

CENTRO PAULA SOUZA
FATEC OURINHOS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

NEEMIAS RAMOS FERREIRA

**ESTUDO DA CONSTRUÇÃO DE UM SISTEMA DE
INFORMAÇÃO USANDO TECNOLOGIAS RESPONSIVAS EM
ARQUITETURA MVC**

OURINHOS – SP

2017

NEEMIAS RAMOS FERREIRA

**ESTUDO DA CONSTRUÇÃO DE UM SISTEMA DE
INFORMAÇÃO USANDO TECNOLOGIAS RESPONSIVAS EM
ARQUITETURA MVC**

Trabalho de Conclusão de Curso de Graduação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Faculdade de Tecnologia, Campus Ourinhos, como requisito parcial para obtenção do título de Tecnólogo.

Orientador(a): Prof. Me. João Mauricio Hypólito

OURINHOS – SP

2017

DEDICATÓRIA

Dedico este trabalho de conclusão de curso aos meus pais que me deram forças e apoiaram durante o decorrer da graduação e projeto, dedico também a todos os professores em especial ao orientador João Mauricio Hypólito, que deu as orientações corretas para que eu pudesse concluir este projeto.

Neemias Ramos Ferreira

RESUMO

Este trabalho apresenta a pesquisa e desenvolvimento de um sistema *web* para conectar pessoas que necessitam solucionar um problema com pessoas que solucionam o mesmo por meio de prestação de serviço. Para tal contexto foi desenvolvido um sistema *web* responsivo com arquitetura de software MVC (*Model View Controller*). Com esta arquitetura, a aplicação no futuro, será possível alterar, fazer manutenção e aumentar o sistema com maior facilidade e organização. No desenvolvimento desta aplicação foram utilizadas diversas ferramentas de modelagem de *software* e algumas linguagens de programação. A utilização do sistema será como forma de divulgar um prestador de serviços, sendo assim, outros usuários poderão se cadastrar e/ou contratar, com base nas informações de contato.

Palavras-Chave: Sistema, *Web*, MVC.

ABSTRACT

This work presents the research and development of a web system to connect people who need to solve a problem with people who solve it through the provision of service. For this context a responsive web system was developed with MVC (Model View Controller) software architecture. With this architecture, the application in the future, it will be possible to change, maintain and increase the system with greater ease and organization. In the development of this application several software modeling tools and some programming languages were used. The use of the system will be as a way to disclose a service provider, so other users can register and / or contract, based on the contact information.

Keywords: **System, Web, MVC.**

LISTA DE FIGURAS

Figura 1 – The MVC Architecture	8
Figura 2 – Aplicativo web PHPMyAdmin	10
Figura 3 - Painel de controle do Xampp	11
Figura 4 - Software de modelagem ASTAH.....	13
Figura 5 - Linguagem de Marcação HTML5	14
Figura 6 - Estrutura Visual exemplificada na Figura 2 em formato de código HTML	15
Figura 7 - Diretório do Bootstrap.....	16
Figura 8 - Exemplo de nomenclaturas do Framework Bootstrap.....	16
Figura 9 - Opções de grid do Bootstrap.....	17
Figura 10 - Programação do botão “Enviar Mensagem”.....	18
Figura 11 - Diagrama de classes. Cadastrar prestador de serviço.	19
Figura 12 - Caso de uso - Prestador de Serviço	20
Figura 13 - Diagrama de atividade - login	21
Figura 14 - Diagrama de sequência - Cliente	22
Figura 15 - Tela Inicial	24
Figura 16 - Tela, Cadastro de Prestador de Serviço	24
Figura 17 - Tela inicial responsiva, com restrição	25
Figura 18 - Tela, cadastro de Prestador de Serviço "bloqueada".....	26
Figura 19 – Programação para exibir mensagem em tela menor que 768 pixels.....	26
Figura 20 - Tabela de classes disponíveis	27
Figura 21 - Clique no botão "Enviar msg" do resultado de busca	28
Figura 22 - Pagina do WhatsApp Web.....	29
Figura 23 - Pagina de autenticação via QR Code do WhatsApp Web	29
Figura 24 - Conversa estabelecida no WhatsApp Web via API do WhatsApp.....	30
Figura 25 - Caso de uso Cliente	37
Figura 26 - Caso de uso Prestador de Serviço.....	39
Figura 27 - Caso de uso Administrador	41
Figura 28 - Modelo entidade relacionamento	42
Figura 29 - Diagrama de atividade - Login	43
Figura 30 - Diagrama de atividade - Buscar	44
Figura 31 - Diagrama de atividade - Entrar em contato	45
Figura 32 - Diagrama de atividade - Atualizar informações.....	46

Figura 33 - Diagrama de sequência – Cliente	47
Figura 34 - Diagrama de sequência - Prestador de Serviço	47
Figura 35 - Diagrama de sequência - Administrador	48

LISTA DE TABELAS

Tabela 1 - Descrição de cenário - Cliente	38
Tabela 2 - Descrição do cenário - Prestador de Serviço	40
Tabela 3 - Descrição do cenário - Administrador	41

LISTA DE ABREVIATURAS

API: Application Programming Interface;

CSS: Cascading Style Sheets;

HTML: *Hyper Text Markup Language*;

IBGE: Instituto Brasileiro de Geografia e Estatística;

IDE: *Integrated Development Environment*.

IHC: Interação Humano-Computador;

MVC: *Model View Controller*;

PHP: *Hypertext PreProcessor*;

PNAD: Pesquisa Nacional por Amostra de Domicílios;

QR CODE: *Quick Response Code*;

RF: Requisitos Funcionais;

SGBD: Sistema de Gerenciamento de Banco de Dados;

SQL: *Structure Query Language*;

TIC: Tecnologia da Informação e Comunicação;

UML: *Unified Modeling Language*;

WWW: *World Wide Web*;

SUMÁRIO

1 INTRODUÇÃO.....	4
2 REFERENCIAL TEÓRICO.....	7
2.1 Descrição das tecnologias propostas para o desenvolvimento do sistema	7
2.1.1 Arquitetura MVC	7
2.1.2 HTML 5.....	9
2.1.3 PHP	9
2.1.4 MySQL.....	10
2.1.5 XAMPP	10
2.1.6 Design responsivo	11
2.1.7 Frameworks	12
2.1.7.1 Twitter Bootstrap.....	12
2.1.7.2 JQuery	13
2.1.8 Astah	13
3 MÉTODO.....	14
3.1 Front-end.....	14
3.2 Back-end	17
3.3 Servidor	18
3.4 Modelagem.....	18
3.4.1 Diagramas	18
4 ANÁLISE DE RESULTADO	23
5 CONCLUSÃO	31
6 REFERÊNCIAS	32
APÊNDICE A – Requisitos Funcionais e Não Funcionais	34
APÊNDICE B – Diagramas de Caso de Uso.....	37

1 INTRODUÇÃO

Segundo notícia do G1¹, a taxa de desemprego subiu para 13,7% no trimestre de janeiro a março, segundo dados divulgados na sexta-feira (31/04/17) pelo Instituto Brasileiro de Geografia e Estatística (IBGE), por meio da Pesquisa Nacional por Amostra de Domicílios (Pnad) contínua. De acordo com o IBGE, essa foi a maior taxa de desocupação da série histórica, iniciada em 2012. No 1º trimestre, o Brasil tinha 14,2 milhões de desempregados, também batendo recorde da série histórica. O número de trabalhadores por conta própria (22,1 milhões de pessoas) registrou estabilidade em relação ao trimestre anterior (outubro a dezembro de 2016) (SILVEIRA, 2017).

Tema

Este trabalho tem como principal proposta desenvolver um estudo sobre tecnologias da informação e comunicação (TIC) que facilitem o desenvolvimento de um sistema de informação na plataforma *World Wide Web* responsiva que facilite o contato entre prestador de serviço e cliente. Este sistema ainda terá uma funcionalidade de pesquisa sobre os serviços que são feitos pelos prestadores de serviços.

Utilizar Tecnologia da Informação e Comunicação para fazer a ponte entre prestador de serviço e seu potencial cliente.

Aplicando, no desenvolvimento do sistema, tecnologia responsiva, um servidor de pagina juntamente com uma linguagem de programação CGI e um banco de dados para o armazenamento das informações. E assim possibilitar ao cliente fazer buscas, por prestadores de serviço cadastrados no sistema, e selecionar a melhor opção para solucionar determinado problema.

¹ Mais Informações consultar: <http://g1.globo.com/economia/noticia/desemprego-fica-em-137-no-1-trimestre-de-2017.ghtml>

Problema

O desenvolvimento de um sistema *web* para organização de prestadores de serviços que oferecem soluções para problemas do cotidiano de uma pessoa ou de uma empresa, entendendo-se como um sistema mediador entre a oferta e a procura destes prestadores de serviços.

Hipótese

A implementação de um sistema *web*, funcionando entre a oferta de serviços e procura pelos mesmos, é capaz de facilitar o contato entre prestador e clientes. Isso pode acontecer de modo rápido, em uma interface limpa e simples.

Objetivos

Estudar e caracterizar como ferramentas de TIC podem dar suporte ao desenvolvimento de um sistema *web* responsivo em plataforma *Model View Controler* (MVC) que facilite o contato entre prestadores de serviços e seus potenciais clientes.

Objetivos específicos

Destaca-se para o desenvolvimento deste trabalho, Objetivos Específicos como:

- Identificar e definir os essenciais requisitos para o desenvolvimento e implementação do sistema *web* responsivo.
- Documentar e propagar o conhecimento, a linguagem e *frameworks* utilizados neste projeto.
- Desenvolver as funcionalidades que destaquem o funcionamento de sistemas responsivos.
- Ao desenvolver o sistema, seguir as etapas do método de desenvolvimento escolhido.

Justificativa

Estreitar o contato entre prestadores de serviços e seus possíveis clientes, esses profissionais buscam maneiras de expor seu trabalho e aumentar sua visibilidade, seja pelo preço, pela qualidade ou pelo diferencial.

Neste contexto, o cliente em potencial leva como critério de escolha, o preço, qualidade e o diferencial ao pesquisar sobre prestadores de serviço.

Cita-se como alguns sistemas já conhecidos que seguem a mesma linha de raciocínio e trabalham com profissionais e seus potenciais clientes, os *websites* BICOS (Disponível em: www.bicos.com.br), GetNinjas (Disponível em: www.getninjas.com.br) e Bougue (Disponível em: www.bougue.com.br).

Sendo estes mediadores independentes tanto para os profissionais quanto para os clientes, todas as ações realizadas não envolvem interferência direta do sistema, sendo possível encontrar o profissional ou o mesmo encontrar um “problema” para solucionar.

O sistema proposto neste trabalho funciona como uma rede de contatos possibilitando o relacionamento direto entre prestador de serviço e cliente.

Os cadastros são independentes e customizados de acordo com os usuários (profissionais ou clientes). O sistema disponibiliza uma página em que todos os prestadores de serviço são listados com informações detalhadas de cada um. O cliente que tem interesse em contratar um serviço, entra em contato com o prestador, por meio das informações expostas pelo mesmo, esclarecer suas dúvidas e possivelmente o contratar para solucionar algum problema.

2 REFERENCIAL TEÓRICO

Este capítulo tem como conteúdo, descrever assuntos relacionados à tecnologia que são utilizadas neste trabalho, bem como informações para expor a importância da arquitetura MVC em um sistema web responsivo.

2.1 Descrição das tecnologias propostas para o desenvolvimento do sistema

Será utilizado uma arquitetura de software e ferramentas específicas para realizar este projeto, tais como: a arquitetura MVC para manter o desenvolvimento organizado e separado em camadas; Bootstrap, um framework para o desenvolvimento de design responsivo; PHP (*Hypertext PreProcessor*) como linguagem de programação, CGI (*Common Gateway Interface*) para fazer as conexões necessárias com o banco de dados MySQL. Para o servidor de páginas será utilizado o XAMPP. O Astah será o programa escolhido para desenvolver a modelagem do sistema. A seguir, de uma forma mais detalhada, a apresentação de cada ferramenta:

2.1.1 Arquitetura MVC

Inventado para ser utilizado na linguagem Smalltalk, por Trygve Reenskaug nos anos 70, o padrão MVC foi se popularizar na web em 2003, com o surgimento do Ruby on Rails (PALERMO, et. al., 2010, p. 7).

Segundo a definição de Gonçalves (2007), a arquitetura MVC é um conceito (paradigma) de desenvolvimento e *design* que tenta separar uma aplicação em três partes distintas. Uma parte, a *model*, está relacionada ao trabalho atual que a aplicação administra, outra parte, a *view*, está relacionada a exibir os dados ou informações dessa aplicação e a terceira parte, *controller*, em coordenar a *model* e a *view* exibindo a interface correta ou executando algum trabalho que a aplicação precisa completar.

Para Santos (2010) o MVC auxilia os desenvolvedores a construir aplicações separando seus principais componentes, a manipulação e armazenamento dos dados, as funções que irão trabalhar com as entradas dos dados e a visualização do

usuário. A arquitetura MVC especifica onde cada tipo de lógica deve estar localizada na aplicação.

MVC é um dos padrões de desenvolvimento mais conhecidos do mercado. Basicamente o projeto que segue tal padrão, tem suas classes separada por três pacotes de responsabilidade (OGLIO,2016).

A principal justificativa de adotarmos o padrão MVC é não misturarmos em uma mesma classe responsabilidades diferentes (OGLIO,2016).

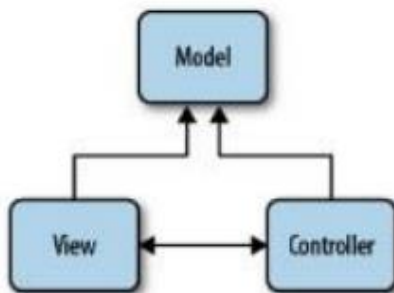
Model (modelo). A classe modelo representa as informações referentes ao banco de dados. Essa classe tem como principal objetivo manter as informações de negócio, regras de negócio do projeto e implementar métodos representando-as (FIGUEIREDO, 2015).

View (visualização). A classe de visualização representa a interface gráfica que interage com o usuário. Esta classe é responsável pela obtenção de dados ao usuário. Não é dever da classe de visualização conter lógica de regra de negócios (FIGUEIREDO, 2015).

Controller (controle). A classe de controle recebe dados das ações do usuário, diretamente da tela de visualização (*View*). Assim, aciona o modelo (*Model*), para pegar os resultados necessários e fazer as validações. Então a *Controller* retorna esses resultados para a *View* apresentar para o usuário (FIGUEIREDO, 2015).

A Figura 1 mostra como o padrão MVC trabalha e se comunica entre as camadas modelo, visão e controle.

Figura 1 – *The MVC Architecture*



Fonte: Chadwick, Snyder, Panda (2012)

2.1.2 HTML 5

HTML 5 é uma linguagem para marcação de hipertexto, em inglês *HyperText Markup Language* (HTML), utilizada para criar *sites*. Segundo Silva (2011), pode-se resumir hipertexto como todo o conteúdo inserido em um documento para a web e que tem como principal característica a possibilidade de se interligar a outros documentos da web. O que torna possível a construção de hipertextos são os links, presentes nas páginas de todos os sites que se visita quando a internet é utilizada.

2.1.3 PHP

O PHP (2017) é uma linguagem de *script open source*, muito utilizada atualmente, é mais focada para o desenvolvimento web, podendo também ser utilizado juntamente com HTML.

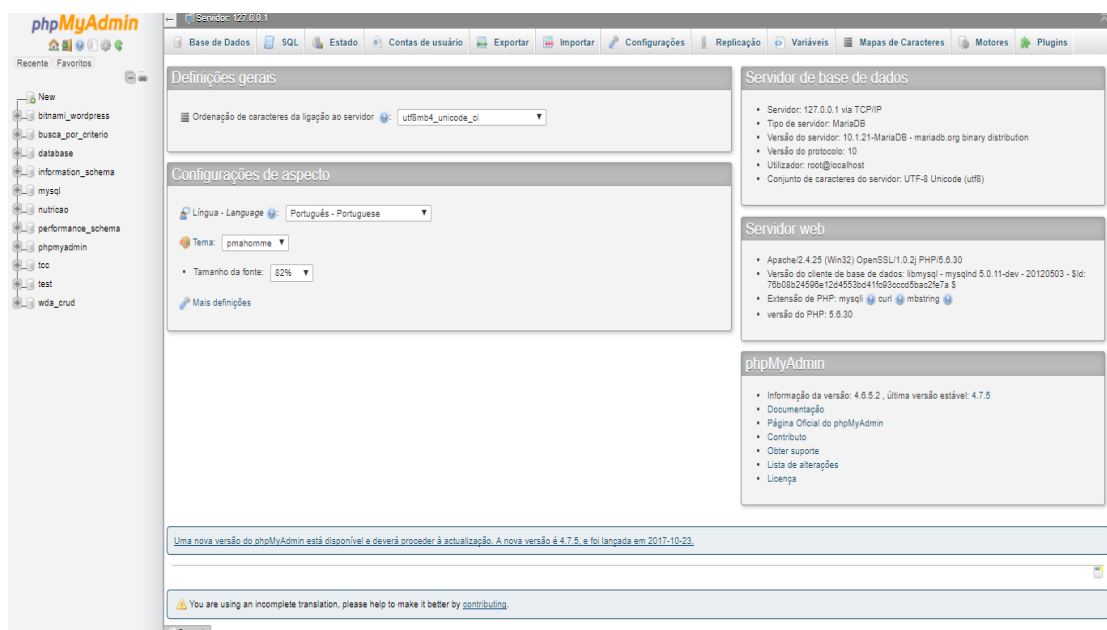
Um dos pontos positivos de se utilizar o PHP no desenvolvimento é a simplicidade, até os mais iniciantes podem utiliza-la, mas também oferece recurso mais avançados para desenvolvedores mais experientes.

Para Lopes (2007) esta linguagem surgiu durante o outono de 1994 por Rasmus Lerdorf, inicialmente seu intuito era para criação de páginas pessoais. Em 1996 a linguagem de programação PHP já estava em aproximadamente 15.000 sites ao redor do mundo, no ano seguinte, 1997, estava presente em cerca de 50.000.

2.1.4 MySQL

Para Ricardo (2013) o MySQL é um sistema gerenciador de banco de dados relacional, e que utiliza da linguagem *Structured Query Language* (SQL), traduzindo, Linguagem de Consulta Estruturada). Também oferece suporte a multiusuários e multitarefas. Este Sistema de Gerenciamento de Banco de Dados (SGBD) foi desenvolvido para trabalhar com projetos de pequeno e médio porte, pois possui uma capacidade aproximada de cem milhões de registros em cada tabela.

Figura 2 – Aplicativo web PHPMYAdmin

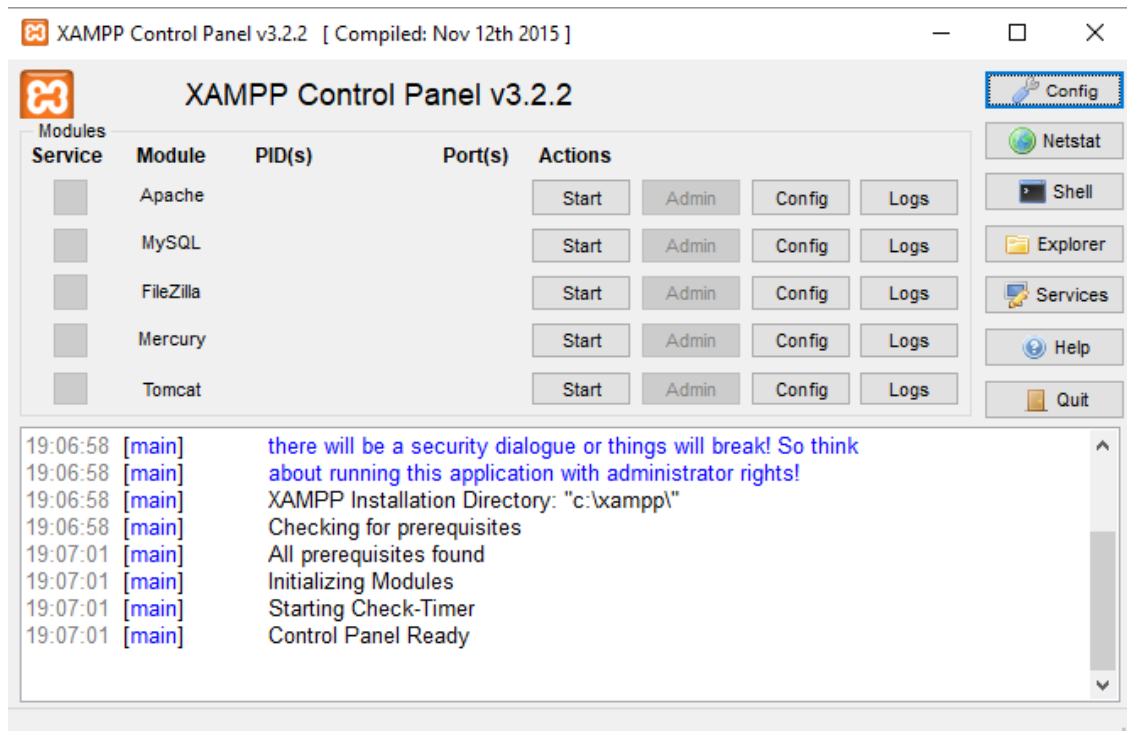


Fonte: (PHPMYADMIN, 2017).

2.1.5 XAMPP

Atualmente existem pacotes de instalação de servidores, com todos os recursos necessários para o desenvolvimento web. Um destes pacotes é o XAMPP, ele poupa tempo do desenvolvedor em configurar todo o ambiente de desenvolvimento de sua máquina, trazendo tudo em um único pacote, permitindo uma instalação única (XAMPP, 2017).

Figura 3 - Painel de controle do Xampp



Fonte: (XAMPP, 2017).

2.1.6 Design responsivo

O princípio de design responsivo, em desenvolvimento para web tem como objetivo adaptar o layout das páginas a qualquer dispositivo, resolução e tela, para garantir ao usuário a melhor experiência possível, possibilitando a leitura e navegação amigável, sem comprometer o conteúdo da página. Silva (2014) explica o conceito de design responsivo:

“Antes de qualquer coisa, é necessário que fique muito claro que design responsivo não diz respeito simplesmente e somente à adaptação do layout ao tamanho da tela. Vai muito além disso, pois o conceito de design responsivo na sua forma ampla deve ser entendido como design capaz de responder às características do dispositivo ao qual é servido. Responder, neste contexto, tem sentido de movimentar-se expandindo e contraindo. Em outras palavras, o design responsivo ou layout responsivo expande e contrai com a finalidade de se acomodar de maneira usável e acessível à área onde é visitado ou, mais genericamente, ao contexto onde é renderizado, seja um smartphone, um tablet, um leitor de tela, um mecanismo de busca etc”. (*ibidem*, p.35).

2.1.7 Frameworks

Se encontra diversas definições para frameworks na literatura.

Para Minetto (2007) um framework de desenvolvimento é uma “base” que se pode desenvolver algo maior ou mais específico. É uma coleção de código fonte, classes, funções, técnicas e metodologias que facilitam o desenvolvimento de novo softwares.

Já segundo Buschmann et all. (1996), um framework é definido como um software parcialmente completo projetado para ser instanciado. O framework define uma arquitetura para uma família de subsistemas e oferece os construtores básicos para criá-los. Também são explicados os lugares ou pontos de extensão (*hot-spots*) nas quais adaptações do código para um funcionamento específico de certos módulos devem ser feitas.

2.1.7.1 Twitter Bootstrap

O principal ponto do uso deste framework, é o suporte a HTML e *Cascading Style Sheets* (CSS), de forma responsiva, porém, também é possível usar nele tecnologias como JavaScript.

Bootstrap é um framework *front-end* desenvolvido pela empresa TWITTER, para facilitar a vida dos desenvolvedores web. Uma de suas maiores vantagens é ter a tecnologia *mobile* (responsivo) (COSTA, 2014).

SILVA (2015) diz que Bootstrap é o mais popular framework JavaScript, HTML e CSS para desenvolvimento de sites e aplicações web responsivas e alinhadas com a filosofia *mobile first*. Torna o desenvolvimento *front-end* muito mais rápido e fácil. Indicado para desenvolvedores de todos os níveis de conhecimento, dispositivos de todos os tipos e projetos de todos os tamanhos.

2.1.7.2 JQuery

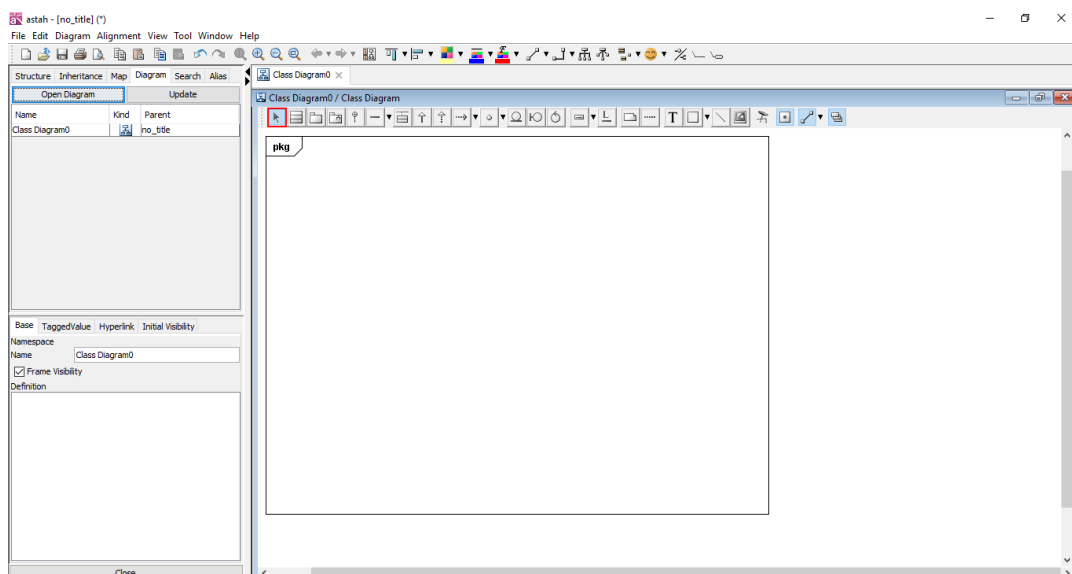
JQuery é um framework JavaScript, em sua documentação oficial, é descrito como uma opção rápida para desenvolvimento web, pequena e rica em recursos JavaScript. Isso torna as tarefas diárias mais fáceis como, por exemplo, passagem de HTML de documentos e manipulação, tratamento de eventos, animação e requisições AJAX. Com uma combinação de versatilidade e capacidade de extensão, jQuery mudou a maneira que milhões de pessoas escrevem JavaScript (JQUERY, 2016).

2.1.8 Astah

Astah é uma ferramenta de modelagem de dados, e diagramas UML – *Unified Modeling Language*. Segundo (CARNEIRO, 2015).

Utilizado para modelagem UML (*Unified Modeling Language*), podendo desenhar diagramas de funcionamento de um sistema a ser criado, como por exemplo diagramas de classe, diagrama de caso de uso, diagramas de sequência, componentes, estrutura de composição, objetos, entre outros. Além da modelagem dos diagramas, a ferramenta oferece ajustes de alinhamento e tamanho dos elementos gráficos, impressão e exportação das imagens.

Figura 4 - Software de modelagem ASTAH



Fonte: (ASTAH, 2016).

3 MÉTODO

Para o desenvolvimento deste trabalho foram realizadas pesquisas bibliográficas específicas, com o objetivo de definir, as estratégias e a maneira de aplicar a tecnologia da informação na hipótese de solução deste trabalho.

Com linguagens de programação *front-end* e *back-end* mais um framework para design responsivo e com um servidor de página, foi possível desenvolver este projeto, descrito abaixo como cada recurso tecnológico foi utilizado.

3.1 Front-end

Para o desenvolvimento do *front-end* do sistema foram utilizadas linguagem de marcação, estilo e de script/programação, exemplificadas a seguir.

HTML 5

Linguagem de marcação usada para descrever as páginas do sistema web. Essa linguagem, como podemos observar na Figura 5, cria uma estrutura visual que podemos observar na Figura 6.

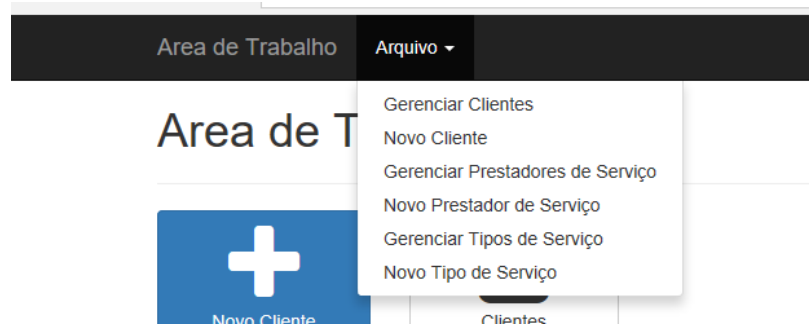
Figura 5 - Linguagem de Marcação HTML5

```
21 </head>
22 <body>
23 <nav class="navbar navbar-fixed-top navbar-inverse navbar-transparente">
```

Fonte: Elaborado pelo próprio autor, 2017.

Na linha 23 vemos um elemento HTML5 de navegação `<nav>`. TEOLI (2014) explica que essa *tag* HTML representa uma seção de uma página que aponta para outras páginas ou para outras áreas da página, ou seja, uma seção com links de navegação.

Figura 6 - Estrutura Visual exemplificada na Figura 5 em formato de código HTML



Fonte: Elaborado pelo próprio autor, 2017.

CSS

Linguagem de estilo usada para trabalhar com a parte visual das páginas web. No qual neste projeto foi escolhido um arquivo pré-compilado do Bootstrap (bootstrap.min.css).

Bootstrap

Framework de código aberto, conta com um conjunto de ferramentas criadas para facilitar o desenvolvimento de sites e sistemas web.

O principal ponto para o uso do Bootstrap, é o suporte a HTML e CSS, de forma responsiva, com arquivos pré-compilados, a Figura 7 ilustra o diretório do Bootstrap após feito o download.

Figura 7 - Diretório do Bootstrap



Fonte: (GETBOOTSTRAP, 2017).

Para usar o Framework, é necessário inserir nomenclaturas específicas, do framework, nas *tags* HTML, como podemos ver na Figura 8.

Figura 8 - Exemplo de nomenclaturas do Framework Bootstrap

```

7  <header>
8
9  <div class="row">
10 <div class="col-sm-6">
11
12 <div class="col-sm-6 text-right h2">
13 <a class="btn btn-primary" href="add.php"><i class="fa fa-plus"></i> Novo Cliente</a>
14 <a class="btn btn-default" href="index.php"><i class="fa fa-refresh"></i> Atualizar</a>
15
16 </div>
17 </div>
18 </header>
19

```

Fonte: Elaborado pelo próprio autor, 2017.

Nomenclaturas como “*col-sm*”, na linha 10 e 13 ainda na figura 8, faz parte de um sistema de grid do Bootstrap que trabalha através de múltiplos dispositivos com uma tabela prática, Figura 9.

Figura 9 - Opções de grid do Bootstrap

Opções de grid

Veja como os aspectos do sistema de grid do Bootstrap trabalha através de múltiplos dispositivos com uma tabela prática.

	Dispositivos extra pequenos Telefones (<768px)	Dispositivos pequenos Tablets (≥768px)	Dispositivos médios Desktops (≥992px)	Dispositivos grandes Desktops (≥1200px)
Comportamento do grid	Horizontal em todos os momentos	Colapsa ao princípio, horizontal acima de breakpoints		
Largura do container	nenhum (auto)	750px	970px	1170px
Prefixo de classe	.col-xs-	.col-sm-	.col-md-	.col-lg-
número de colunas	12			
Largura da coluna	Auto	~62px	~81px	~97px
Largura da calha	30px (15px em cada lado da coluna)			
Encaixável	Sim			
Distância	Sim			
Ordenação de colunas	Sim			

Fonte: (GETBOOTSTRAP, 2017).

Dessa maneira o framework Bootstrap trabalha para redimensionar, *tags* HTML, de acordo com o parâmetro de entrada.

3.2 Back-end

PHP

Para o desenvolvimento do *back-end* do sistema e implementação das regras de negócio, foi utilizada linguagem de programação PHP.

De acordo com o site php.net ²

“Você precisa de três coisas para isto funcionar. O interpretador do PHP(CGI ou módulo do servidor), um servidor web e um navegador web.”

Um dos desafios deste trabalho é “conectar” pessoas, em tese, desconhecidas. Visto que o objetivo do sistema é facilitar o contato entre prestador de serviço e seu potencial cliente, foi programado um botão para que o cliente em potencial, possa

² https://secure.php.net/manual/pt_BR/intro-whatcando.php

clicar e entrar em contato direto, através do aplicativo WhatsApp, com o prestador de serviço. Como ilustra a figura 10.

Figura 10 - Programação do botão “Enviar Mensagem”.

```

56
57
58
59
60
61
62
<!-- Essa api do whatsapp abre uma conversa com um contato não salvo na lista de contatos-->
<a href="https://api.whatsapp.com/send?phone=5514<?php echo $prestadorservico['celular']; ?>"
  target="blank" class="btn btn-sm btn-warning">
  <!-- echo $prestadorservico['celular'] captura do banco de dado o numero do celular do prestador de serviço -->
  <i class="fa fa-pencil"></i> Enviar mensagem
</a>

```

Fonte: Elaborado pelo próprio autor, 2017.

3.3 Servidor

XAMPP

Como explicado no tópico 2.1.4, com Xampp foi configurado todo o ambiente de desenvolvimento local. A partir do painel de controle, inicia o Apache, como o servidor de página, e o MySQL como gerenciador de banco de dados.

MySQL

Como explicado no tópico 2.1.5, o MySQL é um Sistema Gerenciador de Banco de Dados que se utiliza da linguagem SQL. Através do aplicativo web phpMyAdmin – ilustrado anteriormente na Figura 2 – ocorre a administração do MySQL pela internet.

A partir do phpMyAdmin é possível criar, configurar e remover banco de dados, tabelas e campos, executar códigos SQL e manipular toda a estrutura do banco de dados.

3.4 Modelagem

Astah

Astah – citado no tópico 2.1.6 – definido como a ferramenta de modelagem de dados e diagramas UML.

3.4.1 Diagramas

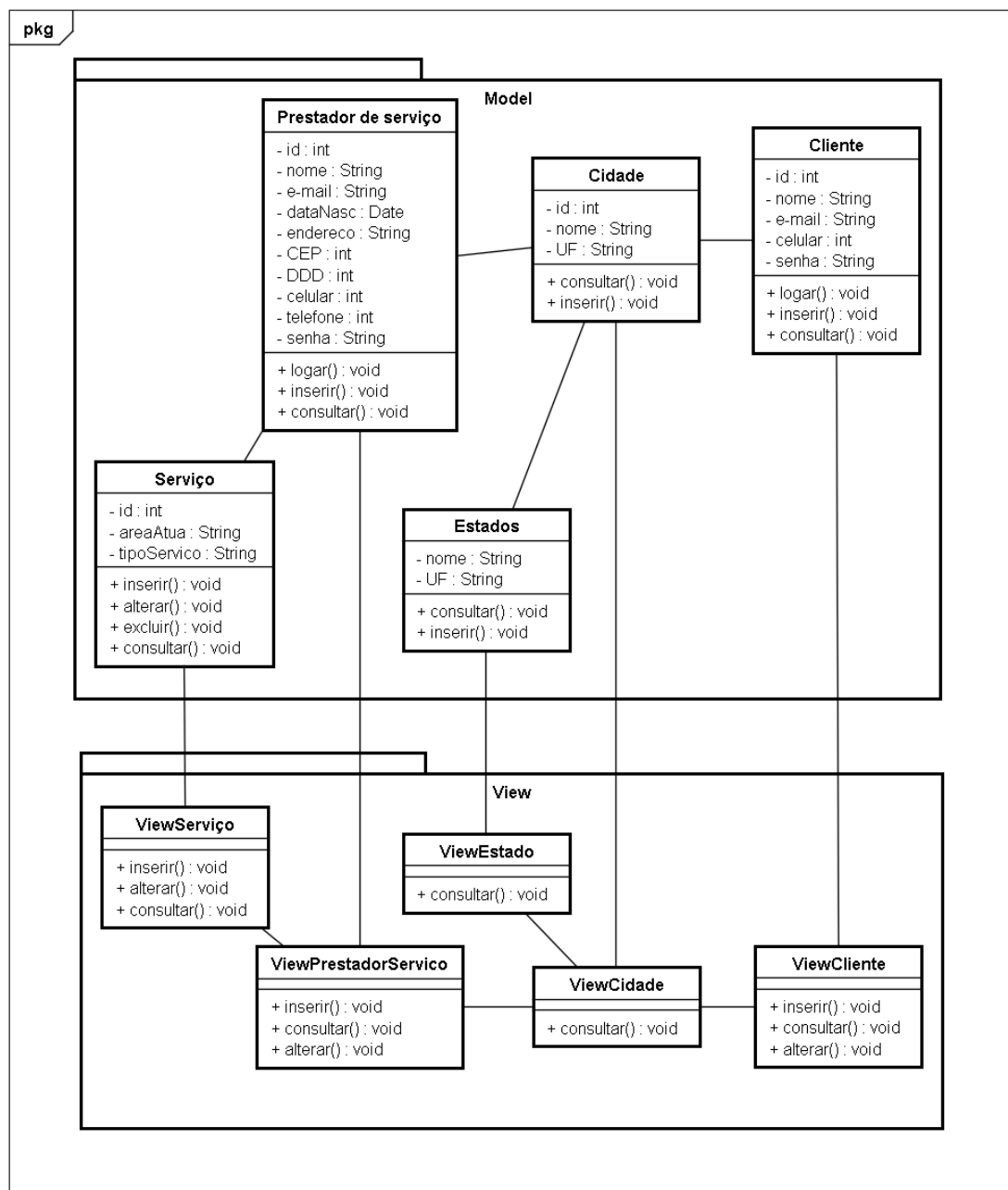
São apresentados os diagramas de caso de uso no apêndice B que relacionam os atores do sistema juntamente com suas principais funções.

Diagrama de Classe

Definido como uma representação das relações entre as classes do sistema juntamente com a estrutura (BELL, 2016).

A Figura 11 apresenta o cadastro de prestador de serviço, através do administrador do sistema, representado por um Diagrama de Classe.

Figura 11 - Diagrama de classes. Cadastrar prestador de serviço.

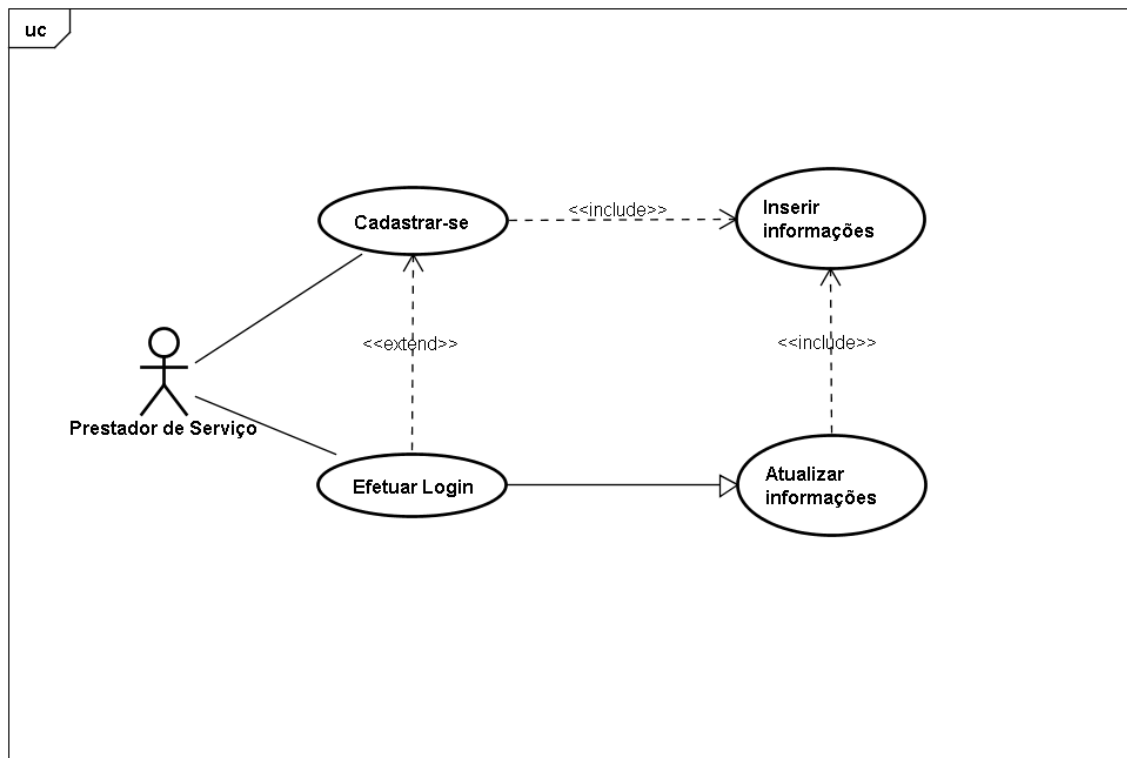


Fonte: Elaborado pelo próprio autor, 2017.

Demais diagramas estão nos apêndices.

A Figura 12 apresenta o cadastro de prestador de serviço, através do próprio usuário, representado por um Diagrama de caso de uso.

Figura 12 - Caso de uso - Prestador de Serviço

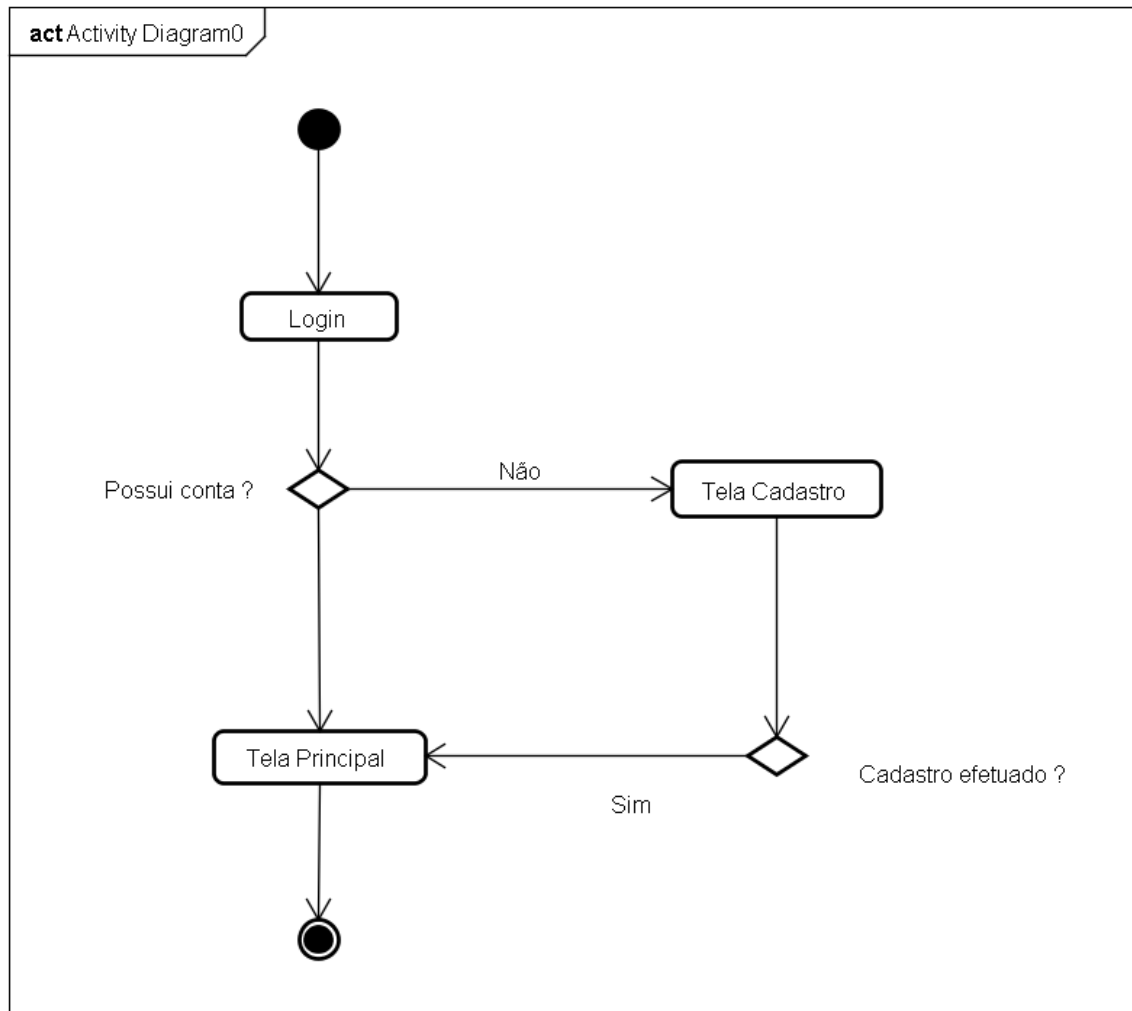


Fonte: Elaborado pelo próprio autor, 2017.

Demais diagramas estão nos apêndices.

A Figura 13 apresenta o *login* de usuário, prestador de serviço ou cliente, através do próprio usuário, representado por um Diagrama de atividade.

Figura 13 - Diagrama de atividade - *login*

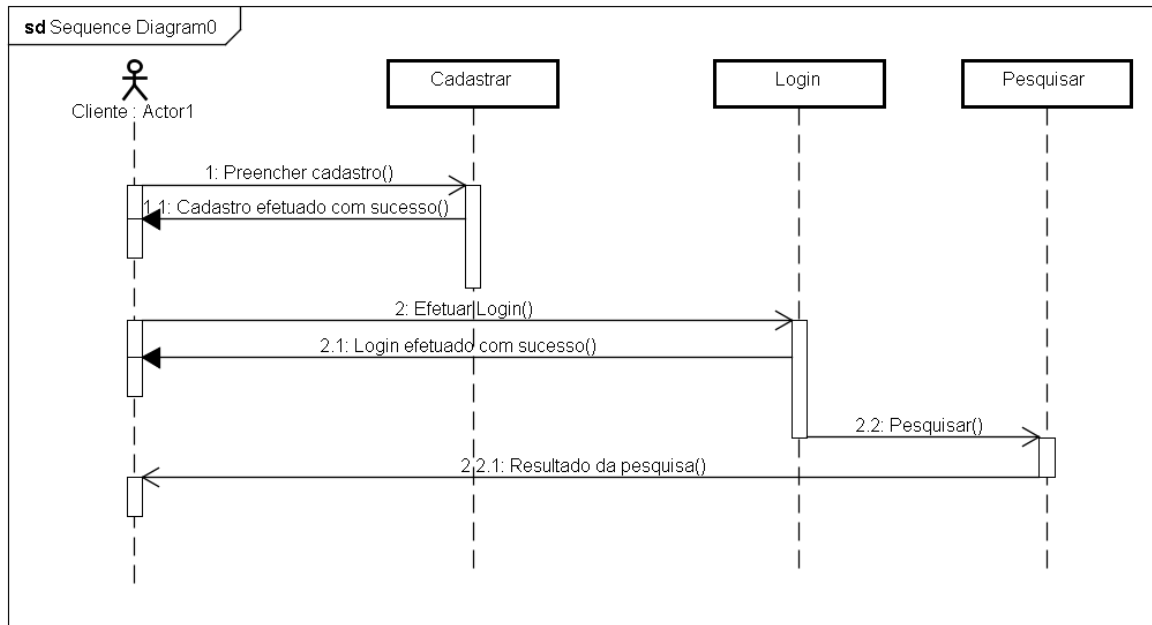


Fonte: Elaborado pelo próprio autor, 2017.

Demais diagramas estão nos apêndices.

A Figura 14 apresenta a sequência de atividade do usuário cliente, representado por um Diagrama de sequência.

Figura 14 - Diagrama de sequência - Cliente



Fonte: Elaborado pelo próprio autor, 2017.

Demais diagramas estão nos apêndices.

4 ANALISE DE RESULTADO

O desenvolvimento do projeto priorizou a responsividade, usabilidade e a facilidade de comunicação entre os atores do sistema.

Por se tratar de um sistema web, a responsividade foi fundamental, principalmente para a restrição de acesso a determinadas páginas, como por exemplo, as páginas de cadastros só podem ser acessadas via desktop ou notebook, caso o usuário acesse por um dispositivo móvel a página não é exibida.

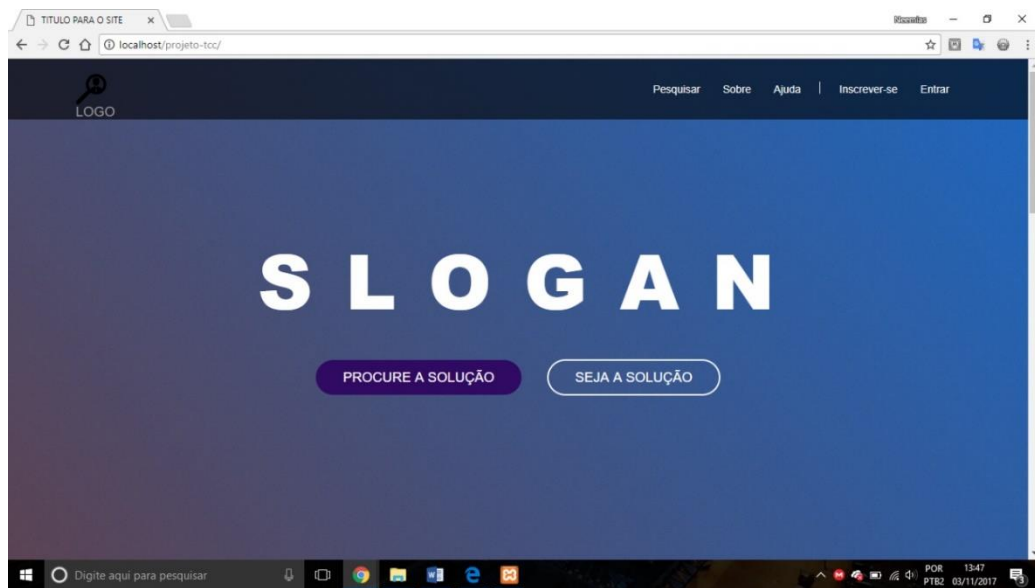
Um sistema simples e direto, totalmente responsivo, que em poucos passos o usuário atinge seu objetivo. A usabilidade do sistema foi pensada nesses fatores, ser simples, direto, fácil e juntamente com os conceitos de IHC (Interação Homem-Computador).

A facilidade de comunicação veio através de uma *Application Programming Interface* (API) do whatsapp, mencionado no capítulo 3.4 anteriormente descrito neste trabalho

Analisando a responsividade

Analisando as telas do sistema, a primeira tela será para escolha do perfil conforme a Figura 15. Se o usuário pretende se cadastrar como prestador de serviço, clicar no botão “SOU A SOLUÇÃO” do lado direito da tela. Se é para apenas fazer uma pesquisa, clicar no botão “BUSCAR SOLUÇÃO” no lado esquerdo. Se for um usuário, poderá executar o *login* do sistema clicando no “Entrar” localizado no canto superior direito.

Figura 15 - Tela Inicial



Fonte: Elaborado pelo próprio autor, 2017.

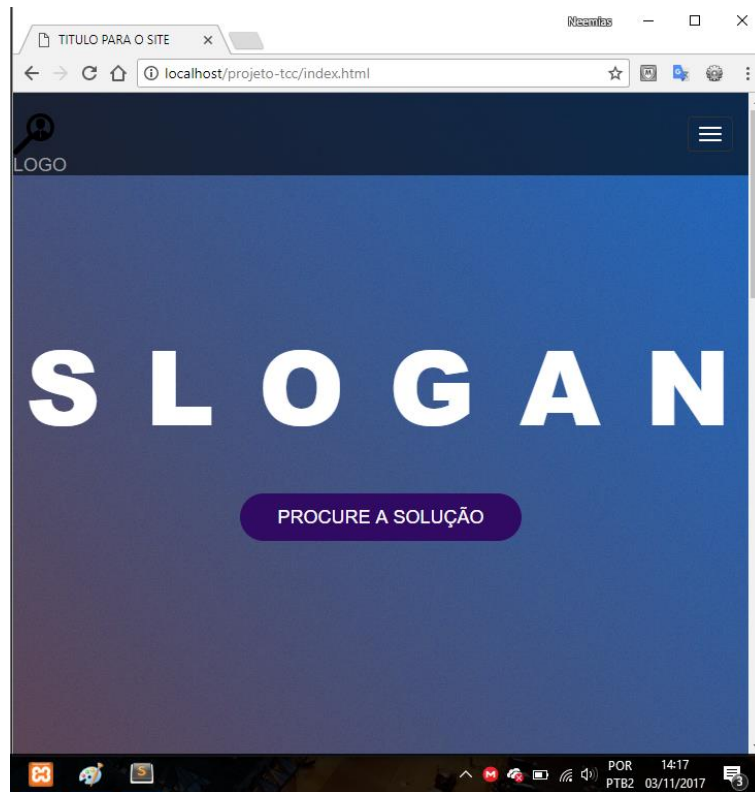
Para o usuário que quer se cadastrar como prestador de serviço, ao clicar no botão “SOU A SOLUÇÃO”, será redirecionado à tela de cadastro de prestadores de serviço, que deverá preencher todos os campos para fins de cadastro conforme a Figura 16.

Figura 16 - Tela, Cadastro de Prestador de Serviço

Fonte: Elaborado pelo próprio autor, 2017.

Porém essa funcionalidade só está disponível para *desktop* como ilustra a Figura 17.

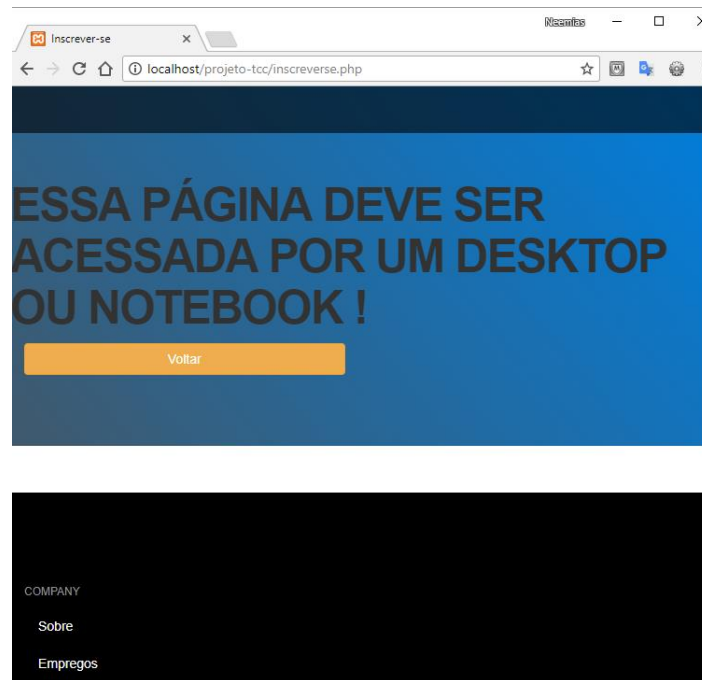
Figura 17 - Tela inicial responsiva, com restrição



Fonte: Elaborado pelo próprio autor, 2017.

No entanto, o usuário pode acessar a página de cadastro através de um link, e caso o mesmo venha acessar esse link através de um dispositivo móvel, não vai conseguir visualizar o formulário, como ilustra a Figura 18.

Figura 18 - Tela, cadastro de Prestador de Serviço "bloqueada"



Fonte: Elaborado pelo próprio autor, 2017.

Esse “bloqueio” se faz através do Bootstrap como mostra a Figura 19 onde, na linha 36 e 37 é usado a classe “*visible-xs*”.

Figura 19 – Programação para exibir mensagem em tela menor que 768 pixels

```

35
36 <h2 class="visible-xs">ESSA PÁGINA DEVE SER ACESSADA POR UM DESKTOP OU NOTEBOOK !</h2>
37 <div class=" visible-xs col-xs-6">
38   <a href="index.html" class="btn btn-warning visible-xs">Voltar</a>
39 </div>

```

Fonte: Elaborado pelo próprio autor, 2017.

A classe “*visible-xs*” quando inserida corretamente em código HTML, torna aquele trecho de código visível somente para dispositivos com tela menor que 768px.

A Figura 20 mostra uma tabela de classes disponíveis, contida na documentação do framework Bootstrap. Nessa tabela podemos identificar e escolher a classe ideal para utilizar em determinados componentes das páginas HTML, como forma de torna-los visíveis ou não.

Figura 20 - Tabela de classes disponíveis

classes disponíveis

Use uma única ou a combinação das classes disponíveis para alternar conteúdo em pontos de interrupção janela de visualização.

	Extra pequenos dispositivos Telefones (<768px)	Dispositivos pequenos Tablets (≥768px)	Dispositivos médias áreas de trabalho (≥992px)	Dispositivos de grandes áreas de trabalho (≥1200px)
<code>.visible-xs-*</code>	Visível	Escondido	Escondido	Escondido
<code>.visible-sm-*</code>	Escondido	Visível	Escondido	Escondido
<code>.visible-md-*</code>	Escondido	Escondido	Visível	Escondido
<code>.visible-lg-*</code>	Escondido	Escondido	Escondido	Visível
<code>.hidden-xs</code>	Escondido	Visível	Visível	Visível
<code>.hidden-sm</code>	Visível	Escondido	Visível	Visível
<code>.hidden-md</code>	Visível	Visível	Escondido	Visível
<code>.hidden-lg</code>	Visível	Visível	Visível	Escondido

Fonte: Elaborado pelo próprio autor, 2017.

Otimizando a Comunicação cliente-prestador

A comunicação entre prestador de serviço e potencial cliente, como exemplificado no tópico 3.2 acima, acontece através de uma conversa via WhasApp (aplicativo de mensagem). Para estabelecer uma conversa, no aplicativo WhasApp, entre o cliente e o prestador de serviço, foi implementado o recurso “Conversa em um clique”.

WhatsApp (2017) explica que o recurso Conversa em um clique do WhatsApp permite iniciar uma conversa com uma pessoa sem que tenha o número de telefone, do mesmo, salvo em na lista de contatos, basta saber o número de telefone dessa pessoa para que possa criar um link onde será possível iniciar uma conversa. Ao clicar no link, uma janela de conversa com essa pessoa se abrirá automaticamente. A Conversa em um Clique funciona tanto em telefone móvel quanto no WhatsApp para computador.

Para criar o link, deve-se usar <https://api.whatsapp.com/send?phone=> seguido pelo número de telefone em formato internacional completo, da outra pessoa com quem estabelecerá uma conversa.

As figuras a seguir ilustram a especificação do recurso “Conversa em um clique”.

Figura 21 - Clique no botão "Enviar msg" do resultado de busca

ID	Nome	Opções
1	neemias	Visualizar Enviar msg

©2017 - TCC

<https://api.whatsapp.com/send?phone=5514996794391>

Fonte: Elaborado pelo próprio autor, 2017.

Logo após clicar no botão “Enviar Mensagem”, é aberta uma nova aba – Figura 22 – carregando uma página do WhatsApp Web mostrando o número da pessoa com a qual está prestes a entrar em contato.

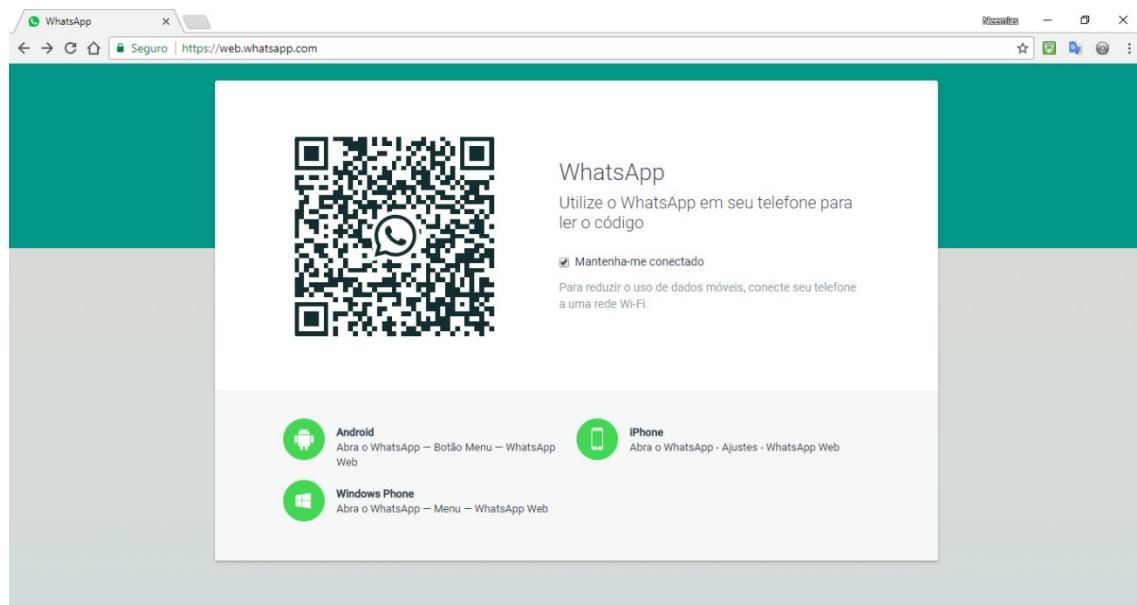
Figura 22 - Pagina do WhatsApp Web



Fonte: Elaborado pelo próprio autor, 2017.

Ao clicar novamente em “Enviar Mensagem” a página é atualizada e o WhatsApp web solicita uma autenticação via *Quick Response Code* (QR Code). Figura 23.

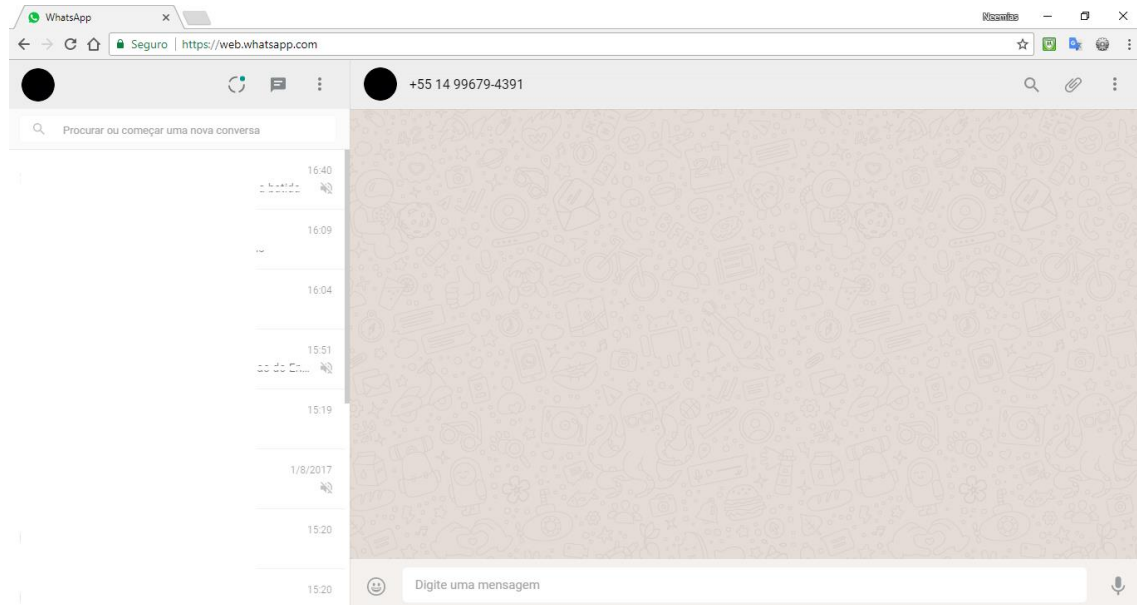
Figura 23 - Pagina de autenticação via QR Code do WhatsApp Web



Fonte: Elaborado pelo próprio autor, 2017.

Após efetuar a leitura do QR Code com o dispositivo móvel em mãos, automaticamente a conversa é aberta, a partir daí está estabelecido o contato com o profissional desejado. Figura 24.

Figura 24 - Conversa estabelecida no WhatsApp Web via API do WhatsApp



Fonte: Elaborado pelo próprio autor, 2017.

5 CONCLUSÃO

O trabalho trata de apresentar um projeto de um sistema de informação usando tecnologia responsiva em arquitetura MVC, para oferecer uma praticidade na busca por profissionais prestadores de serviço.

O objetivo deste trabalho é apresentar a possibilidade de utilizar tecnologia da informação na construção de uma ferramenta responsiva, para facilitar a busca e proporcionar o contato entre prestadores de serviços e seus potenciais clientes.

Todos os objetivos propostos foram alcançados. O sistema utiliza o framework Bootstrap, totalmente responsivo, sendo que atualmente é difícil pensar em um sistema ou site para a web que não seja, total ou parcial, responsivo.

Ao adotar o padrão de arquitetura de software MVC, além de fazer um desenvolvimento organizado, será menos complicado implementar novas funcionalidades e torna o suporte mais eficaz e ágil, embora não se tenha desenvolvido pesquisa sobre este foco.

Com a finalização da construção desse sistema, pretende-se operacionalizar o uso do mesmo comprando um domínio Internet e alugando a hospedagem do sistema em *datacenter*. A princípio será necessário montar uma base de dados com uma quantidade considerável de profissionais cadastrados, para que, ao abrir para o público, os mesmos tenham acesso ao sistema com resultados relevantes em suas buscas.

Com a implementação do sistema, aplicou-se técnicas de empreendedorismo, aprendidas em aula, para que seja elaborado um modelo de negócio consistente e eficaz, onde o sistema atenda às necessidades do seu público-alvo e ao mesmo tempo seja rentável. Buscar maneiras de monetizar o sistema de modo a mantê-lo gratuito ao usuário.

De modo geral o uso de TIC se mostrou eficiente no desenvolvimento de um sistema de informação em arquitetura MVC, como é objetivo do trabalho.

6 REFERÊNCIAS

ASTAH. **Astah is Modeling**, 2016. Disponível em: <<http://astah.net/>>. Acesso em: 12 mai. 2016

BELL, D. **O diagrama de classes**. Uma introdução aos diagramas de estrutura UML 2. 2016. Disponível em <<https://www.ibm.com/developerworks/br/rational/library/content/RationalEdge/sep04/bell/index.html>>. Acesso em 08 out. 2017.

BOOTSTRAP, 2017. **CSS**. Disponível em: <<http://getbootstrap.com.br/css/>>. Acesso em 19 out. 2017.

BUSCHMANN, F., MEUNIER, R., ROHNERT, H., SOMMERLAND, P. & STAL, M (1996): **Pattern-Oriented Software Architectur A System Of Patterns**, JOHN WILEY & SONS, 1996.

CARNEIRO, B. S. **O que é Astah? #PostTécnico**. 2015. Disponível em: <<http://www.startupsstars.com/2015/10/o-que-e-o-astah-posttecnico-por-brunoseabra/>>. Acesso em: 12 mai .2016

CHADWICK, Jess. SNYDER, Todd. PANDA, Hrusikesh. **ASP.NET MVC 4**. 1 ed. Sebastopol: O'Reilly, 2012.

COSTA, G. **O QUE É BOOTSTRAP?**. Disponível em: <<http://www.tutorialwebdesign.com.br/o-que-e-bootstrap/>>. Acesso em: 06 mai 2016.

FIGUEIREDO, E. **Entendendo o padrão MVC na prática**. 2015. Disponível em: <<http://tableless.com.br/entendendo-o-padrao-mvc-na-pratica/>>. Acesso em: 06 mai. 2016.

GONÇALVES, Edson. **Desenvolvendo Aplicações Web com Jsp , Servlets , Javasever Faces , Hibernate , Ejb 3 Persistence e Ajax**. Ciência Moderna Ltda, 2007.

LOPES, S. 2007. **O que é PHP?**. Disponível em:< https://www.oficinadanet.com.br/artigo/659/o_que_e_php>. Acesso em: 06 mai. 2017.

MINETTO, E. **Frameworks para Desenvolvimento em PHP**. São Paulo, Novatec Editora, 2007.

OGLIO, P. D. **PHP Programando com Orientação a Obejetos**.São Paulo: Novatec Editora Ltda. 2016.

PALERMO, Jeffrey et al. **APS.NET MVC 2 In Action**. Manning Publications – Stamford, 2010.

RICARDO, José. **Introdução ao MySQL**. Disponível em:<<http://www.devmedia.com.br/introducao-ao-mysql/27799>>. Acesso em: 10 mai. 2017.

SANTOS, Isaias et al. **Possibilidades e limitações da arquitetura mvc (model – view – controller) com ferramenta ide (integrated development environment)**.

2010. 56f. Trabalho de Conclusão de Curso (Graduação em Ciências da Computação) - Universidade José do Rosário Vellano, Alfenas, Mg.

SILVA, Maurício Samy. **Bootstrap 3.3.5**. São Paulo: Novatec, 2015.

SILVA, Maurício Samy. **HTML5. 1ª ed.** São Paulo: Novatec, 2011.

SILVA, Maurício Samy. **Web Design Responsivo**. São Paulo: Novatec, 2014.

SILVEIRA, D. CAVALLINI, M. 2017. **Desemprego fica em 13,7% no 1º trimestre de 2017 e atinge 14,2 milhões**. Disponível

em:<<http://g1.globo.com/economia/noticia/desemprego-fica-em-137-no-1-trimestre-de-2017.ghml>>. Acesso em: 30 abr. 2017.

TEOLI. **Nav**. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/nav>>. Acesso em: 18 out 2017.

XAMPP. **Apache + MariaDB + PHP + Perl**, 2017. Disponível em: <https://www.apachefriends.org/pt_br/index.html>. Acesso em: 29 dez 2017

WHATSAPP, 2017. **Usando a Conversa em um Clique**. Disponível em: <https://faq.whatsapp.com/pt_br/general/26000030/?category=5245251>. Acesso em: 01 nov. 2017.

APÊNDICE A – Requisitos Funcionais e Não Funcionais

Requisitos Funcionais

RF 01 Apresentar Página Inicial

A página inicial do site contém botões para cadastro de cliente, prestador de serviço. Menu de opções com opção de *login*.

No rodapé, contém informações complementares do site e links para as redes sociais.

RF 02 Cadastro de cliente

Inicialmente na tela do Cliente, pede-se o nome, e-mail, cidade, estado, celular e senha, após a confirmação do e-mail o cliente é redirecionado para a página de *login*.

RF 03 Cadastro de Prestador de Serviço

Inicialmente na tela do Prestador de Serviço, pede-se nome, e-mail, data de nascimento, endereço, bairro, CEP, município, DDD, celular, telefone, área de atuação e tipo de serviço, após a confirmação dos dados cadastrados e do e-mail o prestador de serviço é redirecionado para a página de *login*.

O site deve oferecer uma lista de áreas de atuação e tipos de serviços pré cadastrados pelo administrador do site.

RF 04 Entrar

A tela de *login* deverá solicitar para verificação:

- E-MAIL;
- SENHA.

Após a verificação ser aceita, o usuário deverá ser redirecionado para a tela principal do site.

RF 05 Editar Informações do Perfil de Prestador de Serviço

Esta tela é composta com as informações complementares para preenchimento, sendo todas elas opcionais.

- E-mail (pré cadastrado);
- Nome (pré cadastrado);
- DDD (pré cadastrado);
- Celular (pré cadastrado);
- Telefone (pré cadastrado);
- Habilidades;
- Breve descrição do serviço prestado;
- Links para redes sociais.

RF 06 Apresentar resultado de busca

Este requisito deverá mostrar as informações do Prestador de Serviço para o Cliente, exibindo as informações de cadastro do Prestador de Serviço.

Deverá ser exibido:

- Nome;
- Área de atuação;
- Tipo de serviço;
- Cidade;
- Bairro;
- Enviar Mensagem (celular).

Requisitos Não Funcionais

Requisitos não-funcionais do projeto define prioridades e restrições do sistema. São especificados detalhadamente, tecnologia utilizada, métodos de desenvolvimento entre outros requisitos. Os requisitos não-funcionais geralmente são abstratos de um ponto de vista mensurável, sendo assim abaixo a descrição dos requisitos não-funcionais

RNF 01 Requisitos de Segurança

O sistema deverá possuir senhas e identificação para diferentes tipos de usuário e classifica-los em níveis de permissão da funcionalidade do sistema.

A alteração de senha, só pode ser feita pelo próprio usuário.

RNF 02 Requisitos de Usabilidade

O site deve apresentar uma interface gráfica intuitiva para que a funcionalidades sejam utilizadas sem a necessidade do usuário fazer um treinamento.

RNF 03 Requisitos de Manutenção

Ao implementar o site deve ser estruturado de forma a facilitar a manutenção no futuro.

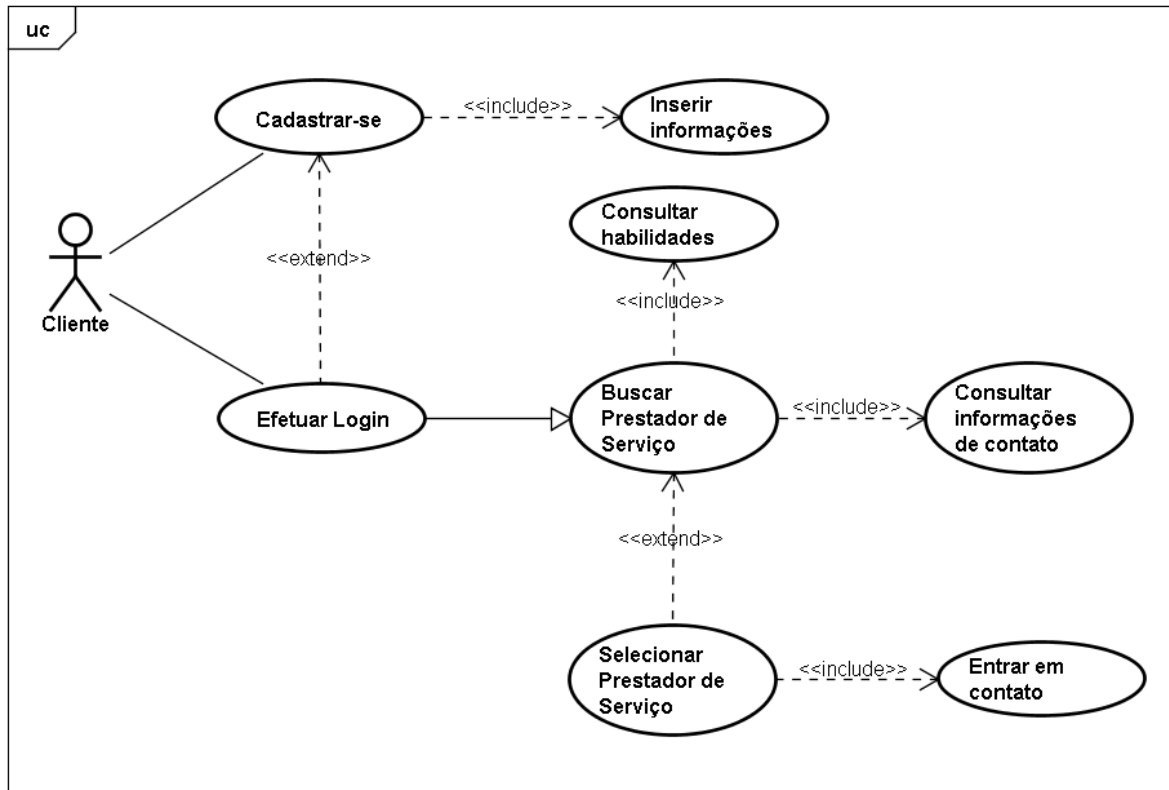
RNF 04 Requisitos Externos

Informações pessoais do administrador não serão acessadas pelos usuários cliente e prestador de serviço, apenas pelo próprio administrador

APÊNDICE B – Diagramas de Caso de Uso

A Figura 25 ilustra o caso de uso de Cliente

Figura 25 - Caso de uso Cliente



Fonte: Elaborado pelo próprio autor, 2017.

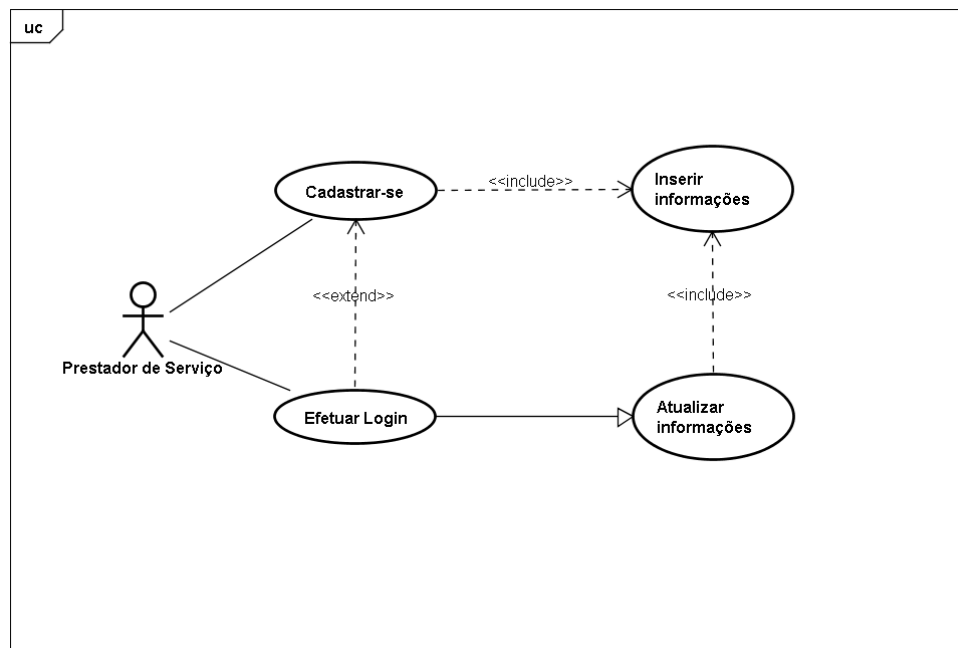
Tabela 1 - Descrição de cenário - Cliente

ITEM	VALUE
Use Case	Cliente
Summary	Busca de profissionais prestadores de serviço
Actor	Usuário Cliente
Precondition	Usuário estar cadastrado
Postcondition	Usuário estar logado
Base Sequence	<p>Usuário deverá se cadastrar e inserir as seguintes informações:</p> <ul style="list-style-type: none"> - Nome - E-mail - Celular - Senha <p>Usuário poderá efetuar busca dos prestadores de serviço que estão cadastrados, podendo filtrar por:</p> <ul style="list-style-type: none"> - Tipo de serviço - Área de atuação - Cidade - Estado
Branch Sequence	Selecionar, no resultado da busca, o prestador de serviço desejado; Entrar em contato
Exception Sequence	Usuário sem cadastro, não consegue efetuar a busca por prestador de serviço
Sub Use Case	
Note	

Fonte: Elaborado pelo próprio autor, 2017.

A Figura 26 ilustra o caso de uso do Prestador de Serviço.

Figura 26 - Caso de uso Prestador de Serviço



Fonte: Elaborado pelo próprio autor, 2017.

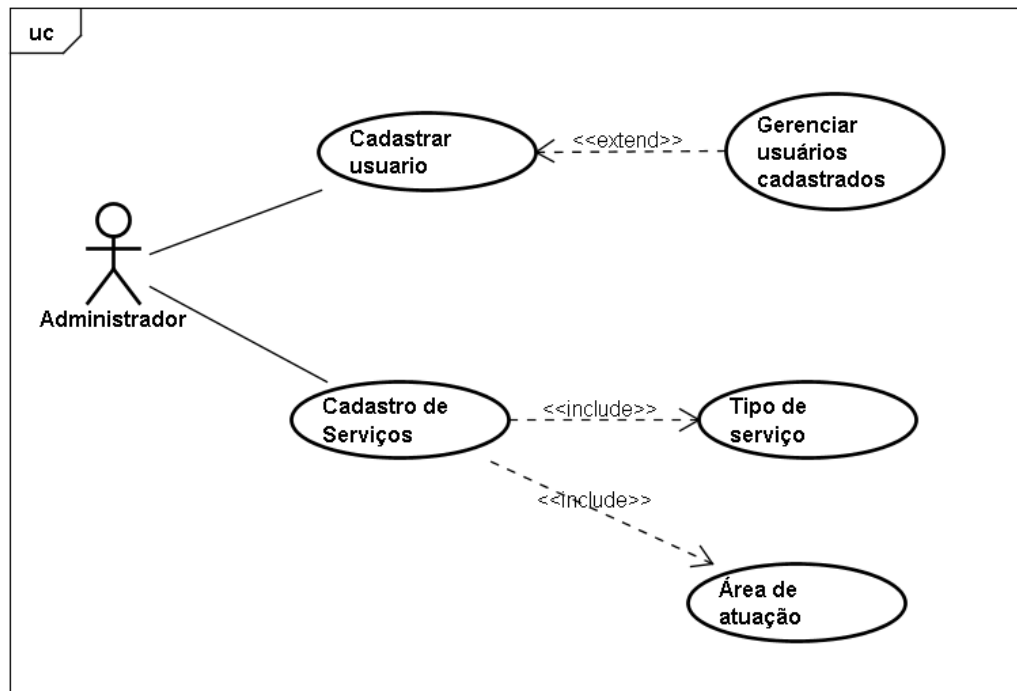
Tabela 2 - Descrição do cenário - Prestador de Serviço

ITEM	VALUE
Use Case	Prestador de Serviço
Summary	Busca de profissionais prestadores de serviço
Actor	Usuário prestador de serviço
Precondition	Usuário estar logado
Postcondition	
Base Sequence	<p>Usuário deverá se cadastrar e inserir as seguintes informações:</p> <ul style="list-style-type: none"> - Nome - E-mail - Data de nascimento - Endereço - Bairro - CEP - DDD - Celular - Telefone - Senha - Área de atuação - Tipo de serviço <p>Usuário poderá atualizar as informações cadastradas, posteriormente.</p>
Branch Sequence	
Exception Sequence	
Sub Use Case	
Note	

Fonte: Elaborado pelo próprio autor, 2017.

A Figura 27 ilustra o caso de uso do Prestador de Serviço.

Figura 27 - Caso de uso Administrador



Fonte: Elaborado pelo próprio autor, 2017.

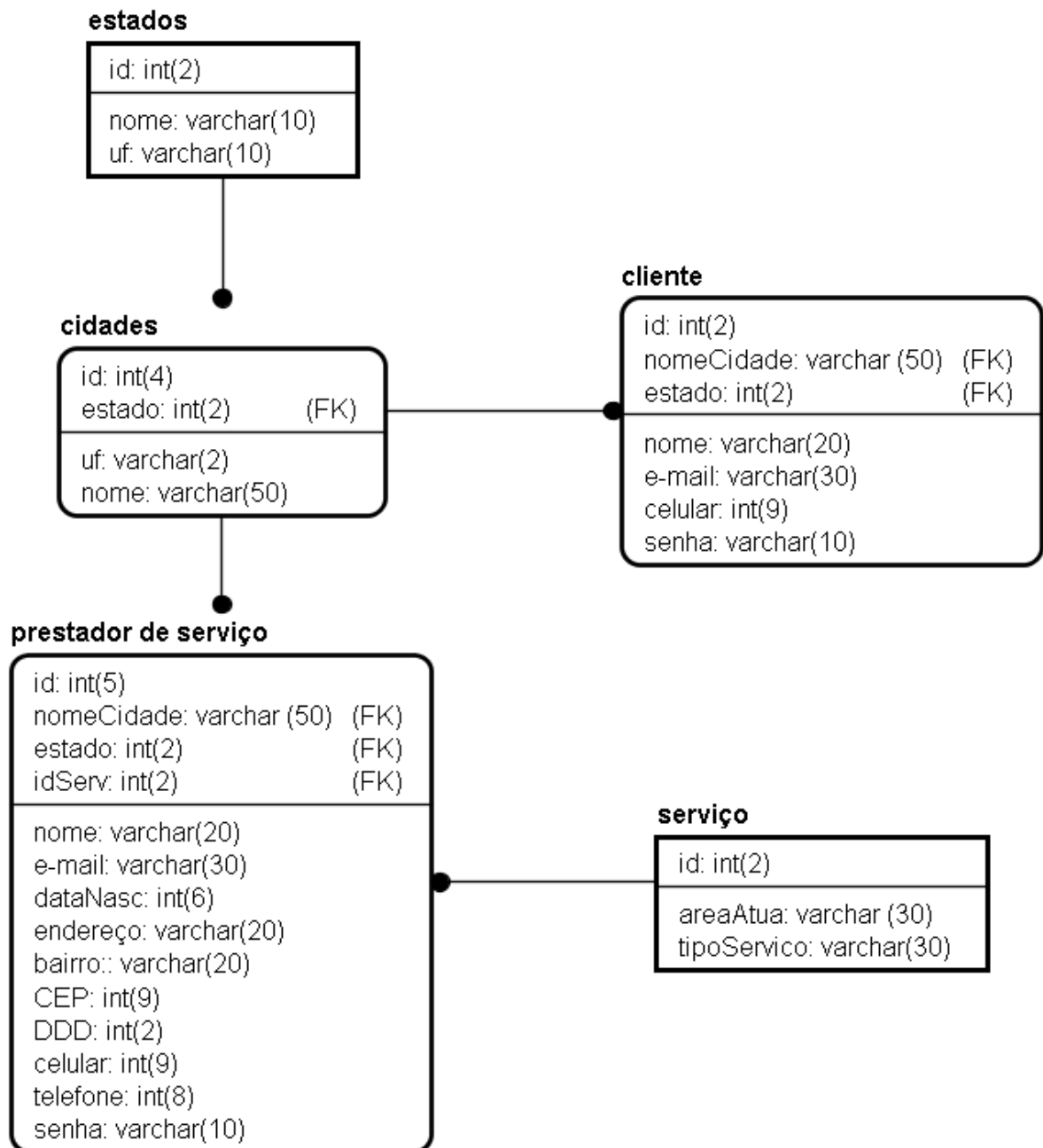
Tabela 3 - Descrição do cenário - Administrador

ITEM	VALUE
Use Case	Administrador
Summary	Gerenciamento do site
Actor	Administrador
Precondition	
Postcondition	
Base Sequence	O Administrador poderá cadastrar usuário, tanto cliente como prestador de serviço, manualmente. Devera gerenciar os cadastros feitos pelo usuário a fim de identificar possíveis inconsistências nos dados informados pelo usuário. Cadastrar tipos de serviços e área de atuação no cadastro de serviços.
Branch Sequence	
Exception Sequence	
Sub Use Case	
Note	

Fonte: Elaborado pelo próprio autor, 2017.

MODELO ENTIDADE RELACIONAMENTO - MER

Figura 28 - Modelo entidade relacionamento



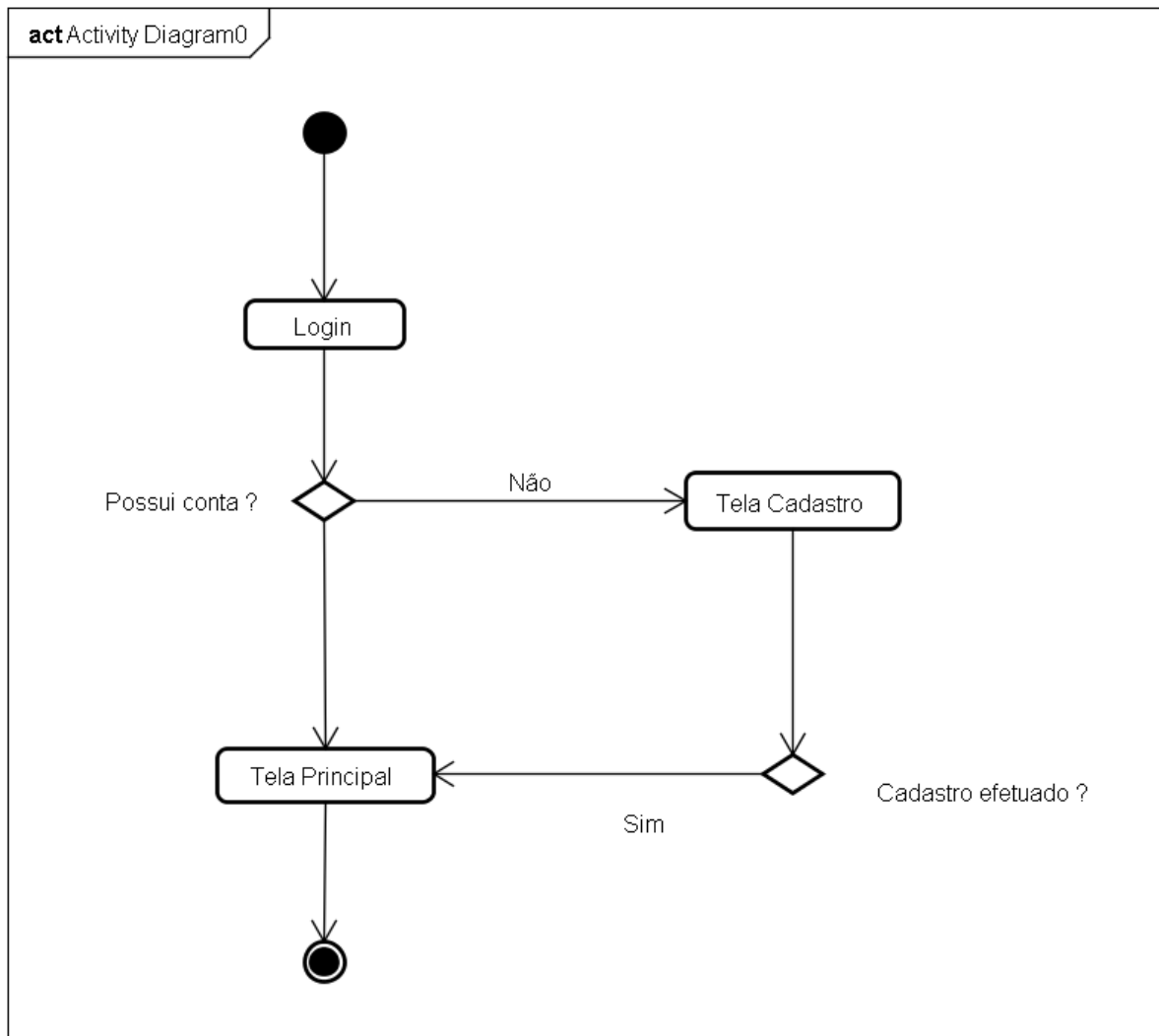
Fonte: Elaborado pelo próprio autor, 2017.

DIAGRAMA DE ATIVIDADE

Login

Demonstra etapas para o processo de *login*.

Figura 29 - Diagrama de atividade - Login

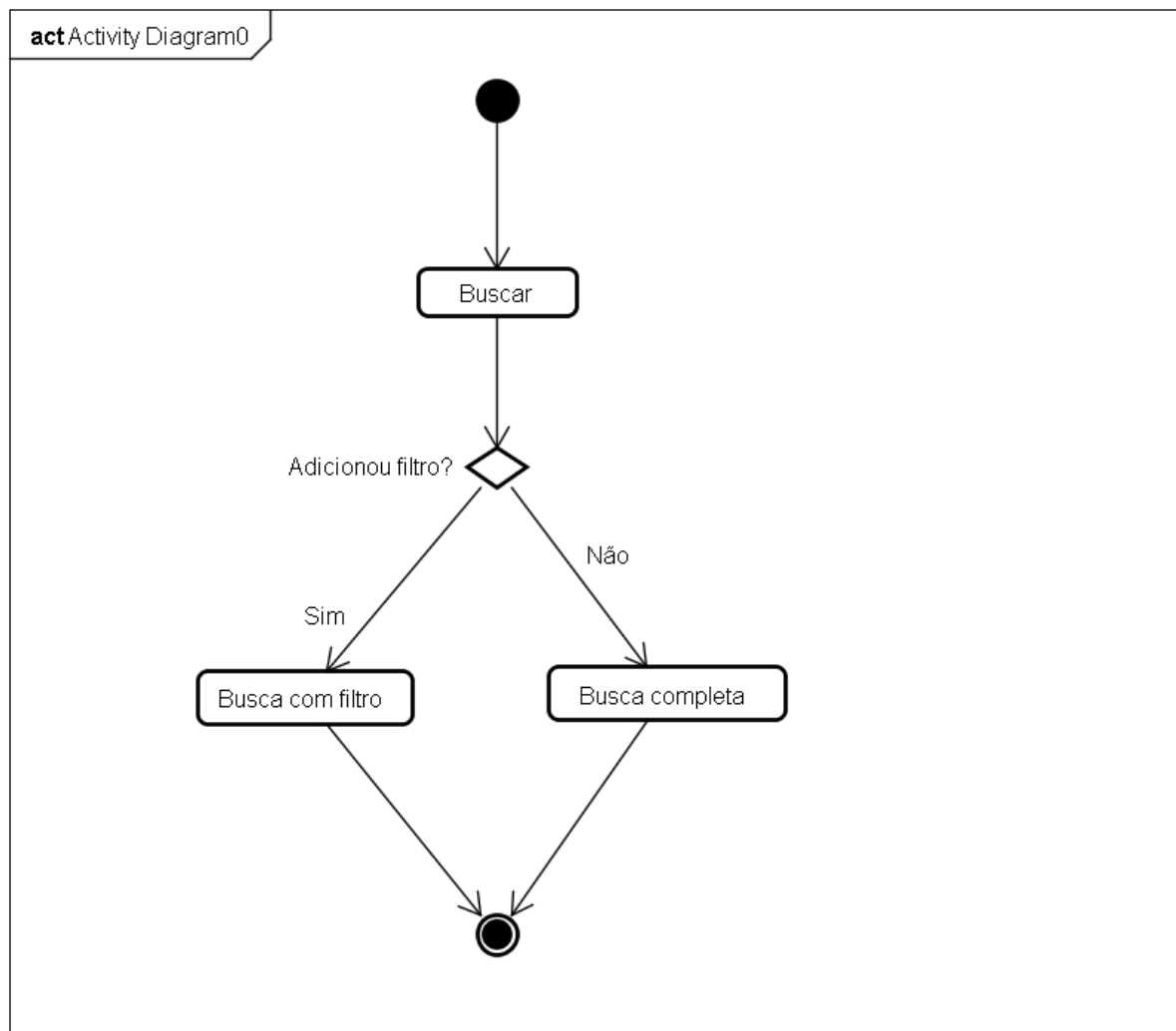


Fonte: Elaborado pelo próprio autor, 2017.

Buscar

Descreve etapa para o processo buscar prestador de serviço.

Figura 30 - Diagrama de atividade - Buscar

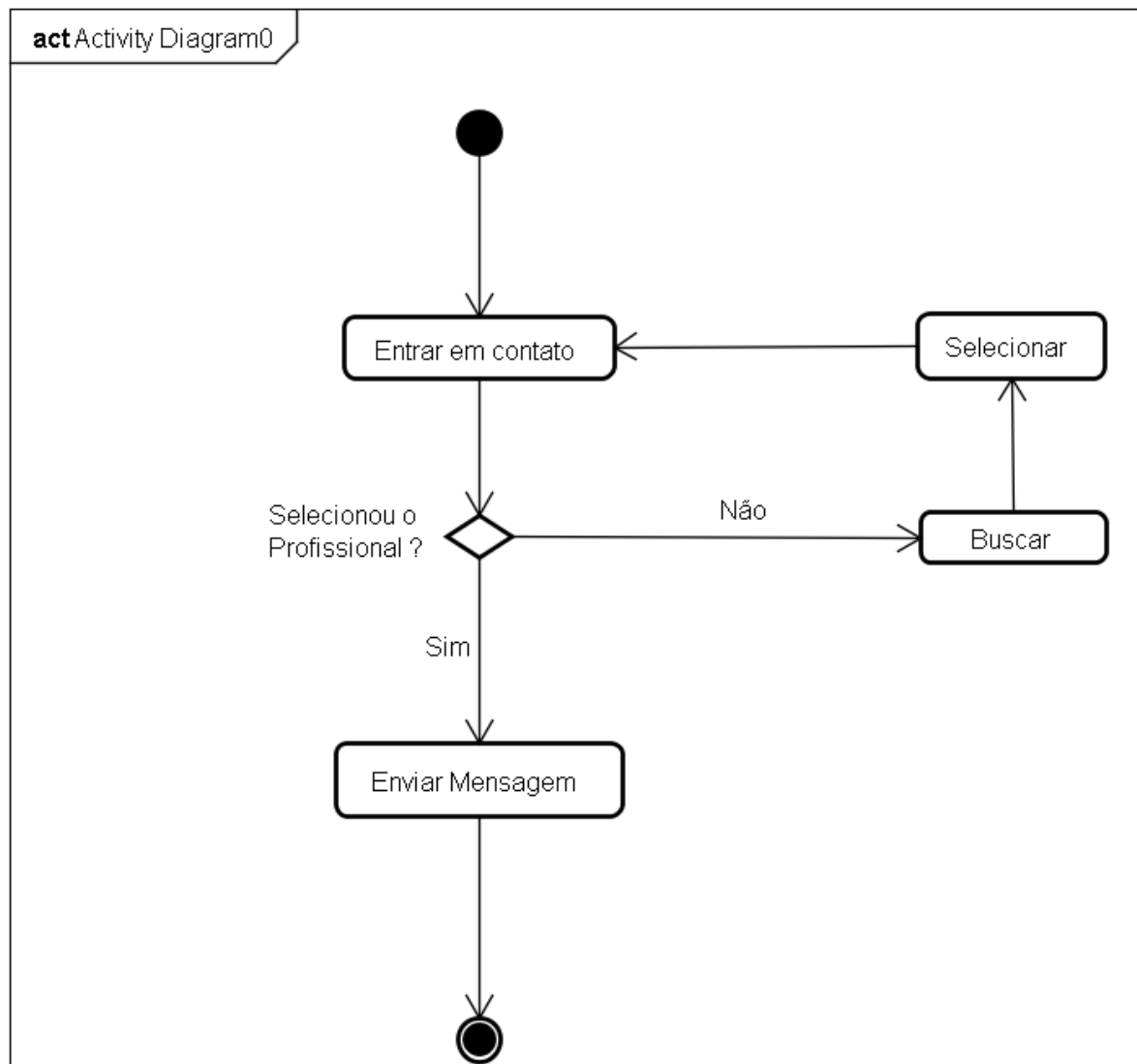


Fonte: Elaborado pelo próprio autor, 2017.

Entrar em contato

Descreve etapas para o processo de entrar em contato com o prestador de serviço

Figura 31 - Diagrama de atividade - Entrar em contato

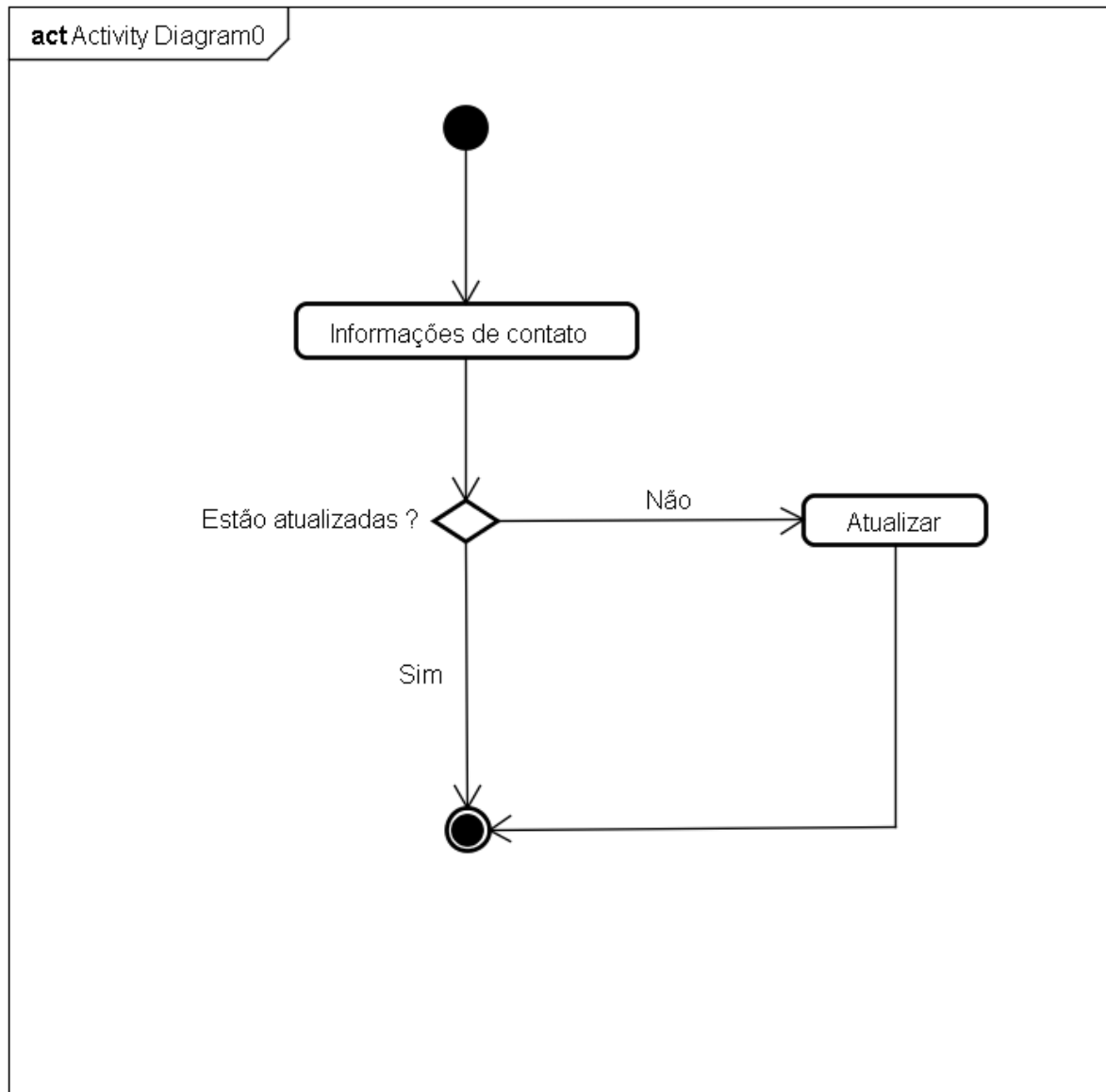


Fonte: Elaborado pelo próprio autor, 2017.

Atualizar informações

Descreve o processo para atualizar informações já cadastradas pelo usuário prestador de serviço e usuário cliente.

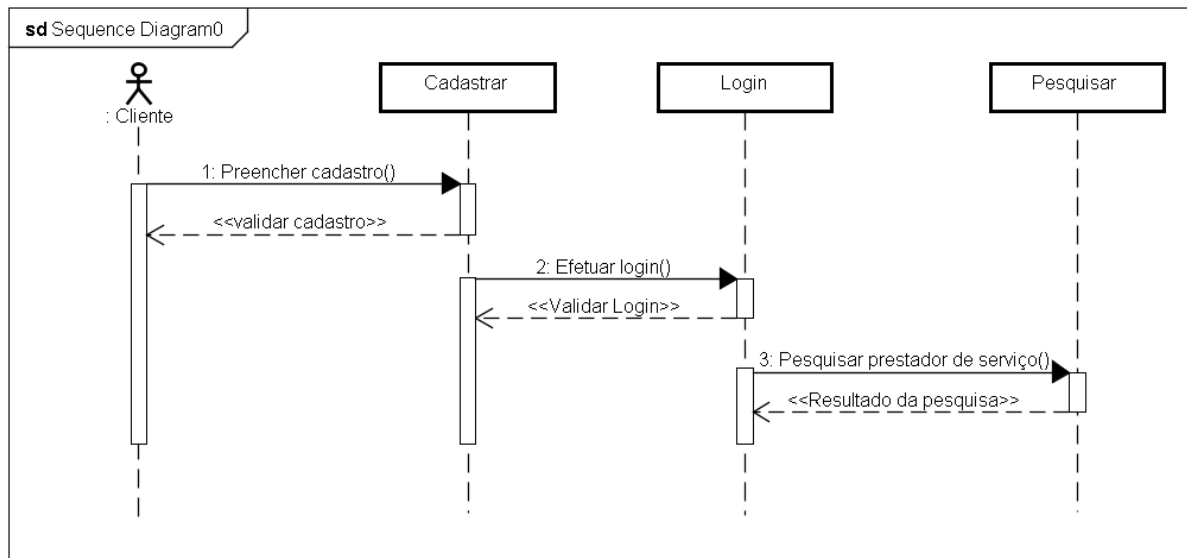
Figura 32 - Diagrama de atividade - Atualizar informações



Fonte: Elaborado pelo próprio autor, 2017.

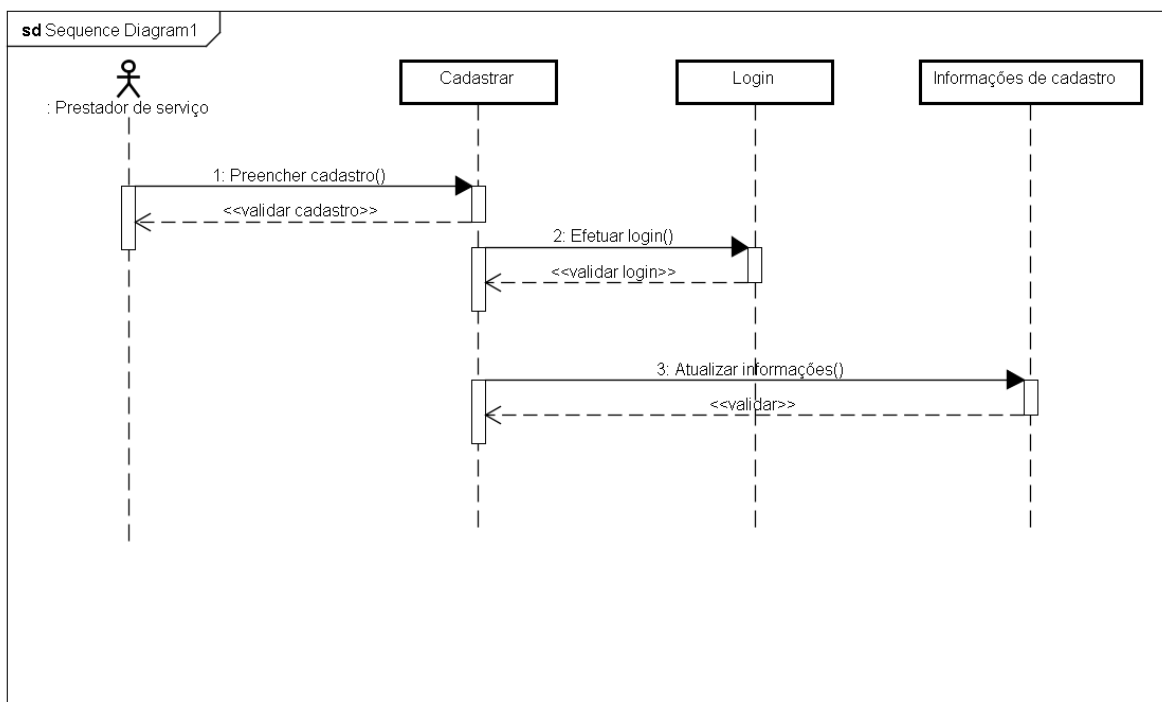
DIAGRAMA DE SEQUÊNCIA

Figura 33 - Diagrama de sequência – Cliente



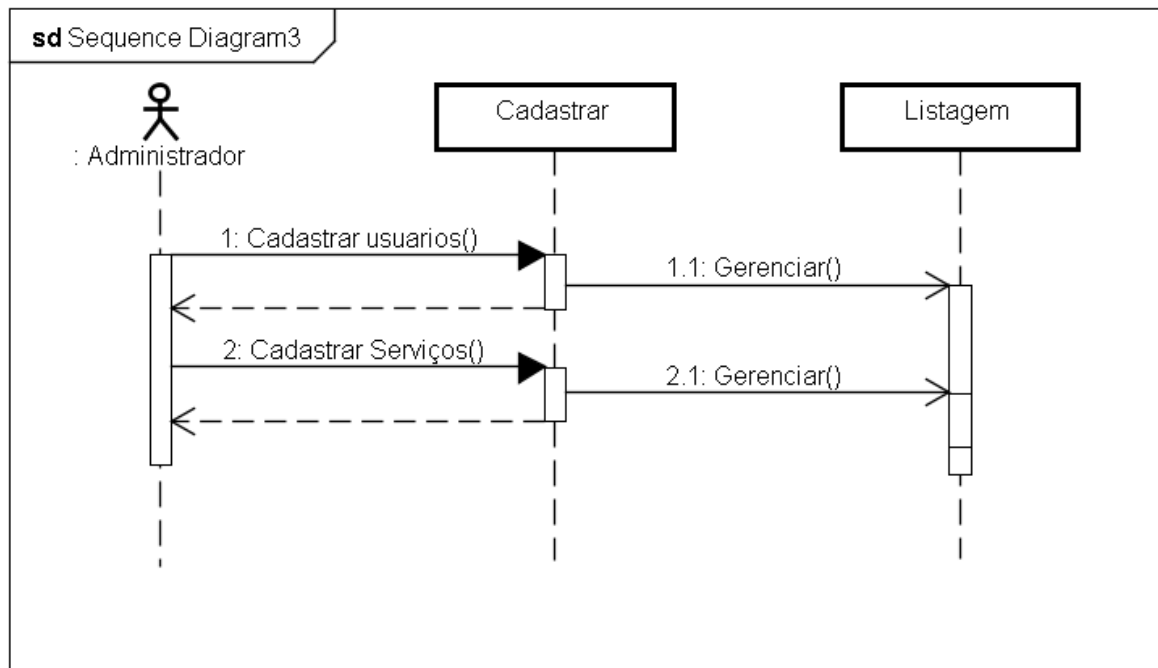
Fonte: Elaborado pelo próprio autor, 2017.

Figura 34 - Diagrama de sequência - Prestador de Serviço



Fonte: Elaborado pelo próprio autor, 2017.

Figura 35 - Diagrama de sequência - Administrador



Fonte: Elaborado pelo próprio autor, 2017.