel da. **Development of a Social Network in the Field of Facility Maintenance**. 2016. 70 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Federal Technology University - Paraná. Ponta Grossa, 2016.

This Undergraduate Thesis presents the development of an evaluation social network in the field of facility maintenance. It allows service providers of a particular region to offer services in this area. Users can rate advertisements with a grade and comments that will be available to other people on the advertiser's profile. The project management was done using the Kanban method and the software development using Extreme Programming (XP) which is an agile development technique.

Keywords: Social Networks. Evaluation Via Social Networking. Facility Maintenance. Sites for Freelance Services.

LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo de funcionamento de uma aplicação Web	16
Figura 2 - Modelo de comunicação entre cliente e servidor HTTP	18
Figura 3 - Exemplo de uma requisição HTTP.....	18
Figura 4 - Exemplo de parte de um cabeçalho de resposta HTTP	19
Figura 5 - Exemplo de um Kanban board.....	26
Figura 6 – Fórmula para ordenar avaliações utilizando o intervalo de confiança de Wilson	28
Figura 7 - Resultado de uma busca no site PetAnjo	30
Figura 8 - Exemplo de uma busca por eletricitas na região de São Paulo no site GetNinjas.....	30
Figura 9 - Quadro Kanban utilizado para gerenciar o projeto desse trabalho	33
Figura 10 - Parte das histórias e releases criadas no Pivotal Tracker.....	34
Figura 11 - Diagrama de Caso de Uso do Sistema	36
Figura 12 - Diagrama de Atividades para tela Login	37
Figura 13 - Diagrama de Atividades Buscar Anúncio	38
Figura 14 - Diagrama de Atividades Avaliar Anúncio	39
Figura 15 - Diagrama de Atividades Editar Perfil.....	40
Figura 16 - Diagrama de Atividades Editar Anúncio.....	41
Figura 17 - Testes da página com o resultado de uma de uma busca por anúncios.	42
Figura 18 - Modelo relacional do banco de dados utilizado na aplicação	44
Figura 19 - Tela Inicial.....	45
Figura 20 - Protótipo da tela Resultado Busca	46
Figura 21 - Tela Resultado Busca	46
Figura 22 - Protótipo da tela perfil do prestador de serviços	47
Figura 23 - Tela perfil do prestador de serviços	48
Figura 24 - Avaliações no perfil de um prestador de serviços	49
Figura 25 - Tela avaliação	49
Figura 26 - Opção de login na tela de Avaliação.....	50
Figura 27 - Fotos de um anúncio no perfil de um prestador de serviços	51
Figura 28 - Tela de Login	52
Figura 29 - Tela criar conta	52
Figura 30 - Protótipo da tela do menu dados básicos	53
Figura 31 - Tela do menu dados básicos	54
Figura 32 - Protótipo da tela do menu contato	54
Figura 33 - Tela do menu contato	55
Figura 34 - Protótipo da tela do menu endereço	56
Figura 35 - Tela do menu endereço	56
Figura 36 - Protótipo da tela do menu “Resumo”	57
Figura 37 - Tela do menu título e resumo	58

Figura 38 - Protótipo da tela do menu experiência	58
Figura 39 - Tela do menu experiência	59
Figura 40 - Protótipo da tela do menu escolaridade	59
Figura 41 - Tela do menu escolaridade	60
Figura 42 - Protótipo da tela de cadastro dos serviços prestados	61
Figura 43 - Tela do menu Serviços	61
Figura 44 - Protótipo da tela do menu fotos	62
Figura 45 - Tela do menu fotos	63
Figura 46 - Exemplo de um álbum de fotos de um anúncio	63

SUMÁRIO

1 INTRODUÇÃO	11
1.1 OBJETIVOS.....	12
1.1.1 Objetivo Geral.....	12
1.1.2 Objetivos Específicos.....	12
1.2 JUSTIFICATIVA.....	12
1.3 ORGANIZAÇÃO DO TRABALHO.....	13
2 REFERENCIAL TEÓRICO	14
2.1 APLICAÇÕES PARA WEB	14
2.1.1 Web 2.0	16
2.1.2 Servidor Web	17
2.1.3 Cliente Web	19
2.2 REDES SOCIAIS	20
2.3 LINGUAGEM UNIFICADA DE MODELAGEM - UML	21
2.4 EXTREME PROGRAMMING - XP	22
2.4.1 Etapas do XP	23
2.5 KANBAN	24
2.5.1 Funcionamento do Kanban.....	25
2.6 FRAMEWORK	26
2.7 MÉTODOS DE ORDENAÇÕES DE AVALIAÇÕES.....	27
2.8 SISTEMAS SIMILARES.....	29
3 DESENVOLVIMENTO.....	32
3.1 GERENCIAMENTO DO PROJETO	32
3.2 GERENCIAMENTO DE DESENVOLVIMENTO.....	34
3.3 MODELAGEM DO SISTEMA.....	35
3.3.1 Diagrama de Caso de Uso.....	35
3.3.2 Diagrama de Atividades.....	37
3.4 TESTES DO SISTEMA.....	41
3.5 MODELAGEM DO BANCO DE DADOS	42
4 SISTEMA.....	45
4.1 PÁGINA INICIAL.....	45
4.2 TELA RESULTADO BUSCA.....	46
4.3 TELA PERFIL DO PRESTADOR DE SERVIÇOS	47
4.4 TELA AVALIAÇÃO.....	49
4.5 TELA FOTOS.....	50
4.6 TELA LOGIN	51
4.7 TELA CRIAR CONTA	52
4.8 ABA PERFIL	53
4.9 ABA ANÚNCIO	57

5 CONCLUSÃO	64
5.1 TRABALHOS FUTUROS.....	64
REFERÊNCIAS	66

1 INTRODUÇÃO

Com o crescimento da utilização da Internet móvel e fixa, a busca por produtos e serviços online tem se destacado nos últimos anos. Serviços que recentemente eram oferecidos somente por empresas físicas, têm sido ofertados por meio de *websites* e aplicativos móveis. Por outro lado, grandes empresas vêm surgindo a partir dessa nova forma de comércio, como é o caso da Amazon e eBay (FORBES, 2016). Por meio da Internet é possível buscar, comparar e comprar produtos ou serviços de uma forma rápida, prática e eficiente, podendo assim economizar tempo e dinheiro (AKST, 2016).

Segundo um levantamento feito pelo IBGE (2015), o acesso à Internet no Brasil aumentou 21,6% de 2008 para 2009, 14,8% de 2009 para 2011 e 6,9% de 2011 para 2012. Outra pesquisa feita pelo Centro de Estudos sobre as Tecnologias da Informação e da Comunicação (CETIC, 2015), aponta que em 2014 metade da população brasileira possuía acesso à Internet considerando também o uso de dispositivos móveis.

Com o crescimento do acesso à Internet no Brasil, as redes sociais estão cada vez mais populares. De acordo com o Facebook, 45% da população brasileira acessava mensalmente a rede social no segundo trimestre de 2014. Isso corresponde a 95 milhões de pessoas usando essa plataforma, fazendo dela a rede social mais usada no Brasil (FACEBOOK PARA EMPRESAS, 2015).

Empresas, profissionais liberais e trabalhadores informais têm utilizado as redes sociais para oferecer serviços nas mais diversas áreas. Entre elas, destacam-se os serviços de reparos residenciais.

O Brasil possui aproximadamente 70 milhões de residências e 18 milhões de prédios comerciais e industriais. Calcula-se que 80% desse total necessitem de manutenção pelo menos uma vez por ano. Entretanto, encontrar profissionais dessa área é difícil. Grandes construtoras têm atraído cada vez mais esse tipo de trabalhador, fazendo com que esse mercado necessite cada vez mais de mão de obra (SEBRAE, 2012).

Percebendo o potencial dessa área, algumas empresas especializadas em manutenções residenciais têm surgido e conquistado mercado. Esse crescimento está ligado principalmente a falta de tempo dos consumidores. A vida agitada nos grandes

centros urbanos faz com que as pessoas optem por contratar essas empresas para realizar tais serviços (SEBRAE, 2012).

Neste contexto, este trabalho apresenta o desenvolvimento de uma rede social, que permitirá aos seus usuários ofertar e/ou buscar serviços na área de pequenos reparos residenciais (alvenaria, elétrica, hidráulica, jardinagem e pintura) de forma rápida, fácil e segura.

1.1 OBJETIVOS

O objetivo geral e específicos estão definidos a seguir.

1.1.1 Objetivo Geral

Desenvolver uma rede social que permitirá aos seus usuários ofertar e/ou buscar serviços na área de pequenos reparos residenciais, tais como: alvenaria, elétrica, hidráulica, jardinagem e pintura.

1.1.2 Objetivos Específicos

Os objetivos específicos estão listados a seguir:

- Fazer o estudo de sistemas similares;
- Fazer o levantamento dos requisitos do sistema;
- Fazer a modelagem do sistema;
- Implementar o sistema;
- Realizar testes no sistema.

1.2 JUSTIFICATIVA

Encontrar pessoas para realizar pequenas manutenções residenciais nas áreas de alvenaria, elétrica, hidráulica, jardinagem e pintura tem sido difícil devido à falta de profissionais disponíveis no mercado de trabalho. Cada vez mais grandes

construtoras têm atraído este perfil de profissional para grandes obras de construção civil, fazendo com que falte mão de obra para pequenos reparos.

Anúncios virtuais de prestação de serviços nessa área são divulgados principalmente nas redes sociais, sites de compras e vendas, e sites de classificados. Porém, encontrá-los é uma tarefa difícil, pois tais sistemas não foram feitos especialmente para essa área. Outro problema, é que poucas informações são divulgadas a respeito da pessoa e seus serviços devido as limitações impostas pelo sistema, fazendo com que o cliente questione a confiabilidade do anúncio. Tais sistemas apresentam também limitações geográficas, isto é, não é possível verificar qual é o prestador de serviço mais próximo ao cliente.

Um sistema específico para essa área ajudaria na inserção de novos trabalhadores, fazendo com que novos postos de trabalho possam ser criados. Além disso, os que já atuam nesse mercado poderiam utilizá-lo para facilitar a busca por trabalho. Por outro lado, a população em geral se beneficiaria, pois a busca e a contratação de um desses profissionais se tornaria mais eficiente.

1.3 ORGANIZAÇÃO DO TRABALHO

Este capítulo apresenta as considerações iniciais, objetivos e justificativa deste trabalho.

No capítulo 2 está o referencial teórico utilizado como base para a implementação do projeto proposto.

No capítulo 3 são apresentados os métodos, ferramentas e *frameworks* utilizados nesse trabalho. O capítulo 4 mostra os resultados obtidos, tais como: estudo de sistemas similares, gerenciamento do projeto, gerenciamento do desenvolvimento, modelagem do sistema, modelagem do banco de dados, testes utilizados, protótipos e descrição das telas criadas. O capítulo 5 contém a conclusão e trabalhos futuros. E por fim são apresentadas as referências bibliográficas utilizadas.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico utilizado para o desenvolvimento desse trabalho. A seção 2.1 mostra o conceito de Web e seu funcionamento. A seção 2.2 apresenta o conceito de redes sociais. A seção 2.3 discute sobre Linguagem Unificada de Modelagem. A seção 2.4 discorre sobre a metodologia *Extreme Programming*. A seção 2.5 apresenta o método Kanban. A seção 2.6 discute sobre *framework* e a arquitetura *Model-View-Controller*. E por fim, na seção 2.7, são apresentados métodos de ordenações de avaliações.

2.1 APLICAÇÕES PARA WEB

Esta categoria de software abrange uma vasta gama de aplicações. Em sua forma mais simples elas podem ser formadas por um conjunto de arquivos de hipertexto interconectados, apresentando informações por meio de texto e informações gráficas limitadas conhecida como páginas Web (PRESSMAN, 2011).

Cada página Web é classificada como *Hypermedia* e é acessível por meio da Internet. O sufixo *media* indica que um documento não possui apenas texto, mas imagens e outros elementos. E o prefixo *hyper* indica que a página contém *links* que referenciam outras páginas (COMER, 2000).

Diferentemente de aplicações *desktop* tradicionais, que são executadas pelo sistema operacional do cliente, aplicações Web são geralmente acessadas por navegadores (*browsers*) por meio de um identificador único conhecido como *Uniform Resource Locator* (URL).

Em termos de plataforma, as aplicações Web apresentam certos benefícios em relação a aplicações *desktop*. Um deles está ligado a consistência da interface de usuário nos diferentes sistemas operacionais, devido a aparência geralmente depender de um navegador. Portanto, desenvolvedores não precisam desenvolver para múltiplas plataformas. Outro benefício é que os usuários podem facilmente acessá-las em diversos dispositivos, pois elas armazenam e processam dados remotamente.

Apesar do seu desenvolvimento ser considerado uma disciplina nova em relação a aplicações *desktop*, elas têm evoluído significativamente em um curto

espaço de tempo. Atualmente elas são sistemas complexos que vão muito além de simples páginas Web, além de oferecerem uma grande variedade de conteúdo e funcionalidade para seus usuários. Elas se tornaram fundamentais para diversos setores da economia como indústrias, bancos, *e-commerce* e governo (BRANDON, 2008).

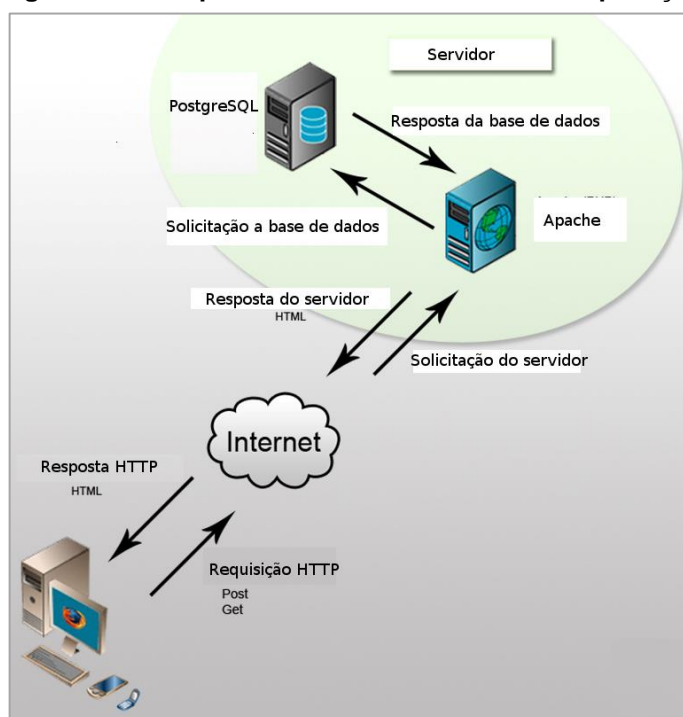
O desenvolvimento de aplicações Web ocorre de forma diferente se comparado com o desenvolvimento de software tradicional. Sua estrutura e funcionalidade evolui ao longo do tempo principalmente depois de sua implantação. Em muitos casos fica difícil ou até mesmo impossível especificar os seus requisitos no início do processo de desenvolvimento (BRANDON, 2008).

Pressman (2011) afirma que aplicações Web apresentam algumas características que podem ser encontradas na maioria das aplicações dessa categoria. Algumas delas estão listadas abaixo:

- Simultaneidade: vários usuários podem acessar ao mesmo tempo a aplicação;
- Carga não previsível: a quantidade de acessos varia com o tempo;
- Disponibilidade: a página deve estar disponível para acesso o tempo todo;
- Uso intensivo de redes: as aplicações Web são acessadas por meio de uma rede;
- Estética: a página deve ter uma boa aparência para causar uma boa impressão ao usuário.

Em uma aplicação Web simples o cliente faz uma requisição ao servidor por meio de um protocolo, depois o servidor acessa uma base de dados, se necessário, e então o servidor responde a solicitação do cliente com algum tipo de dado.

Figura 1 - Exemplo de funcionamento de uma aplicação Web



Fonte: Adaptado de FRANÇA (2010)

A Figura 1 mostra um exemplo de aplicação Web. É possível notar que o cliente, representado pelo monitor no canto inferior esquerdo, faz uma solicitação utilizando o protocolo HTTP a um servidor *Apache*. O servidor acessa a base de dados e retorna uma resposta HTTP em forma de texto HTML ao cliente.

2.1.1 Web 2.0

Com o surgimento da Web 2.0, as aplicações Web têm evoluído e se transformado em sistemas complexos que fornecem recursos e funções para o usuário final. Muitas delas são capazes de interagir com outros sistemas e base de dados (PRESSMAN, 2011).

Devido às novas tecnologias disponibilizadas, foi possível a criação de novos recursos, tais como: redes sociais, jogos online, mapas interativos e lojas virtuais. Desta forma, a Internet evoluiu de apenas páginas informativas para páginas em que o usuário contribui e interage de alguma forma, fazendo com que ele tenha um papel ativo, sendo capaz de produzir conteúdo.

Essa nova versão de Web vem crescendo devido a alguns fatores, tais como: evolução de *hardware*, popularização da Internet e o crescimento de Softwares de

Código Aberto (*Open Source*). Além disso, a utilização do AJAX (*Asynchronous Javascript and XML*), utilizado para fazer requisições a um servidor sem a necessidade de recarregar a página toda, têm impulsionado esse crescimento (DEITEL; DEITEL, 2008). Por meio do AJAX, usuários podem interagir com partes de uma página sem afetar o comportamento do resto dela, permitindo assim, que sejam mais ricas em conteúdo.

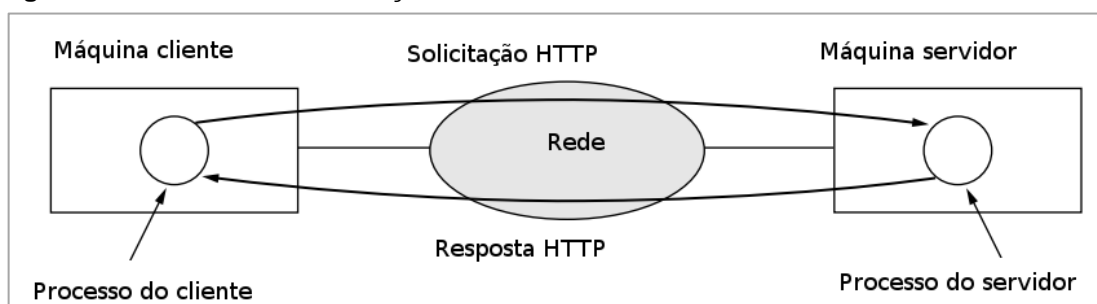
2.1.2 Servidor Web

O termo servidor refere-se a qualquer programa que oferece um serviço que pode ser acessado usando uma rede. Um servidor aceita uma solicitação vinda da rede, processa ela, e retorna o resultado para quem solicitou (COMER, 2000).

Tanenbaum e Steen (2007) define servidor como sendo uma “máquina com alto poder de processamento capaz de processar e responder mensagens recebidas de máquinas clientes”. Por meio de servidores Web como o *Apache Web Server* e *Internet Information Services* instalados na máquina servidor, eles são capazes de aceitar conexões e responder solicitações utilizando o protocolo de comunicação HTTP (*HyperText Transfer Protocol*). Existem diversas categorias de servidores, entre eles, destaca-se os de arquivo, e-mail, proxy e HTTP.

Em um servidor HTTP há dois processos (programas em execução) envolvidos: um na máquina cliente e outro no servidor. A comunicação entre eles é feita por meio de uma rede em que o processo do cliente envia uma mensagem para o processo do servidor. Após o servidor receber a mensagem, ele faz o que o cliente solicitou e envia de volta uma resposta caso seja necessário. A Figura 2 ilustra como funciona esse tipo de servidor (TABENBAUM; WETHERALL, 2010).

Figura 2 - Modelo de comunicação entre cliente e servidor HTTP



Fonte: Adaptado de TANENBAUM; WETHERALL (2010)

Para realizar a comunicação cliente-servidor, o protocolo HTTP utiliza os métodos OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE e CONNECT (W3C, 2016). Os mais utilizados em aplicações Web são detalhados a seguir:

- GET: geralmente utilizado para fazer a leitura de um dado no servidor. Ele retorna ao cliente a informação em formato XML, JSON ou HTML.
- POST: utilizado para criar e inserir novos dados no servidor. Ele retorna 201 em caso de sucesso da requisição.
- PUT: utilizado para alterar ou criar dados no servidor.
- DELETE: apaga um objeto do servidor utilizando uma URL.

Depois de receber e interpretar uma solicitação, o servidor responde ao cliente com uma mensagem de resposta HTTP com a versão do protocolo, código de status e sua descrição, cabeçalho e uma mensagem opcional (FIELDING, et al., 2014). A Figura 3 mostra um exemplo de uma requisição HTTP para buscar o arquivo “search.js” em um servidor local.

Figura 3 - Exemplo de uma requisição HTTP



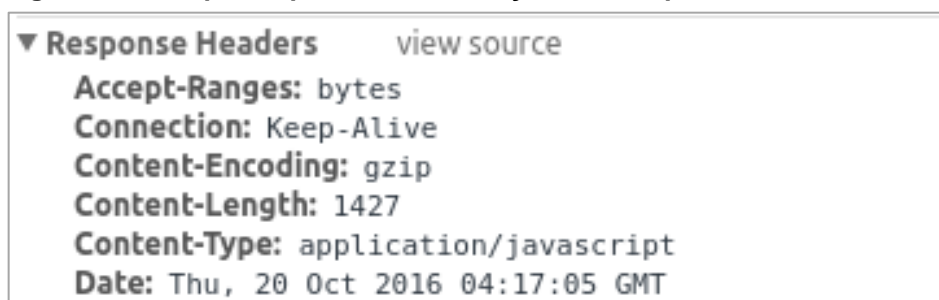
Fonte: Autoria própria

Na Figura 3, o campo “*Request URL*” é o endereço do arquivo “search.js” no servidor. O campo “*Request Method*” é o método do protocolo HTTP. O campo “*Status*

Code” é um código retornado pelo servidor usado para indicar erro ou sucesso em uma requisição. O código 200 indica que não houve erro. E o campo “*Remote Address*” corresponde ao endereço do servidor.

A Figura 4 apresenta parte do cabeçalho de resposta da requisição da Figura 3.

Figura 4 - Exemplo de parte de um cabeçalho de resposta HTTP

A screenshot of a web browser's developer tools showing the 'Response Headers' section. The headers listed are: Accept-Ranges: bytes, Connection: Keep-Alive, Content-Encoding: gzip, Content-Length: 1427, Content-Type: application/javascript, and Date: Thu, 20 Oct 2016 04:17:05 GMT. There is a 'view source' link next to the 'Response Headers' title.

```
▼ Response Headers view source
Accept-Ranges: bytes
Connection: Keep-Alive
Content-Encoding: gzip
Content-Length: 1427
Content-Type: application/javascript
Date: Thu, 20 Oct 2016 04:17:05 GMT
```

Fonte: A autoria própria

O campo “*Accept-Ranges*” da Figura 4 indica o tipo de arranjo de conteúdo parcial que o servidor suporta. Os valores permitidos são “*bytes*” ou “*none*” (MDN, 2016). O campo “*Connection*” permite que a conexão com servidor seja fechada após uma solicitação. Há dois valores utilizados: “*Keep-Alive*” e “*Close*”. O primeiro mantém a conexão aberta com o servidor, e o segundo fechada (MDN, 2016).

2.1.3 Cliente Web

Um cliente é uma máquina que fornece recursos para que usuários possam interagir com servidores remotos. Há basicamente duas formas de interação, a primeira, e mais utilizada, o cliente possui uma parte separada em que ele faz requisições utilizando uma rede. Na segunda, o cliente possui acesso direto aos serviços do servidor (TANENBAUM, 2010).

O papel de cliente Web é feito pelo Navegador. Ele é responsável por buscar uma página solicitada, interpretar o conteúdo e mostra-la na tela para o usuário. Geralmente o conteúdo é formado por uma mistura de texto, imagens, vídeos e *hyperlinks*.

Portanto, o cliente Web é responsável por apresentar ao usuário uma interface em que ele possa visualizar e/ou gerenciar algum tipo de dado disponibilizado por um

servidor remoto. Por meio do protocolo de comunicação, geralmente o HTTP, o usuário solicita e envia informações para um servidor.

2.2 REDES SOCIAIS

Ryan (2011) afirma que o ser humano, mesmo sem perceber, faz parte de várias redes sociais no seu dia a dia. Grupos de amigos, escolas, cidades e estados são exemplos típicos de redes sociais. Uma das características principais dessa forma de interação é o compartilhamento de interesses em comum entre as pessoas. Com o desenvolvimento de tecnologias como rádio, telefone e Internet, foi possível manter essas conexões mesmo com dificuldades, como a distância entre as pessoas.

Com o crescimento da Internet, o termo rede social passou a ser utilizado para classificar aplicações Web como Facebook e MySpace em que usuários são capazes de interagir uns com os outros em tempo real, independente da distância entre eles. Tais aplicações podem permitir, por exemplo, o compartilhamento de conteúdo em forma de texto, imagens e vídeos entre os seus participantes (RYAN, 2011).

Uma das principais características de uma rede social é a possibilidade do usuário criar um perfil com informações pessoais. Seu objetivo é fazer com que outras pessoas da rede se interessem por ele. O perfil funciona como um currículo onde outros usuários podem ver suas atividades de acordo com as configurações de privacidade definidas pelo seu dono.

Ter um bom perfil é essencial para que um usuário possa se destacar na rede. As pessoas tendem a gastar apenas alguns segundos vendo o perfil de alguém e, com base nessas informações, julgam se a pessoa é, por exemplo, confiável ou competente. Um bom perfil deve atender os seguintes requisitos: ter um bom nome de usuário, ser otimizado para cinco segundos, ter uma foto de perfil que mostre quem o usuário é, e possuir o máximo possível de informações a respeito do que é importante para o usuário (KAWASAKI; FITZPATRICK, 2014).

Com a popularização das redes sociais, empresas as têm utilizado como uma forma de marketing para os seus produtos, serviços e marcas, tendo em vista a possibilidade de atingir um grande público de forma rápida e, muitas vezes, mais barata que os meios tradicionais de divulgação como rádio, televisão e mídia impressa.

Atualmente há dezenas de sites de redes sociais com diferentes propósitos e abrangências geográficas. Entre os mais conhecidos, destacam-se o Facebook, Twitter, LinkedIn e Youtube.

As redes sociais podem ser divididas nas seguintes categorias de aplicações: conexões sociais, compartilhamento de conhecimento colaborativo, conteúdo gerado por usuários, mundos virtuais, avaliação, agrupamento e organização de conteúdo, e tangíveis (HARTLEY, 2010).

Uma das categorias, menos popular que as conexões sociais que incluem o Facebook, Twitter e LinkedIn, é as redes sociais de avaliações. Ela inclui sites como Amazon, Yelp e Foursquare. Entre os vários métodos de avaliação usados por esses sites, destaca-se o uso de comentários positivos ou negativos, e uma escala que representa uma nota dada a um produto, serviço ou estabelecimento comercial. Isso faz com que produtos e serviços sejam oferecidos com uma melhor qualidade para que tenha uma boa avaliação do consumidor, pois as avaliações podem influenciar na tomada de decisão do cliente (HARTLEY, 2010).

2.3 LINGUAGEM UNIFICADA DE MODELAGEM - UML

Devido à complexidade envolvida no desenvolvimento de software, é necessário compreender o problema na fase inicial de projeto. Detalhes ignorados podem causar impactos negativos nos prazos e custos, fazendo com que o produto seja entregue atrasado, causando transtornos tanto para o cliente quanto para os desenvolvedores. A modelagem de software surgiu da necessidade da compreensão total de um problema e pode ser feita utilizando uma linguagem específica para modelagem (LIMA, 2009).

Uma das formas de modelar um software é utilizando a Linguagem Unificada de Modelagem (do Inglês, *Unified Modeling Language* – UML). Ela é uma linguagem visual de modelagem usada para visualizar, construir, e documentar um software. Ela pode ser usada em todas as etapas do ciclo de vida, desde o levantamento de requisitos até a manutenção (RUMBAUGH; JACOBSON; BOOCH, 2004).

Essa linguagem passou a ser amplamente utilizada por empresas importantes de software após a finalização da sua criação em 1996. Ela é uma fusão de outras notações utilizadas na época, como a *Object Management Technology* (OMT) e

Notação *Booch*, criadas por James Raumbaugh e Grady Booch, respectivamente (BOGGS; BOGGS, 2002).

Na modelagem utilizando a UML, não há um padrão de modelagem a ser seguido, ela é adaptada para atender as necessidades de cada problema. Portanto, a escolha dos elementos da linguagem fica a critério de quem é responsável pela modelagem do sistema. Ela é usada geralmente no processo de desenvolvimento de software orientado a objetos (RUMBAUGH; JACOBSON; BOOCH, 2004).

A UML não é uma linguagem de programação. Entretanto, ela pode ser usada para tal propósito. Por meio de ferramentas que suportam essa linguagem, modelos UML podem ser transformados em códigos de uma linguagem de programação específica. Porém, a UML não oferece os mesmos recursos de uma linguagem de programação, isso faz com que o desenvolvimento seja mais difícil e demorado (LIMA, 2009).

Para representar e obter múltiplas visões de um sistema, a UML permite a criação de diversos diagramas, dentre eles: diagrama de classes, diagrama de atividades, diagrama de caso de uso, diagrama de implantação e diagrama de componentes. Tais diagramas podem representar o sistema como um todo ou partes específicas dele. Essa abordagem permite que falhas sejam detectadas no período inicial do projeto, diminuindo assim a possibilidade de falhas futuras (LIMA, 2009).

2.4 EXTREME PROGRAMMING - XP

No desenvolvimento de software existe uma categoria de desenvolvimento conhecida como Processos Ágeis de Desenvolvimento, que engloba, entre outros métodos, o *Extreme Programming* (XP).

O XP é um método ágil de desenvolvimento de software indicado para o desenvolvimento de sistemas orientados a objetos em que os requisitos não são claros e mudam com frequência. Uma das suas principais características é a utilização do desenvolvimento incremental em oposição ao processo tradicional em forma de cascata (SOMMERVILLE, 2007).

O cliente é parte fundamental do desenvolvimento utilizando o XP, pois ele é responsável por verificar se o sistema atende os requisitos solicitados. Ele fornece

feedbacks durante o processo de desenvolvimento do sistema, e por meio deles, o software é refinado até que se atinja os seus objetivos (TELES, 2006).

O XP parte do pressuposto de que normalmente o cliente não é capaz de fornecer todos os requisitos que um sistema deve ter. Isso acontece devido à grande quantidade de detalhes que precisam ser fornecidos, ou porque o cliente simplesmente não conhece os requisitos. Com base na utilização de versões parciais do sistema (*releases* curtas), o cliente pode avaliar, sugerir ou criar requisitos. Esse processo é repetido até a versão final do sistema, fazendo com que ele seja validado de forma contínua pelo cliente e não somente no final do desenvolvimento como acontece no modelo tradicional de desenvolvimento (TELES, 2006).

O XP apresenta quatro valores fundamentais (BECK, 2004). Eles estão descritos a seguir:

- *Feedback*: gerado pelo cliente quando ele utiliza uma *release* do sistema e é usado para direcionar o desenvolvimento da próxima iteração;
- *Comunicação*: é necessário que a comunicação seja eficiente para a transmissão dos *feedbacks* entre o cliente e a equipe de desenvolvimento;
- *Simplicidade*: ela está relacionada a atender as necessidades imediatas do cliente sem levar em consideração os impactos futuros gerados por uma modificação no código, reduzindo assim o tempo de desenvolvimento de uma determinada funcionalidade;
- *Coragem*: é preciso ter coragem para assumir o risco de inserir erros no sistema durante a implementação de novas funcionalidades ou refatorar o código.

2.4.1 Etapas do XP

O XP pode ser dividido em algumas etapas, como planejamento, teste e codificação. Na primeira é feito o levantamento de requisitos utilizando cartões escritos pelo cliente, denominadas histórias de usuários. Neles estão contidas as características e os resultados de uma funcionalidade do sistema. Para cada história é associado um custo de desenvolvimento que pode variar de dias a semanas (PRESSMAN, 2011).

As histórias são analisadas e divididas em tarefas (divisão em novos cartões), se necessário, e distribuídas entre os desenvolvedores para serem implementadas. Nessa fase é definido como será a entrega do software. Ela é dividida em partes conhecidas como *releases*, que por sua vez são subdivididas em iterações. Uma das vantagens desse método, é que o cliente pode começar a utilizar imediatamente o sistema após a entrega de cada *release* (TELES, 2006).

A próxima etapa consiste na criação de testes automatizados de unidade e de aceitação antes da implementação das histórias. Os testes de unidade servem para testar classes específicas do sistema. Já o teste de aceitação foca nas funcionalidades do sistema, ou seja, ele testa um conjunto de classes responsáveis por uma funcionalidade (SOMMERVILLE, 2007).

Após o desenvolvimento dos testes, as histórias são implementadas e partes do código que precisam de modificações são refatorados. A cada modificação, é necessário executar os testes criados anteriormente para evitar a inserção de erros no sistema.

O XP utiliza o conceito de programação em par (*pair programming*), onde dois desenvolvedores trabalham juntos utilizando uma única máquina. Enquanto um digita, o outro fica responsável por analisar o código, fazendo com que a probabilidade de erros diminua consideravelmente (BECK, 2004).

2.5 KANBAN

No desenvolvimento de software, Kanban é um método para o melhoramento de processos aplicados no desenvolvimento ágil. Seu foco difere das metodologias SCRUM (SCHWABER; SUTHERLAND, 2013) e XP (seção 2.4). O primeiro possui como foco principal o gerenciamento de projetos, já o segundo o desenvolvimento de software (ANDERSON, 2010).

De acordo com Moura (1999), um sistema de produção com várias etapas pode ser dividido em dois tipos: sistema de empurrar e sistema de puxar. O sistema Kanban utiliza o sistema de puxar, onde um item entra somente quando um outro sai de determinado estágio. Essa ideia foi desenvolvida pela empresa Toyota, no Japão, inspirada no processo de reposição de produtos em uma prateleira de supermercado.

Devido à falta de espaço, um novo produto somente é adicionado a prateleira quando há espaço para ele, ou seja, um outro produto deve sair da prateleira.

Os principais valores do Kanban estão listados a seguir (STELLMAN; GREENE, 2015):

- a) Eliminar desperdício;
- b) Amplificar a aprendizagem por meio de *feedbacks*;
- c) Tomar decisões o mais tarde possível;
- d) Entregar o mais rápido possível;
- e) Ver o todo do projeto.

O Kanban é usado para limitar os trabalhos em progresso da equipe de desenvolvimento. Isso faz com a equipe seja desafiada a resolver problemas para permitir um ritmo constante do fluxo de trabalho. Limitar esse fluxo pode permitir que se atinja maior qualidade nas soluções de problemas e melhorar o desempenho da equipe (ANDERSON, 2010).

Uma das vantagens do Kanban é que ele não necessita de um número mínimo de integrantes no time de desenvolvimento para a sua utilização. No SCRUM, por exemplo, é recomendado ter pelo menos três pessoas na equipe, caso contrário a aplicação dele pode ser comprometida (SCHWABER; SUTHERLAND, 2013).

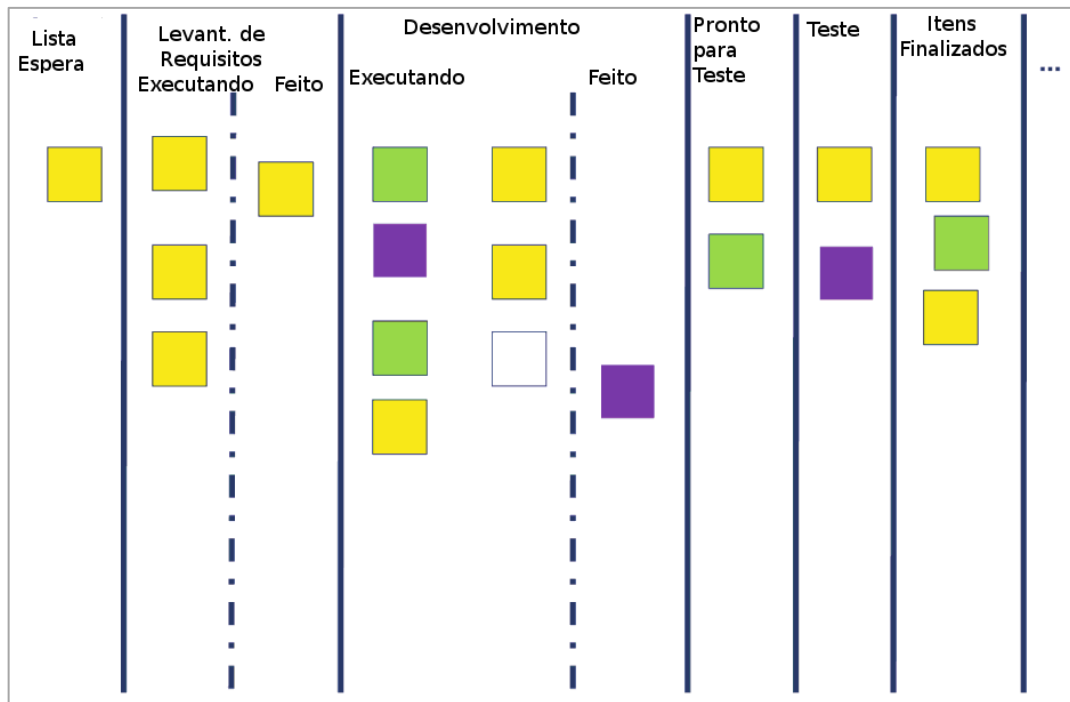
2.5.1 Funcionamento do Kanban

Para utilizar o Kanban no desenvolvimento ágil, é necessário criar um quadro, conhecido como *Kanban board*, e subdividi-lo em colunas em que serão colocados cartões com os trabalhos a serem desenvolvidos. Esse quadro pode ser criado utilizando papel ou algum software como o Trello (TRELLO, 2016), por exemplo. Quando um trabalho é feito, o cartão que corresponde a ele vai para a próxima coluna (etapa) a direita, e um novo cartão pode ser adicionado no seu lugar (HAMMARBERG; SUNDÉN, 2014).

A Figura 5 mostra um exemplo de um *Kanban board*. A primeira coluna (Lista Espera) representa os cartões que serão adicionados ao fluxo do *Kanban board*. A segunda coluna (Levant. de Requisitos) mostra os itens da fase de levantamento de requisitos. Essa coluna é subdivida em duas partes: itens que estão passando pelo processo de levantamento de requisitos (Executando) e itens que já passaram por

esse processo (Feito). A terceira coluna (Desenvolvimento) mostra os itens que estão sendo implementados (Executando) e os itens terminados (Feito). A quarta coluna (Pronto para Teste) funciona como um *buffer* para a próxima etapa. A quinta coluna (Teste) representa os itens que estão sendo testados. E a última coluna (Itens Finalizados) armazena os itens finalizados.

Figura 5 - Exemplo de um *Kanban board*



Fonte: Adaptado de ANDERSON (2010)

Por meio do *Kanban board* é possível visualizar o que deve ser feito, limitar a quantidade de trabalho em progresso e melhorar o fluxo dos cartões nas diversas etapas do desenvolvimento. Os cartões que possuem uma prioridade mais alta podem ser posicionados no topo da sua coluna para que possam ser escolhidos primeiro.

2.6 FRAMEWORK

Pressman e Lowe (2009) define *framework* como sendo um conjunto de bibliotecas e/ou componentes usados para estruturar uma aplicação. No contexto de

aplicações Web, ele é usado, por exemplo, para gerenciar conteúdo, conexões com banco de dados, sessões e apresentação de conteúdo.

Uma das vantagens da utilização de um *framework* é a possibilidade de desenvolver uma aplicação de forma mais rápida e segura, pois funções que são comumente usadas não precisam ser reescritas, o *framework* pode fornecê-las.

Um dos *frameworks* mais utilizados para o desenvolvimento Web é o CodeIgniter. Ele fornece recursos como validações de formulários HTML, abstração no acesso ao banco de dados, gerenciadores de arquivo, entre outras funções de facilitam o desenvolvimento (CODEIGNITER, 2016).

Grande parte dos *frameworks* Web disponíveis atualmente utilizam a arquitetura *Model-View-Controller* (MVC). Nesta arquitetura há uma separação de conceitos e códigos, com o objetivo de facilitar o desenvolvimento e a manutenção de uma aplicação. A *View* é responsável por apresentar conteúdo ao usuário. O *Model* fica responsável pela escrita, validação e leitura dos dados da aplicação. E o *Controller* faz a ligação entre o *Model* e a *View*, e decide qual *Model* e *View* será usado para processar as ações dos usuários (PRESSMAN; LOWE, 2009).

2.7 MÉTODOS DE ORDENAÇÕES DE AVALIAÇÕES

Uma avaliação em um sistema é uma nota dada por um usuário a um determinado item. Ela pode ser feita usando escalas numéricas em forma de estrelas, curtidas (semelhante ao sistema do Facebook), votos positivos ou negativos, entre outras formas de avaliação.

O principal objetivo de fornecer meios para que usuários avaliem itens como produtos e serviços, é deixar em destaque os que têm as melhores avaliações. Sites como Amazon, Mercado Livre e Submarino utilizam alguns algoritmos para ordenar seus produtos, dessa forma os consumidores são beneficiados com informações que auxiliam na tomada de decisão ao comprar determinado item.

De acordo com Miller (2009), existem três principais algoritmos para ordenar itens avaliados:

- 1) A primeira técnica consiste em subtrair o total de avaliações positivas das negativas. O problema desse método é que pode acontecer de itens com uma porcentagem menor de avaliações positivas ficar em uma posição melhor do que outro

com uma porcentagem maior. O site *Urban Dictionary*, por exemplo, utiliza essa técnica.

2) A segunda técnica consiste em dividir o total de avaliações positivas pelas negativas. Com essa técnica, itens com poucas avaliações positivas e nenhuma avaliação negativa aparecem na frente dos que tem muitas avaliações positivas e poucas negativas. O site Amazon, por exemplo, utiliza essa técnica.

3) A terceira solução utiliza o intervalo de confiança de Wilson. Esse método é considerado o estado da arte. Ele consiste em balancear a proporção de avaliações positivas a partir de um número pequeno de observações. A Figura 6 ilustra a fórmula utilizada para esse cálculo. Onde $n = n_{\uparrow} + n_{\downarrow}$ é o total de avaliações positivas mais negativas, $p = n_{\uparrow}/n_{\downarrow}$, $z_{1-\alpha/2}$ é $(1 - \alpha/2)$ onde α tem valor padrão de 0.10 para garantir um intervalo de confiança de 95% (ZHANG, et al., 2011).

Figura 6 – Fórmula para ordenar avaliações utilizando o intervalo de confiança de Wilson

$$s(n_{\uparrow}, n_{\downarrow}) = \frac{p + \frac{z_{1-\alpha/2}^2}{2n} - \sqrt{\frac{z_{1-\alpha/2}^2}{n} \left[p(1-p) + \frac{z_{1-\alpha/2}^2}{4n} \right]}}{1 + \frac{z_{1-\alpha/2}^2}{n}}$$

Fonte: Adaptado de ZHANG, et al. (2011)

Para utilizar o intervalo de confiança de Wilson é necessário que as avaliações utilizem o formato de avaliações positivas ou negativas. Nesse formato o usuário tem duas opções: ou ele avalia algo como positivo ou avalia como negativo.

Avaliações que utilizam escalas de cinco estrelas em que cada estrela representa um número inteiro positivo de 1 a 5 podem ser convertidos para avaliações positivas ou negativas. O Quadro 1 mostra os valores equivalentes para conversão (GUIDO, 2015).

Quadro 1 - Conversão de avaliações no formato de estrela para votos positivos e negativos

Estrela	Voto Negativo	Voto Positivo	Total
1	1	0	1
2	0.75	0.25	1
3	0.5	0.5	1
4	0.25	0.75	1
5	0	1	1

Fonte: Adaptado de GUIDO (2015)

2.8 SISTEMAS SIMILARES

É possível encontrar sistemas similares ao proposto nesse trabalho. Os mais conhecidos são os sites das empresas OLX, GetNinjas e PetAnjo.

Na OLX os usuários podem postar anúncios de prestação de serviços, entretanto, esse não é foco principal da empresa. A grande maioria dos anúncios é relacionado à venda ou troca de produtos. Um usuário pode inserir até dois anúncios gratuitamente nesse sistema (OLX, 2016).

Já a empresa PetAnjo conecta cuidadores de animais de estimação a pessoas que necessitam desses serviços. Ela é focada exclusivamente nesse nicho de mercado. Quando um usuário faz uma busca no site, ele apresenta uma lista de anúncios e a localização dos anunciantes em um mapa, facilitando assim a contratação de uma pessoa da região. O sistema desenvolvido nesse trabalho utiliza um mecanismo de busca semelhante ao utilizado por esse site. A Figura 7 ilustra o resultado de uma busca nesse sistema (PETANJO, 2016).

Figura 7 - Resultado de uma busca no site PetAnjo

1 Elizabeth G. R\$ 60,00 /Hospedagem
Adoro animais e acima de tudo os respeito muito.
PQ Bottolandia, São Paulo-SP

Tenho um cocker há 12 anos (Nick), é meu grande companheiro. Ele é super manso e amoroso. Ele está vacinado e toma medicamento pra o coração agora que está ficando velho. Cuido dele com muito amor, é um filho para mim. Minha casa é...

[Ver Perfil](#)

2 Cristina G. R\$ 65,00 /Hospedagem
Cães amigos fazem parte da minha história
Jd. Monte Kemel, São paulo-SP

Olá, tenho horários bem flexíveis para cuidar de seu pet durante a hospedagem familiar em minha casa. A casa é ampla e bem arejada. Tenho dois quintais para banho de sol e brincadeiras. Tenho dois cachorros poodles...

[Ver Perfil](#)

Fonte: Adaptado de PETANJO (2016)

A terceira empresa, GetNinjas, fornece um mecanismo de busca de prestadores de serviços em seu site. A maioria dos usuários são da região de São Paulo. Após um usuário fazer uma busca, ele recebe por e-mail uma lista dos profissionais recomendados para executar o serviço. Para cadastrar um anúncio no site é necessário pagar um dos planos disponibilizados pela empresa. A Figura 8 mostra um exemplo de uma busca por eletricitistas na região de São Paulo – SP (GETNINJAS, 2016).

Figura 8 - Exemplo de uma busca por eletricitistas na região de São Paulo no site GetNinjas

Bem vindo de volta, Daniel!
Enviamos um email para você poder acessar sua conta e gerenciar seus pedidos.

GetNinjas Dúvidas? (11) 3791-7315 [entrar](#)

Assist. Técnica Aulas Autos Consultoria Design e Tecnologia Eventos Moda e Beleza Reformas Saúde Serviços Domésticos

Estamos buscando os **melhores profissionais** para você

Em breve você vai receber no seu email uma lista com eletricitistas perto de você.

Fonte: Adaptado de GETNINJAS (2016)

De acordo com a Figura 8, é possível verificar que o site GetNinjas oferece serviços em várias áreas, não restringindo somente a reparos residenciais. Podem ser encontrados serviços relacionados a eventos, aulas particulares, serviços de moda e beleza, entre outros.

3 DESENVOLVIMENTO

O desenvolvimento do sistema foi feito utilizando a metodologia de gerenciamento Kanban (ANDERSON, 2010), juntamente com a metodologia ágil de desenvolvimento XP (Extreme Programming) (TELES, 2006). Foi utilizado a ferramenta Trello para a criação do quadro e dos cartões Kanban (TRELLO, 2016). E para o planejamento das releases do XP e criação das histórias foi utilizado a ferramenta Pivotal Tracker (PIVOTAL TRACKER, 2016).

Os diagramas UML (RUMBAUGH; JACOBSON; BOOCH, 2004) foram modelados na ferramenta StarUML 2.6.0 (StarUML 2.6.0, 2016).

A programação no lado servidor foi feita em PHP 7 utilizando o *framework* CodeIgniter 3.0 (CodeIgniter 3.0, 2016). No lado cliente foi utilizado o *framework* para JavaScript AngularJS 1.5.8 (AngularJS, 2016) e o *framework* para CSS Bootstrap 3.7 (Bootstrap, 2016).

Toda a implementação do sistema foi feita utilizando a IDE (*Integrated development Environment*) PhpStorm 2016.2.1 (PhpStorm, 2016). Para a base de dados foi utilizado o Sistema Gerenciador de Banco de Dados PostgreSQL 9.6 (PostgreSQL, 2016).

Para auxiliar no levantamento de requisitos, foi feito um estudo de sistemas similares. E para testar o sistema foi utilizado o Desenvolvimento Orientado a Testes (do inglês, *Test Driven Development* – TDD) com o *framework* de testes para JavaScript Jasmine 2.5.0 (Jasmine, 2016). Esse *framework* permite que testes de unidade sejam criados para testar códigos do *framework* AngularJS.

Antes da implementação foram desenvolvidos protótipos para as principais telas do sistema utilizando a ferramenta Google Drawings do Google Drive (Google Drive, 2016).

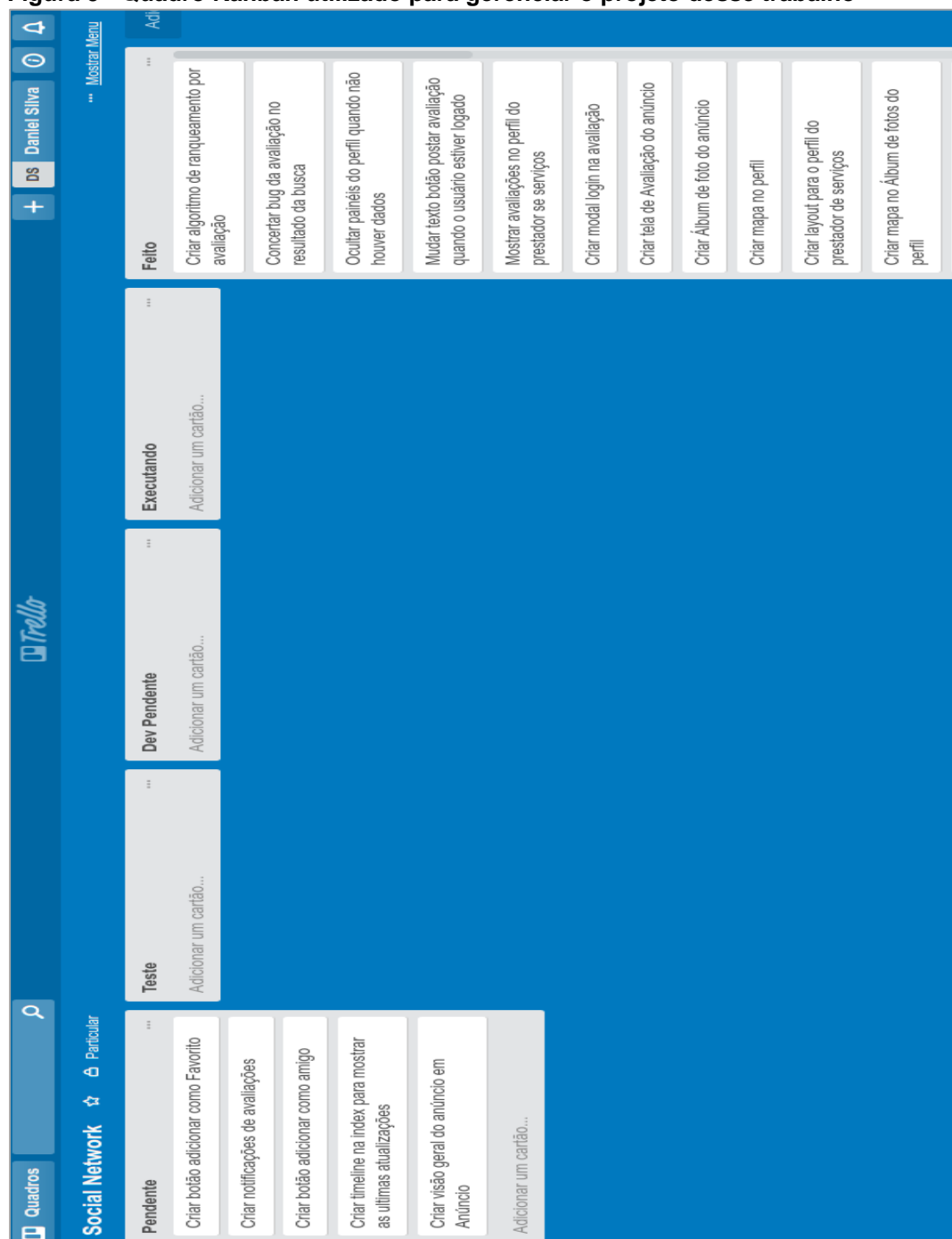
3.1 GERENCIAMENTO DO PROJETO

Todo o desenvolvimento do sistema foi realizado utilizando a metodologia de gerenciamento Kanban (Capítulo 2.7). O quadro Kanban e os cartões foram criados usando a ferramenta online Trello (TRELLO, 2016). Com ela é possível criar múltiplas colunas e inserir uma quantidade indefinida de cartões simulando dessa forma um quadro Kanban. A cada release do XP (Capítulo 2.6) os cartões do Kanban referentes

aquela release foram criados e adicionados a essa ferramenta.

A Figura 9 mostra o quadro Kanban, no Trello, utilizado para fazer o gerenciamento do projeto desse trabalho.

Figura 9 - Quadro Kanban utilizado para gerenciar o projeto desse trabalho



Fonte: Autoria própria

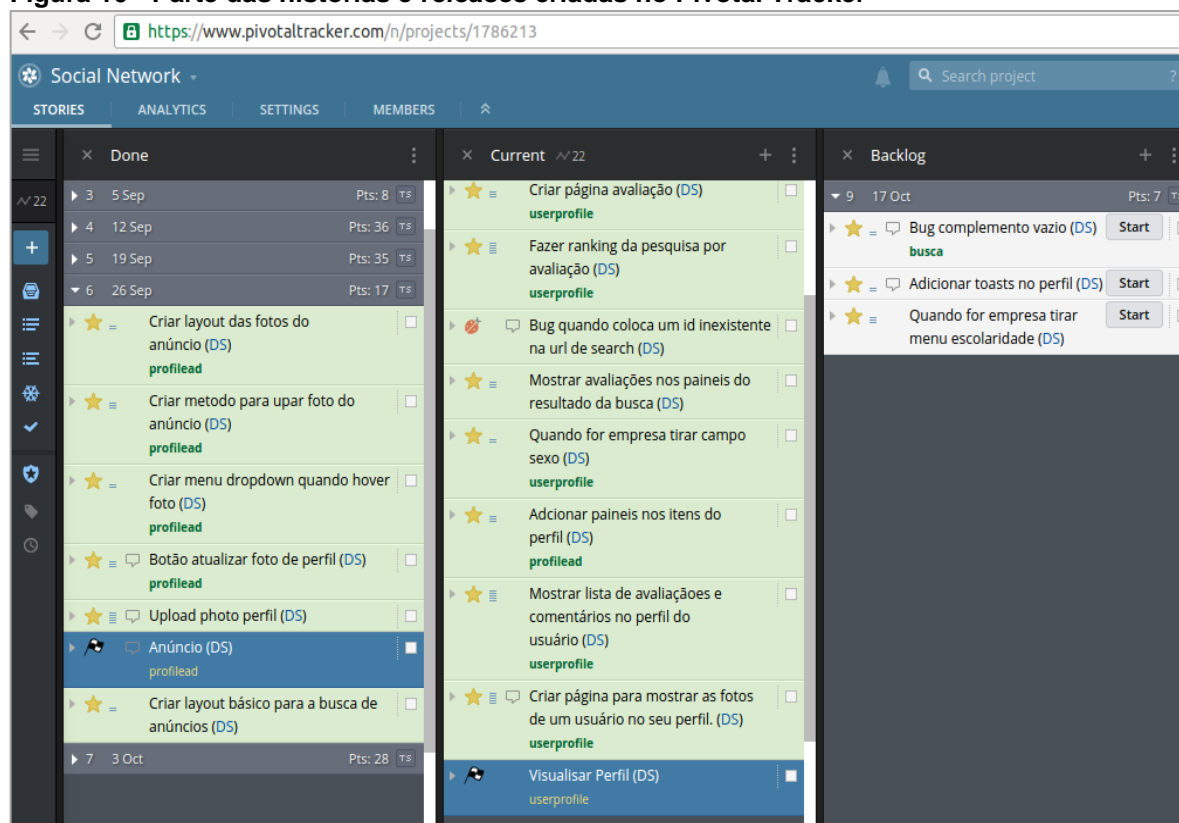
A coluna “Pendente” representa os itens que estão na lista de espera para a criação de testes. A coluna “Teste” armazena os cartões cujos testes estão sendo

desenvolvidos. Quando os testes referentes a um item são finalizados, ele vai para a coluna “Dev Pendente” para aguardar sua implementação. A coluna “Executando” refere-se aos cartões que estão sendo implementados. E por fim, a coluna “Feito” guarda os itens finalizados.

3.2 GERENCIAMENTO DE DESENVOLVIMENTO

Além da aplicação da metodologia Kanban, foi também utilizado o método ágil de desenvolvimento XP (Capítulo 2.6). A criação dos cartões (histórias) foi feita em uma ferramenta de gerenciamento de projetos online Pivotal Tracker (PIVOTAL TRACKER, 2016). Essa ferramenta permite, entre outras coisas, fazer o planejamento das *releases* e criar as histórias. A Figura 10 mostra parte das histórias e *releases* criadas nessa ferramenta. Os itens que possuem um marcador no formato de estrela são as histórias, e os que possuem um marcador semelhante a uma bandeira são as *releases*.

Figura 10 - Parte das histórias e *releases* criadas no Pivotal Tracker



Fonte: Autoria Própria

O projeto foi dividido em cinco *releases*, cada uma com duração de aproximadamente três semanas, sendo que cada uma das duas últimas teve duração de apenas uma semana e meia. Na primeira *release* foi feito a tela inicial, a criação de conta e *login* de usuários. A segunda *release* consistiu na criação do perfil. No final dela o usuário era capaz de cadastrar contato, endereço e mudar a foto de perfil. Na terceira *release* foram criadas as telas para o cadastro de anúncio, nelas eram possíveis adicionar um título para o anúncio, experiência de trabalho, escolaridade, serviços oferecidos, e criar um álbum de fotos.

Na quarta *release* foi criado um mecanismo de busca de anúncios usando os filtros definidos na página inicial. No final dessa etapa, uma lista de anúncios era mostrada de acordo com as avaliações recebidas. Além disso, foi criado um mapa para facilitar a localização de um prestador de serviços. E por fim, na quinta *release* foi criado a visualização do perfil de um usuário. Nessa etapa também foi criado a parte da avaliação do prestador de serviços por meio de uma nota de um a cinco em forma de estrelas e um comentário opcional.

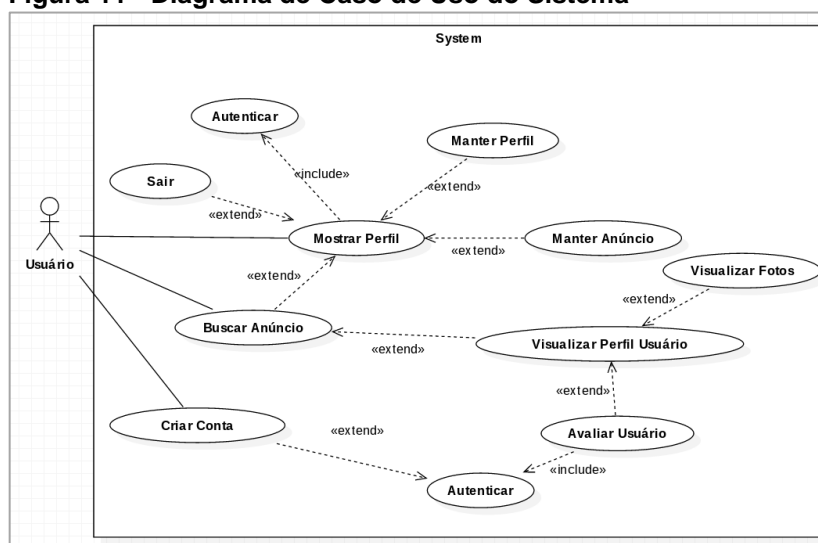
3.3 MODELAGEM DO SISTEMA

Nesta seção são apresentados os diagramas UML criados para modelar o sistema.

3.3.1 Diagrama de Caso de Uso

Após a criação dos protótipos das telas do sistema, elaborou-se o Diagrama de Caso de Uso. Ele está ilustrado na Figura 11.

Figura 11 - Diagrama de Caso de Uso do Sistema



Fonte: Autoria própria

De acordo com a Figura 11, o caso de uso “Autenticar” é responsável pela validação do e-mail e senha de um usuário para que ele possa ter acesso ao sistema. Caso ele não tenha uma conta, é necessário que ele cadastre uma nova utilizando o caso de uso “Criar Conta”. Caso o acesso ao sistema seja autorizado, ele é redirecionado para o caso de uso “Mostrar Perfil”. Esse caso de uso é responsável por mostrar os dados cadastrados no perfil. Já o caso de uso “Sair” encerra a sessão do usuário e redireciona ele para o caso de uso “Autenticar”.

No caso de uso “Manter Perfil”, é possível, por exemplo, alterar o nome do usuário, cadastrar contato e endereço. Já no caso de uso “Manter Anúncio”, o usuário é capaz de criar um título para o anúncio, cadastrar experiência, escolaridade, informar os serviços oferecidos e inserir fotos. O pré-requisito para a utilização desses dois casos de uso anteriores é a autenticação do usuário.

O caso de uso “Buscar Anúncio” é responsável por apresentar ao usuário uma lista de anúncios ordenados por avaliações recebidas. Ele pode ser acessado por qualquer pessoa, independente da autenticação. Quando um usuário se interessa por um anúncio em específico e clica no seu título ou foto, ele é redirecionado para o caso de uso “Visualizar Perfil Usuário”. É nesse caso de uso que estão disponibilizadas informações como experiência do prestador de serviços, escolaridade, lista completa dos serviços oferecidos, contato, avaliações, entre outros dados.

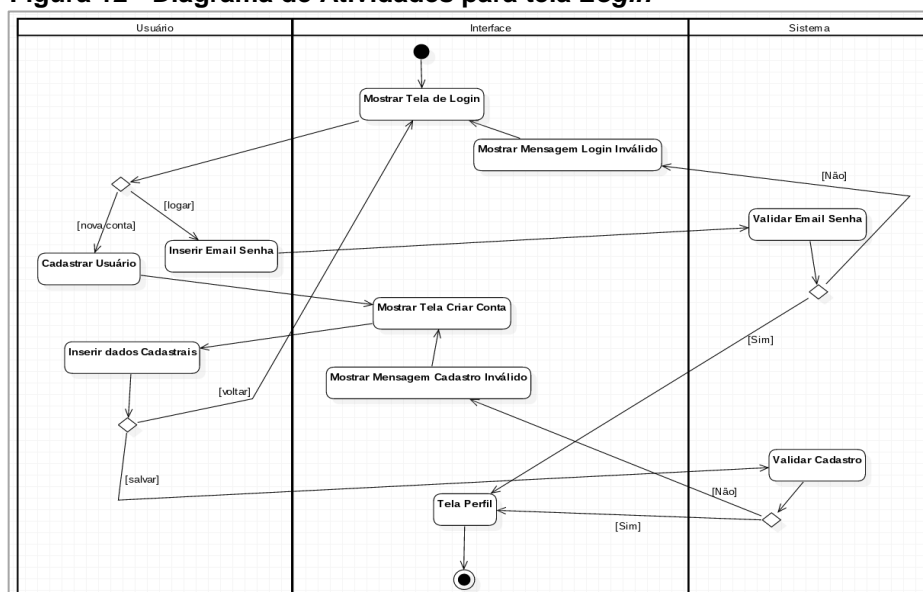
A avaliação de um prestador de serviços é realizada por meio do caso de uso “Avaliar Usuário” que é acessado usando o caso de uso “Visualizar Perfil Usuário”.

Nele é possível dar uma nota de um a cinco em forma de estrela para o anúncio, e escrever um comentário. O caso de uso “Autenticar” é responsável por autenticar o usuário após ele tentar salvar uma avaliação. Caso ele já esteja autenticado, esse caso de uso não é utilizado. E por fim, o caso de uso “Visualizar Fotos” mostra o álbum de fotos de um anunciante.

3.3.2 Diagrama de Atividades

A Figura 12 ilustra os passos necessários para que um usuário consiga se autenticar no sistema. O fluxo inicia no módulo Interface onde o sistema apresenta ao usuário a tela de *login*. Ele tem duas opções: inserir e-mail e senha ou cadastrar um novo usuário. Caso a opção escolhida seja inserir e-mail e senha, os dados são validados no módulo Sistema que o redireciona para a tela “Perfil” ou mostra uma mensagem de erro indicando que os dados fornecidos são inválidos. Outra opção é cadastrar um novo usuário, para isso o módulo Interface apresenta ao usuário a tela de criar conta. Se os dados inseridos forem válidos, o módulo Interface mostra a tela “Perfil”, caso contrário ele apresenta uma mensagem de erro indicando que os dados são inválidos. O fluxo das atividades é finalizado quando o usuário consegue acessar a tela “Perfil”.

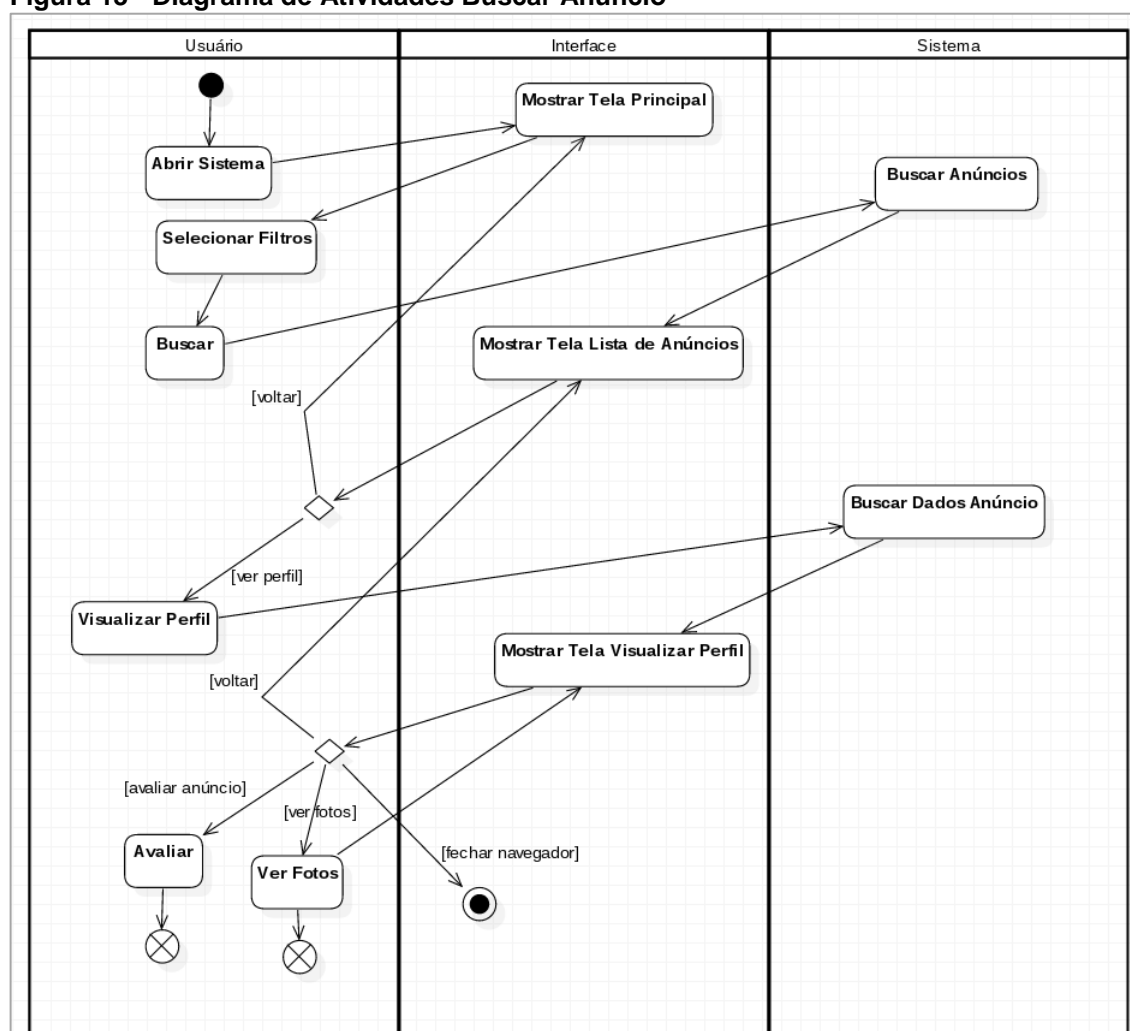
Figura 12 - Diagrama de Atividades para tela Login



Fonte: Autoria própria

A Figura 13 mostra o diagrama de atividades responsável pelo processo de busca de um anúncio. O fluxo inicia-se no módulo Usuário onde o sistema é aberto, e como resposta o módulo Interface apresenta a Tela Principal. Após o usuário selecionar os filtros de busca disponíveis, o módulo Sistema faz uma busca por anúncios e a Interface apresenta os resultados. O usuário tem a opção de visualizar o perfil do anunciante ou voltar para a página anterior. Caso ele opte por visualizar o perfil, o módulo Interface apresenta os dados do Perfil retornado pelo Sistema. O usuário tem as opções de avaliar um anúncio, visualizar fotos ou encerrar o fluxo das atividades.

Figura 13 - Diagrama de Atividades Buscar Anúncio

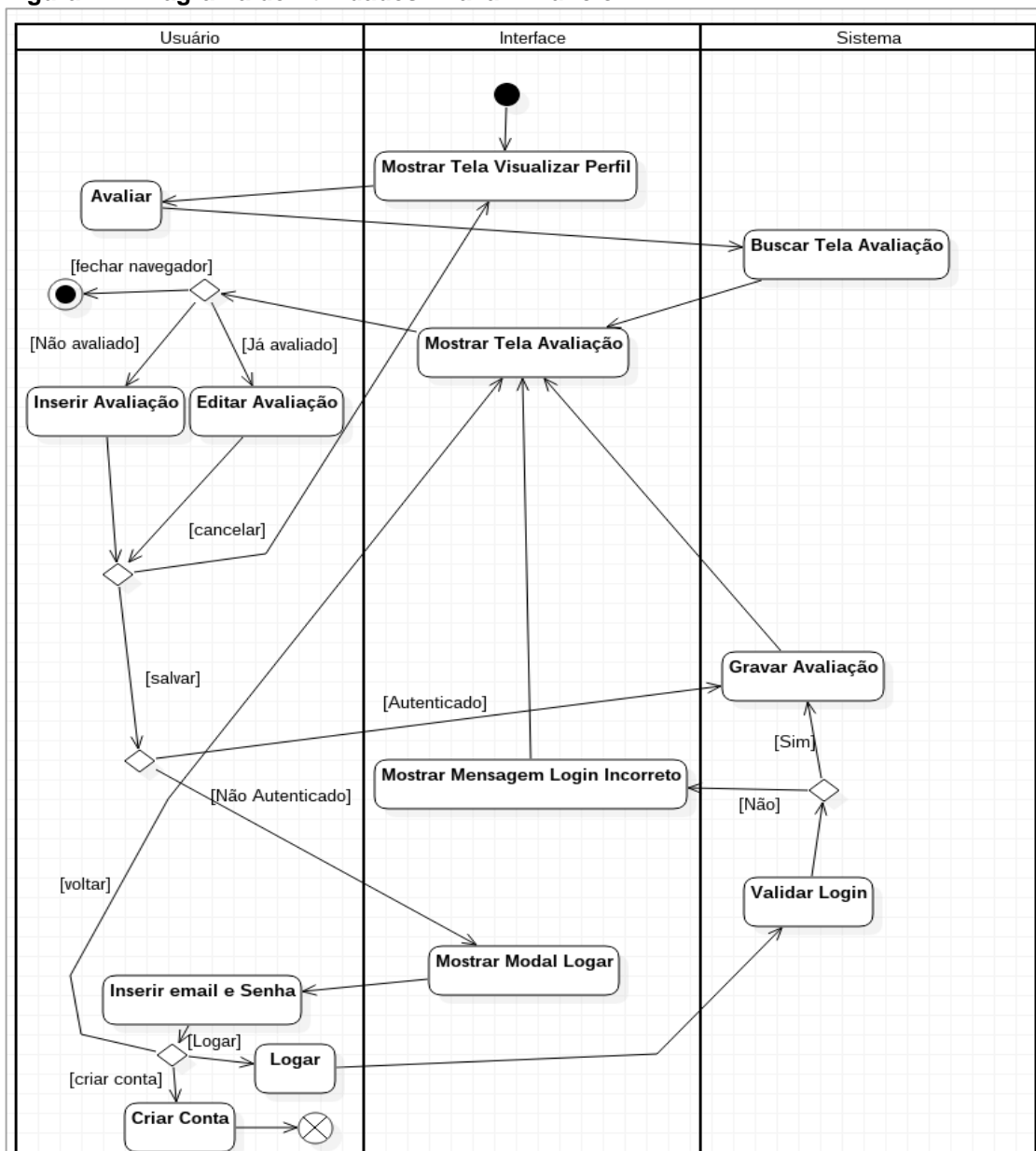


Fonte: Autoria própria

O processo de avaliação de um anúncio é mostrado no diagrama de atividades da Figura 14. O fluxo começa quando o módulo Interface apresenta a tela do perfil do prestador de serviços ao usuário. Nela ele escolhe avaliar, e então a

Interface mostra a Tela de Avaliação. O usuário pode editar uma avaliação ou criar uma nova. Para salvar, é necessário que ele esteja autenticado no sistema. Se ele não estiver, a Interface mostra uma opção de *login* para ele inserir e-mail e senha, além da opção Criar Conta. Após informar os dados para o *login*, o sistema valida eles e salva a avaliação caso os dados informados estejam corretos. Independente da avaliação ser salva ou não, o usuário permanece na Tela de Avaliação, encerrando assim o fluxo de atividades.

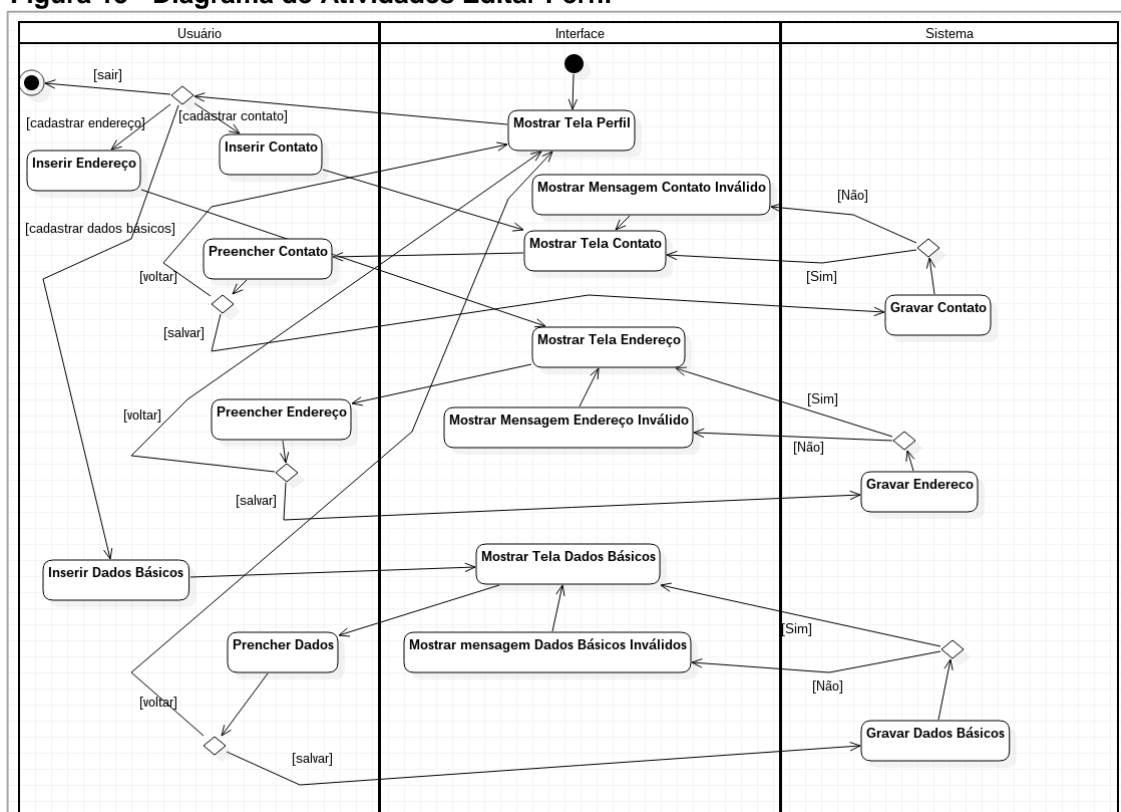
Figura 14 - Diagrama de Atividades Avaliar Anúncio



Fonte: Autoria própria

A Figura 15 ilustrado o diagrama de atividades relacionado em editar os dados do perfil de um usuário. O fluxo começa quando o módulo Interface apresenta ao usuário a Tela Perfil. Ele tem as opções de inserir dados como nome, contato e endereço. O cadastro desses dados segue um padrão que consiste no preenchimento dos formulários onde o usuário tem a opção de voltar para a Tela Perfil ou salvar os dados. Quando ele escolhe salvar os dados, o sistema faz uma validação deles e os salva. O módulo interface apresenta uma mensagem de erro quando os dados informados são inválidos.

Figura 15 - Diagrama de Atividades Editar Perfil



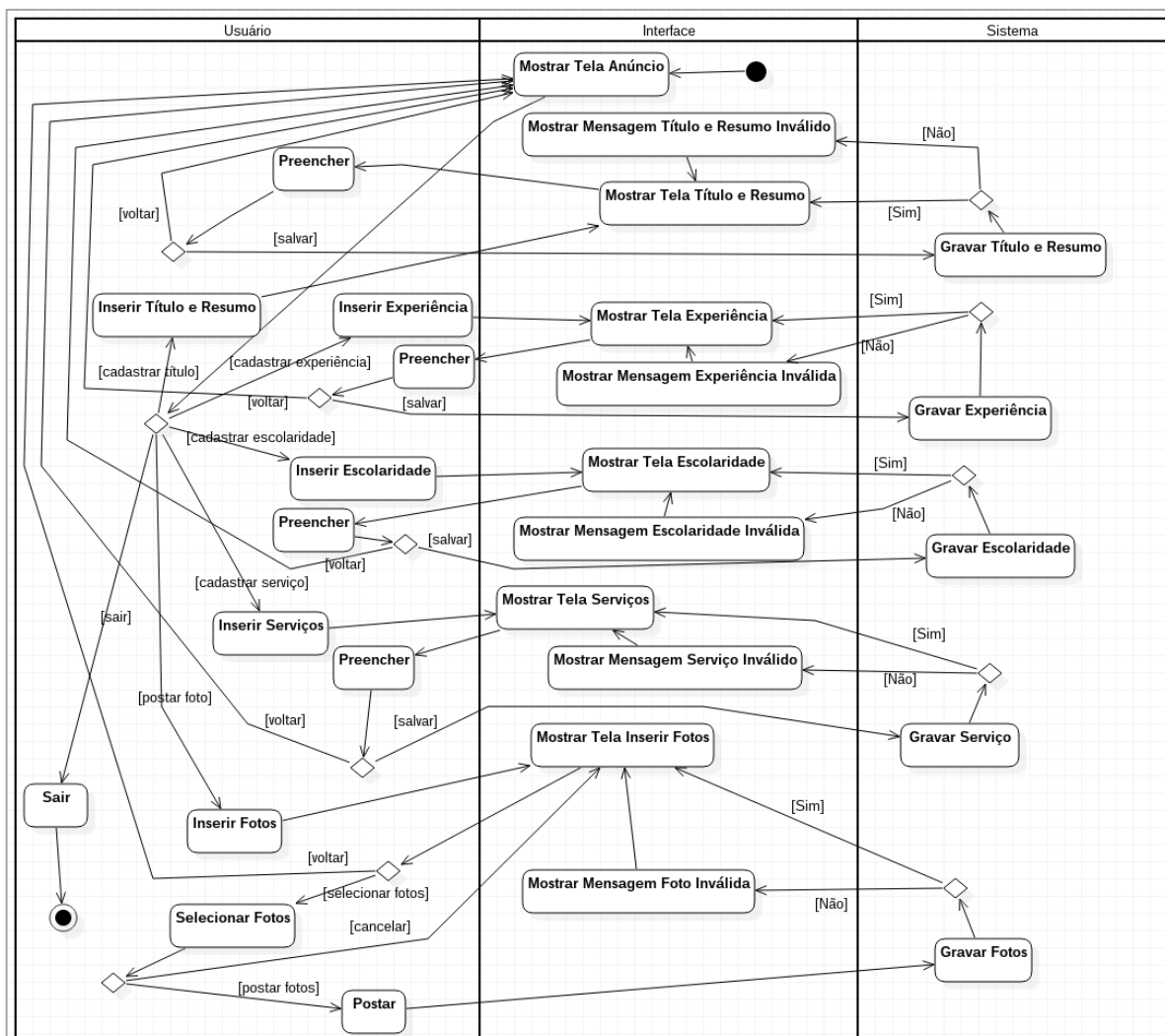
Fonte: Autoria própria

A Figura 16 apresenta o diagrama de atividades relacionado a ação de inserir um anúncio. Primeiramente o módulo Interface mostra a Tela Anúncio, após isso o usuário tem as opções de Inserir Título e Resumo, Experiência, Escolaridade, Serviços e Fotos. Em todas as opções, com exceção de Inserir Fotos, o usuário insere os dados cadastrais e, o módulo Sistema salva eles no banco de dados ou apresenta uma mensagem de erro na tela quando eles forem inválidos.

Na atividade Inserir Fotos, o usuário tem a opção de visualizar as fotos antes

de postar elas, isso é feito por meio da atividade Mostrar Fotos Seleccionadas no módulo Interface. Ele tem duas opções: Postar ou Cancelar. Se a opção escolhida for Postar, o módulo Sistema salva as fotos e o usuário permanece na Tela Inserir Fotos, caso contrário a pré-visualização das fotos é abortada e o usuário permanece na mesma tela.

Figura 16 - Diagrama de Atividades Editar Anúncio



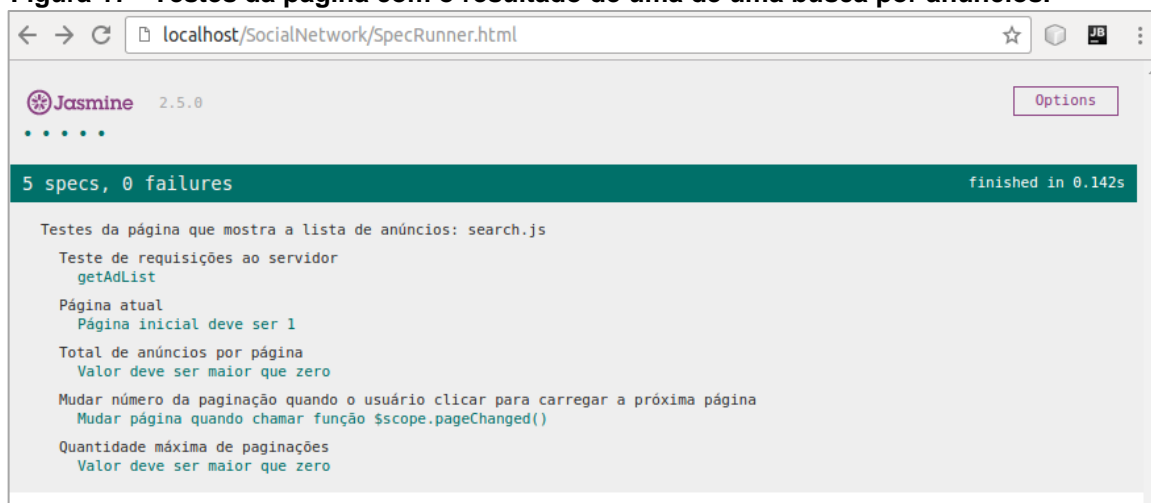
Fonte: Autoria própria

3.4 TESTES DO SISTEMA

Os testes do sistema foram feitos utilizando o *framework* Jasmine 2.5.0 (Jasmine, 2016). Esse *framework* permite a criação de testes de unidade para códigos *JavaScript* do *framework* AngularJS (AngularJS, 2016). A Figura 17 mostra alguns testes desenvolvidos para a página que contém a lista de anúncios após um usuário

fazer uma busca.

Figura 17 - Testes da página com o resultado de uma de uma busca por anúncios.



Fonte: Autoria própria

No exemplo da Figura 17, foram criados cinco testes. Eles são chamados de *specs* pelo *framework*. No primeiro foi verificado se os dados dos anúncios retornados pela requisição feita ao servidor estão corretos. No segundo é testado se a paginação do resultado de uma busca está definida com o valor numérico um. Já no terceiro, é testado o total de anúncios mostrados em uma página, esse valor deve ser um número inteiro maior que zero. O quarto é responsável por verificar se o número da paginação muda quando o usuário clica para carregar os anúncios da próxima página. E o quinto testa a quantidade máxima de números presentes na barra de paginação do resultado de uma busca.

3.5 MODELAGEM DO BANCO DE DADOS

O modelo foi criado especificamente para o Sistema Gerenciador de Banco de Dados (SGBD) PostgreSQL. Tal SGBD foi escolhido por possuir um avançado recurso de gerenciamento de multiusuários, conhecido como *Multi-Version Concurrency Control* (MVCC). O MVCC não utiliza bloqueios para o controle de transações, isso significa que leituras não precisam aguardar escritas e vice-versa, fazendo com que o SGBD tenha uma maior performance (PostgreSQL, 2016).

A Figura 18 ilustra o modelo relacional do banco de dados utilizado na aplicação. Optou-se por usar termos em inglês para nomear as tabelas e seus atributos para facilitar a internacionalização do projeto.

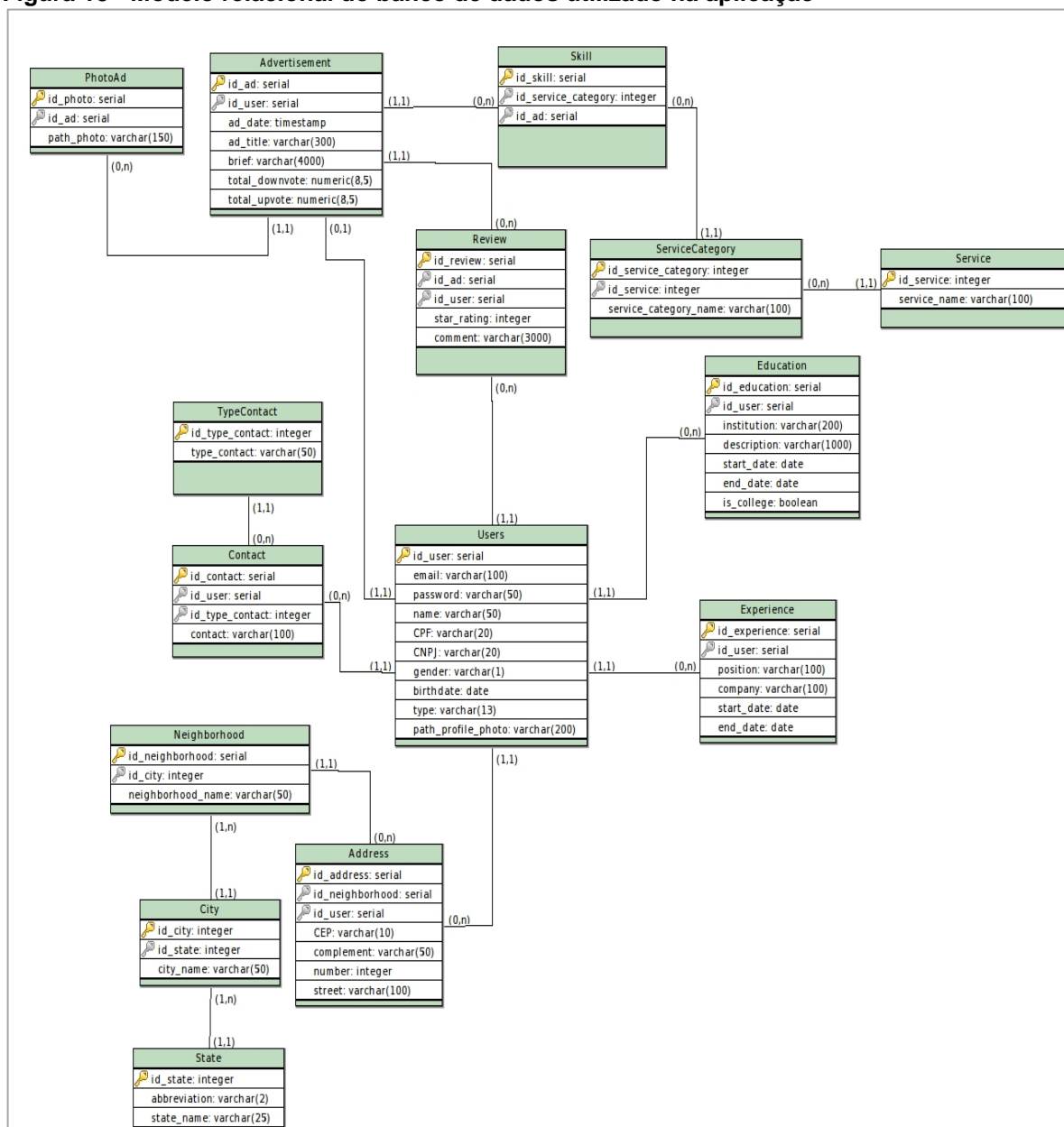
A tabela “*Users*” é responsável por armazenar os dados básicos de um novo usuário como e-mail, senha e nome. Já as tabelas “*Address*”, “*Neighborhood*”, “*City*” e “*State*” são responsáveis por armazenar o endereço de um usuário.

A tabela “*Experience*” contém a experiência de um usuário, nela ele pode colocar, por exemplo, qual foi o seu último emprego, empresa em que trabalhou, data de início e fim, e cargo. O relacionamento entre as tabelas “*Users*” e “*Experience*” é de 0 para n possibilitando que um usuário insira n experiências. A tabela “*Contact*” e “*TypeContact*” armazena o contato e tipo de contato, essa última possui valores pré-definidos (comercial, residencial e celular). E por meio da tabela “*Education*”, um usuário pode cadastrar n escolas ou universidades que ele frequentou.

Quando um usuário cria uma conta no sistema, ele tem a opção de criar um anúncio. Os dados do anúncio como resumo, título e data são salvos na tabela “*Advertisement*”. O usuário também tem a opção de criar uma galeria de fotos para o anúncio, elas são salvas em uma pasta no servidor e seu caminho é armazenado na tabela “*PhotoAd*”.

A tabela “*Review*” é responsável pelas avaliações relacionadas a um anúncio. Ele pode ter 0 ou n avaliações, e cada avaliação é composta por uma nota de 0 a 5 e um comentário opcional que são salvos nos campos “*star_rating*” e “*comment*”, respectivamente. O relacionamento entre a tabela “*Review*” e “*Users*” é de 1 para 1, pois uma avaliação é de apenas um usuário.

Figura 18 - Modelo relacional do banco de dados utilizado na aplicação



Fonte: Autoria própria

Os serviços prestados por um usuário são armazenados na tabela “Skill”. Ela possui um relacionamento de 1 para 1 com “ServiceCategory” que é responsável pelas subcategorias de serviço. A tabela “ServiceCategory” possui um relacionamento de 1 para 1 com a tabela “Service” que armazena uma lista de serviços disponíveis. Caso um usuário ofereça o serviço de instalações elétricas residenciais, por exemplo, a tabela “Service” armazena a categoria “Eletricidade” e a tabela “ServiceCategory” armazena “Instalações elétricas residenciais”.

4 SISTEMA

De acordo com seção 2.2, sistemas que permitem avaliações de conteúdos por meio de comentários e notas podem se enquadrar na categoria de redes sociais de avaliações. O sistema desenvolvido nesse trabalho se enquadra nessa categoria.

Antes de iniciar o desenvolvimento do sistema, foram criados protótipos das principais telas. Esta seção apresenta os protótipos e as telas criadas a partir deles.

4.1 PÁGINA INICIAL

A Figura 19 ilustra a tela inicial do sistema. Não foi criado protótipo para essa tela.

Figura 19 - Tela Inicial

A imagem mostra a interface de usuário da página inicial. No topo, há uma barra de navegação com o texto 'Home' à esquerda e três links à direita: 'Minha Conta' (com ícone de perfil), '+ Criar Conta' e 'Inserir Anúncio' (em um botão azul). Abaixo, o conteúdo principal é centralizado e contém quatro menus suspensos empilhados verticalmente: 'Selecione um estado', 'Selecione uma cidade', 'Selecione um serviço' e 'Detalhes do serviço'. Cada menu possui uma seta para baixo no canto direito. Logo abaixo dos menus, há um botão azul com o texto 'BUSCAR' em branco.

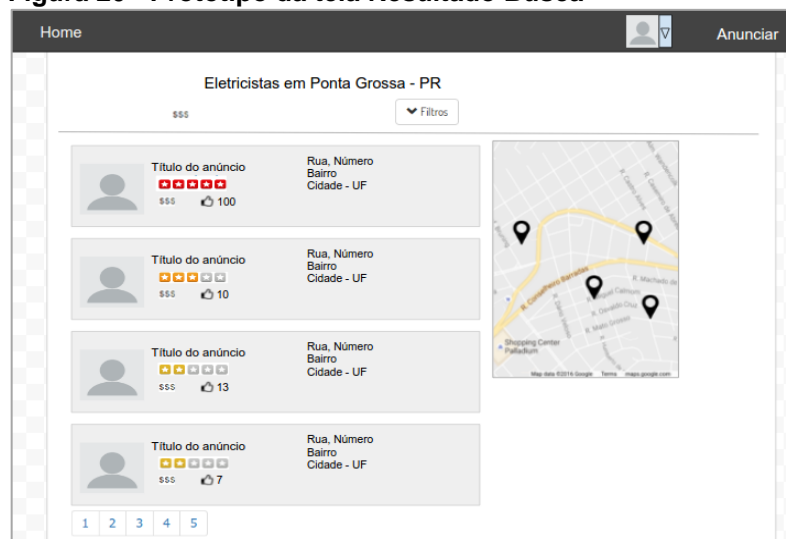
Fonte: Autoria própria

Nela um usuário pode utilizar filtros para buscar prestadores de serviço. As opções de filtro são por estado, cidade, serviço e subcategoria de serviço. Caso o usuário não selecione nenhum filtro e clique em “buscar”, o sistema mostrará uma lista com os últimos anúncios postados. Para realizar buscas, não é necessário estar autenticado no sistema.

4.2 TELA RESULTADO BUSCA

A Figura 20 mostra o protótipo da tela com o resultado de uma busca por prestadores de serviços.

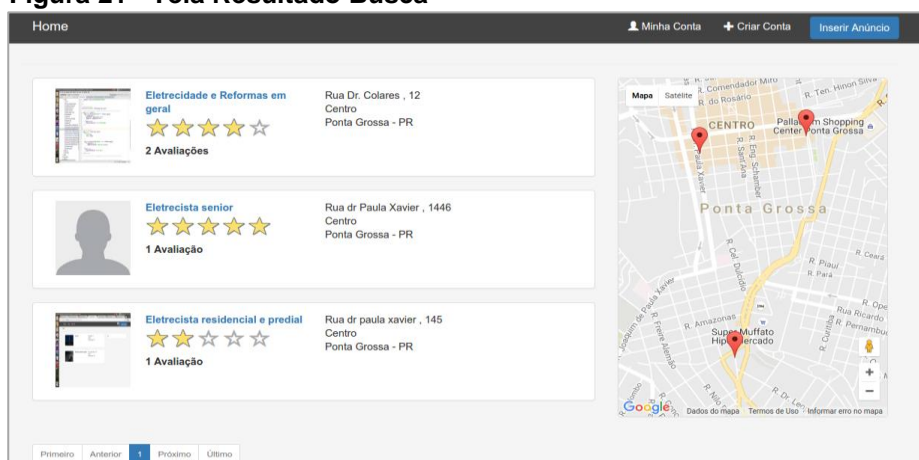
Figura 20 - Protótipo da tela Resultado Busca



Fonte: Autoria Própria

A Figura 21 mostra a tela criada a partir do protótipo da Figura 20. Ela apresenta o resultado de uma busca por eletricistas na cidade de Ponta Grossa - PR. Cada página apresenta uma lista com cinco anúncios de prestadores de serviço. Os anúncios são ordenados por meio do intervalo de confiança de Wilson (seção 2.10) utilizando as avaliações como parâmetro.

Figura 21 - Tela Resultado Busca



Fonte: Autoria própria

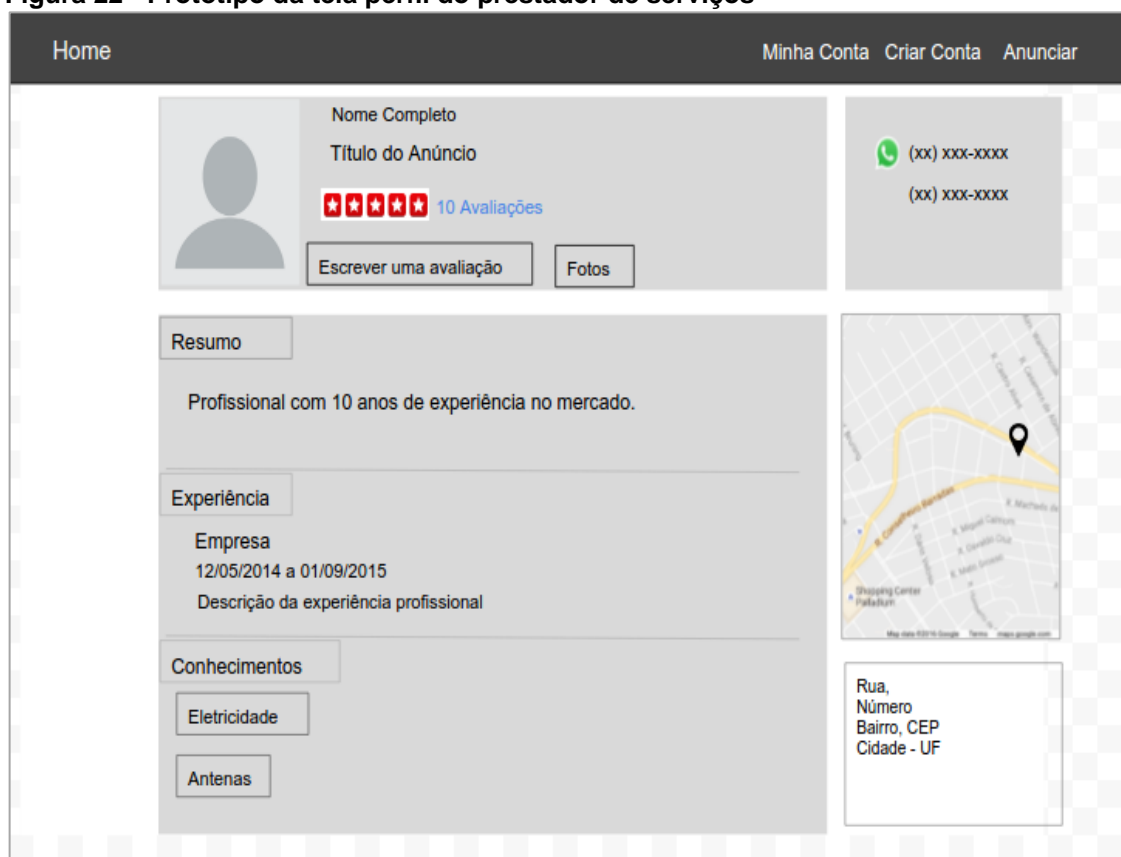
Quando o usuário passa o mouse por cima de um anúncio, o marcador de endereço no mapa move para cima e para baixo destacando-o entre os demais marcadores, facilitando a identificação da localização.

Nessa tela é possível verificar a quantidade de avaliações que um anúncio recebeu, bem como uma representação gráfica da nota (estrelas), que é calculada usando a média de todas as avaliações. Cinco estrelas representam nota máxima. Para obter-se mais informações sobre um anúncio, o usuário pode clicar no anúncio para acessar o perfil do prestador de serviços.

4.3 TELA PERFIL DO PRESTADOR DE SERVIÇOS

A Figura 22 mostra o protótipo da tela do perfil do prestador de serviços.

Figura 22 - Protótipo da tela perfil do prestador de serviços



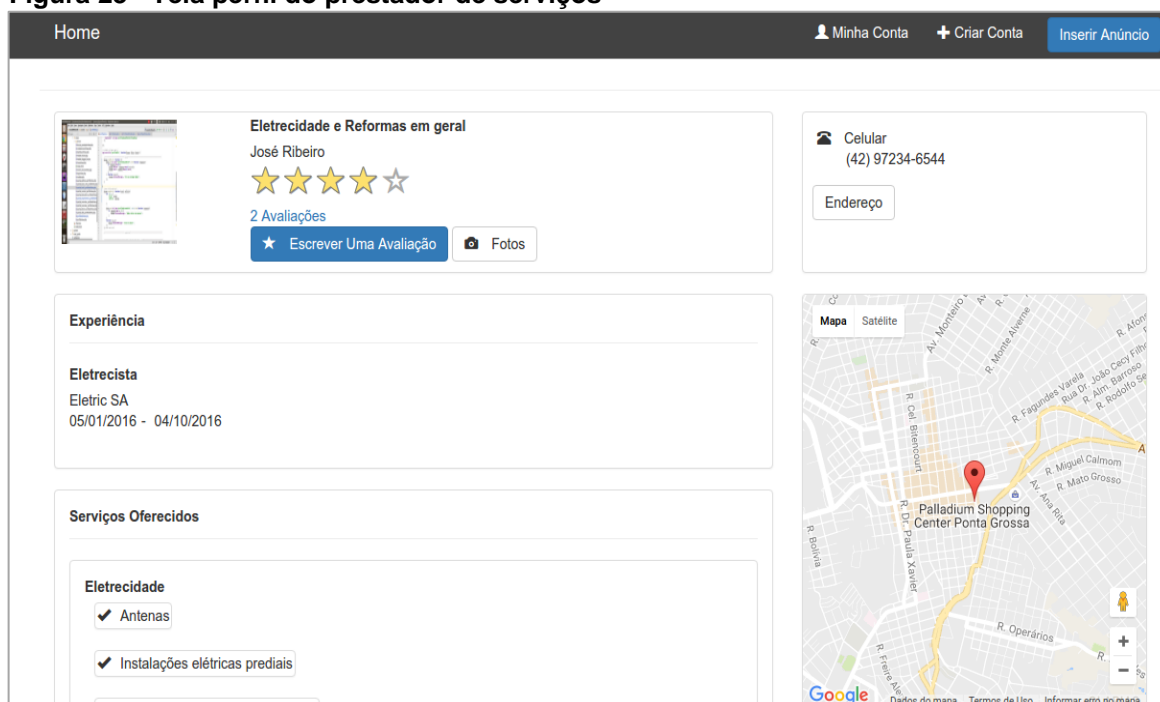
Fonte: Autoria própria

A partir do protótipo da Figura 22 foi criada a tela ilustrada na Figura 23. Ela mostra o perfil de um prestador de serviços. Usuários podem visualizar informações

como serviços oferecidos, resumo do anúncio, escolaridade, experiência, e uma lista com as avaliações recebidas, entre outras informações.

Quando o usuário clica no botão “Escrever uma Avaliação”, ele é redirecionado para a Tela Avaliação (Seção 4.7.1). Existe também a possibilidade de o usuário visualizar as fotos disponibilizadas pelo prestador de serviço, para isso é necessário clicar no botão “Fotos”. Há um terceiro botão que foi chamado de “Endereço”, ele foi criado para facilitar a localização do endereço do prestador de serviço. Quando clicado, a página rola para baixo deixando o endereço, que está localizado abaixo do mapa, no topo da página.

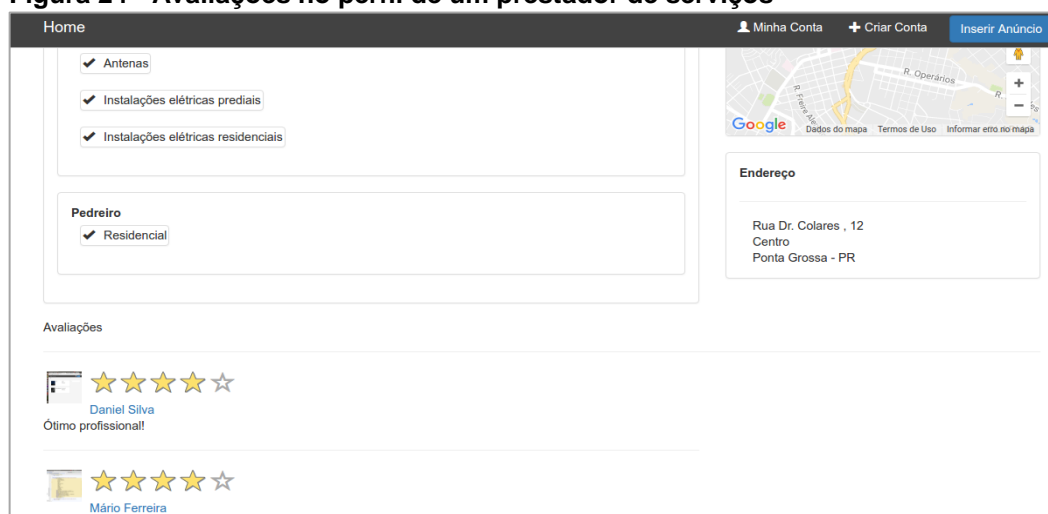
Figura 23 - Tela perfil do prestador de serviços



Fonte: Autoria própria

A Figura 24 mostra as avaliações no perfil do prestador de serviços da Figura 23. O usuário “Mário Ferreira” avaliou ele com quatro estrelas, e “Daniel Silva” avaliou com o comentário “Ótimo Profissional!” e quatro estrelas. Uma avaliação é composta por uma nota, de um a cinco em forma de estrelas, e um comentário opcional.

Figura 24 - Avaliações no perfil de um prestador de serviços

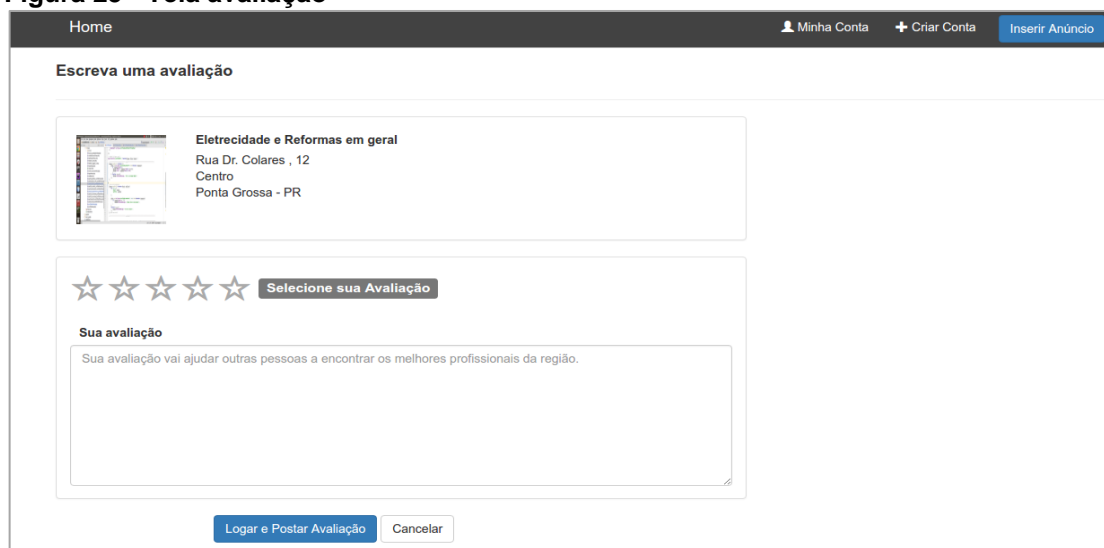


Fonte: Autoria própria

4.4 TELA AVALIAÇÃO

Esta tela é apresentada ao usuário quando ele clica no botão “Escrever uma Avaliação” localizado no perfil do prestador de serviços. Nela é possível fazer uma avaliação por meio de uma nota de um a cinco representada por meio de estrelas. Além disso, há a possibilidade de escrever um comentário que ficará posteriormente visível no perfil do prestador de serviço, para que outras pessoas possam visualizar avaliações anteriores. Para que a avaliação seja salva, é necessário que o usuário esteja autenticado no sistema. A Figura 25 ilustra a tela de avaliação.

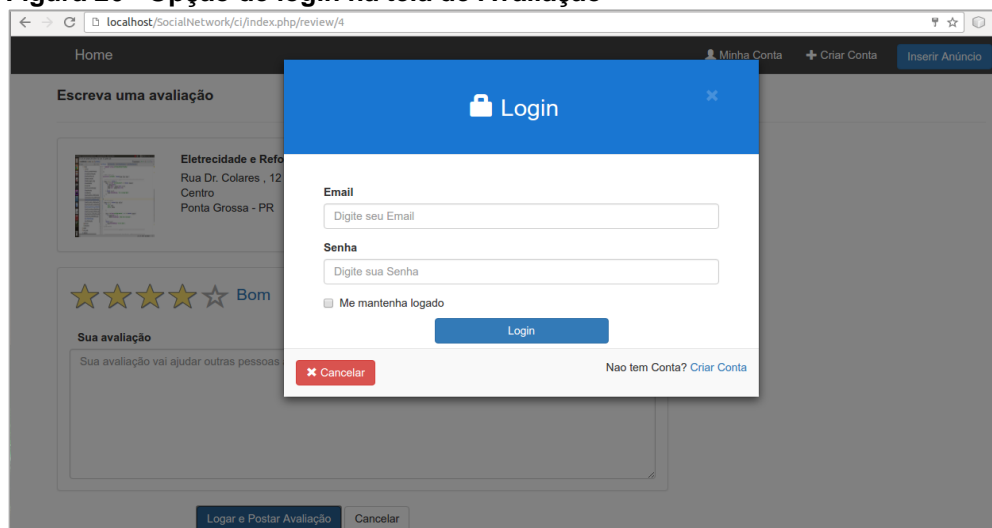
Figura 25 - Tela avaliação



Fonte: Autoria própria

A Figura 26 mostra a resposta do sistema quando um usuário clica no botão “Logar e Postar Avaliação”. Caso ele informe e-mail e senha corretamente, a avaliação feita por ele será salva e mostrada no perfil do prestador de serviços. A alteração de uma avaliação é possível a qualquer momento, basta que ele submeta outra avaliação com as novas modificações.

Figura 26 - Opção de login na tela de Avaliação

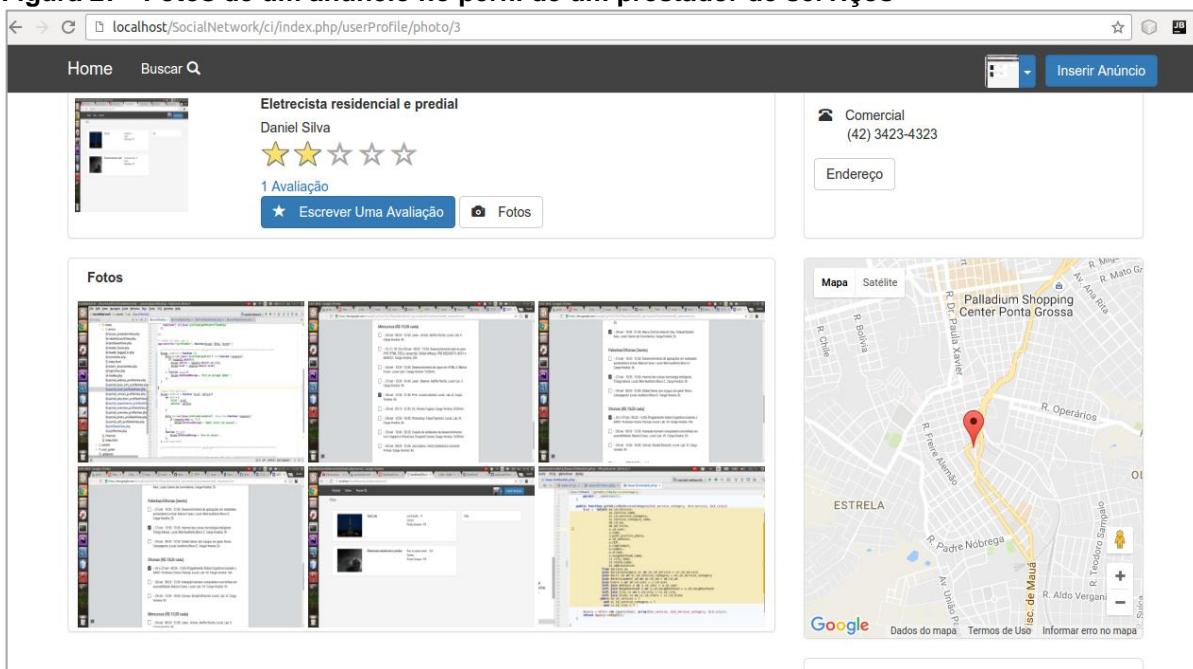


Fonte: Autoria própria

4.5 TELA FOTOS

A Figura 27 mostra as fotos adicionadas por um prestador de serviços a um anúncio. Essa tela é acessada usando o botão “Fotos” no perfil de um prestador de serviços.

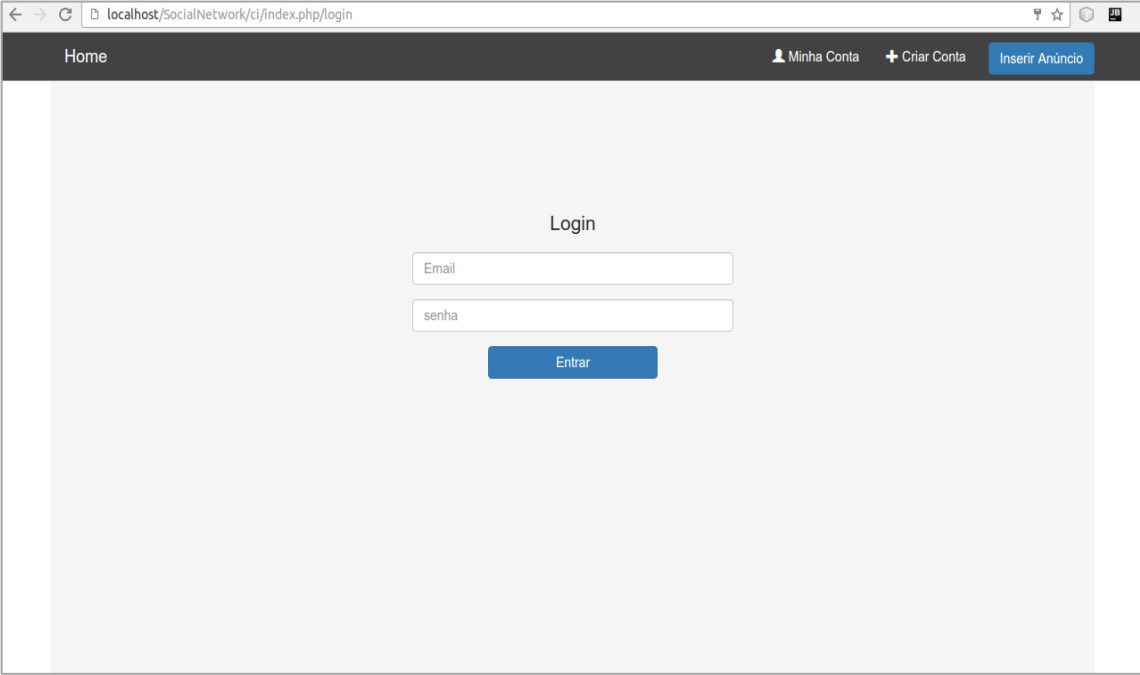
Figura 27 - Fotos de um anúncio no perfil de um prestador de serviços



Fonte: Autoria própria

4.6 TELA LOGIN

A tela de *login* ilustrada na Figura 28 é apresentada ao usuário quando ele clica em “Minha Conta” na parte superior direita do sistema. Para que ele consiga a autenticação, é necessário que ele possua e-mail e senha previamente cadastrados.

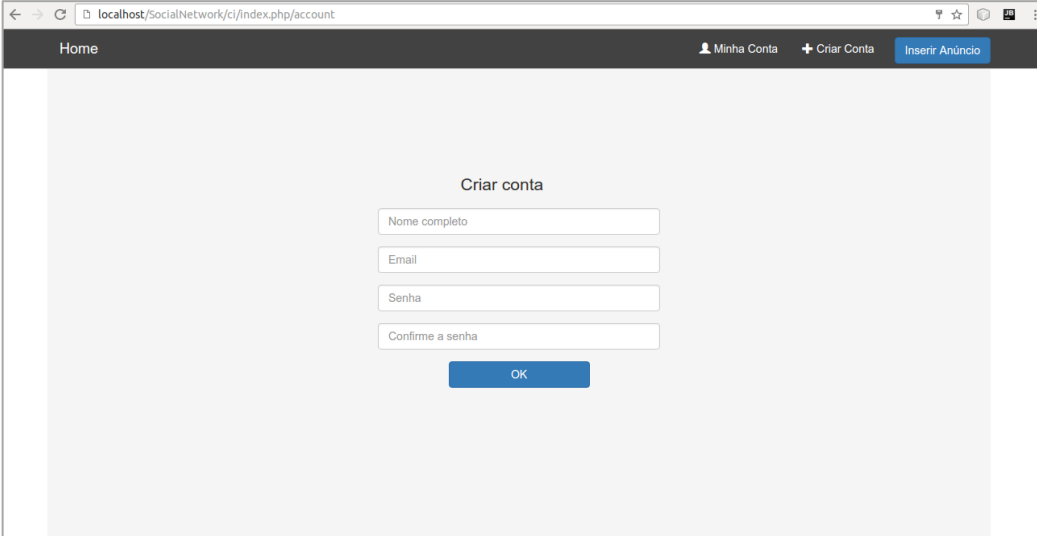
Figura 28 - Tela de Login

The screenshot shows a web browser window with the address bar containing 'localhost/SocialNetwork/ci/index.php/login'. The page features a dark navigation bar at the top with the text 'Home' on the left and three links: 'Minha Conta', 'Criar Conta', and 'Inserir Anúncio' on the right. The main content area is a light gray rectangle centered on the page. It is titled 'Login' and contains two input fields: 'Email' and 'senha'. Below these fields is a blue button labeled 'Entrar'.

Fonte: Autoria própria

4.7 TELA CRIAR CONTA

Para que um anúncio seja criado, é necessário que o usuário possua uma conta. A Figura 29 ilustra a tela em que o usuário é capaz de criar uma nova conta. Para isso, ele deve informar nome, e-mail e senha que são campos obrigatórios.

Figura 29 - Tela criar conta

The screenshot shows a web browser window with the address bar containing 'localhost/SocialNetwork/ci/index.php/account'. The page features a dark navigation bar at the top with the text 'Home' on the left and three links: 'Minha Conta', 'Criar Conta', and 'Inserir Anúncio' on the right. The main content area is a light gray rectangle centered on the page. It is titled 'Criar conta' and contains four input fields: 'Nome completo', 'Email', 'Senha', and 'Confirme a senha'. Below these fields is a blue button labeled 'OK'.

Fonte: Autoria própria

4.8 ABA PERFIL

Uma vez autenticado no sistema, o usuário pode acessar a aba “Perfil” e cadastrar dados pessoais, tais como: nome, CPF, sexo, entre outros dados. Nos protótipos ela se chama “Minha Conta”. A Figura 30 ilustra o protótipo da tela do menu “Dados básicos”.

Figura 30 - Protótipo da tela do menu dados básicos

O protótipo da tela do menu "Dados básicos" apresenta uma interface com uma barra superior escura contendo o link "Home" à esquerda, um ícone de perfil de usuário no centro e o link "Anunciar" à direita. Abaixo da barra, há uma barra de navegação com os itens "Dashboard", "Minha Conta" (destacado em verde) e "Meu Anúncio".

À esquerda, há um menu lateral com um ícone de perfil de usuário e os itens "Visão Geral", "Dados básicos" (destacado em verde), "Contato" e "Endereço".

À direita, o formulário de "Dados básicos" contém os seguintes campos e opções:

- Nome: Campo de texto com o placeholder "Nome Completo".
- Eu sou: Opções de rádio para "Autônomo" e "Empresa".
- CPF ou CNPJ: Campo de texto com o placeholder "CPF ou CNPJ".
- Sexo: Opções de rádio para "Masculino" e "Feminino".

Um botão azul "Salvar" está posicionado na base do formulário.

Fonte: Autoria própria

A Figura 31 ilustra a tela do menu “Dados básicos” criada com base no protótipo da Figura 30.

Figura 31 - Tela do menu dados básicos

The screenshot shows a web browser window with the URL `localhost/SocialNetwork/ci/index.php/profile/load_profile/#/`. The page has a dark header with 'Home' and 'Buscar' on the left, and a user profile icon and 'Inserir Anúncio' button on the right. Below the header, there are three menu items: 'Dashboard', 'Perfil' (highlighted in green), and 'Anúncio'. The main content area contains a form for editing basic data. On the left, there is a sidebar menu with 'Visão Geral', 'Dados Básicos' (highlighted in green), 'Contato', and 'Endereço'. The form fields are: 'Nome' (Daniel Silva), 'Eu sou' (radio buttons for 'Autônomo' and 'Empresa'), 'CPF/CNPJ' (123123245), and 'Sexo' (radio buttons for 'Masculino' and 'Feminino'). A 'Salvar' button is at the bottom of the form.

Fonte: Autoria própria

Quando o usuário clica no menu “Contato”, ele pode cadastrar um tipo de contato (comercial, residencial ou celular) e um número de telefone. A Figura 32 mostra o protótipo criado para esse menu.

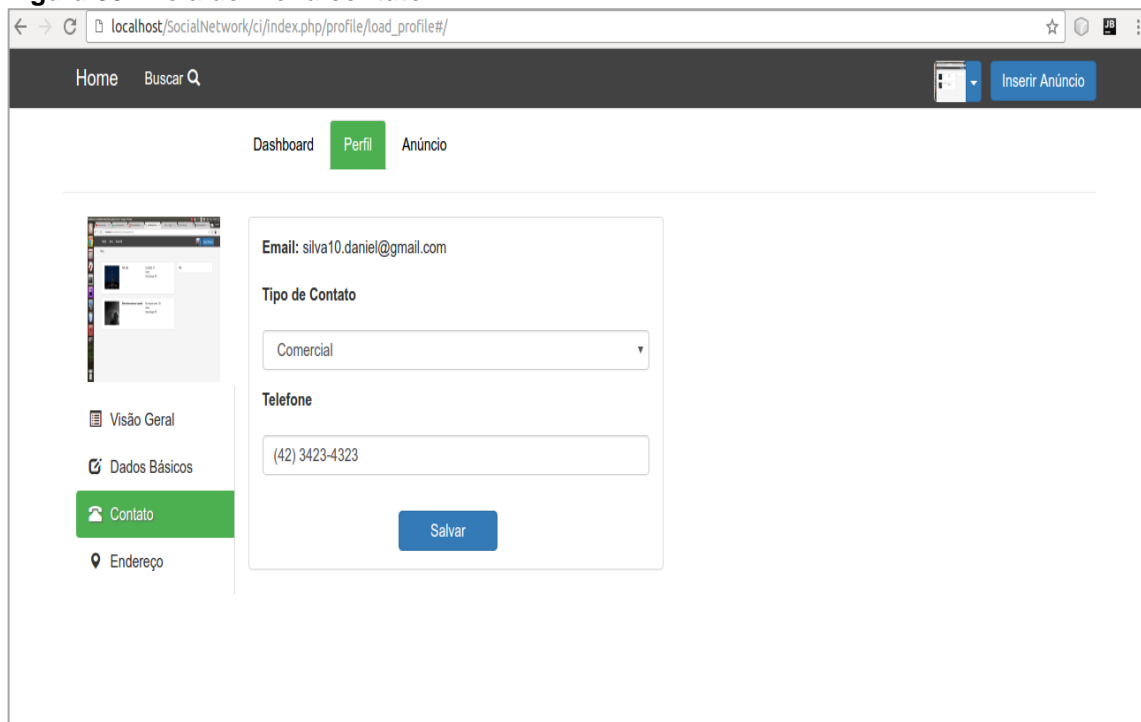
Figura 32 - Protótipo da tela do menu contato

The prototype shows a web browser window with the URL `localhost/SocialNetwork/ci/index.php/profile/load_profile/#/`. The page has a dark header with 'Home' on the left and a user profile icon and 'Anunciar' button on the right. Below the header, there are three menu items: 'Dashboard', 'Minha Conta' (highlighted in green), and 'Meu Anúncio'. The main content area contains a form for editing contact information. On the left, there is a sidebar menu with 'Visão Geral', 'Dados básicos', 'Contato' (highlighted in green), and 'Endereço'. The form fields are: 'Email' (Email), 'Tipo de Contato' (dropdown menu), and 'Telefone' (Telefone). A 'Salvar' button is at the bottom of the form.

Fonte: Autoria própria

Com base no protótipo da Figura 32, foi criado a tela da Figura 33 para o menu “Contato”.

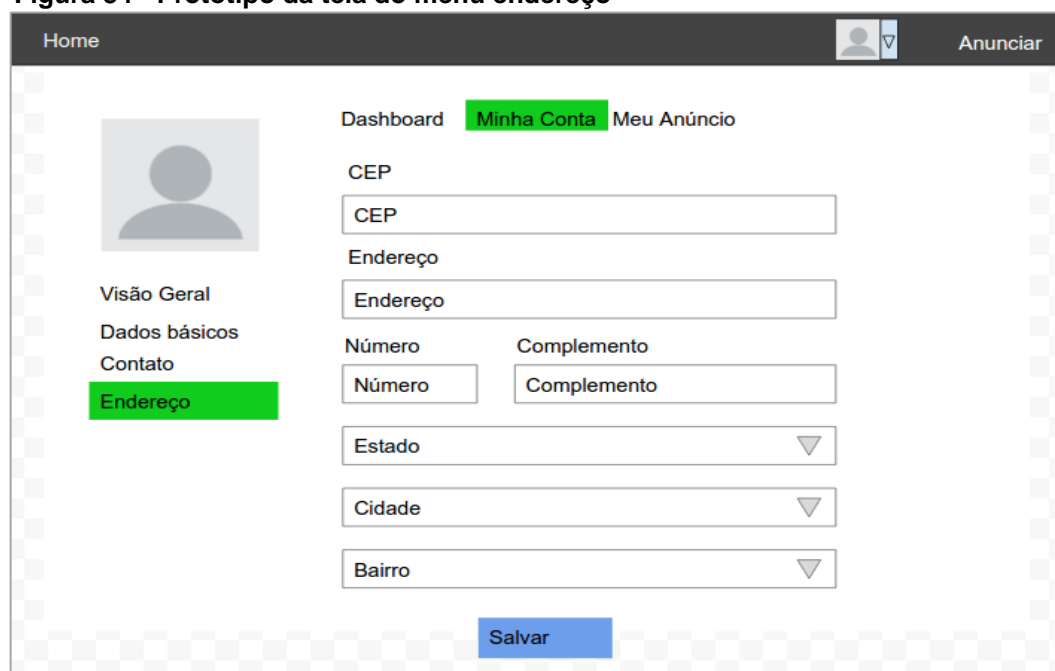
Figura 33 - Tela do menu contato



The screenshot shows a web browser window with the URL `localhost/SocialNetwork/cj/index.php/profile/load_profile/#/`. The page has a dark header with "Home" and "Buscar" on the left, and a mobile menu icon and "Inserir Anúncio" button on the right. Below the header, there are three tabs: "Dashboard", "Perfil" (highlighted in green), and "Anúncio". The main content area is divided into a left sidebar and a main form. The sidebar contains a profile picture placeholder and four menu items: "Visão Geral", "Dados Básicos", "Contato" (highlighted in green), and "Endereço". The main form contains the following fields: "Email: silva10.daniel@gmail.com", "Tipo de Contato" (a dropdown menu with "Comercial" selected), and "Telefone" (a text input field containing "(42) 3423-4323"). A blue "Salvar" button is positioned below the phone number field.

Fonte: Autoria própria

O último menu da aba perfil é “Endereço”, nele é possível inserir CEP, rua, número, complemento, estado, cidade e bairro. Esses dados são usados para mostrar a localização do usuário, no perfil dele, por meio do *Google Maps*. Os campos estado, cidade e bairro possuem valores previamente cadastrados. A Figura 34 apresenta o protótipo da tela desse menu.

Figura 34 - Protótipo da tela do menu endereço

Home Anunciar

Dashboard **Minha Conta** Meu Anúncio

Visão Geral
Dados básicos
Contato
Endereço

CEP

Endereço

Número Complemento

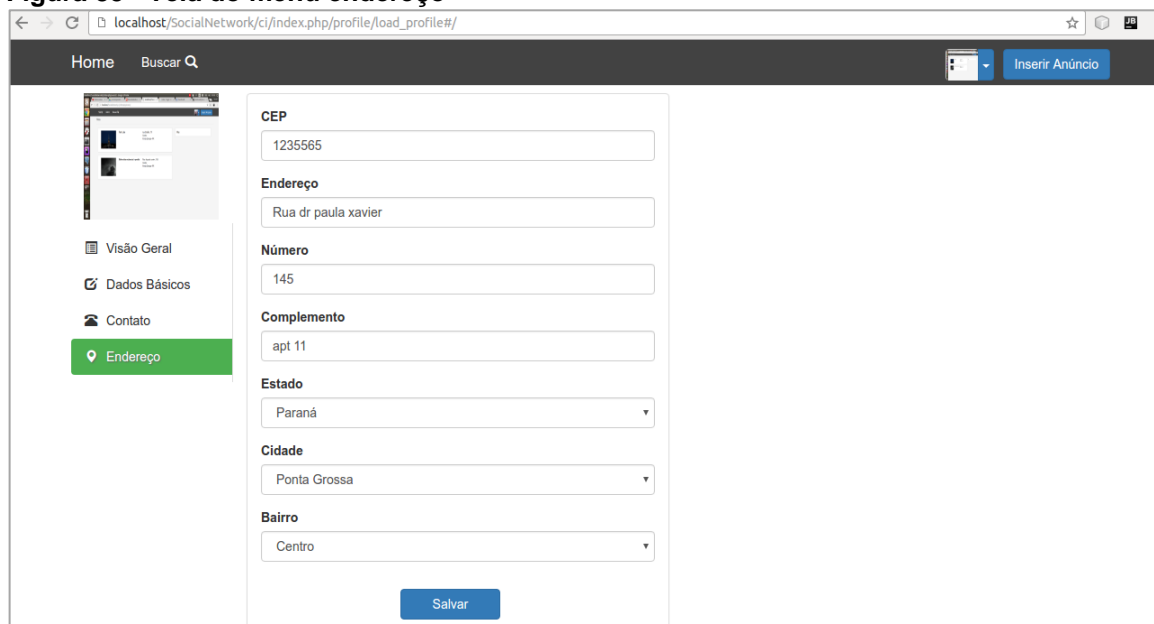
Estado

Cidade

Bairro

Fonte: Autoria própria

A Figura 35 ilustra a tela do menu “Endereço” criado a partir do protótipo da Figura 34.

Figura 35 - Tela do menu endereço

Home Inserir Anúncio

Visão Geral
Dados Básicos
Contato
Endereço

CEP

Endereço

Número Complemento

Estado

Cidade

Bairro

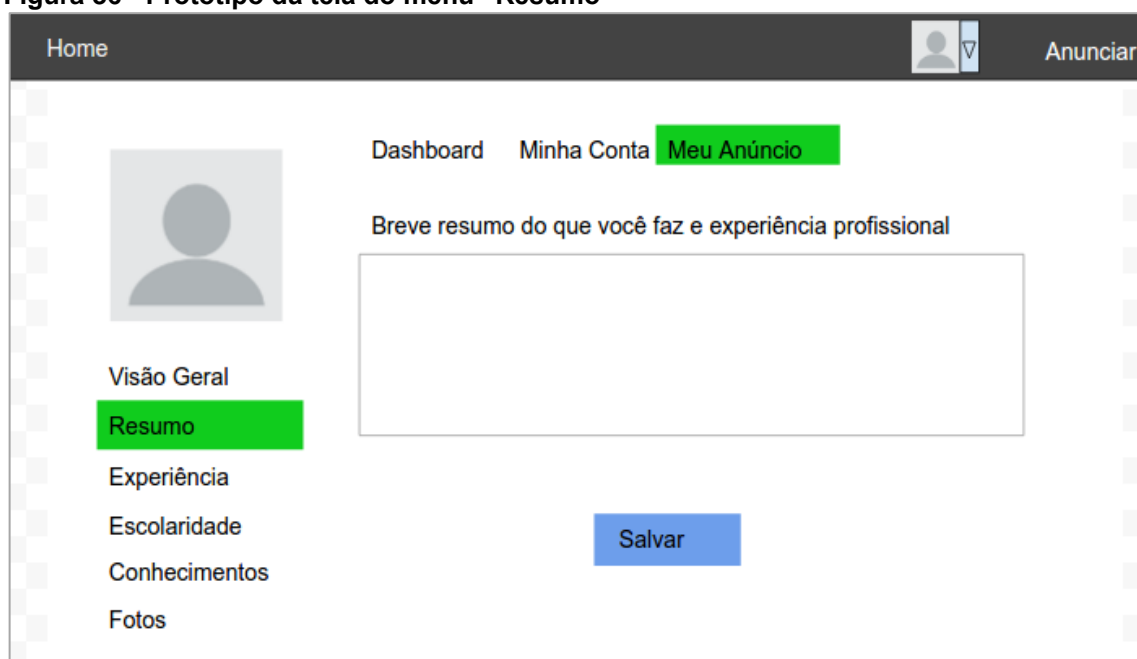
Fonte: Autoria própria

4.9 ABA ANÚNCIO

Outro item acessível no sistema quando um usuário está autenticado é a aba “Anúncio”, nela é possível cadastrar dados referentes a um anúncio de prestação de serviços. O usuário pode inserir título para o anúncio, experiência, escolaridade, serviços que ele presta e fotos.

Na tela do menu “Resumo”, é possível cadastrar título e resumo para o anúncio. O título aparecerá em destaque na lista de anúncios quando alguém estiver procurando um prestador de serviços. E o resumo aparecerá no topo da página do perfil do usuário. Ele pode conter informações como descrição dos serviços prestados e experiência profissional. A Figura 36 mostra o protótipo da tela do menu “Resumo”.

Figura 36 - Protótipo da tela do menu “Resumo”



Fonte: Autoria própria

A Figura 37 ilustra a tela “Título e Resumo” criada com base no protótipo da Figura 36.

Figura 37 - Tela do menu título e resumo

Fonte: Autoria própria

Caso o prestador de serviços não seja uma empresa, ele pode cadastrar uma quantidade indeterminada de experiências, ou seja, ele pode inserir dados referente a trabalhos anteriores. A Figura 38 apresenta o protótipo da tela do menu “Experiência”.

Figura 38 - Protótipo da tela do menu experiência

Fonte: Autoria própria

A Figura 39 apresenta a tela criada a partir do protótipo da tela do menu “Experiência” (Figura 38). Nela é possível inserir informações como: cargo, empresa,

e período de trabalho. O sistema permite o cadastro de uma quantidade indeterminada de experiências.

Figura 39 - Tela do menu experiência

The screenshot shows a web browser window with the URL `localhost/SocialNetwork/ci/index.php/profileAd#`. The page has a navigation bar with 'Home', 'Buscar', and 'Inserir Anúncio'. Below the navigation bar, there are tabs for 'Dashboard', 'Perfil', and 'Anúncio'. The main content area is titled 'Experiência' and contains a form to add a new experience. The form has the following fields and options:

- Cargo:** Eletrecista
- Empresa:** Eletrecidade SA
- Autônomo:**
- Período:** De 10/02/2013 Até 07/05/2016
- Trabalho aqui:**

At the bottom of the form are two buttons: 'Salvar' (blue) and 'Excluir' (red).

Fonte: Autoria própria

O prestador de serviços também pode inserir múltiplas instituições que ele frequentou, tais como: escolas e universidades. A Figura 40 mostra o protótipo da tela escolaridade.

Figura 40 - Protótipo da tela do menu escolaridade

The screenshot shows a web browser window with the URL `localhost/SocialNetwork/ci/index.php/profileAd#`. The page has a navigation bar with 'Home', 'Anunciar', and a user profile icon. Below the navigation bar, there are tabs for 'Dashboard', 'Minha Conta', and 'Meu Anúncio'. The main content area is titled 'Escolaridade' and contains a form to add a new educational institution. The form has the following fields and options:

- Escola:** Escola
- Descrição:** (empty text area)
- Período:** de de Até Até
- Estudo aqui:**
- Universidade/Faculdade:**

At the bottom of the form is a blue 'Editar' button.

Fonte: Autoria própria

A Figura 41 ilustra a tela do menu “Escolaridade” criada a partir do protótipo da Figura 40. Nessa tela o usuário pode inserir instituição, descrição e duração do

curso.

Figura 41 - Tela do menu escolaridade

The screenshot shows a web browser window with the URL `localhost/SocialNetwork/ci/Index.php/profileAd#`. The page has a dark header with 'Home' and a search bar. Below the header, there are navigation tabs: 'Dashboard', 'Perfil', and 'Anúncio'. A sidebar on the left contains menu items: 'Visão Geral', 'Título e Resumo', 'Experiência', 'Escolaridade' (highlighted in green), 'Serviços', and 'Fotos'. The main content area displays a form titled '+ Adicionar Escolaridade'. The form fields are: 'Instituição' (text input with 'UTFPR'), 'Universidade/Faculdade' (checkbox checked), 'Descrição' (text area with 'Engenharia Elétrica'), and 'Período' (date inputs 'De 06/01/2015' and 'Até 10/02/2016' with an 'Estudo aqui' checkbox). At the bottom of the form are 'Salvar' and 'Excluir' buttons.

Fonte: Autoria própria

Para que o anúncio seja incluído nas buscas por prestadores de serviços, é necessário que o usuário informe os serviços que ele presta. A Figura 42 apresenta o protótipo da tela onde os serviços são cadastrados.

Figura 42 - Protótipo da tela de cadastro dos serviços prestados

Fonte: Autoria própria

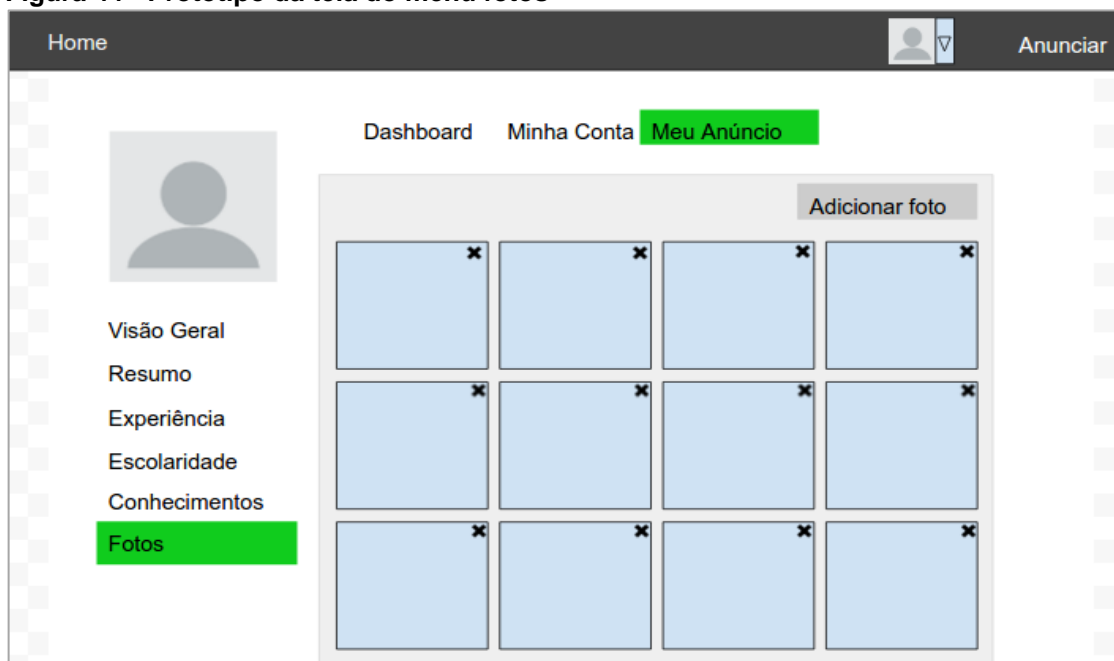
A Figura 43 ilustra a tela desenvolvida a partir do protótipo da Figura 42. Para cadastrar um serviço, o usuário deve selecionar um serviço e categoria, e clicar em “Adicionar”. Esses dados são valores previamente cadastrados no sistema. A exclusão dele é realizada por meio do botão “X” ao lado de cada item incluído.

Figura 43 - Tela do menu Serviços

Fonte: Autoria própria

Há também a possibilidade de inserir fotos no anúncio. Por meio delas, o prestador de serviços pode, por exemplo, adicionar fotos de serviços anteriores realizados por ele. As fotos de um anúncio podem ser visualizadas no perfil do anunciante (botão Fotos). A Figura 44 apresenta o protótipo da tela de fotos de um anúncio.

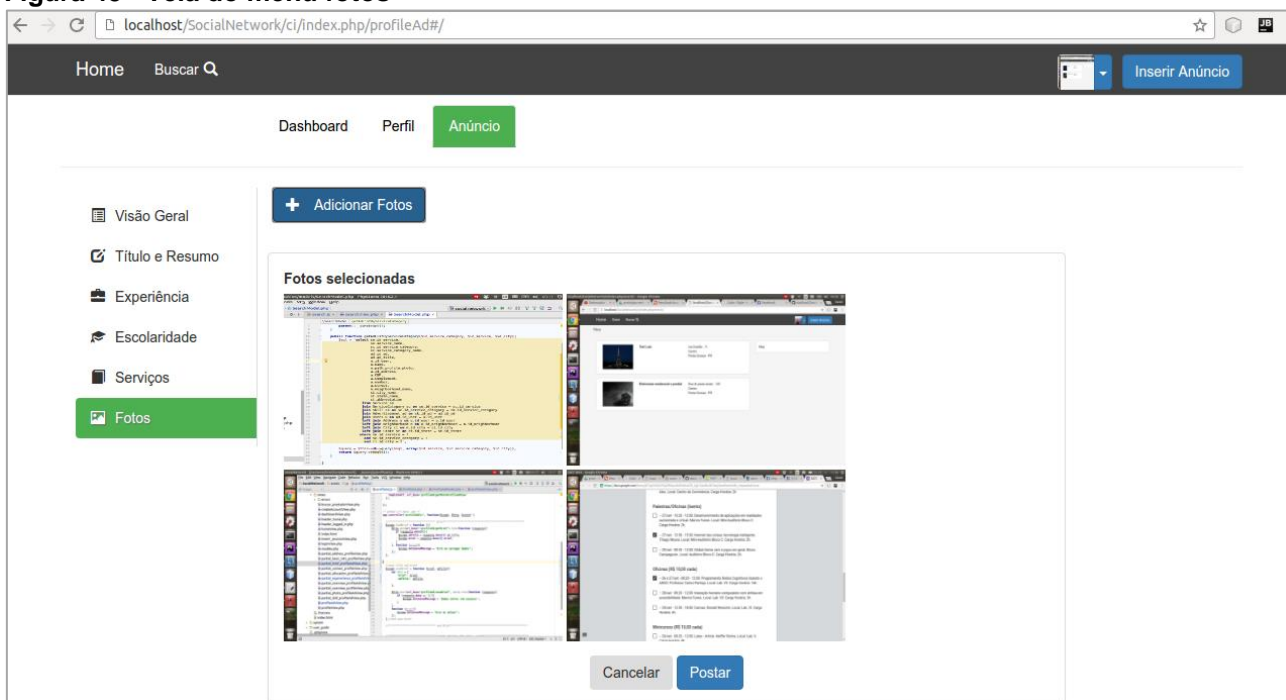
Figura 44 - Protótipo da tela do menu fotos



Fonte: Autoria própria

A Figura 45 apresenta a tela do menu “Fotos” criada com base no protótipo da Figura 44. Para inserir fotos, o usuário deve clicar no botão “Adicionar Fotos”, selecionar uma ou mais por vez e, depois disso o sistema mostrará as fotos selecionadas, e então o usuário poderá cancelar ou clicar no botão “Postar”. A exclusão de uma foto é feita clicando em “remover” quando o mouse for passado por cima da foto escolhida.

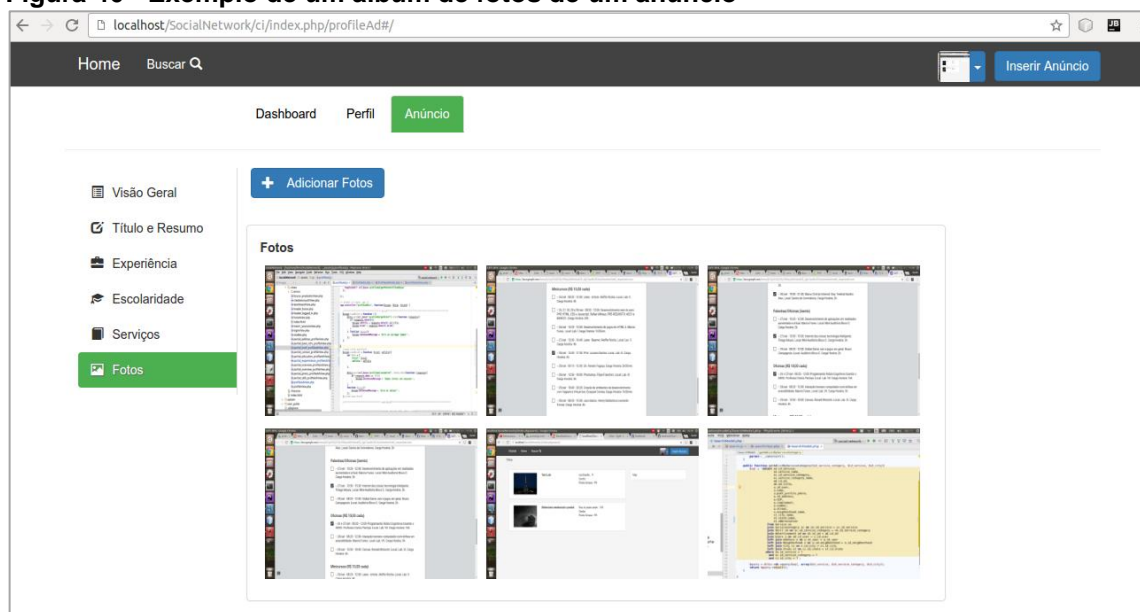
Figura 45 - Tela do menu fotos



Fonte: Autoria própria

A Figura 46 mostra um exemplo de álbum de fotos em anúncio criado por um usuário.

Figura 46 - Exemplo de um álbum de fotos de um anúncio



Fonte: Autoria própria

5 CONCLUSÃO

O sistema desenvolvido nesse trabalho permite que usuários encontrem prestadores de serviços na área de pequenos reparos residenciais de forma rápida, fácil e segura. Eles podem fazer buscas utilizando filtros que retornam anúncios que oferecem exatamente o serviço procurado.

Para facilitar a localização do endereço de um anunciante, foi utilizado o *Google Maps*. Nele é possível visualizar quem está mais próximo do usuário. Isso pode fazer com que o custo por um determinado serviço seja menor, tendo em vista que o deslocamento do prestador de serviços também será menor.

Com a aplicação do algoritmo de ordenação utilizando o intervalo de confiança de Wilson, os primeiros anúncios apresentados aos usuários após uma busca são os que tem as melhores avaliações. Dessa forma é possível identificar os melhores profissionais e aumentar as chances de contratar alguém confiável. Outra forma de garantir uma maior segurança é fornecer mais informações sobre o anunciante. No sistema desenvolvido é possível inserir informações como experiência, escolaridade e fotos, além de permitir a visualização do perfil do anunciante.

Por meio do uso de metodologias como Kanban e XP, pode-se aumentar a qualidade do software desenvolvido, favorecendo dessa forma os usuários do sistema que terão acesso a um produto com menos chances de apresentar falhas.

5.1 TRABALHOS FUTUROS

A seguir são apresentadas algumas sugestões para] trabalhos futuros:

- Implementar uma linha do tempo na página inicial com as últimas atualizações dos usuários;
- Mostrar notificações ao prestador de serviços quando ele receber uma avaliação;
- Permitir que um usuário adicione outro como amigo;
- Implementar um sistema de *chat* para a comunicação entre o cliente e o prestador de serviços;
- Criar anúncios exclusivos para empresas;
- Fazer a validação do contato via mensagem de texto;

- Criar um perfil exclusivo para usuários não anunciantes;
- Mostrar no perfil do usuário as avaliações feitas por ele;
- Criar um sistema de busca em que o usuário digita o serviço e a cidade e o sistema faz sugestões;
- Permitir o pagamento online dos serviços utilizando cartões de débito e/ou crédito; e
- Criar um sistema de avaliação por preço.

REFERÊNCIAS

AKST, Daniel. **Want to Pay Less in Your Online Shopping? Search More.** Disponível em: <<http://www.wsj.com/articles/want-to-pay-less-in-your-online-shopping-search-more-1466172018>> Acesso em: 1 set. 2016.

ANDERSON, David J. **Kanban: Successful Evolutionary Change for Your Technology Business.** Sequim, WA, EUA: Blue Hole Press, 2010.

AngularJS. **HTML enhanced for web apps!**. 2016. Disponível em: <<https://angularjs.org/>> Acesso em: 05 ago. 2016.

BECK, Kent. **Programação extrema explicada: acolha as mudanças.** Porto Alegre: Bookman, 2004.

BOGGS, Wendy; BOGGS, Michael. **Mastering UML com Rational Rose.** Rio de Janeiro, RJ: Alta Books, 2002.

Bootstrap. **Getting started.** 2016. Disponível em: <<http://getbootstrap.com/getting-started/#download>> Acesso em: 05. ago. 2016.

BRANDON, Daniel M. **Software Engineering for Modern Web Applications: Methodologies and Technologies.** Hershey, PA, EUA: IGI Global, 2008.

CETIC. **Pesquisa Sobre o Uso das Tecnologias de Informação e Comunicação nos Domicílios Brasileiros.** São Paulo, 2015. Disponível em: <http://cetic.br/media/docs/publicacoes/2/TIC_Domicilios_2014_livro_eletronico.pdf>. Acesso em: 16 mar. 2016.

CODE IGNITER. **Codelgniter User Guide.** 2016. Disponível em: <http://www.codeigniter.com/user_guide/>. Acesso em: 10 jun. 2016.

Codelgniter 3.0. **Download Codelgniter.** 2016. Disponível em: <<http://www.codeigniter.com/download>>. Acesso em: 27 mar. 2016.

COMER, Douglas. **InternetWorking with TCP/IP: principles, protocols, and architectures.** 4. ed. Upper Saddle River, NJ, EUA: Prentice-Hall, 2000.

DEITEL, P. J.; DEITEL, H. M. AJAX, **Rich Internet Applications and Web Development for Programmers**. Pearson, 2008.

FACEBOOK PARA EMPRESAS. **45% da População Brasileira Acessa o Facebook Mensalmente**. 2015. Disponível em: <<https://www.facebook.com/business/news/BR-45-da-populacao-brasileira-acessa-o-facebook-pelo-menos-uma-vez-ao-mes>>. Acesso em: 20 mar. 2016.

FIELDING, R et al. **Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing**. 2014. Disponível em: <<https://tools.ietf.org/html/rfc7230>>. Acesso em: 09 mar. 2016.

FORBES. **A Comparative Look At The Valuation Of Amazon, Alibaba and eBay**. 2016. Disponível em: <<http://www.forbes.com/sites/greatspeculations/2015/10/09/a-comparative-look-at-the-valuation-of-amazon-alibaba-and-ebay/#5bdcb796747e>> Acesso em: 1 set. 2016.

FRANÇA, Flávio A. de. **Tutorial – Entendendo Java para Web (Parte 1)**. 2010. Disponível em: <<https://flavioaf.wordpress.com/tag/mvc/>>. Acesso em: 10 jun. 2016.

GETNIJAS. **Encontre Profissionais para Facilitar sua Vida**. 2016. Disponível em: <<https://www.getninjas.com.br/>> Acesso em: 12 out. 2016.

Google Drive. **A safe place for all your files**. 2016. Disponível em: <<https://www.google.com/drive/>> Acesso em: 11 ago. 2016

GUIDO, Rus. **How to Build a 5 Star Rating System with Wilson Interval in MySQL**. 2015. Disponível em: <<http://www.onextrapixel.com/2015/09/24/how-to-build-a-5-star-rating-system-with-wilson-interval-in-mysql/>> Acesso em: 21 set. 2016.

HAMMARBERG, Marcus; SUNDÉN, Joakim. **Kanban in Action**. Shelter Island, NY, EUA: Manning Publications Co, 2014.

HARTLEY, Darin. **10 Steps to Successful Social Networking for Business**. Danvers, MA, EUA: ASTD Press, 2010.

IBGE. **Pesquisa Nacional por Amostra de domicílios**. Rio de Janeiro, 2015. Disponível em: <<http://biblioteca.ibge.gov.br/visualizacao/livros/liv94414.pdf>>. Acesso em: 16 mar. 2016.

Jasmine. **Downloads**. 2016. Disponível em: <<https://github.com/jasmine/jasmine/releases>> Acesso em: 1 set. 2016.

KAWASAKI, Guy; FITZPATRICK, Peg. **The Art of Social Media: Power Tips for Power Users**. New York, NY, EUA: Portfolio, 2014.

LIMA, Adilson da Silva. **UML 2.2: Do Requisito à Solução**. 4. ed. São Paulo: Érica, 2009.

MDN. **Accept-Ranges**. Disponível em <<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Accept-Ranges>> Acesso em 20 out. 2016.

MDN. **Connection**. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Connection>> Acesso em: 20 out. 2016.

MILLER, Evan. **How Not To Sort By Average Rating**. 2009. Disponível em: <<http://www.evanmiller.org/how-not-to-sort-by-average-rating.html>>. Acesso em: 14 set. 2016.

MOURA, Reinaldo A. **Kanban: a simplicidade do controle da produção**. 5. ed. São Paulo: IMAM, 1999.

OLX. **Regras**. 2016. Disponível em: <<https://olxbrasil.zendesk.com/hc/pt-br/articles/211375529-Regras>> Acesso em: 12 out. 2016.

PhpStorm. **Download PhpStorm**. 2016. Disponível em: <<https://www.jetbrains.com/phpstorm/download/#section=linux-version>> Acesso em: 10 ago. 2016.

PETANJO. **PetAnjo**. 2016. Disponível em: <<https://petanjo.com/>> Acesso em: 10 out. 2016.

PIVOTAL TRACKER. **Get Started**. 2016. Disponível em: <<https://www.pivotaltracker.com/signup/new>> Acesso em: 11 jul. 2016.

PostgreSQL. **Concurrency Control**. 2016. Disponível em: <<https://www.postgresql.org/docs/current/static/mvcc-intro.html>> Acesso em: 01 out. 2016.

PostgreSQL. **Downloads**. 2016. Disponível em: <<https://www.postgresql.org/download/>> Acesso em: 15 ago. 2016.

PRESSMAN, Roger S. **Engenharia de Software: uma Abordagem Profissional**. 7. ed. New York, NY, EUA: McGraw-Hill, 2011.

PRESSMAN, Roger S.; LOWE, David. **Web Engineering: a practitioners's Approach**. New York, NY, EUA: McGraw-Hill, 2009.

RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady. **The Unified Modeling Language Reference Manual**. 2. ed. Boston, MA, EUA: Pearson, 2004.

RYAN, K. Peter. **Digital and Information Literacy: Social Networking**. New York, NY, EUA: Rosen Central, 2011.

SCHWABER, Ken; SUTHERLAND, Jeff. **Guia do Scrum**. 2013. Disponível em: <<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Acesso em: 08 jun. 2016.

SEBRAE. **Como Montar um Serviço de Reparos Residenciais**. 2012. Disponível em: <<http://www.sebrae.com.br/sites/PortalSebrae/ideias/Como-montar-um-servi%C3%A7o-de-reparos-residenciais>>. Acesso em: 31 mar. 2016.

SOMMERVILLE, Ian. **Engenharia de software**. 8. ed. São Paulo, SP: Pearson Addison-Wesley, 2007.

StarUML 2.6.0. **Download**. 2016. Disponível em: <<http://staruml.io/download>>. Acesso em: 23 mar. 2016.

STELLMAN, Andrew; GREENE, Jennifer. **Learning Agile**. Sebastopol, CA, EUA: O'Reilly Media, 2015.

TANENBAUM, Andrew S.; STEEN, Maarten Van. **Distributed Systems: Principles and Paradigms**. 2. ed. Upper Saddle River, NJ, EUA: Pearson, 2007.

TANENBAUM, Andrew S.; WETHERALL, David J. **Computer Networks**. 5. ed. Pearson, 2010.

TELES, Vinícius Manhães. **Extreme programming: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade**. São Paulo, SP: Novatec, 2006.

TRELLO. **Criar uma Conta Trello**. 2016. Disponível em: <<https://trello.com/signup>> Acesso em: 10 jul. 2016.

W3C. **Method Definitions**. 2016. Disponível em: <<https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>>. Acesso em: 08 mar. 2016.

ZHANG, Dell et al. **How to Count Thumb-Ups and Thumb-Downs: User-Rating based Ranking of Items from an Axiomatic Perspective**. 2011. Disponível em: <http://www.dcs.bbk.ac.uk/~dell/publications/dellzhang_ictir2011.pdf> Acesso em: 14 set. 2016.