

UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS  
FACULDADE DE COMPUTAÇÃO  
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Maykon Araújo de Souza

**CONCEPÇÃO, ESPECIFICAÇÃO, IMPLEMENTAÇÃO E AVALIAÇÃO DE APLICAÇÃO  
WEB PARA PRESTADORES DE SERVIÇOS**

Belém  
2017

Maykon Araújo de Souza

**CONCEPÇÃO, ESPECIFICAÇÃO, IMPLEMENTAÇÃO E AVALIAÇÃO DE APLICAÇÃO  
WEB PARA PRESTADORES DE SERVIÇOS**

Trabalho de Conclusão de Curso apresentado para obtenção do título de Bacharel em Sistemas de Informação pela Faculdade de Computação do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará.

Orientador: Prof. Dr. Alfredo Braga Furtado.

Belém  
2017

Souza, Maykon Araújo de

Concepção, Implementação e Avaliação de Aplicação Web para Prestadores de Serviço / (Maykon Araújo de Souza); orientador, Alfredo Braga Furtado – 2017.

149f. il. 28cm.

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal do Pará. Instituto de Ciências Exatas e Naturais. Faculdade de Computação. Belém, 2017.

1. WebApp. 2. Booszinga. 3. Prestadores de Serviço.

CDD \_\_\_\_ . ed. \_\_\_\_\_

Maykon Araújo de Souza

**CONCEPÇÃO, ESPECIFICAÇÃO, IMPLEMENTAÇÃO E AVALIAÇÃO DE APLICAÇÃO  
WEB PARA PRESTADORES DE SERVIÇOS**

Trabalho de Conclusão de Curso apresentado para obtenção do título de Bacharel em Sistemas de Informação pela Faculdade de Computação do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará.

Data da aprovação: Belém-PA. \_\_\_\_/\_\_\_\_/\_\_\_\_

Conceito:

Banca Examinadora

---

Prof. Dr. Alfredo Braga Furtado  
Faculdade de Computação/ICEN/UFPA – Orientador

---

Prof. Dr. Raimundo Viégas Júnior  
Faculdade de Computação/ICEN/UFPA – Membro

---

Prof. M. Sc. José Maria Nascimento Bitar  
Faculdade de Computação/ICEN/UFPA – Membro

## AGRADECIMENTOS

A Deus, pela dádiva da vida e por todas as portas que ele abriu e abrirá na minha vida e por nunca me abandonar nos bons e maus momentos.

À minha mãe, Raimunda Araújo de Souza, pelo amor, apoio, compreensão e carinho, essenciais nos momentos de dificuldades e a meu pai, Miguel Pinto de Souza, por sempre apoiar os seus filhos a estudarem mesmo sem ter tido a oportunidade de estudar e ao meu irmão Magno Araújo de Souza pelo companheirismo e amizade durante a vida.

Ao meu orientador, Professor Alfredo Braga Furtado, pela disponibilidade e qualidade da orientação prestada e por sempre ter uma boa história para contar.

A minha namorada Gabrielle, por estar ao meu lado nesse momento importante da minha vida, por toda paciência, compreensão, carinho, companheirismo e amor.

Ao meu amigo e sócio, Everson P. A. Costa, que me ajudou na parte do *designer* da ferramenta apresentada neste trabalho.

A todos os meus amigos que contribuíram com sua amizade e paciência quando mais precisei, o caminho percorrido ainda está no começo, mas sem amigos de verdade eu não teria chegado tão longe.

A todos, que direta ou indiretamente, contribuíram até aqui em minha jornada acadêmica, porque ninguém realiza nada sozinho. Um agradecimento especial aos colegas que compõem comigo a nova geração da *Gang of Four*: Deivison, Marcos Paulo e Júlio. Agradecimento especial também ao Professor Dr. Bianchi Serique Meiguins que me motivou e ajudou bastante no início do curso. No contexto acadêmico, sempre há pessoas que, com pequenas ou grandes ações, são responsáveis pela materialização de um projeto ou simplesmente de um sonho.

“Ainda que eu andasse pelo vale  
da sombra da morte, não temerei  
mal algum, porque tu estás  
comigo; a tua vara e o teu cajado  
me consolam. ”  
(Salmos 23:4)

## RESUMO

Este trabalho faz um estudo sobre o desenvolvimento (envolvendo concepção, especificação, implementação e avaliação) de uma aplicação *WebApp*, cujo objetivo é a intermediação entre interessados em serviços e prestadores de serviços<sup>1</sup>. A ferramenta é chamada de Booszinga<sup>2</sup>. Os prestadores de serviços, em sua grande maioria, não possuem capital financeiro para divulgar seu novo empreendimento em curto prazo e a ferramenta Booszinga atende esta necessidade. A ferramenta auxilia prestadores de serviços na criação de uma página *web*, disponibilizada em uma área administrativa, na qual eles podem cadastrar categorias de atuação, portfólio, perfil profissional, preços praticados e descrever suas experiências. Essas informações ficam disponibilizadas na plataforma online. Potenciais clientes podem consultar e encontrar o prestador de serviço que melhor atenda suas necessidades, para fechamento de negócio. O sistema foi desenvolvido utilizando metodologia ágil com uma influência da modelagem interativa e incremental, utilizando UML para especificação. A implementação foi feita na linguagem PHP, com o auxílio dos *Frameworks Codeigniter, Bootstrap e JQuery* e a base de dados criada em MySQL, utilizando o sistema de gestão de base de dados (SGBD), *phpmyadmin*. A avaliação da usabilidade da ferramenta foi feita por especialistas em IHC e por usuários finais, e atestou que os objetivos iniciais foram alcançados satisfatoriamente.

PALAVRAS-CHAVES: WebApp, Booszinga, Prestadores de serviços.

---

<sup>1</sup> É o coletivo usado para representar empreendedores e autônomos.

<sup>2</sup> Nome criado usando a combinação da variação da palavra em inglês *Boss (chefe)* e o bordão Bazinga usado pelo personagem Sheldon Cooper (Jim Parsons) na série *The Big Bang Theory* ou Teoria do *Big Bang* (título no Brasil).

## **ABSTRACT**

This work makes a study about the development (involving design, specification, implementation and evaluation) of a WebApp application, whose objective is the intermediation between interested in services and service providers. The tool is called Booszinga. The majority of service providers do not have the financial capital to advertise their new business in the short term, and the Booszinga tool meets this need. The tool assists service providers in the creation of a web page, available in an administrative area, in which they can register performance categories, portfolio, professional profile, prices practiced and describe their experiences. This information is made available on the online platform. Potential clients can consult and find the service provider that best meets their needs, for business closing. The system was developed using agile methodology with an influence of interactive and incremental modeling, using UML for specification. The implementation was done in the PHP language, with the help of the Codeigniter, Bootstrap and JQuery frameworks and the database created in MySQL, using the database management system (DBMS), phpmyadmin. The evaluation of the usability of the tool was done by specialists in IHC and by end users, and stated that the initial objectives were achieved satisfactorily.

**KEYWORDS:** Web App, Booszinga, Service providers.

## LISTA DE SIGLAS

IDE	Ambiente de Desenvolvimento Integrado
IHC	Interação Humano-Computador
MIT	Instituto de Tecnologia de Massachusetts
MVC	Modelo – Visão – Controlador
PHP	<i>Hypertext Preprocessor</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Linguagem de Consulta Estruturada
UFSC	Universidade Federal de Santa Catarina
UML	Linguagem de Modelagem Unificada

# SUMÁRIO

1. INTRODUÇÃO .....	12
1.1 Objetivos .....	13
1.1.1 Objetivo geral .....	13
1.1.2 Objetivos específicos .....	14
1.3 Justificativa.....	14
1.4 Organização do trabalho .....	15
1.5 Contribuições do trabalho.....	15
2. CONCEPÇÃO, ESPECIFICAÇÃO, IMPLEMENTAÇÃO E AVALIAÇÃO DE APLICAÇÃO WEB 16	
2.1 Concepção e Especificação da Ferramenta .....	16
2.2 Implementação.....	17
2.3 Avaliação da aplicação Web .....	18
3. TRABALHOS RELACIONADOS .....	20
3.1 Portal “Quem Conserta” .....	21
3.2 AchouServiços!.....	21
3.3 Bicos.....	22
3.4 Sr. Lupa.....	23
3.5 Clube do Autônomo .....	23
3.6 Agilize .....	24
4. REFERENCIAL TEÓRICO .....	25
4.1 Engenharia de software.....	25
4.1.1 Processo de Software.....	26
4.1.2 Modelo de Processo de Software .....	26
4.1.2.1 Movimento Ágil.....	27
4.1.2.1.1 Scrum .....	27
4.1.3 Especificação de Software.....	30
4.1.4 Documentar Requisitos Usando Modelos .....	30
4.1.4.1 UML .....	30
4.1.4.1.1 Diagrama de casos de uso.....	32
4.1.4.1.2 Diagrama de pacotes .....	33
4.1.4.1.3 Diagrama de classes.....	34

4.1.4.1.4	Diagrama de sequência .....	35
5.	FERRAMENTA E TECNOLOGIAS UTILIZADAS .....	37
5.1	PHP.....	37
5.2	Web Framework CodeIgniter.....	38
5.2.1	MVC .....	39
5.3	jQuery .....	40
5.4	MySQL.....	41
5.4.1	PHPMyAdmin .....	41
5.5	Trello.....	42
6.	ESPECIFICAÇÃO DE REQUISITOS, ANÁLISE, PROJETO E IMPLEMENTAÇÃO DA FERRAMENTA.....	44
6.1	Documento de Especificação de Requisitos do Booszinga .....	44
6.1.1	Ambiente do sistema .....	46
6.1.2	Objetivos do produto.....	47
6.1.3	Benefícios do produto .....	47
6.1.4	Modelos de Caso de Uso.....	48
6.1.4.1	Subsistema Painel Administrativo .....	49
6.1.4.2	Subsistema Plataforma de busca.....	54
6.4.5	Requisitos não funcionais .....	56
6.2	Documento de Especificação de Análise do Booszinga .....	57
6.2.1	Diagrama de Pacotes do Booszinga.....	57
6.2.2	Diagrama de Classes do Booszinga .....	58
6.2.3	Diagramas de Sequência do Booszinga .....	61
6.2.3.1	UC02: Manter Perfil.....	61
6.2.3.2	UC06: Área de buscas .....	62
6.2.4	Dicionário de Dados do Booszinga .....	63
6.3	Documento de Especificação de Projeto do Booszinga.....	65
6.3.1	Projeto de Interação Humano Computador.....	66
6.3.2	Projeto de Banco de Dados .....	68
6.3.3	Projeto de Componentes .....	72
6.4	Implementação da Ferramenta Booszinga .....	75
6.4.1	Planejamento.....	75
6.4.2	Banco de Dados .....	77
6.4.3	Ambiente de Desenvolvimento .....	80

6.4.4	Finalização do Desenvolvimento do Booszinga .....	82
7.	AVALIAÇÃO DA FERRAMENTA.....	83
7.1	Usabilidade .....	83
7.2	Funcionalidade.....	86
7.3	Confiabilidade .....	88
7.4	Eficiência.....	91
8.	CONCLUSÃO .....	94
	REFERÊNCIAS.....	95
	APÊNDICE A – DOCUMENTO DE ESPECIFICAÇÃO DE REQUISITOS .....	97
	APÊNDICE B – DOCUMENTO DE ESPECIFICAÇÃO DE ANÁLISE .....	107
	APÊNDICE C – DOCUMENTO DE ESPECIFICAÇÃO DE PROJETOS.....	119
	APÊNDICE D – DOCUMENTO DE APRESENTAÇÃO DO BOOSZINGA .....	137
	APÊNDICE E – QUESTIONÁRIO DE VALIDAÇÃO DO BOOSZINGA .....	146

## 1. INTRODUÇÃO

Gasparin (2011) aponta que, em 2008, houve o estouro da crise dos bancos nos Estados Unidos, marcada pela quebra do banco Lehman Brothers: isto fez com que investidores de todo o mundo tirassem suas aplicações em ações de empresas, de bancos e de títulos de governos, incluindo os do Brasil.

Para Sant'ana (2016), os efeitos da crise no Brasil repercutiram da seguinte maneira:

o aumento no número de demissões e o desejo de ser dono do próprio negócio fizeram com que quatro em cada dez brasileiros (39,3%) escolhessem o empreendedorismo como fonte de renda em 2015. Dados da pesquisa Global Entrepreneurship Monitor (GEM) mostram que existem 52 milhões de pessoas no país com idade entre 18 e 64 anos envolvidas na criação ou manutenção de algum negócio, o maior número registrado desde 2002.

Com esse cenário de desemprego, observou-se grande demanda por abertura de negócios, em especial na área de prestação de serviços. Uma questão se impôs: como dar visibilidade a prestadores de serviços por meio da internet, de modo a facilitar tanto a divulgação dos serviços como a conquista de potenciais clientes? A resposta que obtivemos foi por meio da ideia da criação de uma aplicação Web, em que os prestadores de serviços poderiam divulgar seus negócios de forma gratuita. Por meio de pesquisas nessa plataforma online, consumidores encontrariam os serviços que suprissem suas necessidades. Isso resultaria em ganhos para ambas as partes.

Para entender a importância da internet nos dias atuais, Cintra (2009, p. 6) diz que:

a grande parte do mundo está conectada pela internet. Relações pessoais, sociais e profissionais são firmadas pelo uso da grande rede. Isso é possível graças à facilidade, rapidez e eficiência desse recurso, fazendo com que grandes e pequenas empresas a usem para fechar seus negócios. É uma nova forma de se chegar ao consumidor final. O novo consumidor assiste a menos televisão, ouve menos rádio e opta por ver as notícias pela internet, onde são mais atualizadas em um espaço menor de tempo.

A internet é um conglomerado de informações sobre os mais variados assuntos, então o internauta faz uso de buscadores que indexam páginas na

web ou realizam buscas específicas em classificados para encontrar o que deseja em meio a uma profusão de materiais disponíveis.

Com o objetivo de promover a interação entre os prestadores de serviços e os consumidores, foi desenvolvida a ferramenta online Booszinga, que é alvo deste trabalho. Este *WebApp* (aplicativo Web) funciona como um classificado de prestadores de serviços e foi concebida como uma alternativa de baixo custo para divulgação de serviços por parte dos prestadores.

Com respeito ao desenvolvimento da ferramenta: buscou-se identificar um modelo de processo a ser utilizado no trabalho. Processo de software é definido por Sommerville *apud* Furtado e Costa Jr. (2010, p. 59) como “o conjunto ordenado de atividades que levam a resultados intermediários e que tem como fim a produção de software de qualidade”.

O processo é importante e deve ser levado em consideração, pois advém de planejamento e organização das atividades a estabilidade necessária para a criação de um software que atinja seus objetivos. Sem planejamento adequado do que se pretende fazer ou de quais objetivos alcançar, além de se ter descontrole e desorganização, obtem-se ao final um software de qualidade duvidosa e total desconhecimento do caminho trilhado para sua construção.

Para tanto, o modelo de processo de desenvolvimento de software adotado neste trabalho compreende a formulação de documentos de especificação de requisitos, de análise e de projeto. Para cumprimento das atividades também foi adotada a metodologia ágil, com o uso da ferramenta online Trello<sup>3</sup> para gerenciamento e organização das atividades.

Por fim, a ferramenta desenvolvida foi avaliada com a utilização do método de avaliação heurística, descrito adiante.

## 1.1 Objetivos

### 1.1.1 Objetivo geral

Desenvolvimento da *WebApp* Booszinga para promover a interação entre prestadores de serviços e consumidores finais pela internet.

---

<sup>3</sup> Trata-se de um aplicativo online que permite gerenciar projetos e tarefas. Esta ferramenta é um organizador de tarefas e eventos, inspirado na metodologia Scrum, processo de desenvolvimento para gerenciar projetos e desenvolvimento ágil de softwares ([www.seomaster.com.br](http://www.seomaster.com.br)). [SILVA *et al*, 2015]. Disponível em <http://www.trello.com>.

### 1.1.2 Objetivos específicos

Fazer o levantamento bibliográfico sobre os conteúdos relacionados à proposta;

Produzir a especificação necessária para o desenvolvimento do software, como documentos de especificação de requisitos, de análise e de projeto que possibilitem a implementação do software;

Operacionalizar a avaliação de usabilidade do software por potenciais usuários e por especialistas e realizar os ajustes necessários.

### 1.3 Justificativa

Quando identificada a necessidade de desenvolver uma ferramenta que possibilitasse o encontro entre prestadores de serviços e consumidores por meio da Internet, percebeu-se a inexistência de uma ferramenta conhecida que atendesse esse propósito.

Nos dias atuais, se um consumidor quer encontrar informações sobre algum determinado serviço do seu interesse, este recorre a ferramentas de buscas conhecidas como: Google, Bing, Yahoo, etc. Existe uma possibilidade de o usuário encontrar o que deseja, mas isso pode ser mais facilitado se estiver a sua disposição uma ferramenta online específica para isso.

Do outro lado temos o prestador de serviço. Este deve ter um *Site* na internet para aparecer nos buscadores citados. Criar um *site* e utilizar Marketing Digital para ter relevância entre diversos concorrentes. O Booszinga foi criado para atender a necessidade do prestador de serviço divulgar seus serviços na Internet de maneira barata ou gratuita. A ferramenta disponibiliza uma área administrativa em que o prestador de serviço pode descrever um pouco mais sobre o seu negócio, preços praticados, portfólio e informações de contato. Essas informações cadastradas geram uma página online em formato de *Site* “*one page*”<sup>4</sup> e pode ser acessado na plataforma por meio das buscas de consumidores.

A capitalização da aplicação será feita por meio da assinatura de Planos de Marketing, ou seja, o prestador de serviço que queira um destaque ou um

---

<sup>4</sup> São sites formados por apenas uma página.

posicionamento melhor nas buscas: devem assinar um Plano de Marketing que atenda sua necessidade, caso não queira destaque, ele será ranqueado e apresentado para o consumidor da mesma maneira que os demais que optaram pelo uso gratuito da ferramenta. Essa abordagem torna a aplicação sustentável, pois esse modelo é aplicado em buscadores como o Google por meio do Google *Adwords* que é um dos produtos do Google que vende o destaque dos *Sites* por meio de palavras chaves.

Este trabalho consistiu na concepção, implementação e avaliação da plataforma online Booszinga. Essa ferramenta de baixo custo facilita a interação entre prestadores de serviços e consumidores finais, a fim de que negócios sejam fechados e tragam benefícios para todos os envolvidos.

#### 1.4 Organização do trabalho

Este trabalho está estruturado da seguinte maneira: o Capítulo 2 apresenta Concepção, Implementação e Avaliação de Aplicação Web. O Capítulo 3 apresenta os Trabalhos Relacionados. O Capítulo 4 apresenta o Referencial Teórico. O Capítulo 5 apresenta Ferramentas e Tecnologias Utilizadas. O Capítulo 6 apresenta Especificação de Requisitos, Análise, Projeto e Implementação da Ferramenta. O Capítulo 7 apresenta Avaliação da Ferramenta. O trabalho é finalizado com a Conclusão e os Apêndices.

#### 1.5 Contribuições do trabalho

A principal contribuição do trabalho é mostrar claramente todas as etapas que compreendem o processo de desenvolvimento de uma *WebApp*, desde a concepção inicial do negócio retratado que vai atender dada parcela de usuários, até a avaliação final por estes usuários do software desenvolvido, ratificando tudo o que foi realizado em termos de escolha de procedimentos, ferramentas e métodos empregados.

## 2. CONCEPÇÃO, ESPECIFICAÇÃO, IMPLEMENTAÇÃO E AVALIAÇÃO DE APLICAÇÃO WEB

Este capítulo tem por objetivo expor o processo de desenvolvimento da aplicação Web. O processo de desenvolvimento é composto basicamente por 3 etapas: (i) concepção, (ii) implementação e (iii) avaliação. Cada etapa possui um objetivo e um foco bem definido.

Na etapa de concepção são definidas as diretrizes gerais da *WebApp*. Compreende a análise do modelo de negócio e a descrição das ferramentas utilizadas na especificação da aplicação. A etapa de implementação consiste na execução dos passos do processo de concepção e a descrição das ferramentas que foram utilizadas. A etapa de avaliação contempla a descrição da abordagem usada para avaliação da aplicação.

### 2.1 Concepção e Especificação da Ferramenta

A fase de concepção tem o objetivo de identificar o escopo inicial do projeto e verificar se é viável a continuidade do projeto, para depois produzir a documentação necessária para implementação da aplicação. A concepção da ferramenta proposta neste trabalho começou com a percepção do cenário econômico do país. O momento de crise proporcionou um *insight*<sup>5</sup>. A crise fez muitos profissionais perderem seus empregos e que jovens recém-formados na faculdade não tivessem chance ao primeiro emprego. Então, observou-se que o novo cenário fomentou o aumento de prestadores de serviços, autônomos e empreendedores que, agora, precisam de um lugar para divulgar seus serviços, a fim de formar sua base de clientes. A partir dessa percepção, partiu-se para a parte da pesquisa de mercado e elaboração de um Plano de Negócio Simplificado. Após verificar que a continuidade da aplicação online seria viável, a próxima etapa no processo foi o levantamento dos requisitos e funcionalidades que a ferramenta deveria ter para atender o que lhe é proposto.

Na conclusão desta etapa são elaborados os seguintes documentos:

---

<sup>5</sup> Insight é um substantivo com origem no idioma inglês e que significa compreensão súbita de alguma coisa ou determinada situação.

- Documento de especificação de requisitos:
  - Ambiente do sistema;
  - Objetivos do produto;
  - Benefícios do produto;
  - Modelos de caso de uso;
  - Restrições do software;
- Documento de especificação de análise:
  - Diagrama de classes;
  - Diagrama de sequência;
  - Dicionário de dados;
- Documento de especificação de projeto:
  - Plataforma computacional;
  - Arquitetura do sistema;
  - Projeto de interação humano-computador;
  - Projeto de banco de dados;
  - Projeto de componentes

Após a produção dos documentos listados, iniciou-se a implementação da ferramenta. São pressupostos da aplicação sua abrangência e o fato de lidar com usuários com variados níveis de instrução. Então, para fazer o nivelamento dos usuários do sistema, considerou-se a criação de uma área de ajuda na aplicação e que esteja disponível para todos os usuários e para resolver casos atípicos pode ser criado um suporte com atendimento online para os clientes da ferramenta, ou no primeiro momento disponibilizar um e-mail de contato para tirar as dúvidas.

## 2.2 Implementação

A fase de implementação é a fase mais demorada do processo de desenvolvimento, já que são realizadas as atividades de criação de banco de dados, codificação e testes iniciais. Tem como objetivo específico a criação da aplicação baseado na documentação elaborada na etapa de concepção.

Esta fase é constituída de quatro atividades básicas:

- Banco de dados: Realizar a criação das tabelas e relacionamentos especificados na documentação de especificação de projeto. É

importante que essa etapa seja a primeira a ser realizada, tendo em vista que se trata de toda a base da aplicação.

- Configuração do ambiente de desenvolvimento: Consiste em definir a linguagem de programação utilizada, o IDE (*Integrated Development Environment* – Ambiente de Desenvolvimento Integrado), liberação de permissões de acesso e ferramentas de apoio ao desenvolvimento como serviço de versionamento de código, dentre outras.
- Construção da aplicação: Fica a cargo dos *designers* e programadores. Consiste no desenvolvimento da ferramenta por meio do uso das tecnologias definidas no passo anterior. Essa atividade é responsável pela elaboração da interface com o usuário; para isso são usados as diretrizes e os princípios de Interação Humano-computador (IHC). No final dessa atividade, uma aplicação Web está pronta para os procedimentos finais.
- Finalização: Essa é a última atividade realizada durante a fase de implementação. O desenvolvedor verifica a necessidade de ajustes e realiza alguns testes básicos com o objetivo de validar se o sistema atende em um primeiro momento o que foi proposto. Não se trata de um teste definitivo; este é realizado por um profissional de testes que não tem relação com quem desenvolveu a aplicação. As documentações produzidas durante a fase de concepção podem ser atualizadas, se houver necessidade de ajustes descobertos durante a fase de implementação. Ao término dessa etapa, o software está pronto para avaliação.

### 2.3 Avaliação da aplicação Web

A aplicação Web foi construída com o objetivo de promover a interação entre prestadores de serviços e consumidores. Para saber se a ferramenta realmente atenderia a necessidade do público alvo foi realizada uma avaliação de usabilidade.

A técnica de coleta de informação empregada na avaliação de usabilidade foi a aplicação de questionário. Esta técnica investigativa foi escolhida por ser uma forma barata de coletar informações, por atingir uma grande população e

de não necessitar da presença de um entrevistador. Este último aspecto – ausência do entrevistador – exige que o questionário contendo questões claras, objetivas e não ambíguas. O questionário aplicado é formado tanto por perguntas abertas e fechadas.

Na fase de avaliação foram escolhidos entre cinco a dez usuários cadastrados na plataforma e para estes foram enviados o questionário de avaliação.

A avaliação levou em consideração a “árvore de requisitos de qualidade” de Olsina et al (1999) *apud* Pressman (2011, p. 339): “que identifica um conjunto de atributos técnicos – usabilidade, funcionalidade, confiabilidade e eficiência – que levam a *WebApps* de alta qualidade”<sup>6</sup>. No Capítulo 6 o questionário aplicado neste projeto é apresentado e descrito.

---

<sup>6</sup> Foi retirado o tópico facilidade de manutenção, pois não cabe ao usuário responder perguntas de nível de implementação de software.

### 3. TRABALHOS RELACIONADOS

O avanço tecnológico chegou a tal ponto que a criação de um produto ou de uma ideia genuinamente nova é rara.

Antecedendo o desenvolvimento desta ferramenta, realizamos pesquisas com o objetivo de encontrar trabalhos relacionados ou sistemas online semelhantes ao apresentado neste trabalho. No Quadro 1 são apresentadas as expressões de buscas utilizadas e os resultados obtidos. Mais à frente será detalhado cada item deste Quadro.

Quadro 1. Expressões de busca por trabalhos relacionados.

Expressão usada na busca	Plataforma de busca	Resultados	Tipo
Desenvolvimento site prestador serviço	Google Acadêmico ( <a href="https://scholar.google.com.br/">https://scholar.google.com.br/</a> )	PORTAL “QUEM CONSERTA” ( <a href="http://projetos.inf.ufsc.br/arquivos_projetos/projeto_243/ProjetoII.pdf">http://projetos.inf.ufsc.br/arquivos_projetos/projeto_243/ProjetoII.pdf</a> )	TCC
Encontre prestador de serviço	Google ( <a href="https://www.google.com.br/">https://www.google.com.br/</a> )	AchouServiços! ( <a href="https://www.achouservicos.com.br/">https://www.achouservicos.com.br/</a> ) <hr/> Bicos ( <a href="https://www.bicos.com.br/">https://www.bicos.com.br/</a> )	Site
Encontre autônomos	Google ( <a href="https://www.google.com.br/">https://www.google.com.br/</a> )	Sr. Lupa ( <a href="https://www.srlupa.com.br/">https://www.srlupa.com.br/</a> ) <hr/> Clube do Autônomo ( <a href="http://clubedoautonomo.com.br/site/">http://clubedoautonomo.com.br/site/</a> ) <hr/> Agilizze ( <a href="http://agilizze.com/">http://agilizze.com/</a> )	Site

### 3.1 Portal “Quem Conserta”

Desenvolvido em 2004 pelo discente Diego Teixeira de Mello da UFSC (Universidade Federal de Santa Catarina), com orientação do Professor Rui César Quinhões Pinto e co-orientação do Professor Edvardo Bonfim Rodrigues Junior.

Segundo Mello (2004), o Portal *Quem Conserta* permite a aproximação de profissionais executores de serviços do público consumidor, que busca localizar estes profissionais por meio de pesquisas nas páginas do portal, utilizando imagens e recursos gráficos e selecionando-os de cada grupo de especialização (eletricista, mecânico, bombeiro hidráulico, etc.).

Alguns recursos interessantes são citados como ranqueamento dos resultados das pesquisas baseados em média de pontuação e um acesso ao *site* do PROCON para saber se aquele profissional requisitado tem alguma reclamação registrada no órgão.

O trabalho relacionado à ferramenta “Quem Conserta” é do ano de 2004. As informações contidas no trabalho estão desatualizadas. O trabalho chega a informar o link <http://www.quemconserta.com.br>, mas encontra-se fora de operação. Foi observado que não existe preocupação em monetizar a aplicação. Não existe Plano de Negócios construído ou algum interesse em fazer um estudo de viabilidade da ferramenta.

Como diferenciais da ferramenta desenvolvida neste trabalho em relação ao descrito no artigo, apontamos: as versões das tecnologias Web utilizadas são atualizadas. A implementação da aplicação *WebApp* proposta está disponível para uso em <http://www.booszinga.com>. A ferramenta foi desenvolvida a partir de plano de negócios que havia atestado que a sua construção era viável.

### 3.2 AchouServiços!

A *WebApp* AchouServiço! funciona de uma forma similar à que é proposta neste trabalho. Dispõe de um leiaute atualizado e funcional. Algumas das funcionalidades da ferramenta são: salvar os profissionais favoritos, ver amigos que já avaliaram profissionais, fazer avaliações exige um cadastro prévio na

plataforma.

A ferramenta segue os seguintes passos para atender os objetivos:

1. Consumidor Final

- a. Pesquise o profissional que você precisa, você terá acesso a todos os profissionais cadastrados no *site*.
- b. Analise o perfil do profissional, suas avaliações e trabalhos realizados.
- c. Feche o negócio diretamente com o profissional, sem intermediários.

2. Prestador de Serviço

- a. Cadastre-se no *site* como profissional para poder divulgar seus serviços e preencha todo o seu perfil, descrição e fotos.
- b. Divulgue sua página para seus conhecidos, clientes e compartilhe nas redes sociais.

De todos os resultados obtidos, esta aplicação Web segue os mesmos princípios e tem os mesmos objetivos e funcionalidades do Booszinga.

### 3.3 Bicos

Segundo o *site* da ferramenta Bicos, trata-se de:

“Uma plataforma digital inovadora que traz um ambiente seguro e dinâmico para a busca de mão de obra qualificada em assuntos domésticos.

Ela foi fundada em dezembro de 2014 pelo publicitário Marcos de Arruda Botelho.

Nosso objetivo é facilitar a vida dos nossos clientes, conectando quem precisa de uma força em casa a prestadores de serviço competentes, de forma inovadora e ágil.

Bicos é simples, rápido e fácil. ”

A aplicação apresenta uma página um pouco confusa, pois não é direta para atender o objetivo que o *site* propõe. Para começar a pesquisar profissionais é necessário encontrar e clicar no botão “encontre oportunidade de serviço”. Ao clicar no botão mencionado, há redirecionado para uma página que apresenta um resultado prévio de alguns serviços que precisam de pessoas para atendê-los.

O *site* oferece uma metodologia oposta à proposta pelo Booszinga, já que

o consumidor final procura por oportunidades de serviços e na solução proposta neste trabalho o consumidor final procura por serviços que atendam às suas necessidades.

### 3.4 Sr. Lupa

Segundo as informações encontradas no *site* da ferramenta. O Sr. Lupa é

“uma plataforma comunitária confiável destinada a profissionais autônomos que anunciam seus serviços e clientes que possuem alguma necessidade. A intenção é fazer com que se descubram uns aos outros e façam conexões de maneira rápida, ágil e assertiva, solucionando seus problemas através de uma experiência única; seja de um computador, de um celular ou de um tablet. “

A plataforma atende os requisitos de qualidade citados por Pressman (2011, p. 339) que são: “usabilidade, funcionalidade, confiabilidade e eficiência”. Possui um *design* limpo e direto. A aplicação faz o intermédio e a negociação entre o autônomo e o prestador de serviço por meio de créditos comprados na plataforma. O Booszinga não usa essa técnica, deixa livre a negociação entre prestador de serviço e consumidor final.

### 3.5 Clube do Autônomo

Segundo informações encontradas no *site* da ferramenta, o Clube do Autônomo tem como objetivo “suprir ambas as necessidades, para quem busca ou oferece esses serviços. Ser o elo entre os melhores profissionais e quem busca serviço de qualidade. Ser o melhor *site* de classificados para divulgação de serviços”.

Diferentemente da plataforma anterior, esta não atende os requisitos de qualidade de uma *WebApp*, além de não estar preocupada com a adequação da tela em outros tipos de dispositivos como celulares ou *tablets*.

Apesar do leiaute inadequado e ultrapassado, a ferramenta é direta e atende os objetivos propostos, mas isso não é o suficiente para tratá-la como parâmetro de melhorias para este trabalho.

### 3.6 Agilizze

O *Site* Agilizze propõe a busca, cadastro de profissionais e cadastro de vagas de emprego. A aplicação é bem direta em cumprir esse objetivo. Já na tela principal há um resumo dos últimos profissionais cadastrados, as vagas de empregos disponíveis e um campo para buscar profissionais ou cadastrar profissionais.

Igual a plataforma citada anteriormente, esta também não atende os requisitos de qualidade de uma *WebApp*. Consegue atender o que propõe, mas de maneira não atrativa.

Após fazer uma análise da concorrência, foi possível ver que o Booszinga tem entrada neste mercado, sendo possível ir para o próximo passo, que é o levantamento bibliográfico necessário para o desenvolvimento do trabalho, assunto este que é tratado no próximo capítulo.

## 4. REFERENCIAL TEÓRICO

Este Capítulo tem por objetivo descrever os principais conceitos, métodos e ferramentas empregadas neste trabalho. O capítulo está organizado da seguinte forma: Primeiramente a área em que se enquadra o desenvolvimento da ferramenta é definida - Engenharia de Software. Em seguida, são apresentadas as características das *WebApps*, fechando o Capítulo.

### 4.1 Engenharia de software

SWEBOK *apud* Furtado e Costa Jr. (2010, p.51) define Engenharia de Software como “a aplicação de uma abordagem sistemática, disciplinada, quantificável ao desenvolvimento à operação e à manutenção de software, isto é, a aplicação de engenharia ao software”. Acrescenta ainda que a Engenharia de Software inclui o estudo de abordagens que levem a disciplina de engenharia à área de software. Pressman (2011) aponta que, além de disciplina, deve-se acrescentar adaptabilidade e agilidade, qualidades exigidas pela atual demanda de computação.

Sommerville (2011) define a engenharia de software como uma disciplina da engenharia que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação.

Sommerville (2011) afirma que a área de entretenimento, incluindo a indústria da música, jogos de computador, cinema e televisão, faz uso intensivo de software. Portanto, a engenharia de software é essencial para o funcionamento de sociedades nacionais e internacionais.

Observamos então a importância da engenharia de software para a sociedade e o mundo globalizado. O software ganhou um papel principal nesse novo mundo. Para evitar perdas e retrabalhos durante o processo de criação de software, faz-se necessário o uso das técnicas de engenharia de software.

#### 4.1.1 Processo de Software

Sommerville (2011, p. 5) define um processo de software como “uma sequência de atividades que leva à produção de um produto de software”. Pressman (2011, p. 40) tem uma visão parecida e define um processo de software como “um conjunto de atividades, ações e tarefas realizadas na criação de algum produto de trabalho.”

Cada instituição pode adotar o modelo de processo de software que julgar necessário. Nada impede o uso de modelos ultrapassados, já que não existe um modelo ideal.

#### 4.1.2 Modelo de Processo de Software

Sommerville (2011) diz que o modelo de processo de software é uma descrição simplificada de um processo de software, que é apresentada a partir de uma perspectiva específica. Os modelos, pela sua natureza, são simplificações; e, assim, um modelo de processo de software é uma abstração do processo real que está sendo descrito.

Furtado e Costa Jr (2010) citam que os modelos de processo de software mais utilizados são:

o [Pressman, 2006], [Sommerville, 2008], [Gustafson, 2003], [SWEBOK, 2004]:

- Modelo cascata;
- Modelo de prototipagem;
- Modelo incremental;
- Modelos evolucionários;
- Modelo baseado em componentes reutilizáveis;
- Modelo espiral.

Furtado e Costa Jr (2010) mostram ainda que outros modelos de processos que precisam ser citados: modelo de engenharia de software sala limpa, modelo de engenharia da web, métodos formais, modelo de desenvolvimento orientado a aspectos.

#### 4.1.2.1 Movimento Ágil

O Movimento Ágil é um modelo de processo não prescritivo, ou seja, não estabelece uma estruturação e ordenação de tarefas [Furtado e Costa Jr., 2010]. Sommerville (2011) afirma que a metodologia prescritiva dominava a área de engenharia de software no final da década de 1980, e que a maneira para conseguir o melhor software era por meio de um planejamento cuidadoso do projeto, qualidade da segurança formalizada, dentre outras coisas que levavam a um desenvolvimento de software rigoroso e controlado.

Um grande número de desenvolvedores estava insatisfeito com esse tipo de abordagem pesada; então, na década de 1990 propuseram os “métodos ágeis”.

Sommerville (2011, p. 39) apresenta o manifesto ágil. Esse manifesto afirma:

Estamos descobrindo melhores maneiras de desenvolver softwares, fazendo-o e ajudando outros a fazê-lo. Através desse trabalho, valorizamos mais: Indivíduos e interações do que processos e ferramentas;  
Software em funcionamento do que documentação abrangente;  
Colaboração do cliente do que negociação de contrato;  
Respostas a mudanças do que seguir um plano;  
Ou seja, embora itens à direita sejam importantes, valorizamos mais os que estão à esquerda.

O manifesto ágil ganhou bastante força e adeptos desde o seu lançamento. Isso pode ser observado nas empresas de desenvolvimento de software e nas comunidades de desenvolvedores, que são responsáveis por eventos e circuitos de palestras que divulgam e apontam as vantagens de ser ágil.

Sommerville (2011, p. 40) cita os métodos ágeis mais conhecidos: Scrum (COHN, 2009; SCHWABER, 2004; SCHWABER e BEEDLE, 2001), que será tratado adiante. Extreme Programming (BECK, 1999; BECK 200), Crystal (COCKBURN, 2001; COCKBURN, 2004), etc.

##### 4.1.2.1.1 Scrum

O Scrum, segundo um dos seus autores, é um *framework* para desenvolver e manter produtos complexos. Schawaber e Sutherland (2013, p. 3) definem o Scrum da seguinte maneira:

Scrum não é um processo ou uma técnica para construir produtos; em vez disso, é um *framework* dentro do qual você pode empregar vários processos ou técnicas. O Scrum deixa claro a eficácia relativa das práticas de

gerenciamento e desenvolvimento de produtos, de modo que você possa melhorá-las.

O Scrum é formado por artefatos e eventos. Os artefatos são:

- *Backlog do Produto*: é uma lista ordenada de tudo que é necessário no produto, tendo como fonte os requisitos;
  - *Backlog da Sprint*: é uma lista de tarefas que serão realizadas em uma determinada *Sprint*;
  - *Incremento*: é a soma de todas as tarefas realizadas em determinada *Sprint* com as tarefas de *Sprints* anteriores;
- Os eventos do Scrum são:
- *Sprint*: são ciclos com duração de um mês ou menos. Este assunto é melhor tratado mais à frente nesta seção.
  - *Revisão da Sprint*: realizado ao final de cada *Sprint*, para avaliar se os objetivos foram alcançados. O refinamento do *Backlog* também é feito nesta fase.
  - *Retrospectiva da Sprint*: é um evento de reavaliação por parte do Time Scrum, onde este pode ver em quais pontos podem melhorar. Este evento ocorre após a revisão da *Sprint*.

Schawaber e Sutherland (2013) definem o Time Scrum com os seguintes componentes: Product Owner, Time de Desenvolvimento e o Scrum Master.

O Product Owner é o responsável por maximizar o valor do produto e do trabalho do Time de Desenvolvimento. Uma das responsabilidades do Product Owner é o gerenciamento do Backlog do Produto. Sommerville (2011) define o *backlog* como o ponto de partida do produto. Trata da lista do trabalho a ser feito no projeto. O *backlog* é revisado durante a fase de avaliação da *Sprint*, e as prioridades e os riscos são identificados. O cliente que está envolvido no projeto, pode introduzir novos requisitos ou tarefas no *backlog*.

Schawaber e Sutherland (2013) afirmam que o Time de Desenvolvimento é o responsável pela entrega de uma versão usável que potencialmente incrementa o produto ao final de cada *Sprint*.

O Scrum Master é definido por Schawaber e Sutherland (2013) como o responsável por garantir que o Scrum seja entendido e aplicado. O Scrum Master tem o papel de garantir que o Time Scrum adira à teoria, práticas e regras do Scrum.

Sommerville (2011) aborda sobre as três fases do Scrum: a primeira fase é de planejamento geral, em que os objetivos gerais são estabelecidos. Em seguida, ocorre uma série de ciclos de *Sprint*, estes ciclos incrementam o sistema. Finalmente, a última fase do projeto encerra o projeto, fazendo uma

avaliação geral das lições aprendidas.

A fase principal deste *framework* são os ciclos de *Sprint*. Schawaber e Sutherland (2013) afirmam que a *Sprint* é o coração do Scrum, com uma duração de um mês ou menos. É nesta fase que ocorre todo o esforço de desenvolvimento para entregar um incremento do sistema. Ao final de cada *Sprint* é realizada uma revisão desta, com o objetivo de avaliar o que foi feito e o que pode melhorar. No final da *sprint* é entregue um incremento.

Por se tratar de um *framework*, o Scrum pode ser adaptado à necessidade. Na conclusão do guia do Scrum os autores Schawaber e Sutherland (2013) deixam claro que o Scrum funciona em sua totalidade, e os papéis definidos no guia são imutáveis. Então para que o resultado seja Scrum, o *framework* deve ser usado na sua totalidade. O fluxo de funcionamento do Scrum pode ser visto na Figura 1.

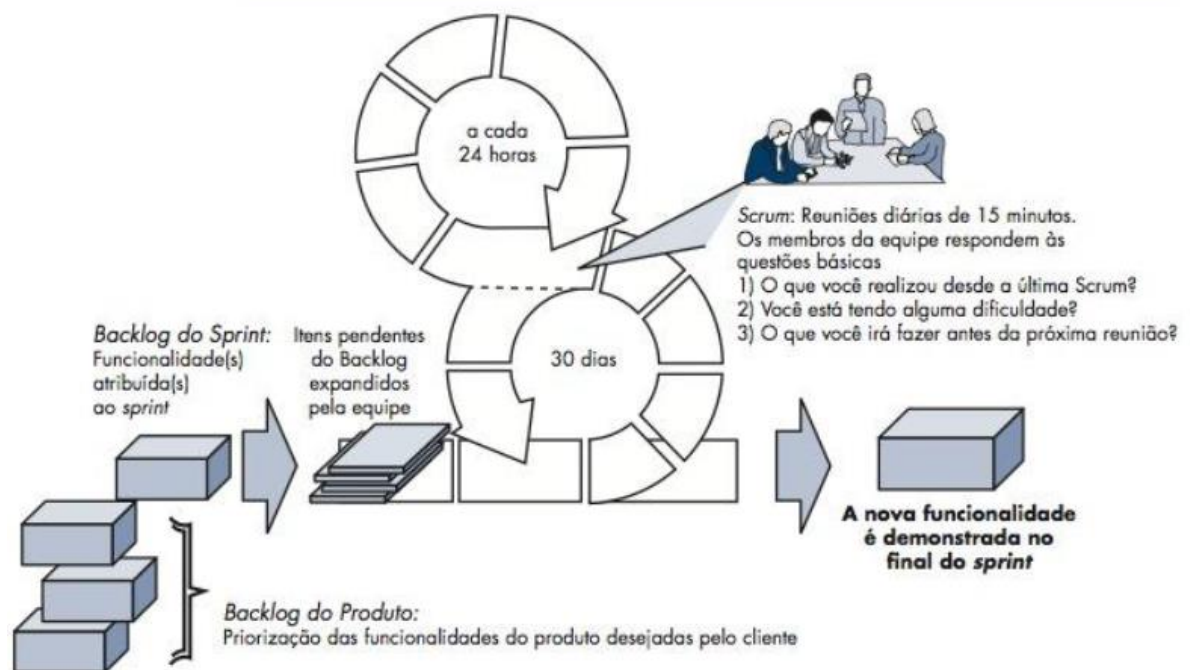


Figura 1. Fluxo do processo do Scrum.

Fonte: (Pressman, 2011, p. 96)

### 4.1.3 Especificação de Software

Sommerville (2011, p. 24) define especificação de software ou engenharia de requisitos da seguinte forma:

é o processo de compreensão e definição dos serviços requisitados do sistema e identificação de restrições relativas à operação e ao desenvolvimento do sistema. A engenharia de requisitos é um estágio particularmente crítico do processo de software, pois erros nessa fase inevitavelmente geram problemas no projeto e na implementação do sistema.

Para entender o que será o software ou qual será seu objetivo é necessário fazer o levantamento dos requisitos. Para POHL e RUPP (2012), a etapa de eliciação de requisitos é a atividade central da engenharia de requisitos e que a base é formada pelo conhecimento sobre o contexto do sistema, com o uso de fontes de requisitos como: *Stakeholders*<sup>7</sup>, documentos e sistemas em operação.

### 4.1.4 Documentar Requisitos Usando Modelos

A documentação de requisitos será de uso de diversos profissionais durante o desenvolvimento do software. Para que haja um entendimento claro do que está documentado, se faz necessário o uso de uma linguagem padrão e conhecida por todos. A UML (*Unified Modeling Language*) é frequentemente usada para construir modelos de requisitos [OMG 2007] *apud* Pohl e Rupp (2012, p. 89).

#### 4.1.4.1 UML

A UML é definida por Furtado e Costa Jr (2010) como “uma linguagem para especificação de artefatos de software, assim como para modelagem de negócios”.

---

<sup>7</sup> São pessoas ou organizações que influenciam os requisitos de um sistema.

O guia do usuário escrito por Booch *et al* (2006, p. 17) que são autores da UML tem uma definição mais completa:

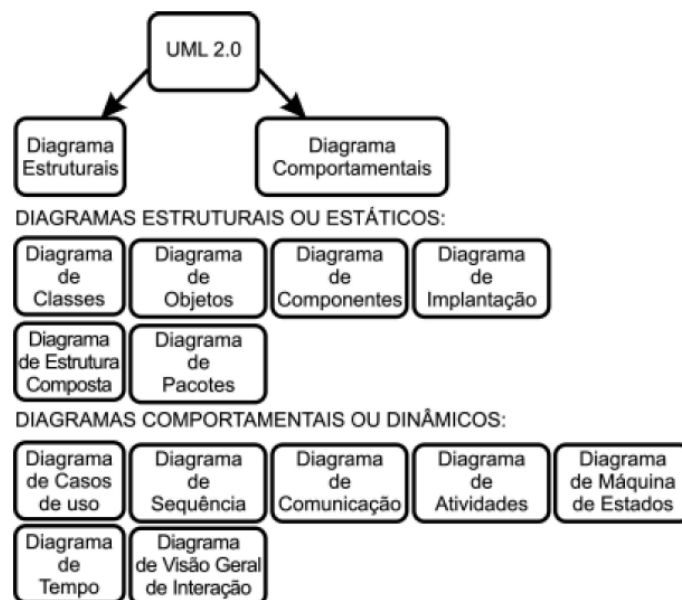
é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexo de software. A UML proporciona uma forma-padrão para preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de banco de dados e componentes de software reutilizáveis.

Ela é composta por itens, relacionamentos e diagramas que facilitam um melhor entendimento da modelagem. São eles:

- Os itens podem ser:
  - Estruturais (classe, interface, casos de uso e componentes);
  - Comportamentais (interações, máquinas de estado);
  - Grupos de elementos (pacotes, *frameworks* e subsistemas);
  - Anotações (notas).
- Existem quatro tipos de relacionamentos na UML:
  1. Dependência;
  2. Associação;
  3. Generalização
  4. Realização.
- Segundo Booch *et al* (2006, p. 26) o diagrama faz a combinação entre os itens e os relacionamentos para formar um conjunto de representação gráfica para dar entendimento sobre o sistema. A UML na sua versão 2.0 inclui 13 diagramas:
  1. Diagrama de classes
  2. Diagrama de objetos
  3. Diagrama de componentes
  4. Diagrama de estrutura compostas
  5. Diagrama de casos de uso
  6. Diagrama de sequências
  7. Diagrama de comunicações
  8. Diagrama de gráfico de estados

9. Diagrama de atividades
10. Diagrama de implantação
11. Diagrama de pacote
12. Diagrama de temporização
13. Diagrama de visão geral da interação

A Figura 2 mostra os diagramas separados em dois subgrupos: Estruturais e comportamentais.



**Figura 2. Diagramas da UML 2.0.**

**Fonte: (Furtado & Costa Jr, 2010, p. 106)**

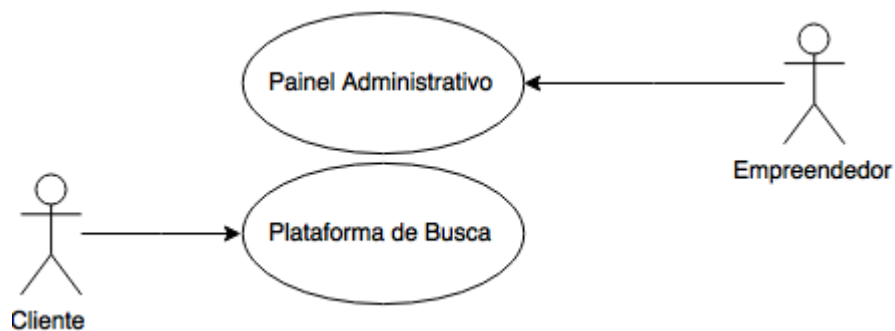
Os diagramas usados neste trabalho são: diagrama de casos de uso, diagrama de pacotes, diagrama de classes e diagrama de sequência. A maioria dos diagramas citados são indicados por Furtado e Costa Jr (2010) como os que mais se destacam pela grande frequência de uso na especificação de software.

#### 4.1.4.1.1 Diagrama de casos de uso

Para Furtado e Costa Jr (2010), o diagrama de casos de uso é o primeiro diagrama a ser construído pelo analista de sistema no seu trabalho de especificação. Os autores definem o diagrama de casos de uso como “o diagrama da UML utilizado para especificar os requisitos funcionais de um

sistema”.

O diagrama de casos de uso de um sistema apresenta todas as interações possíveis que deverão estar descritas nos requisitos do sistema. É fácil entender a simbologia do diagrama de casos de uso. Temos os atores representados por “bonecos-palito” que podem ser pessoas ou outros sistemas. As classes de interação são representadas por uma elipse. As ligações entre atores e interação é feita por meio de linhas. Pontas de flechas podem ser adicionadas opcionalmente para mostrar a direção da interação Sommerville (2011). Na Figura 3 é ilustrado um caso de uso do Booszinga. Este caso de uso faz a representação do contexto do sistema, temos os atores: Empreendedor que faz uso do Painel Administrativo e o Cliente que interage com a Plataforma de Busca.



**Figura 3. Diagrama de casos de uso do Booszinga**

#### 4.1.4.1.2 Diagrama de pacotes

O diagrama de pacotes é uma forma de representar em alto nível a forma como o sistema está distribuído em módulos ou subsistemas menores. Booch et al (2006, p. 168) define pacotes como:

um mecanismo de propósito geral para organizar elementos em grupos. Os pacotes ajudam a organizar os elementos em modelos, de maneira que você seja capaz de compreendê-los com maior facilidade. Os pacotes também permitem controlar o acesso a seus conteúdos, de modo que você possa controlar as costuras existente na arquitetura do sistema.

O pacote é representado graficamente por uma pasta como mostra a Figura 4. Neste exemplo está representado o submódulo da Área Administrativa. Esse pacote foi criado para organizar e separar a área administrativa do restante da aplicação, ou seja, o que estiver no pacote admin é relativo ao gerenciamento e administração das informações por parte do prestador de serviços e o que estiver fora é relacionado às buscas dos consumidores.

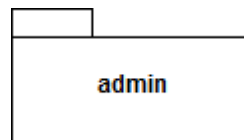


Figura 4. Exemplo de diagrama de pacotes do Booszinga.

#### 4.1.4.1.3 Diagrama de classes

O diagrama de classes é tido como o diagrama mais importante da UML, devido estar sempre presente em toda análise de sistema. Booch *et al* (2006) o definem da seguinte forma: “um diagrama de classes mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos. Graficamente, um diagrama de classes é uma coleção de vértices e arcos”.

Segundo Fowler & Scott (1997) *apud* Furtado e Costa Jr (2010, p. 196) afirma que “um diagrama de classes descreve os tipos de objetos encontrados no sistema (ou seja, as classes dos objetos) e os diversos tipos de relações estáticas (associações herança) que existem entre esses diversos tipos”.

A Figura 5 mostra a classe Pagina e seus atributos por meio do diagrama de classes.

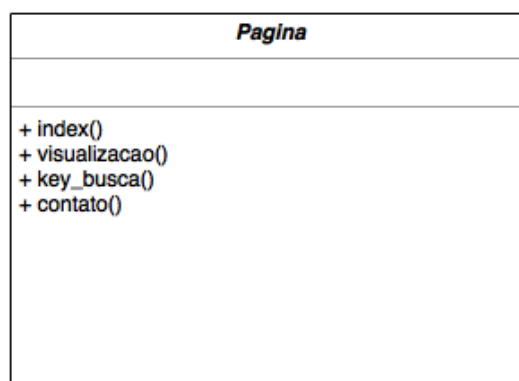


Figura 5. Exemplo de um diagrama de classes do Booszinga.

#### 4.1.4.1.4 Diagrama de sequência

O diagrama de sequência faz parte do conjunto de Diagramas de Interações. Furtado e Costa Jr (2010, p. 214) definem que este diagrama

mostra as interações entre os objetos, organizadas sequencialmente no tempo, para realizar uma funcionalidade do sistema. Este diagrama está associado com os diagramas de casos de uso: as funcionalidades que ele expressa são aquelas que aparecem nos diagramas de casos de uso. A construção do diagrama de sequência deve respeitar o que dispõe o diagrama de classes da aplicação.

Sommerville (2011) mostra a forma de leitura do diagrama de sequência: o diagrama de sequência pode ser lido da seguinte forma: na parte superior temos os atores e objetos. Interações entre objetos são feitas por linhas anotadas. A linha de vida do objeto é indicada por um retângulo seguido de uma linha tracejada que a leva ao objeto. As sequências de interações devem ser lidas de cima para baixo.

A Figura 6 mostra um exemplo de diagrama de sequência usado neste trabalho. O diagrama começa com o ator Empreendedor informando e-mail e senha na Interface de Tela de *Login*. Na sequência é usado a função “validar\_login” da classe de controle Pessoa. A classe modelo Pessoa\_model recupera a informação sobre o login informado por meio da função “select\_login” que tem como parâmetros de entrada: e-mail e senha. O controlador Pessoa recebe os dados do usuário e repassa para Tela de Login. Caso as credenciais estejam corretas é liberado o acesso à Interface do Painel Administrativo, caso contrário o acesso é negado e o ator permanece na Tela de Login.

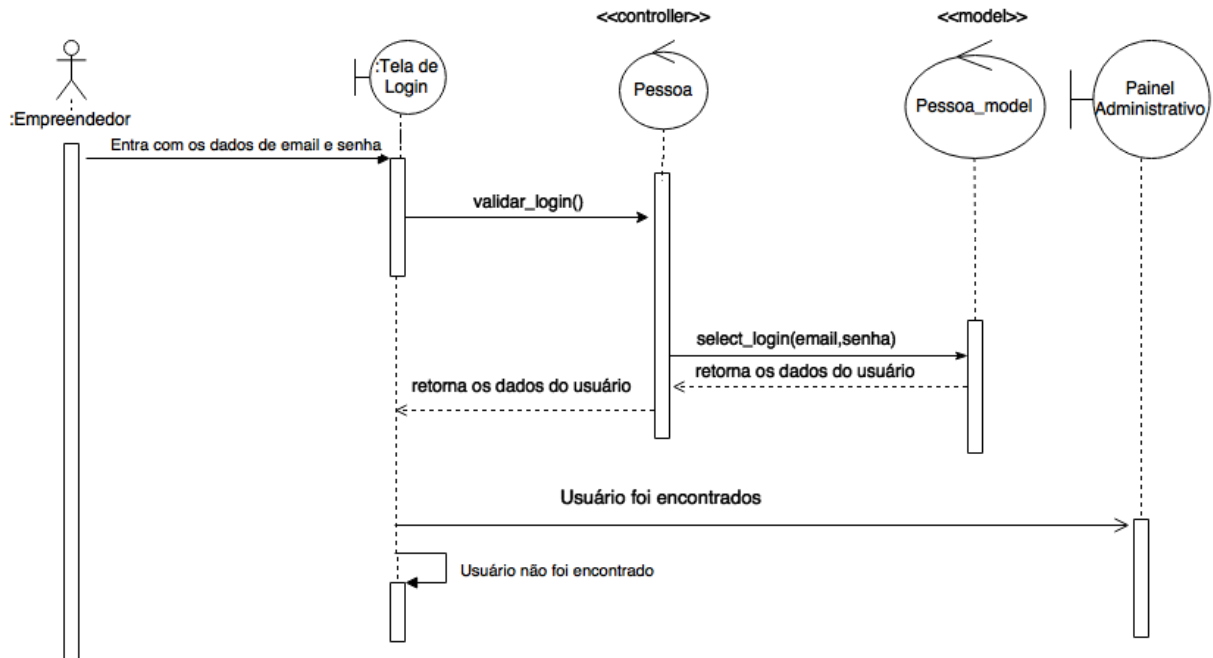


Figura 6. Exemplo de diagrama de sequência do Booszinga.

Depois de levantada toda a bibliografia necessária para o desenvolvimento do trabalho. O próximo passo foi a definição das tecnologias utilizadas, assunto este abordado no próximo Capítulo.

## 5. FERRAMENTA E TECNOLOGIAS UTILIZADAS

Este Capítulo tem como objetivo mostrar uma breve descrição das principais ferramentas e tecnologias utilizadas no desenvolvimento e implementação da *WebApp* Booszinga. O sistema foi desenvolvido na linguagem PHP, com o auxílio dos *Frameworks Codeigniter* e *JQuery* e a base de dados foi criada em MySQL, utilizando o sistema de gestão de base de dados (SGBD), *phpmyadmin*. Para aplicar a metodologia de processo ágil foi usado a ferramenta *Trello*. Essas tecnologias e ferramentas são descritas em seguida.

### 5.1 PHP

Trata-se de uma linguagem de programação *Web Open Source*<sup>8</sup> bastante difundida. No *Site* da linguagem mostra a seguinte definição: o PHP (um acrônimo recursivo para *PHP: Hypertext Preprocessor*) é uma linguagem de *script open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML. A Figura 7 mostra um exemplo simples da linguagem.

Para o desenvolvimento da aplicação deste trabalho foi usada a versão 5 do PHP. Atualmente, encontra-se liberada a versão 7.

É considerada uma linguagem que executa do lado do servidor. Quando você acessa uma página PHP por meio de seu navegador, todo o código PHP é executado no servidor, e os resultados são enviados para seu navegador. Portanto, o navegador exibe a página já processada, sem consumir recursos de seu computador. Niederauer (2011) lembra que as linhas de programação PHP não podem ser vistas por ninguém, já que elas são executadas no próprio servidor, e o que retorna é apenas o resultado do código executado.

---

<sup>8</sup> A definição de Open Source foi criada por Bruce Perens e Eric Raymond, sendo atualmente mantido pela *Open Source Initiative*, adicionando significado adicional ao termo. Como exemplo, temos a possibilidade de não somente ter acesso total ao Código Fonte, mas também o direito de usá-lo. Caso o direito de uso irrestrito seja negado, a licença é categorizada como uma licença para Software Comercial.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Exemplo</title>
  </head>
  <body>

    <?php
      echo "Olá, eu sou um script PHP!";
    ?>

  </body>
</html>
```

Figura 7. Exemplo de código em PHP embutido no HTML.

## 5.2 Web Framework CodeIgniter

Quando o interesse é ganho de produtividade, pode-se utilizar *Frameworks*, o que evita que a tarefa comece do zero. Gabardo (2012) afirma que um *Framework* é um conjunto de ferramentas e funções desenvolvido por um grupo de pessoas e/ou empresa (s), a fim de solucionar problemas e necessidades comuns, criando uma plataforma prática e segura que servirá como base para o desenvolvimento de seu *website* ou aplicação.

O *CodeIgniter* está sob a licença do MIT (*Massachusetts Institute of Technology*), que segundo Sabino e Kon (2009):

trata-se de uma licença permissiva, afirmando que qualquer pessoa que obtém uma cópia do software e seus arquivos de documentação associados pode lidar com eles sem restrição, incluindo sem limitação os direitos a usar, copiar, modificar, mesclar, publicar, distribuir, sublicenciar e/ou vender cópias do software. As condições impostas para tanto são apenas manter o aviso de *copyright* e uma cópia da licença em todas as cópias ou porções substanciais do software.

O *framework* é descrito em seu guia de usuário da seguinte maneira: “CodeIgniter é um toolkit (conjunto de ferramentas) para quem constrói aplicações web usando PHP. Seu objetivo é lhe dar a capacidade de desenvolver projetos com muito mais rapidez do que se você estivesse escrevendo código-fonte do zero, oferecendo um conjunto rico de bibliotecas

para tarefas de uso comum, assim como uma interface simples e estrutura lógica para acessar as bibliotecas. O *CodeIgniter* permite que você se concentre na criatividade e no desenvolvimento do seu projeto, minimizando a quantidade escrita de código-fonte necessária para executar a mesma tarefa.”

Para o desenvolvimento e implementação da ferramenta proposta neste trabalho, foi usada a versão 3.0 do *Framework* que é a sua versão atual até o momento.

Constam no guia do usuário do *CodeIgniter* os seguintes objetivos e características:

- **Instanciamento dinâmico.** No *CodeIgniter*, componentes são carregados e rotinas executadas somente quando preciso, ao invés de globalmente;

- **Junção de componentes.** Os componentes do *framework* são intercomunicativos, proporcionando alto índice de reutilização e flexibilidade dos sistemas baseados/derivados;

- **Singularidade dos componentes.** No *CodeIgniter*, cada classe – e respectivas funções – é autônoma, o que permite elevar o grau de utilidade e o sistema, como um todo, ter mais performance.

O *framework* utiliza o MVC (*Model-View- Controller*) como padrão de design de projetos de software.

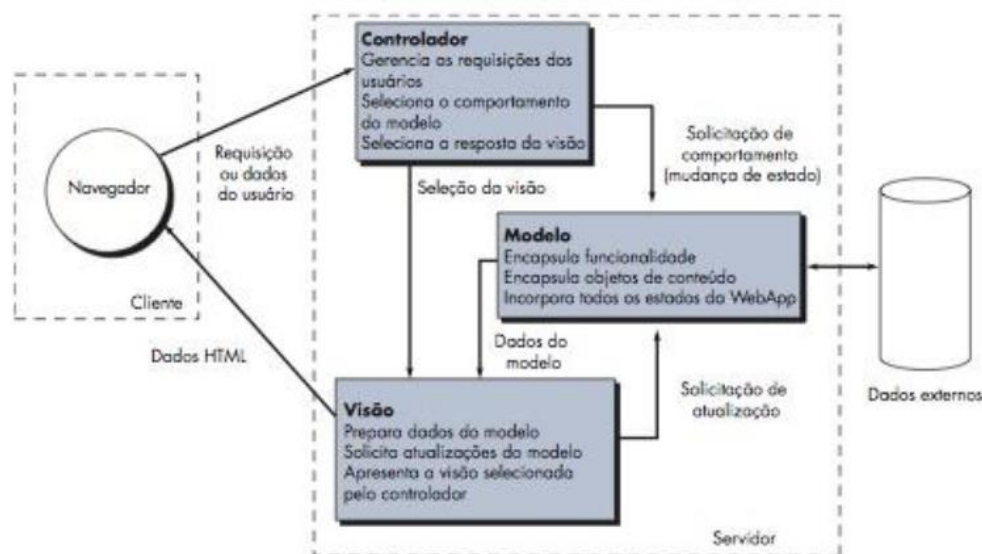
### 5.2.1 MVC

O padrão trabalha com três camadas que se intercomunicam. Gabardo (2012) os apresentam da seguinte maneira:

- *Model*: são os componentes da camada de abstração de dados.
- *View*: é a camada de visualização, recebem os dados dos *controllers* e não tem acesso direto aos *models*.
- *Controllers*: é o responsável pela comunicação entre a camada de dados (*models*) e a camada de apresentação (*views*).

A Figura 8 mostra o funcionamento do modelo MVC de forma genérica. O usuário entra com os dados. Os dados são recebidos por algum método no controlador. Este, por sua vez, faz as requisições necessárias para a camada

modelo. Findo o processo, as informações requisitadas são enviadas a camada de visão pelo controlador.



**Figura 8. Arquitetura MVC**

**Fonte: (Jacobson, 2002 *apud* Pressman, 2011, p. 349 )**

Sommerville (2011, p. 110) afirma que “essa abordagem em camadas apoia o desenvolvimento incremental de sistemas”. O modelo MVC facilita a divisão das equipes na hora do desenvolvimento. Equipes separadas geograficamente podem trabalhar no mesmo projeto fazendo o uso deste padrão. Por exemplo: Uma equipe em São Paulo pode desenvolver as classes modelos, enquanto outra equipe em Belém desenvolve os controladores e as visões da aplicação.

### 5.3 jQuery

O *jQuery* é uma biblioteca *Javascript*<sup>9</sup>, a seguinte afirmação é encontrada em seu *Site*: trata-se de uma biblioteca rápida, compacta e rica em funções. Tem como objetivo a facilitação e a simplificação do código escrito em *Javascript*. Desta forma existe ganho de produtividade, pois haverá a necessidade de se escrever menos código em comparação ao uso do

<sup>9</sup> Linguagem de programação baseada em scripts e padronizada pela ECMA International (associação especializada na padronização de sistemas de informação). Foi criada por Brendan Eich (Netscape) e surgiu em 1995 como linguagem de script client-side de páginas web.

*Javascript* dito como puro. Foi observado um grande ganho no uso dessa biblioteca no uso da técnica de programação *Ajax*<sup>10</sup> muito usada no Booszinga.

## 5.4 MySQL

O MySQL é um servidor e gerenciador de banco de dados (SGBD) relacional, de licença dupla (sendo uma delas de software livre), projetado inicialmente para trabalhar com aplicações de pequeno e médio portes, mas hoje atendendo a aplicações de grande porte e com mais vantagens do que seus concorrentes. Milani (2006) afirma que o MySQL possui todas as características que um banco de dados de grande porte precisa, sendo reconhecido por algumas entidades como o banco de dados open source com maior capacidade para concorrer com programas similares de código fechado, tais como SQL Server (da Microsoft) e Oracle.

Para fazer uso do MySQL de maneira mais simplificada e com uso de interface gráfica foi usada ferramenta *PHPMyAdmin*.

### 5.4.1 PHPMyAdmin

Bento (2014, p. 55) afirma que esta é uma ferramenta escrita em PHP usada para gerenciar o MySQL, com opções para criar novos bancos, usuários, tabelas, inserir, pesquisar e remover registros etc.

Existem várias ferramentas como o *PHPMyAdmin*, mas esta já vem como padrão na maioria dos Kit de instalação de ambiente de desenvolvimento como *WampServer*<sup>11</sup>.

---

<sup>10</sup> É acrônimo em língua inglesa de "Asynchronous Javascript and XML", que em português significa "Javascript e XML Assíncronos". Designa um conjunto de técnicas para programação e desenvolvimento web que utiliza tecnologias como Javascript e XML para carregar informações de forma assíncrona.

<sup>11</sup> É uma aplicação que instala um ambiente de desenvolvimento web no Windows. Com ele você pode criar aplicações web com Apache2, PHP e banco de dados MySQL. Além disso, é possível gerenciar facilmente seus bancos de dados com a ferramenta *PhpMyAdmin* que faz parte do pacote

## 5.5 Trello

O *Trello* é uma ferramenta para gerenciamento de projetos e tarefas. Sua interface trabalha com o uso de “*Boards*” que são quadros que reúnem as tarefas de interesse de uma organização, grupo ou pessoa em particular. Não existe restrição de uso para a ferramenta. Para o Booszinga, a ferramenta foi utilizada para servir como um quadro Kanban online, a fim de seguir a metodologia do *SCRUM*<sup>12</sup>. Foi criado um cartão, onde foi colocado tudo relacionado ao desenvolvimento do Booszinga (Figura 9).

Para esse trabalho foram criadas as *Boards* no *Trello* (Figura 10): Sprint, Andamento, Validação e Concluído. A aplicação Booszinga passou por todos os estágios citados até sua conclusão.

O *Trello* é uma ótima ferramenta para trabalho em equipe. Como pode ser observado na Figura 8 é possível adicionar membros ao cartão e o manter atualizado do que ele precisa fazer dentro da tarefa. Além de outras funções como criar *checklists* e anexar documentos e imagens que sejam relevantes para o projeto.

---

<sup>12</sup> É uma metodologia ágil para gestão e planejamento de projetos de software. No Scrum, os projetos são divididos em ciclos (tipicamente mensais) chamados de Sprints. O Sprint representa um Time Box dentro do qual um conjunto de atividades deve ser executado.

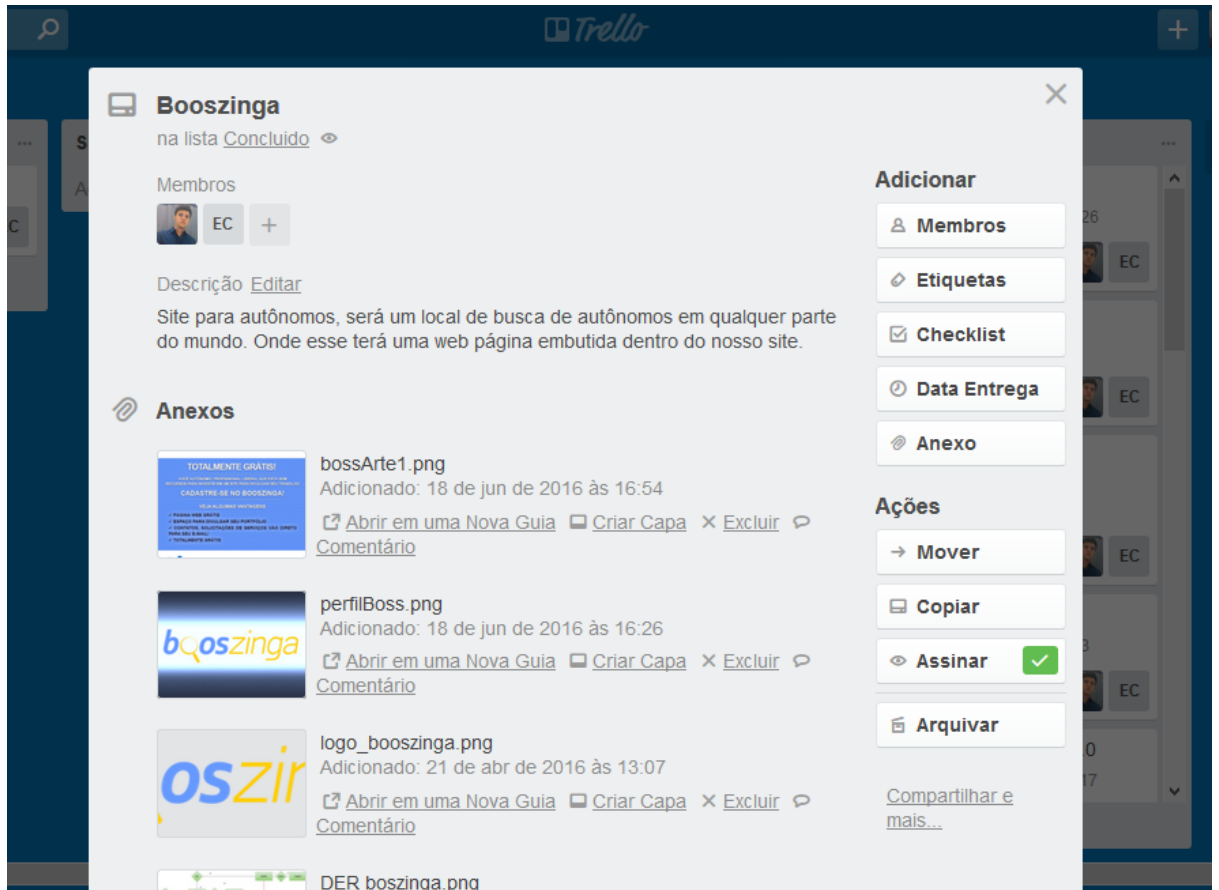


Figura 9. Cartão do Booszinga na ferramenta Trello.

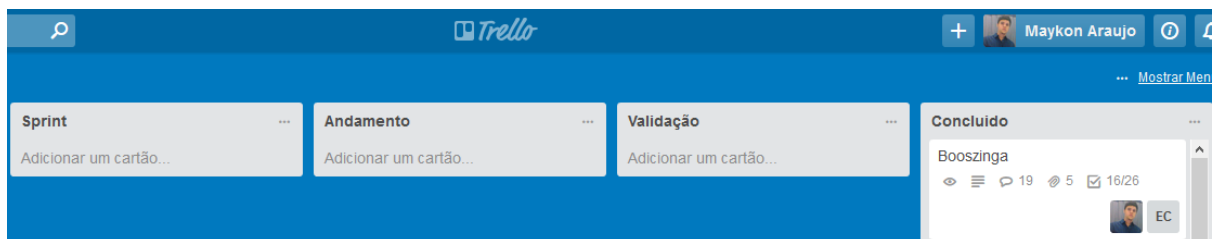


Figura 10. Boards criadas no Trello.

Até aqui foi apresentado o referencial teórico deste trabalho. Estes conceitos são importantes para o entendimento da criação de um produto de software. Sem esse embasamento teórico, tal tarefa seria impossível de realizar ou tenderia ao fracasso. Com esses conhecimentos adquiridos é possível especificar e implementar o Booszinga. O próximo Capítulo trata destes tópicos.

## 6. ESPECIFICAÇÃO DE REQUISITOS, ANÁLISE, PROJETO E IMPLEMENTAÇÃO DA FERRAMENTA

Este capítulo tem como objetivo tratar da fase de especificação da ferramenta (incluindo, requisitos, análise, projeto) e sua implementação.

O processo de desenvolvimento de software começa com a coleta de requisitos. Nessa fase inicial é construído o documento de especificação de requisitos. Aqui serão definidos o escopo, os objetivos do software e a elaboração dos diagramas de casos de uso. Finda esta etapa, a próxima é a construção dos documentos de especificação de análise. Neste documento são definidos os diagramas de pacotes, de classes, de sequências e dicionário de dados (contém breve descrição sobre os dados manipulados pelo software). Logo após é produzido o documento de especificação de projeto. Neste documento são definidas a plataforma computacional, a arquitetura do sistema, o projeto de interação humano-computador (IHC), o projeto de banco de dados e o projeto de componentes. Com as informações coletadas e os documentos elaborados, o desenvolvedor tem o suporte necessário para implementar a ferramenta.

### 6.1 Documento de Especificação de Requisitos do Booszinga


No processo de desenvolvimento de software é preciso ter uma ideia inicial do que se deseja construir. Esse processo no início pode ser nebuloso, ou seja, os envolvidos não sabem ao certo do que o projeto se trata. Então a definição de um escopo inicial se faz necessário. A especificação de requisitos é uma das atividades mais importantes do processo de desenvolvimento de um software.

A aplicação Web proposta neste trabalho não tem um cliente definido. Então para entendimento melhor da necessidade dos possíveis clientes e usuários da plataforma, foram definidas duas personas<sup>13</sup>. Uma persona será a definição genérica do prestador de serviço e a outra do consumidor.


---

<sup>13</sup> Persona é uma técnica de usabilidade, que consiste na criação de perfis e personificação de grupo de usuários.

Quadro 2. Persona prestador de serviço

<b>Persona prestador de serviços</b>	
	<p>Nome: Wilson Idade: 30 anos</p> <p>Wilson é um homem, casado, de 30 anos, possui formação superior em Ciências Contábeis e mora no Brasil. Possui experiência em processos contábeis para pessoas físicas, pequenas e grandes empresas. É reconhecido pela habilidade de facilitar a abertura de novas empresas e normalizar empresas que estão com problemas no fisco. Adoraria que fosse mais fácil:</p> <ul style="list-style-type: none"> <li>● Divulgar seus serviços na internet;</li> <li>● Conseguir fechar mais negócios em outras partes do país;</li> <li>● Fazer um baixo investimento em divulgação na internet;</li> <li>● Fazer uso de um sistema autoexplicativo e simples de usar;</li> </ul>

Quadro 3. Persona consumidor

<b>Persona consumidor</b>	
	<p>Nome: Gabrielle Idade: 25 anos</p> <p>Gabrielle é uma mulher, solteira, de 25 anos, Empresária do ramo de artes. Possui uma galeria de artes. Possui experiência em atendimento ao público. É reconhecida como uma grande artista no meio em que trabalha. Adoraria que fosse mais fácil:</p> <ul style="list-style-type: none"> <li>● Encontrar um prestador de serviço para organizar as finanças da galeria;</li> <li>● Gerenciar as questões fiscais da sua galeria;</li> <li>● Usar a internet para encontrar o que deseja.</li> </ul>

No Quadro 2 temos a persona de Wilson que representa todos os prestadores de serviços que poderão usar o Booszinga. Wilson é alguém com nível superior e está interessado em divulgar seus serviços na Internet. Ele deseja ter um baixo custo investindo na divulgação do seu negócio na Internet. Do outro lado temos a persona de Gabrielle (Quadro 3) que é uma empresária no ramo das artes. Possui uma galeria e gostaria de contratar o serviço de

contabilidade para gerir as finanças e cuidar das questões fiscais.

Após definir as personas, foi definido o ambiente do sistema. O ambiente do sistema é uma visão geral da *WebApp*. Falbo (2011) *apud* Furtado e Costa Jr. (2010, p. 255) define o ambiente do sistema como uma “breve descrição do ambiente do sistema, com menção às tarefas executadas pelos setores envolvidos) com a informatização, suas entradas, seus produtos e possíveis interações com outros sistemas”.

A definição do objetivo do produto é importante durante a produção do documento de requisitos. Este item apresenta o escopo do sistema a ser desenvolvido. Esse tópico é importantíssimo por apresentar a abrangência do sistema: os subsistemas que o compõem são indicados Falbo (2011) *apud* Furtado e Costa Jr. (2010).

Uma lista de eventos com as funcionalidades da aplicação deve constar na seção de Benefícios do produto. O documento de requisitos contém os diagramas de casos de uso com suas respectivas descrições e detalhamentos. Finalizando o documento temos os requisitos não funcionais que são definidos por Sommerville (2011, p. 60) como "requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários". A seguir serão apresentadas algumas dessas informações e os principais casos de usos. O documento de requisitos do Booszinga encontra-se no Apêndice A.

#### 6.1.1 Ambiente do sistema

Este sistema é uma *WebApp* de cadastro e busca de prestadores de serviço. O Booszinga tem como objetivo promover a interação entre prestadores de serviços e consumidores por meio da Internet.

O Empreendedor é uma pessoa interessada em divulgar seus serviços de maneira online. Na maioria dos casos, ele poderá pertencer a várias categorias de serviço, por exemplo: programador, contador, advogado, etc. É de sua responsabilidade a entrada de informações e os cadastramentos realizados na plataforma.

O Cliente é uma pessoa que deseja encontrar um prestador de serviços para atender a sua necessidade. Ele deseja buscar essa informação na internet

e espera por um retorno rápido e preciso. A plataforma não intermedeia as negociações entre cliente e empreendedor.

A aplicação disponibiliza um ambiente administrativo, no qual o empreendedor faz o cadastro do seu perfil, portfólio e informações gerais como: preços, experiências e contatos profissionais. Quando o cliente realizar uma busca na plataforma por meio de palavras chaves, o sistema deve retornar uma lista dos empreendedores cadastrados que se encaixem na busca realizada. Essa lista é composta por uma imagem do empreendedor, seu nome e um resumo da descrição do seu perfil, fornecidos por ele. Ao clicar sobre o empreendedor que lhe interessa, o cliente é levado para a página do empreendedor composta por: foto de perfil, nome, descrição, preços praticados, portfólio, formulário para entrar em contato, redes sociais, endereço e números de telefones. Reforçamos que essas informações não são obrigatórias e são de responsabilidade do empreendedor.

### 6.1.2 Objetivos do produto

Esta aplicação Web objetiva promover a interação entre prestadores de serviços e consumidores, com a possibilidade de fechamento de negócios. Essa interação ocorre por meio da busca com base em palavras chaves por parte do consumidor que retornará uma lista de prestadores de serviços que atendem os requisitos informados.

### 6.1.3 Benefícios do produto

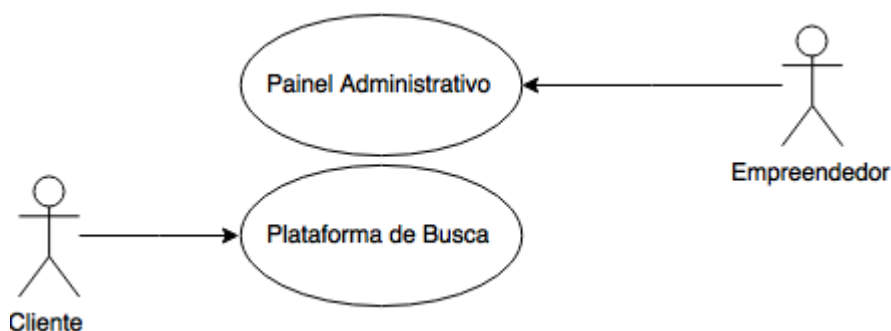
As seguintes funcionalidades estão presentes na aplicação desenvolvida:

- Painel administrativo
  - Cadastro e manutenção do perfil do empreendedor;
  - Cadastro e manutenção de categorias;
  - Cadastro e manutenção de portfólio;
  - Cadastro e manutenção das informações adicionais relacionadas às categorias;
  - Recuperação de senha e dados;
  - Visualização da página do empreendedor.

- Área de buscas utilizada pelo consumidor
  - Busca de prestadores de serviço por meio de palavras-chave;
  - Acesso a página com perfil do empreendedor;
  - Envio de mensagem via formulário;
  - Salvar informações de quantidade de visualizações;
  - Salvar palavras-chave pesquisadas para determinado resultado.

#### 6.1.4 Modelos de Caso de Uso

Nesta *WebApp*, há dois atores interagindo com ele: o ator Empreendedor e o ator Cliente. No contexto do sistema temos dois subsistemas: Painel Administrativo e Plataforma de Busca, conforme Figura 11 a seguir.



**Figura 11. Diagrama de Casos de Uso de Contexto.**

A Figura 11 apresenta o caso de uso de contexto do sistema. O ator cliente tem acesso a Plataforma de Busca e o Empreendedor tem acesso ao Painel Administrativo. Mais à frente temos esse caso de uso decomposto em outros diagramas.

#### 6.1.4.1 Subsistema Painel Administrativo

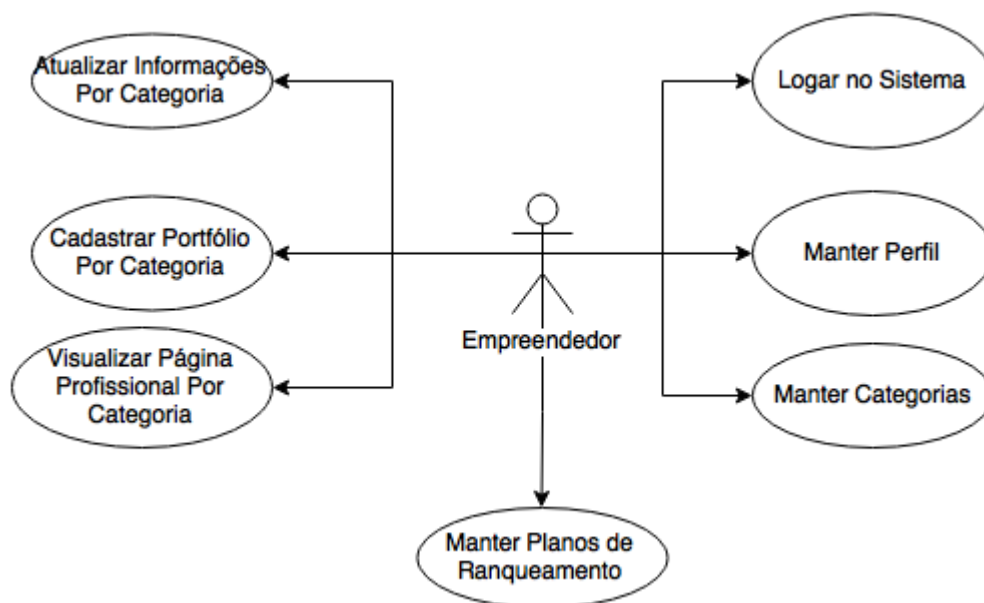


Figura 12. Diagrama de Casos de Uso do Subsistema Administrativo

#### ● CASO DE USO 1: LOGAR NO SISTEMA

**Descrição:** O empreendedor deverá entrar com login e senha e o sistema deverá validar suas credenciais permitindo acesso ou não ao sistema.

**Ator primário:** Empreendedor

**Precondições:** O empreendedor já está cadastrado no sistema.

**Fluxo Principal:**

1. O empreendedor entra com as credenciais de login e senha pré cadastradas.
2. O sistema processa as informações e valida o login.
3. Caso o login seja válido o usuário entra no sistema administrativo.
4. Caso o login seja inválido o usuário não entra no sistema administrativo.

**Pós-Condições:** nenhuma

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Diário.

- **CASO DE USO 2: MANTER PERFIL**

**Descrição:** O empreendedor mantém o seu perfil (inclusão e atualização de informações pessoais, inclusão e atualização de informações profissionais relacionadas a uma determinada categoria).

**Ator primário:** Empreendedor

**Precondições:** O usuário deve estar logado no sistema.

**Fluxo Principal:**

1. O empreendedor atualiza seus dados pessoais e profissionais através de um formulário.
2. O empreendedor salva as informações.

**Pós-Condições:** O sistema deve mostrar uma mensagem sobre o sucesso ou fracasso da operação.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Eventual.

**Fluxo alternativo (3): Desvincular da Categoria.**

- a. Empreendedor escolhe a opção de se desvincular da categoria
- b. O sistema deve verificar se a categoria já tem informações atreladas como portfólios ou informações profissionais.

**Pós-Condições:** Informar o usuário sobre o sucesso ou fracasso dessa operação.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Eventual.

- **CASO DE USO 3: MANTER CATEGORIA**

**Descrição:** O empreendedor vincula-se a uma ou mais categorias.

**Ator primário:** Empreendedor

**Precondições:** O usuário deve estar logado no sistema.

**Fluxo Principal:**

1. O empreendedor vincula-se a uma ou mais categorias
2. O empreendedor desvincula-se de uma ou mais categorias.

**Pós-Condições:** Caso ele já tenha informações cadastradas naquela categoria, o sistema deve retornar uma mensagem de erro.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Eventual.

● **CASO DE USO 4: ATUALIZAR INFORMAÇÕES POR CATEGORIA**

**Descrição:** O empreendedor, para cada nova categoria cadastrada, deverá atualizar as informações referentes a esta.

**Ator primário:** Empreendedor

**Precondições:** O usuário deve estar logado no sistema.

**Fluxo Principal:**

1. O empreendedor insere os dados sobre seu perfil profissional (sobre seu trabalho, redes sociais, faixa de preços e tempo de experiência) relacionado a categoria que ele selecionou.
2. O usuário salva as informações.

**Pós-Condições:** O sistema deve retornar uma mensagem de sucesso ou fracasso para essa operação.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Eventual.

- **CASO DE USO 5: CADASTRAR PORTFÓLIO POR CATEGORIA**

**Descrição:** O empreendedor a cada nova categoria cadastrada, poderá cadastrar seu portfólio referentes a esta.

**Ator primário:** Empreendedor

**Precondições:** O usuário deve estar logado no sistema.

**Fluxo Principal:**

1. O empreendedor faz upload de até 5 imagens por vez.
2. O empreendedor cadastra títulos e descrições das imagens.
3. O empreendedor exclui e altera informações das imagens.
4. O empreendedor marcar uma imagem como destaque, para essa ser a primeira imagem na ordem.

**Pós-Condições:** O sistema deve retornar uma mensagem de sucesso ou fracasso para essa operação.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Mensal.

- **CASO DE USO 6: VISUALIZAR PÁGINA PROFISSIONAL POR CATEGORIA**

**Descrição:** O empreendedor visualiza a sua página na web referente a uma determinada categoria.

**Ator primário:** Empreendedor

**Precondições:** O usuário deve estar logado no sistema.

**Fluxo Principal:**

1. O empreendedor clica na opção de visualizar página referente a uma determinada categoria.

2. Um modelo da sua página com as informações que ele já cadastrou é apresentado.

**Pós-Condições:** nenhuma.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Diário.

● **CASO DE USO 7: MANTER PLANO DE RANQUEAMENTO**

**Descrição:** O empreendedor mantém o cadastro de planos.

**Ator primário:** Empreendedor

**Precondições:** O usuário deve estar *logado* no sistema.

**Fluxo Principal:**

1. O empreendedor escolhe um plano de ranqueamento.
2. O empreendedor escolhe 5 palavras-chave que se enquadram no seu perfil.
3. O empreendedor salva as informações.

**Pós-Condições:** O sistema deve retornar uma mensagem de sucesso ou fracasso para essa operação.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Eventual.

#### 6.1.4.2 Subsistema Plataforma de busca

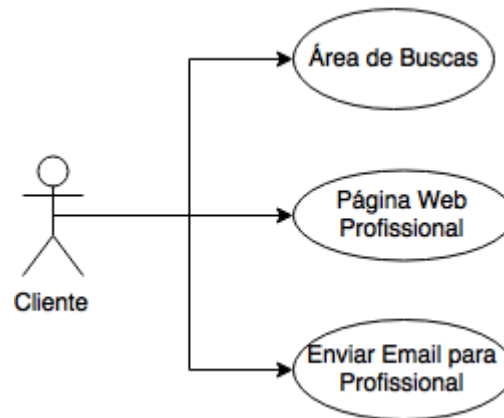


Figura 13. Diagrama de Casos de Uso do Subsistema Plataforma de Busca

#### ● CASO DE USO 1: ÁREA DE BUSCAS

**Descrição:** O cliente pesquisa por meio de palavras chaves e encontra os serviços ou produtos de seu interesse.

**Ator primário:** Cliente

**Precondições:** nenhuma.

**Fluxo Principal:**

1. O cliente insere palavras chaves em um campo de busca único.
2. O sistema retorna o profissional que mais se adequa ao que o cliente procura, levando em consideração as seguintes regras de ranqueamento:
  - a. Classificação Orgânica:
    - i. Do cadastro mais antigo;
    - ii. Do cadastro com atualizações mais recentes.;
    - iii. Com perfil mais completo;
    - iv. Mais informações de portfólio;
    - v. Com maior número de visualizações de perfil;
  - b. Classificação Paga
    - i. Plano Premium

- ii. Plano Ouro
- iii. Plano Prata
- iv. Plano Avulso

**Pós-Condições:** O sistema deve listar os profissionais baseado nas regras de ranqueamento.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Diário.

#### ● CASO DE USO 2: PÁGINA WEB PROFISSIONAL

**Descrição:** O cliente escolhe um dos profissionais listados e entra na sua página web.

**Ator primário:** Cliente

**Precondições:** nenhuma.

**Fluxo Principal:**

1. O cliente clica no resumo do profissional do seu interesse
2. O sistema redireciona o cliente para a página do profissional escolhido
3. O sistema mostra a página dos clientes com todos os dados cadastrados pelo mesmo

**Pós-Condições:** nenhuma

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Diário.

#### ● CASO DE USO 2: ENVIAR EMAIL PARA O PROFISSIONAL

**Descrição:** O cliente entra em contato com o profissional através do formulário de e-mail na página do cliente.

**Ator primário:** Cliente

**Precondições:** Empreendedor deve ter um e-mail pré-cadastrado.

**Fluxo Principal:**

1. O cliente insere as suas informações (nome, e-mail e telefone) e a mensagem que deseja enviar para o profissional no formulário de contato.
2. Ele clica no botão enviar.
3. A mensagem é enviada para o e-mail pré-cadastrado do empreendedor.

**Pós-Condições:** O sistema deve retornar uma mensagem de sucesso ou fracasso para essa operação.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Diário.

#### 6.4.5 Requisitos não funcionais

**Segurança:** O sistema deve prover facilidade para autenticação de todos os usuários, com a digitação de usuário e senha. As operações realizadas numa seção deverão registrar quem as executam.

**Desempenho:** Todas as operações realizadas deverão ser executadas em no máximo 1 segundo.

**Interface:** Deve apresentar boa usabilidade, já que os usuários farão uso diariamente.

**Portabilidade:** O sistema deve permitir ampla portabilidade, de modo a executar em ambientes operacionais diversos.

Com o Documento de Especificação de Requisitos pronto, foi possível ter clareza em que consistiria a ferramenta a ser implementada, com a execução das etapas posteriores do processo de desenvolvimento. Então, o próximo passo foi a produção do Documento de Especificação de Análise.

## 6.2 Documento de Especificação de Análise do Booszinga

Nesta etapa do processo de desenvolvimento da WebApp temos uma visão mais clara do que se trata a aplicação. O principal objetivo do Documento de Especificação de Análise é a identificação e a criação dos diagramas de classes da aplicação, com base nos casos de usos produzidos pelo Documento de Especificação de Requisitos.

Furtado e Costa Jr. (2010, p. 319) relacionam alguns passos para a elaboração do Documento de Especificação de Análise:

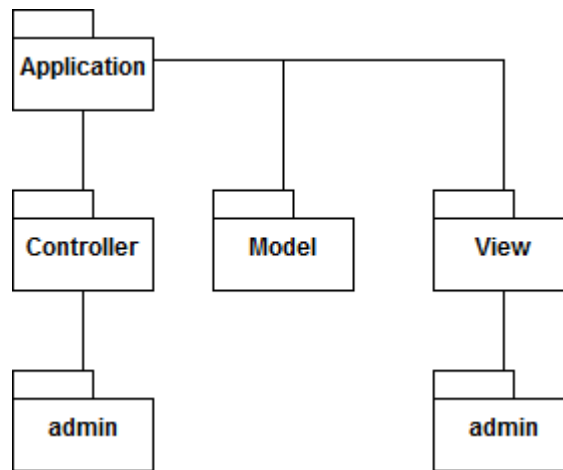
- a) com base no Documento de Especificação de Requisitos que se terá desenvolvido na etapa anterior (visto no capítulo anterior), buscam-se identificar as classes/objetos relevantes para a aplicação que se quer desenvolver; estas classes/objetos aparecem na especificação dos casos de uso. Vamos ler cuidadosamente os casos de uso à procura dos potenciais objetos/classes do sistema;
- b) os objetos de uma classe vão relacionar-se com objetos de outras Classes – podemos reconhecer aí uma associação entre classes, ou uma estrutura de generalização/especialização, ou ainda uma estrutura todo-parte. Estamos falando em identificar estes relacionamentos, com vista à construção do diagrama de classes da aplicação;
- c) como passo final, para cada classe identificada, buscamos relacionar seus atributos e métodos, de modo que se possam atender a todos os requisitos da aplicação.

Para organizar os subsistemas da aplicação foi produzido o diagrama de pacotes. Com a identificação dos pacotes, foi criado o diagrama de classe referente aos subsistemas definidos. O diagrama de classe é o principal artefato em um documento de análise. Para complementar esse diagrama foram construídos outros artefatos: o diagrama de sequência e o dicionário de dados.

### 6.2.1 Diagrama de Pacotes do Booszinga

Este diagrama tem como objetivo garantir uma visualização *top-down* do sistema, provendo um agrupamento lógico das classes.

Na Figura 14 é apresentado o diagrama de pacotes do Booszinga. Neste diagrama pode ser vista a estrutura do padrão MVC. As classes que estão dentro do pacote admin são referentes ao painel administrativo, e as classes que estão fora deste pacote são referentes ao ambiente de buscas.



**Figura 14. Diagrama de pacotes do Booszinga.**

Na Figura 14, temos o pacote *Application*, em que estão contidas todas as classes e os arquivos da aplicação. No pacote *Controller*, temos as classes de controle da plataforma de buscas, juntamente com o pacote admin. Neste pacote estão as classes de controle do sistema administrativo. No pacote modelo não existe a divisão de classes pertencentes ou não ao pacote admin. Isso ocorre porque neste pacote estão as classes que abstraem o banco de dados, ou seja, as classes deste pacote representam as entidades da aplicação, e estas são usadas tanto na plataforma de buscas, quanto no sistema administrativo. O pacote *View* segue a mesma divisão mencionada anteriormente, mas aqui não teremos classes e sim as interfaces da WebApp.

## 6.2.2 Diagrama de Classes do Booszinga

Nesta seção será apresentado um dos principais diagramas de classes da WebApp. A Figura 15 representa as classes modelo. As classes modelo representam as entidades, ou seja, são todas as informações que a aplicação deve manter no banco de dados.

No início do projeto, por meio do Documento de Especificação de Requisitos, foram identificadas as entidades: Categoria, Pesquisa, Pessoa e Portfólio. Baseada nessas entidades foram criadas as classes da Figura 14. Nessas classes temos todas as operações de CRUD (Create, Read, Update, Delete), que são as operações padrão de um banco de dados.

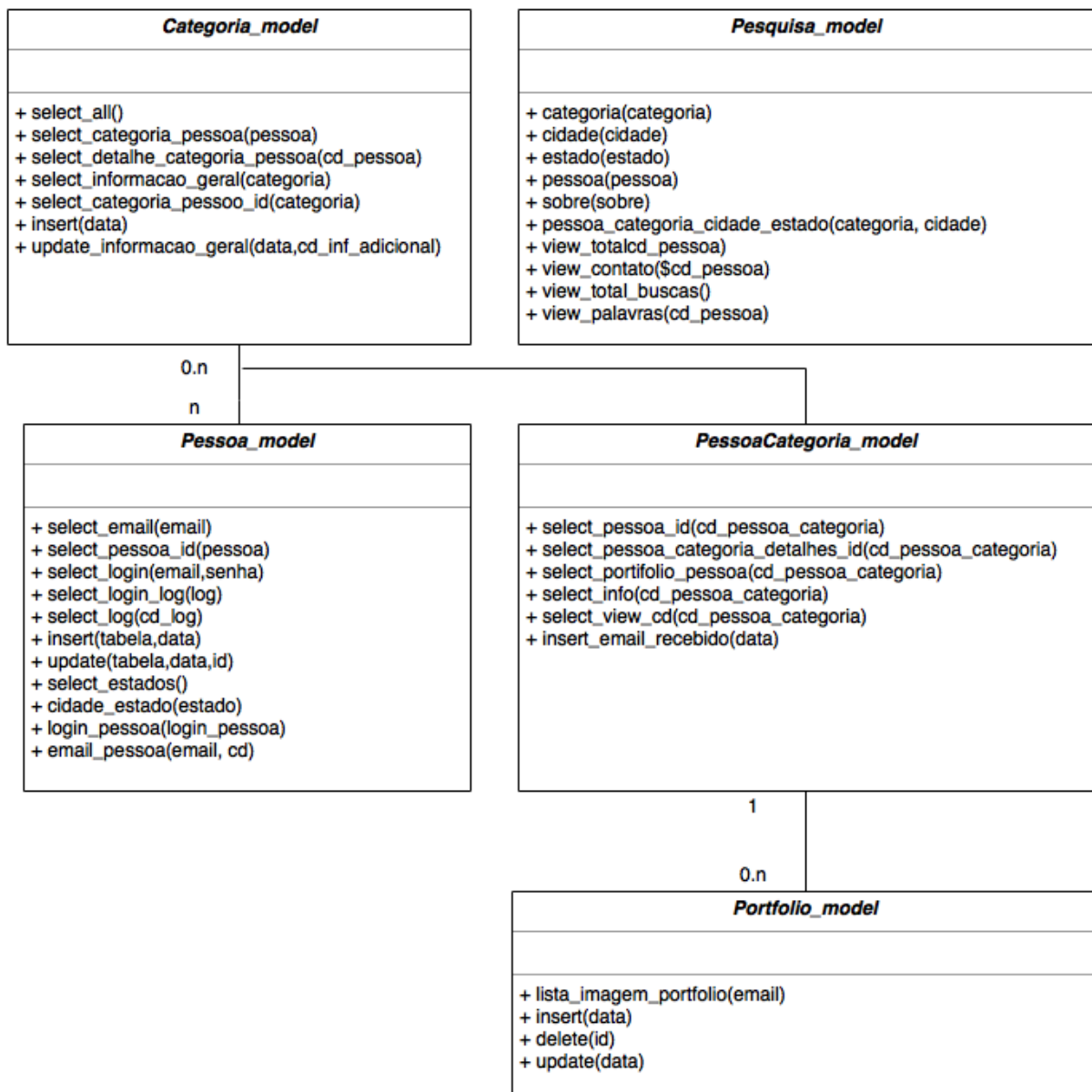
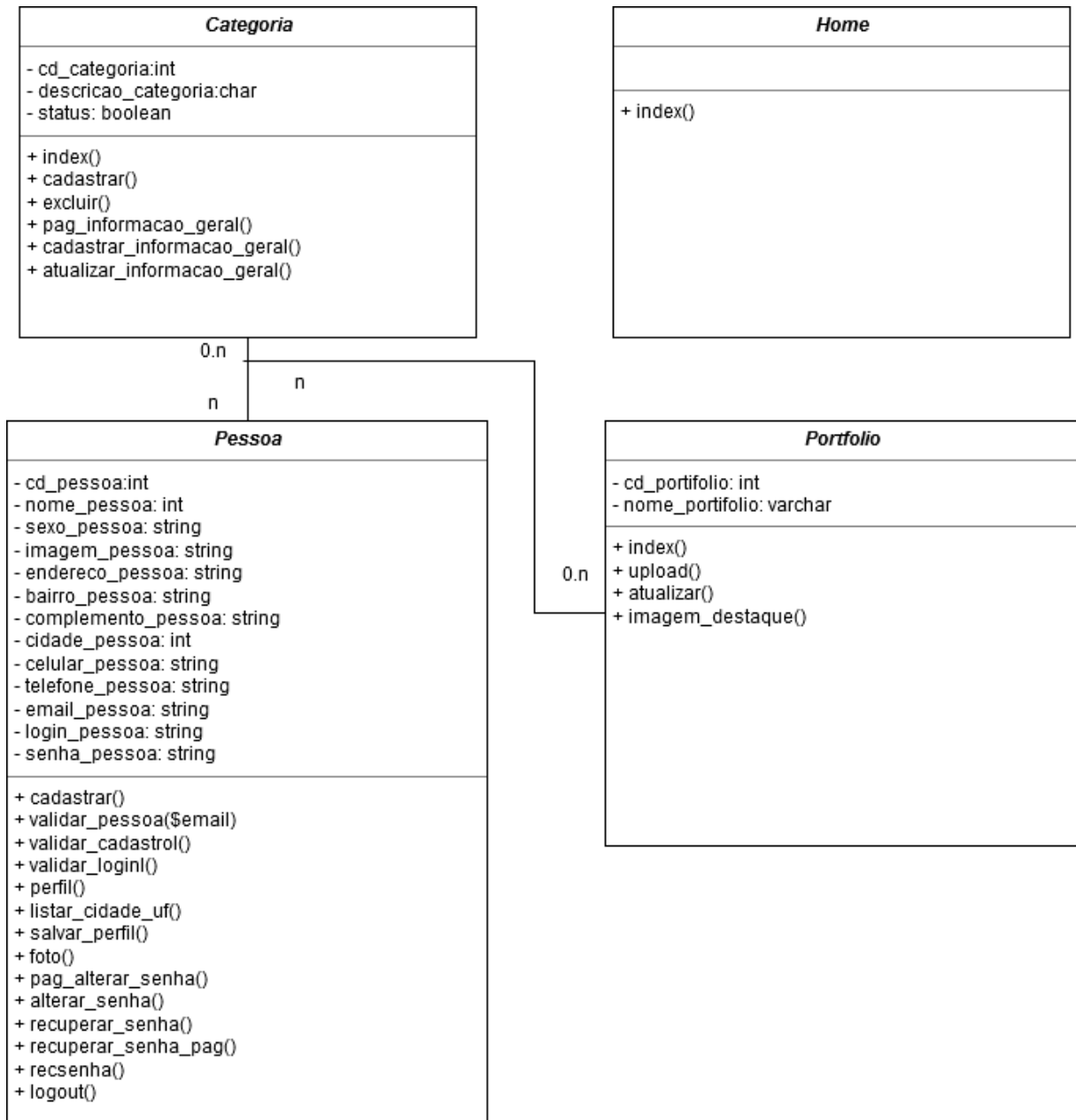


Figura 15. Classes modelo do Booszinga

As classes presentes nos controladores seguem o mesmo padrão descrito anteriormente. Estas usam métodos que se comunicam com as classes modelo. Isto pode ser visto na Figura 16.



**Figura 16. Diagrama de Classes do pacote application/controller/admin.**

Para melhor entendimento por parte do desenvolvedor, foram criados diagramas de seqüências. Estes diagramas - como explicado em capítulos anteriores, e afirmado por Sommerville (2011, p. 87): “tem por objetivo modelar as interações entre os atores e os objetos em um sistema e as interações entre os próprios objetos”.

### 6.2.3 Diagramas de Sequência do Booszinga

Os diagramas de sequência apresentados nesta seção apresentam suas interações baseadas nos casos de uso do Documento de Especificação de Requisitos.

#### 6.2.3.1 UC02: Manter Perfil

Este é um caso de uso do sistema administrativo. Esta funcionalidade tem relação com o Perfil do prestador de serviço, ou seja, este descreve em linhas gerais: os serviços oferecidos, os preços praticados no mercado, o tempo que está em atividade, as redes sociais e os contatos. A Figura 17 detalha as interações que o ator Empreendedor deve seguir para conseguir manter seu cadastro atualizado no sistema. Manter uma informação é realizar todas as operações de manipulação de dados do banco de dados: criar, atualizar e excluir.

A sequência dos passos se inicia com o ator entrando com suas informações no formulário de cadastro, que está disponível na tela de cadastro; a submissão desses dados faz uma chamada ao método `salvar_perfil`, que está presente na classe `Pessoa` do pacote `Controller`; esse método usa a função `update` da classe `Pessoa_model` do pacote `model`. Findo o processo, a função retorna uma mensagem para o usuário, informando sucesso ou fracasso na operação.

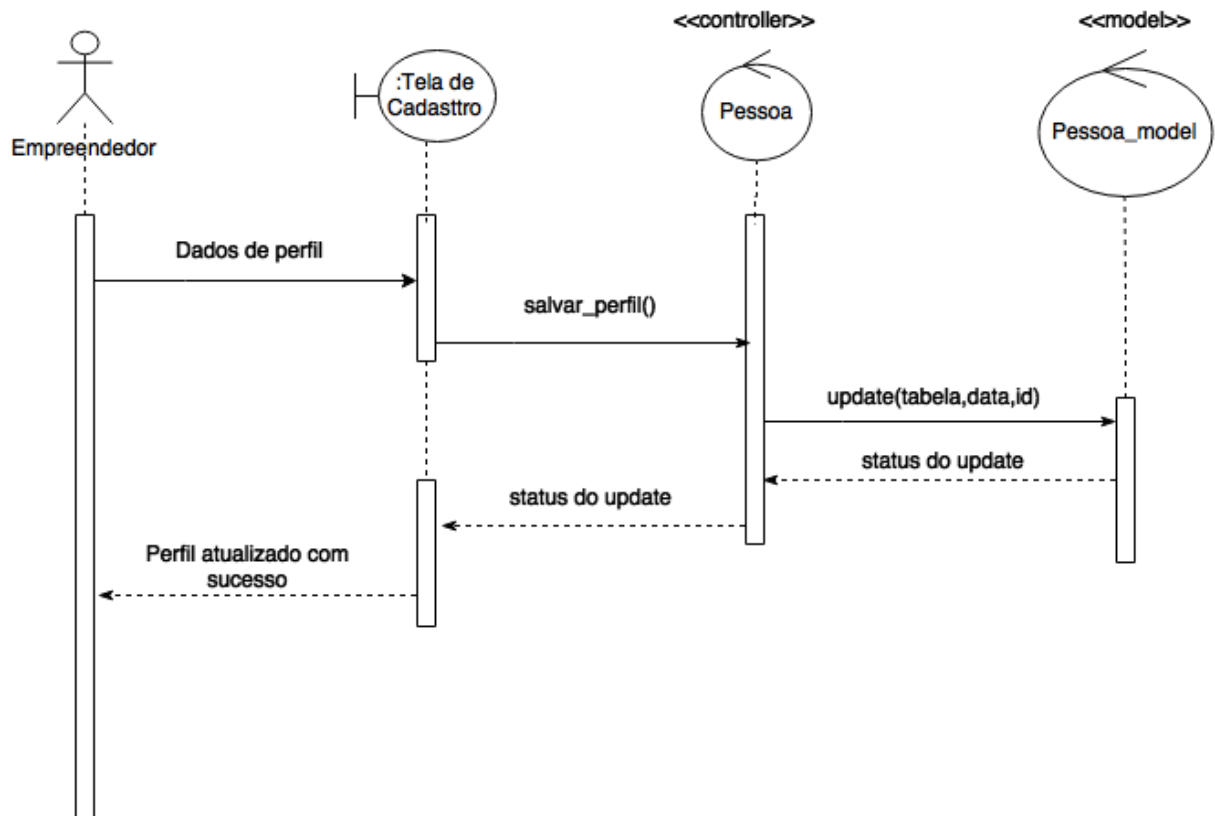


Figura 17. Diagrama de Sequência do funcionamento do UC02

Os demais diagramas de sequência usados para demonstrar as interações de manter informações seguem os mesmos passos do diagrama apresentado acima.

O próximo diagrama de sequência trata da plataforma de busca. Este diagrama mostra as interações que o ator Cliente realiza ao fazer uma pesquisa na plataforma.

#### 6.2.3.2 UC06: Área de buscas

A área de buscas é a parte do sistema onde o consumidor realiza buscas de prestadores de serviços, por meio de palavras chaves que identificam a sua necessidade. A área de buscas é a tela inicial (*home*) da *WebApp*. A figura 18 detalha os caminhos que o ator Cliente terá que seguir para realizar buscas por prestadores de serviços. A interação inicia com o ator inserindo as palavras chaves no campo de buscas da aplicação; a submissão dos dados informados faz a chamada do método *pesquisa*, que está presente na classe *Home* do pacote *Controller*; este método usa as funções *categoria*, *estado*, *cidade* e

sobre, que fazem parte da classe Pesquisa\_model do pacote *Model*. A função retorna uma lista contendo os dados encontrados.

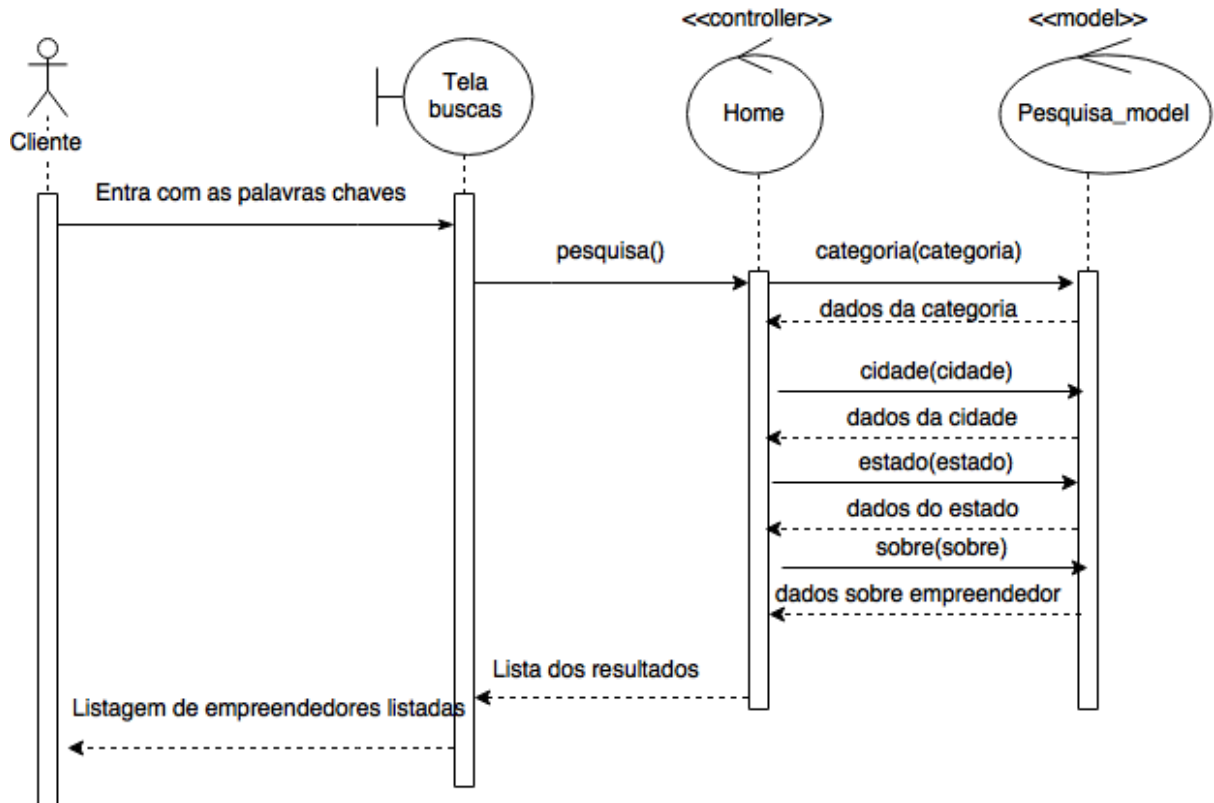


Figura 18. Diagrama de Sequência do UC06.

Foram retratados aqui dois diagramas de sequência da aplicação. Um referente ao sistema administrativo e o outro a área de acesso do consumidor. Para fechar o entendimento do diagrama de classes temos o dicionário de dados.

#### 6.2.4 Dicionário de Dados do Booszinga

O Dicionário de Dados é uma ferramenta que pode ser utilizada para complementar os diagramas da UML. Furtado e Costa Jr. (2010, p. 287), afirmam:

O Dicionário de Dados é ferramenta importante para documentação do sistema, pois descreve todos os dados sobre o sistema. A notação usada nesta metodologia permite descrever itens de dados, depósitos de dados,

registros de dados e fluxos de dados.

Abaixo é apresentado o Dicionário de dados da *WebApp*:

**CATEGORIA\_MODEL = {categoria\_model}**

**Categoria\_model** = \*acessa e registra os dados de categorias no banco de dados\*.

@cd\_categoria + nome\_categoria

**PESQUISA\_MODEL = {pesquisa\_model}**

**Pesquisa\_model** = \*acessa os dados relacionados a categorias, cidades, pessoas e portfólio no banco de dados\*.

**PESSOA\_MODEL = {pessoa\_model}**

**Pessoa\_model** = \*acessa e registra os dados dos usuários no banco de dados\*.

@cd\_usuario + nome\_usuario + senha\_usuario + email\_usuario + telefone\_usuario + endereco\_usuario + bairro\_usuario + cidade\_usuario + uf\_usuario + data\_cadastro

**PESSOA\_CATEGORIA\_MODEL = {pessoa\_categoria\_model}**

**Pessoa\_categoria\_model** = \*acessa e registra os dados das pessoas vinculados a categorias no banco de dados\*.

@cd\_pessoa\_categoria + categoria + pessoa + informacao\_adicional + faixa\_preco

**PORTFOLIO\_MODEL = {portfolio\_model}**

**Portfolio\_model** = \*acessa e registra os dados do portfólio no banco de dados\*.

@cd\_portfolio + url\_imagem + titulo\_imagem + descricao\_imagem + data\_cadastro

**CATEGORIA\_MODEL = {categoria\_model}**

**Categoria\_model** = \*acessa e registra os dados de categorias no banco de dados\*.

@cd\_categoria + nome\_categoria

A figura 19 mostra o significado da simbologia utilizada no Dicionário de Dados.

Símbolo	Significado
=	É composto de
+	E
()	Opcional (pode estar presente ou ausente)
{ }	Iteração
[ ]	Escolha em uma das alternativas
**	Comentário
@	Identificador (chave) em um depósito
/	Separa opções alternativas na construção

Figura 19. Notação para o Dicionário de Dados.

Fonte: (Furtado & Costa Jr, 2010, p. 288)

O Dicionário de Dados fecha o documento de Especificação de Análise que se encontra completo no Apêndice B. O próximo passo foi a construção da Especificação de Projeto.

### 6.3 Documento de Especificação de Projeto do Booszinga

Este documento contém a Especificação de Projeto para o Sistema Booszinga. Este documento deve ser claro o suficiente para a equipe de desenvolvimento, de tal forma que não seja necessário recorrer ao Analista para esclarecimento de dúvidas.

A arquitetura do sistema já foi definida nos diagramas de classes e seus complementos. O único requisito de plataforma computacional exigido para acesso da *WebApp* é um dispositivo com acesso à internet e configurações mínimas para abrir um navegador web.

Uma parte importante desse documento é o Projeto Interação Humano Computador. Serão apresentados alguns protótipos de tela importantes da

*WebApp*.

### 6.3.1 Projeto de Interação Humano Computador

Esta seção irá apresenta as principais telas da aplicação. Barbosa e Silva (2010) definem a usabilidade, a experiência do usuário, a acessibilidade e a comunicabilidade como características de interface e interação. Na Figura 20, é apresentado o protótipo de tela da página inicial do Booszinga. O propósito desta tela é atender os requisitos de qualidade comentados anteriormente. Nesta página o consumidor pode fazer a busca pelo prestador de serviço - informando as palavras chaves no campo de busca, o prestador de serviço tem a opção de acessar a tela de autenticação ou criar uma conta.

Quando o consumidor acessa o *site*, ele já tem a ideia do objetivo da plataforma, não existem distrações. O leiaute é limpo e proporciona ótima usabilidade e comunicabilidade. O *designer* consegue passar a mensagem para o usuário, e este consegue fazer uso da *WebApp* sem dificuldades, o que lhe fornece uma ótima experiência, lhe proporcionando um sentimento de satisfação.



**Figura 20. Página Inicial do Booszinga.**

Na Figura 21, temos um exemplo de protótipo de tela do Sistema Administrativo, que diz respeito ao cadastro de perfil do prestador de serviço.

Esta página fornece um formulário de cadastro, onde o prestador de serviço fornece suas informações básicas pessoais, com a opção de adicionar uma foto ao perfil. A Tabela 1 explica com mais detalhes os campos e suas funcionalidades.

Figura 21. Tela de cadastro de perfil de prestador de serviço no sistema administrativo

Tabela 1. Tabela de Detalhamentos da tela de Visualização de Portfólio.

Tela de cadastro de perfil do prestador de serviço				
SEQ	Nome do Campo	Descrição	Formato	Tamanho
1	Nome	Campo para informar o nome completo.	Caracteres	50
2	Gênero	Campo para escolha do gênero do usuário.	N/A	N/A
3	Endereço	Campo para informar o endereço do usuário.	Alfanumérico	100
4	Bairro	Campo para informar a o bairro do	Alfanumérico	50

		usuário.	co	
5	Complemento	Campo para informar o complemento do endereço do usuário.	Alfanumérico	100
6	Estado	Campo para escolher o estado do usuário.	N/A	N/A
7	Cidade	Campo para escolher a cidade do usuário baseado no estado escolhido.	N/A	N/A
8	Celular	Campo para informar o celular do usuário.	Alfanumérico	15
9	WhatsApp	Campo para informar o número do <i>WhatsApp</i> do usuário.	Alfanumérico	15
10	Email	Campo para informar o email do usuário.	Alfanumérico	50
11	Login	Campo para informar o login do usuário.	Alfanumérico	50

Foram apresentados dois protótipos de telas, um da área de buscas e outro do sistema administrativo. No apêndice C está o documento completo e podem ser visualizados os demais protótipos de telas.

Na próxima subseção será apresentado o Projeto de Banco de Dados. A base para o desenvolvimento de uma aplicação sólida, é um banco de dados bem planejado.

### 6.3.2 Projeto de Banco de Dados

O Projeto de Banco de dados é essencial na implementação de um sistema. Deve conter as tabelas e os relacionamentos relevantes para a aplicação. A Figura 22, apresenta o Diagrama de Entidade Relacional do Booszinga. Logo em seguida vem o detalhamento de cada tabela deste diagrama.



- **id** chave primária
- estado chave estrangeira referenciando a chave primária **id** da tabela boo\_estado;
- boo\_log\_acesso(**cd\_log\_acesso**, cd\_pessoa, data\_log, ip\_acesso, tempo\_saida)
  - **cd\_log\_acesso** chave primária
  - cd\_pessoa chave estrangeira referenciando a chave primária **cd\_pessoa** da tabela boo\_pessoa;
- boo\_informacao\_adicional(**cd\_inf\_adicional**, cd\_pessoa\_categoria, email\_profissional, site\_profissional, facebook\_profissional, twitter\_profissional, linkedin\_profissional, tempo\_experiencia, descricao\_profissional, preco\_profissional, url\_personalizada)
  - **cd\_inf\_adicional** chave primária
  - cd\_pessoa\_categoria chave estrangeira referenciando a chave primária **cd\_pessoa\_categoria** da tabela boo\_pessoa\_categoria;
- boo\_portfolio(**cd\_portfolio**, cd\_pessoa\_categoria, nome\_portfolio, data\_cadastro\_portfolio, data\_atualizacao\_portfolio)
  - **cd\_portfolio** chave primária
  - cd\_pessoa\_categoria chave estrangeira referenciando a chave primária **cd\_pessoa\_categoria** da tabela boo\_pessoa\_categoria;
- boo\_pessoa\_categoria(**cd\_pessoa\_categoria**, cd\_pessoa, cd\_categoria, data\_cadastro)
  - **cd\_pessoa\_categoria** chave primária
  - cd\_categoria chave estrangeira referenciando a chave primária **cd\_categoria** da tabela boo\_categoria
- cd\_pessoa chave estrangeira referenciando a chave primária **cd\_pessoa** da tabela boo\_pessoa;
- boo\_imagem(**cd\_imagem**, cd\_pessoa\_categoria, url\_imagem, titulo\_imagem, descricao\_imagem, destaque\_imagem, data\_cadastro\_imagem)
  - **cd\_imagem** chave primária
  - cd\_pessoa\_categoria chave estrangeira referenciando a chave

- primária **cd\_pessoa\_categoria** da tabela boo\_pessoa\_categoria;
- boo\_email\_recebido(**cd\_email\_recebido**, cd\_pessoa\_categoria, remetente, telefone, data\_email, mensagem)
    - **cd\_email\_recebido** chave primária
    - cd\_pessoa\_categoria chave estrangeira referenciando a chave primária **cd\_pessoa\_categoria** da tabela boo\_pessoa\_categoria;
  - boo\_comentario(**cd\_comentario**, texto\_comentario, data\_comentario, ip\_comentario, navegador\_comentario, navegador\_comentario, regioao\_comentario, cd\_pessoa\_categoria)
  - cd\_comentario chave primária
    - cd\_pessoa\_categoria chave estrangeira referenciando a chave primária **cd\_pessoa\_categoria** da tabela boo\_pessoa\_categoria;
  - boo\_pessoa\_plano\_categoria(**cd\_pessoa**, **cd\_plano**, **cd\_categoria**)
    - **cd\_pessoa** chave primária e chave estrangeira referenciando a chave primária **cd\_pessoa** da tabela boo\_pessoa
    - **cd\_plano** chave primária e chave estrangeira referenciando a chave primária **cd\_plano** da tabela boo\_plano
    - **cd\_categoria** chave primária e chave estrangeira referenciando a chave primária **cd\_categoria** da tabela boo\_categoria;
  - boo\_local(**cd\_local**, nome\_local, acrescimo\_local)
    - **cd\_local** chave primária
  - boo\_chave\_pessoa(**cd\_chave\_pessoa**, cd\_pessoa\_categoria, chave\_busca, data\_busca, ip\_busca\_
    - **cd\_chave\_pessoa** chave primária
    - cd\_pessoa\_categoria chave estrangeira referenciando a chave primária **cd\_pessoa\_categoria** da tabela boo\_pessoa\_categoria;
  - boo\_view\_page(**cd\_view\_page**, cd\_pessoa\_categoria, data\_visualizacao, ip\_visualizacao)
    - **cd\_view\_page** chave primária
    - cd\_pessoa\_categoria chave estrangeira referenciando a chave

primária **cd\_pessoa\_categoria** da tabela boo\_pessoa\_categoria;

- boo\_plano(**cd\_plano**, nome\_plano, valor\_plano, validade\_plano, local\_plano)
  - **cd\_plano** chave primária
- ci\_session(id,ip\_address, timestamp, data)

Com o Diagrama de Entidade e Relacionamento pronto, fechamos o Documento de Especificação de Projeto com a descrição dos componentes de softwares criados para atender os requisitos da aplicação.

### 6.3.3 Projeto de Componentes

Furtado e Costa Jr. (2010) afirmam que esta etapa tem por objetivo especificar detalhadamente cada componente/módulo do sistema, de modo que seja possível ao programador implementá-lo adequadamente. Aqui existe uma preocupação com uma boa especificação de componente, ou seja, o programador deve ser capaz de implementar o componente, sem a necessidade de procurar o projetista para tirar dúvidas sobre a especificação.

A seguir é apresentada a especificação do componente HOME, este é um dos componentes mais complexos da aplicação, pois é responsável pelo algoritmo de buscas da plataforma, por isso é importante estar detalhado neste trabalho. Os demais seguem o padrão de manter itens de dados, já conhecido pelos profissionais da área de tecnologia.

- **Especificação do Componente HOME**

1) FUNÇÃO: Executar a pesquisa dos prestadores de serviços baseados nas palavras-chaves informadas e retornar uma lista de resultados mais próxima possível do esperado pelo usuário.

2) PSEUDOCÓDIGO DO PROGRAMA HOME

1. Início.
2. Fazer inicializações
3. Receber palavras-chave

SE palavras-chaves não for nula ENTÃO  
     executar pesquisar

SENÃO

    exibir “Informe ao menos uma palavra-chave”

FIM SE

4. Finalizações

5. Sair

pesquisar

Inicio

inicializa variáveis

    ENQUANTO houver palavras chaves, faça:

        categoria ← pesquisar-categoria

    cidade ← pesquisar-cidade

    estado ← pesquisar-estado

    sobre ← pesquisar-sobre

    pessoa ← pesquisar-pessoa

    FIM ENQUANTO

    melhor\_caso ← interseccao(categoria,  
cidade,estado,sobre,pessoa)

    caso\_medio ← i nterseccao(categoria, cidade, estado)

    caso\_normal ← interseccao(categoria, cidade)

    pior\_caso ← interseccao(categoria, pessoa)

    resultado ← merge(melhor\_caso, caso\_medio, caso\_normal,  
pior\_caso)

/\*

Critério para ordenação

A ordenação será trabalhada com pesos

exemplo abaixo:

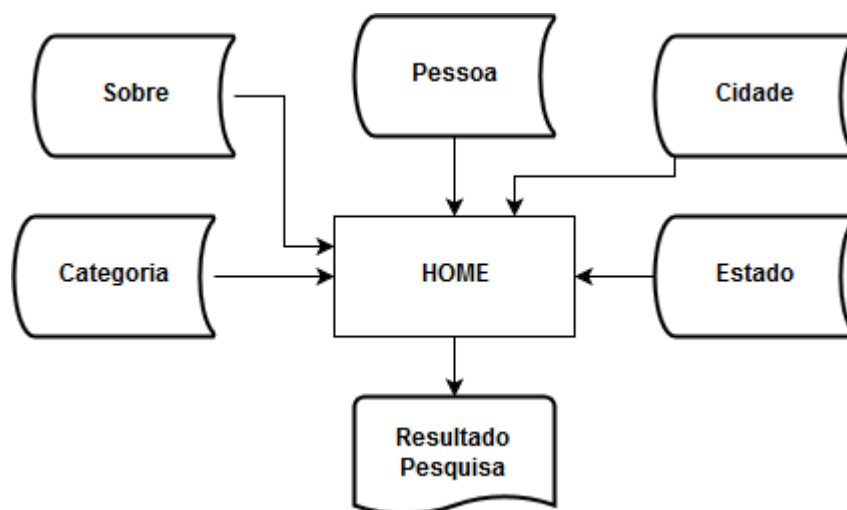
- 1 - Usuário mais antigos - 6
- 2 - Acesso recente no painel administrativo - 5
- 3 - % do perfil - 4
- 4 - Quantidade de visualizações de página - 3
- 5 - Quantidade de fotos no portfólio - 2
- 6 - Quantidade de informações adicionais - 1

fórmula usada OF =  $(o1*6) + (o2*5) + (o3*4) + (o4+3) + (o5+2) + (o6+1)$

\*\*\*o1 = Ordenação 1, o2 = Ordenação 2 sucessivamente.

\*/

Na Figura 23. Temos a rotina gráfica do componente home. É possível observar que as entidades: Sobre, Pessoa, Cidade, Estado, Categoria, são entradas de informações que são processadas pelo componente e gera um resultado.



**Figura 23. Rotina gráfica do Componente Home**

Este documento finaliza a etapa de análise da ferramenta. O Documento de Especificação de Projeto se encontra completo no Apêndice C. Com todos os documentos produzidos nesta seção: Documento de Especificação de Requisitos, Documento de Especificação de Análise e Documento de

Especificação de Projetos, a aplicação foi implementada. A implementação da ferramenta será apresentada com mais detalhes a seguir.

#### 6.4 Implementação da Ferramenta Booszinga

Esta é a parte mais esperada do projeto, o momento da construção da aplicação. O objetivo desta seção é apresentar o passo a passo da construção do Booszinga: planejamento, criação do banco de dados, configuração do ambiente de desenvolvimento e finalização.

Nesta etapa, a gerência do projeto obedeceu ao *framework* Scrum. Como mostrado no Capítulo 3, o modelo de processo Scrum é direcionado para trabalho em equipe, formada por: Product Owner, Scrum Master e o Time de Desenvolvimento. Apesar disto, este trabalho foi desenvolvido por uma pessoa (o autor), que representou todos os papéis do Time Scrum. Os artefatos e os eventos do Scrum (*Backlog*, *Backlog da Sprint*, *Sprint*, *Revisão da Sprint* e *Retrospectiva da Sprint*) fizeram parte do processo de desenvolvimento da aplicação.

##### 6.4.1 Planejamento

O planejamento foi realizado por meio da definição das tarefas do produto no *Backlog* do Produto (Figura 24). Estas atividades usam como fonte de requisitos o Documento de Especificação de Requisitos. As tarefas foram separadas pelos módulos que compõem o sistema: Módulo Administrativo e Módulo de Busca.

Para o Módulo Administrativo foram elencadas as seguintes atividades: criar sistema de login e recuperação de senha; manter cadastro de usuários; manter vinculação de usuário com sua respectiva categoria; manter cadastro de informações profissionais baseados nas categorias; manter cadastro de portfólio; visualizar página do prestador de serviço.

Para o Módulo Administrativo foram levantadas as seguintes tarefas: definição do algoritmo de busca; criação da área de buscas de prestadores de serviço na página inicial da aplicação; visualização da página do prestador de serviço; cadastrar o número de vezes que a página do prestador de serviço foi

visitada; cadastrar as palavras chaves usadas na pesquisa de um determinado prestador de serviço.

As tarefas do *Backlog* do Produto foram adicionadas em uma *Sprint* mensal, e foram refinadas a cada Retrospectiva do *Backlog*. Desta forma, foi possível ter incrementos mensais do produto. Finda a última *Sprint*, o Booszinga estava pronto para ser disponibilizado para os seus usuários.

Após a definição do planejamento, o próximo passo foi o levantamento do Banco de Dados, assunto este que será tratado a seguir.

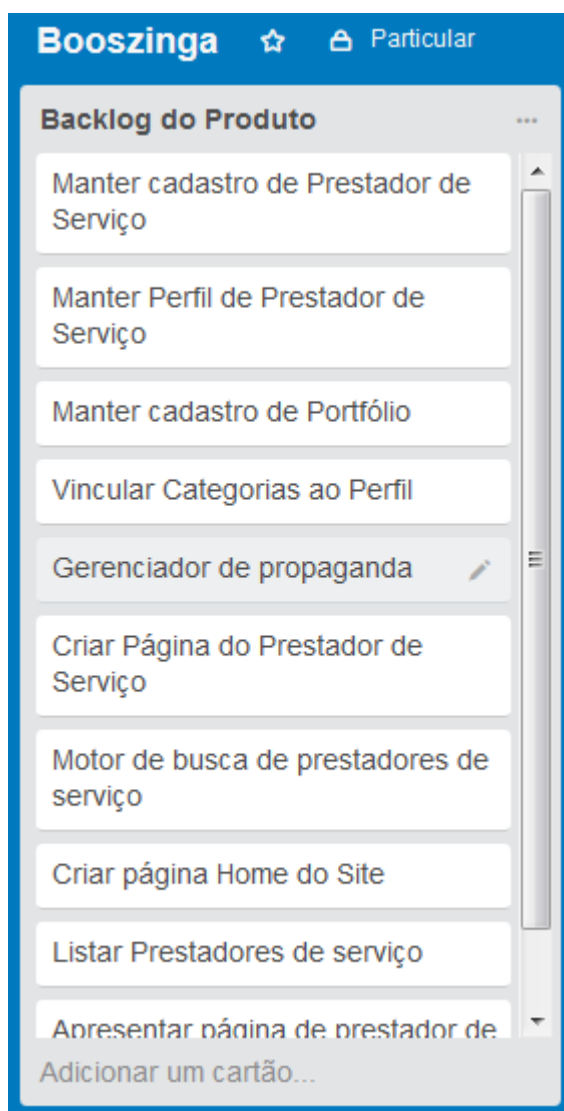


Figura 24. Backlog do Booszinga.

#### 6.4.2 Banco de Dados

O processo de implementação da aplicação começa com a construção das tabelas do banco de dados, usando como base o Diagrama de Entidade e Relacionamento presente no Documento de Especificação de Projetos. O banco de dados foi construído por meio de um *script* em SQL, conforme mostra a Figura 25. Neste *script* é apresentado a SQL que gerou a tabela `boo_pessoa`, que representa a entidade Prestador de Serviço.

```
CREATE TABLE IF NOT EXISTS `boo_pessoa` (
  `cd_pessoa` int(11) NOT NULL AUTO_INCREMENT,
  `nome_pessoa` varchar(50) NOT NULL,
  `sexo_pessoa` char(1) NOT NULL COMMENT 'm-masculino| f-feminio',
  `imagem_pessoa` varchar(50) DEFAULT NULL,
  `endereco_pessoa` varchar(100) DEFAULT NULL,
  `bairro_pessoa` varchar(30) DEFAULT NULL,
  `complemento_pessoa` text,
  `cidade_pessoa` int(11) DEFAULT NULL,
  `celular_pessoa` varchar(20) DEFAULT NULL,
  `telefone_pessoa` varchar(20) DEFAULT NULL,
  `whatsapp_pessoa` varchar(11) DEFAULT NULL,
  `email_pessoa` varchar(50) NOT NULL,
  `login_pessoa` varchar(50) NOT NULL,
  `senha_pessoa` varchar(64) NOT NULL,
  `url_pessoa` varchar(50) DEFAULT NULL,
  `site_pessoa` varchar(100) NOT NULL,
  `facebook_pessoa` varchar(80) DEFAULT NULL,
  `verificacao_pessoa` int(11) NOT NULL COMMENT '0-nao validado | 1 - validado',
  `ordem` int(11) NOT NULL DEFAULT '0',
  `data_cadastro_pessoa` datetime NOT NULL,
  PRIMARY KEY (`cd_pessoa`),
  UNIQUE KEY `email_pessoa` (`email_pessoa`,`login_pessoa`,`url_pessoa`),
  KEY `cidade_pessoa` (`cidade_pessoa`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=14 ;
```

Figura 25. Script SQL da tabela `boo_pessoa`.

A execução do *script* completo gerou as tabelas da Figura 26. Estas tabelas são apresentadas por meio do PhpMyAdmin.

Tabela ▲
<input type="checkbox"/> boo_categoria
<input type="checkbox"/> boo_chave_pessoa
<input type="checkbox"/> boo_cidade
<input type="checkbox"/> boo_comentario
<input type="checkbox"/> boo_email_recebido
<input type="checkbox"/> boo_estado
<input type="checkbox"/> boo_imagem
<input type="checkbox"/> boo_informacao_adicional
<input type="checkbox"/> boo_local
<input type="checkbox"/> boo_log_acesso
<input type="checkbox"/> boo_pais
<input type="checkbox"/> boo_pessoa
<input type="checkbox"/> boo_pessoa_categoria
<input type="checkbox"/> boo_pessoa_cidade_categoria
<input type="checkbox"/> boo_pessoa_plano_categoria
<input type="checkbox"/> boo_plano
<input type="checkbox"/> boo_portifolio
<input type="checkbox"/> boo_view_page
<input type="checkbox"/> ci_sessions
<b>19 tabelas</b>

Figura 26. Tabelas do banco de dados do Booszinga.

Estão presentes na Figura 26, quase todas as tabelas contidas no Diagrama de Entidade e Relacionamento. Para melhor entendimento, será descrito o objetivo das principais tabelas da aplicação.

- **boo\_pessoa:** tabela responsável por manter as informações padrões de cadastro dos Prestadores de Serviço, como: nome, contatos, endereço, imagem de perfil, etc.
- **boo\_categoria:** tabela responsável por manter o cadastro das informações das categorias do sistema. Categorias são tipos de Prestadores de Serviços: advogado, contador, Administrador, etc. Essa tabela é padrão do Booszinga, ou seja, ela não é mantida pela aplicação, ela já contém as principais categorias de prestadores de

serviço. Para manter o controle, caso o Prestador de Serviço não se encaixar em nenhuma das 225 categorias pré-cadastradas, este deve enviar um e-mail para o suporte, solicitando o cadastramento da mesma.

- **boo\_pessoa\_categoria:** A aplicação deve permitir que um Prestador de Serviço possa se cadastrar em várias categorias, criando uma relação de cardinalidade das tabelas boo\_pessoa e boo\_categoria de muitos para muitos (n:n). Para tal, foi criada esta tabela associativa, onde sua chave primária é chave estrangeira nas demais tabelas do sistema.
- **boo\_informação\_adicional:** tabela responsável por manter informações complementares do Prestador de Serviço. Essas informações são relacionadas a um Prestador de Serviço e sua categoria, ou seja, se o Prestador de Serviço for cadastrado em mais de uma categoria, a aplicação permitirá ele cadastrar uma informação adicional para cada uma delas. As informações adicionais são: descrição do profissional, experiências na área, média de preços praticados, etc.
- **boo\_plano:** tabela responsável por manter as informações dos tipos de planos de destaque oferecidos pelo negócio. Esta tabela é padrão, ou seja, já vem com os planos pré-cadastrados, e não é mantida pela aplicação.
- **boo\_pessoa\_plano\_categoria:** tabela associativa, responsável por manter a informação de adesão de um determinado Prestador de Serviço a um plano de negócios.
- **boo\_chave\_pessoa:** tabela responsável por manter as informações sobre as palavras chaves pesquisadas, usadas por consumidores para encontrar um Prestador de Serviço. Por exemplo: um consumidor X faz uso da palavra-chave “contador” que retorna uma lista de resultados relacionados a ela. Ao selecionar um Prestador de Serviço, a informação da palavra chave pesquisada e o Prestador de Serviço selecionado, são mantidos nesta tabela. Então, o Prestador de Serviço tem acesso a informação de quais palavras chaves foram usadas para encontrarem seu perfil.

Após a criação do banco de dados, o próximo passo na implementação da ferramenta é a definição do ambiente de desenvolvimento. Assunto este tratado a seguir.

### 6.4.3 Ambiente de Desenvolvimento

A escolha de um bom ambiente de desenvolvimento, formado por ferramentas e tecnologias eficientes, pode influenciar diretamente no tempo de desenvolvimento e na produtividade da equipe. Atualmente, por padrão o IDE (*Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado) já oferecem as opções de identificação da sintaxe por meio de cores, indentação de códigos, auto complete, etc., o que facilita o trabalho da equipe.

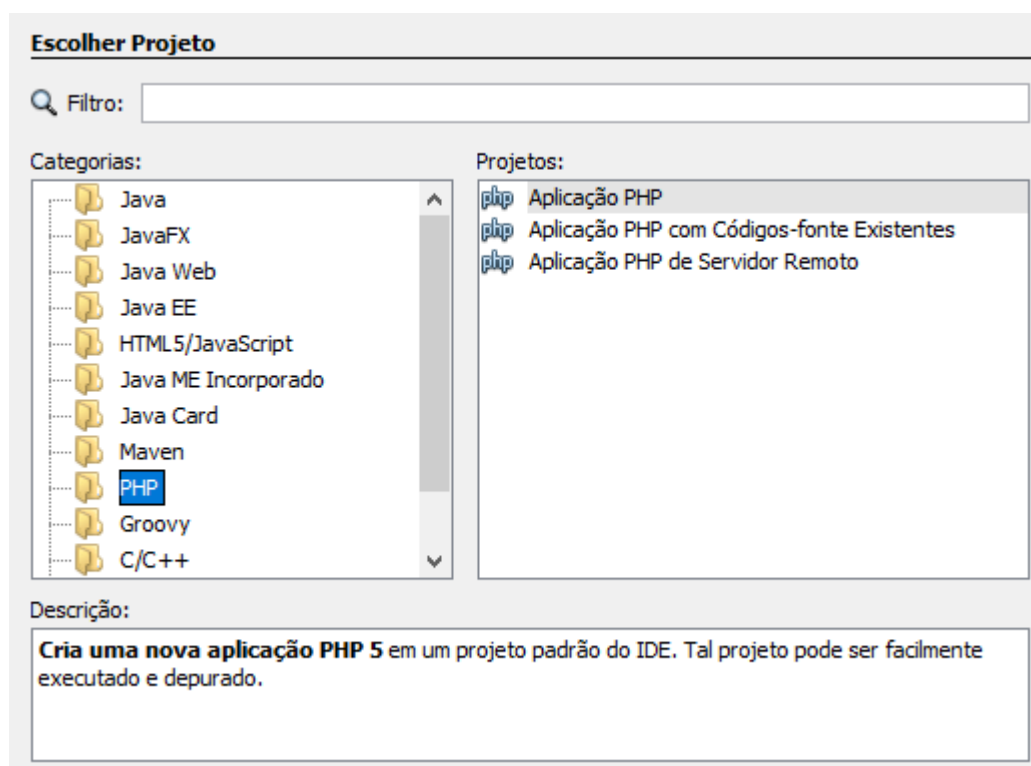


Figura 27. Janela do NetBeans IDE para escolha de tipos de projetos.

Para o desenvolvimento do Booszinga, foi utilizada a ferramenta NetBeans IDE. Por fornecer suporte ao PHP (Figura 27), que é a linguagem de programação utilizada neste trabalho. Na Figura 29 é possível ver um exemplo de código fonte do Booszinga no NetBeans IDE seguindo os padrões mencionados anteriores.

A aplicação WEB foi publicada em um servidor online. Então, de qualquer máquina conectada à internet e por meio da opção “Aplicação PHP de Servidor Remoto” (Figura 27), é possível baixar o código fonte da aplicação e desenvolver, como é mostrado na Figura 28.

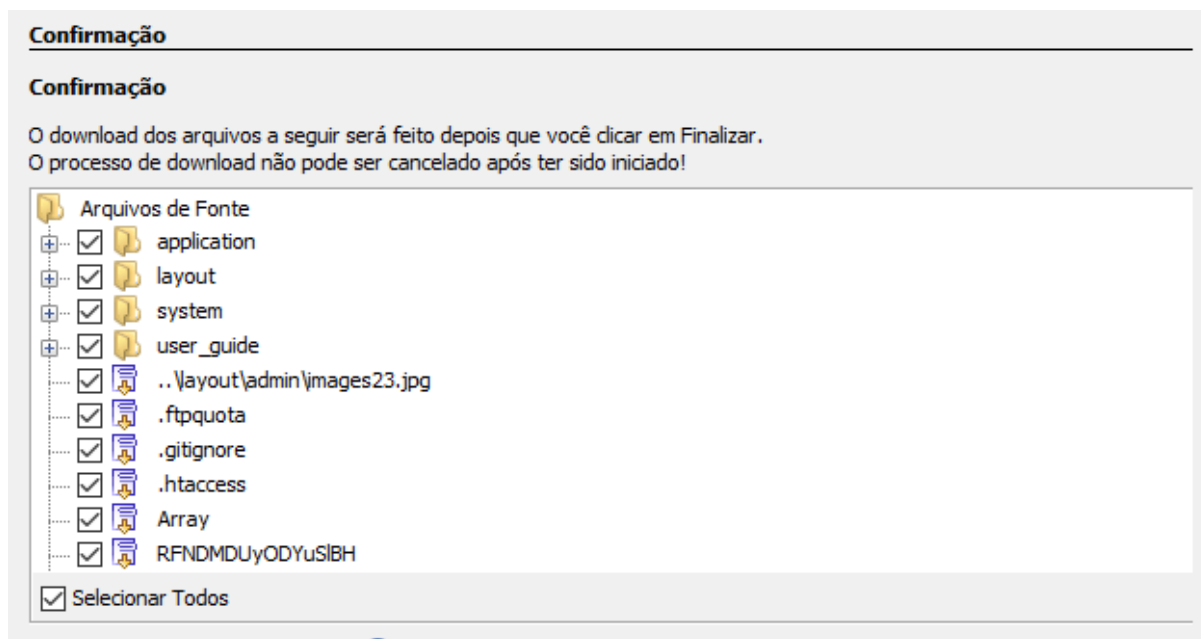


Figura 28. Janela do NetBeans IDE apresentando os arquivos dos códigos fontes do Booszinga que podem ser baixados.

A imagem mostra o editor de código fonte do NetBeans IDE com o código PHP de uma classe Categoria. O código é o seguinte:

```

1 <?php
2
3 defined('BASEPATH') OR exit('No direct script access allowed');
4
5 class Categoria extends CI_Controller {
6
7     public function __construct() {
8         parent::__construct();
9         $this->load->model("Categoria_model", "categoria");
10        $this->load->model("Pessoa_model", "pessoa");
11        if (!$this->session->userdata('session_id') || !$this->session->userdat
12            redirect(base_url("admin"));
13        }
14    }
15
16    public function index() {
17        $categoria = $this->categoria->select_all();
18        $categoria_menu = $this->categoria->select_detalhe_categoria_pessoa($th
19        $pessoa = $this->pessoa->select_pessoa_id($this->session->userdata("cd
20        $data = array(
21            "pessoa" => $pessoa,

```

Figura 29. Exemplo de código fonte no NetBeans IDE

O ambiente de desenvolvimento teve o apoio da ferramenta NetBeans IDE, que além de oferecer o suporte já mencionado, oferece também a opção

de versionamento de código por meio do plugin Git. O *site* do plugin o define da seguinte forma “O Suporte ao Git do IDE permite executar tarefas de controle de versão diretamente de seu projeto dentro do IDE”. Com o suporte do plugin é possível voltar a versões anteriores do código fonte, mesclar novas e antigas modificações, entre outras funções que facilitam e agilizam o processo de desenvolvimento.

#### 6.4.4 Finalização do Desenvolvimento do Booszinga

A etapa de finalização consiste na realização de testes e publicação da aplicação em ambiente de produção, ou seja, a disponibilização do Booszinga para uso online. O projeto foi realizado apenas por um desenvolvedor, então por não contar com uma equipe de desenvolvimento, foi realizado o teste de validação. A realização do teste de validação tem como objetivo segundo Pressman (2010) *apud* Furtado e Costa Jr. (2010) a validação dos requisitos, ou seja, a aplicação deve cumprir o que prometeu. Nesta etapa, todas as funcionalidades devem estar em conformidade com o Documento de Especificação de Requisitos.

Após a realização do teste e correções de falhas apresentadas, o sistema foi publicado na *Internet* no endereço <http://www.booszinga.com>. Para garantir que a migração do ambiente de desenvolvimento para o ambiente de produção foi um sucesso, foi realizado novamente o teste de validação.

A aplicação contínua online até a defesa deste trabalho. Para um melhor entendimento sobre o que foi desenvolvido neste trabalho, um documento de apresentação do Booszinga foi criado e está disponível no Apêndice D. Mesmo sem a divulgação ou um investimento em marketing, ela conta atualmente com vinte e três usuários ativos, espalhados pelo Brasil. Para confirmar se a *WebApp* atende as necessidades desses usuários, foi aplicado um formulário de avaliação, assunto este tratado no próximo capítulo.

## 7. AVALIAÇÃO DA FERRAMENTA

Este Capítulo tem o objetivo de apresentar os resultados da avaliação do Booszinga. A *WebApp* foi avaliada por vinte usuários da plataforma. Para estes foi enviado por e-mail o formulário que está no Apêndice E.

Dada a premência de tempo para fechamento, foram selecionados alguns usuários (de acordo com seu perfil) para fazer a avaliação, a despeito de se saber que é altamente desejável que o número de avaliadores seja bem superior, para garantir que o maior número de problemas e obstáculos de usabilidade seja identificado (e devidamente sanado) antes da implantação definitiva da ferramenta.

A avaliação levou em consideração os requisitos de qualidade: usabilidade, funcionalidade, confiabilidade e eficiência. Para cada item foram formuladas duas perguntas. Mais à frente os requisitos de qualidades são detalhados, assim como também são apresentados os resultados da avaliação da ferramenta.

### 7.1 Usabilidade

Nielsen & Loranger (2007, p. XVI) *apud* Furtado e Costa Jr. (2010, p. 59) definem usabilidade como: “um atributo de qualidade relacionado à facilidade do uso de algo.” Ou seja, a interface deve ser autoexplicativa. O projetista responsável pelo desenvolvimento da interface do sistema, deve ser capaz de transmitir, por meio da interface da aplicação, a maneira como a ferramenta deve ser usada. O usuário precisa gostar de usar a ferramenta, e ter facilidade em gravar o seu funcionamento, isto é, na próxima vez que o usuário usar a ferramenta, ele saberá exatamente o que fazer, porque conseguiu facilmente aprender como a aplicação funciona.

As duas perguntas relacionadas à usabilidade do Booszinga foram:

1. Ao acessar o Booszinga, é possível perceber clara e rapidamente o objetivo da ferramenta?
  - a. Sim, completamente;
  - b. Sim, mas algumas coisas não ficaram claras;
  - c. Não.

## 2. O Booszinga possui estética atraente e intuitiva?

- a. Sim, é esteticamente atraente e intuitivo, ou seja, é possível entender a funcionalidade de uma determinada interface somente pelo designer.
- b. Sim é esteticamente atraente, mas não é intuitivo, ou seja, às vezes não é possível entender o que determinada interface possibilita fazer.
- c. Não é atraente, mas é intuitivo.
- d. Não é atraente e nem intuitivo.

A primeira pergunta tem como objetivo constatar se o Booszinga é autoexplicativo, ou seja, sem apoio de nenhum manual ou ajuda qualquer, o usuário entra na ferramenta e entende, de imediato, por meio da interface, qual é a sua proposta. A Figura 30 mostra que para 60% dos usuários avaliadores a ferramenta consegue transmitir de maneira clara e rápida o seu objetivo. Já 30% dos usuários ficaram em dúvida sobre o assunto tratado pelo Booszinga. Aqui, constatou-se que ocorreu um número elevado de avaliadores que não entendeu a proposta da ferramenta. A solução para reduzir esse percentual foi criar uma área na aplicação que descreva o funcionamento do Booszinga. Os outros 10% não tinham a mínima noção de qual era a proposta da *WebApp*.

Usabilidade: Ao acessar o Booszinga, é possível perceber clara e rapidamente o objetivo da ferramenta?

20 respostas

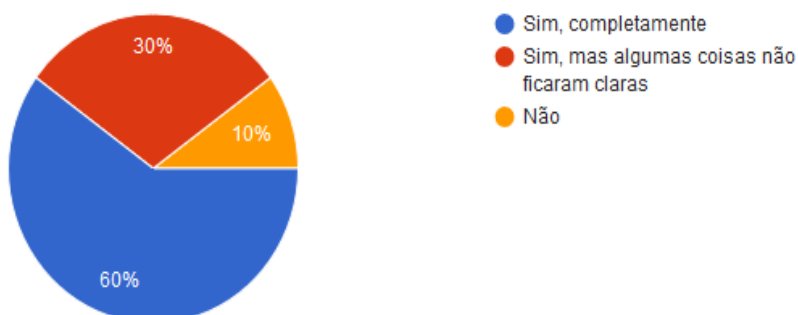


Figura 30. Resultado gráfico da primeira pergunta sobre o item de usabilidade.

A Figura 31 mostra o resultado da segunda pergunta sobre o item de usabilidade. Esta pergunta tinha como objetivo saber do usuário sua opinião sobre a estética do Booszinga, ou seja, se a ferramenta é agradável visualmente. Este item é importante, pois uma aplicação não deve ser somente funcional, ela deve ser esteticamente atraente para o usuário. Para 60% dos usuários, o Booszinga é atraente e intuitivo, ou seja, além de ter uma boa estética é possível entender o funcionamento da ferramenta por meio da sua interface. Para 25% dos usuários, a *WebApp* é esteticamente atraente, mas não consegue transmitir a forma como deve ser usada por meio da sua interface, isto é, não é intuitiva. Mas 15% avaliaram de maneira contrária, acreditam que o Booszinga não é atraente, mas é intuitivo.

Os resultados mostram que o Booszinga consegue atrair a atenção da maioria dos usuários e ao mesmo tempo consegue ser fácil de usar. O leiaute é o responsável por esse resultado satisfatório. A interface da ferramenta é limpa e objetiva, mas isso tem um efeito reverso para outros usuários, uma vez que usuários menos experientes com uso de software precisam de informações adicionais ou algum tipo de guia para entender o funcionamento de uma determinada aplicação.

### Usabilidade: O Booszinga possui estética atraente e intuitiva?

20 respostas

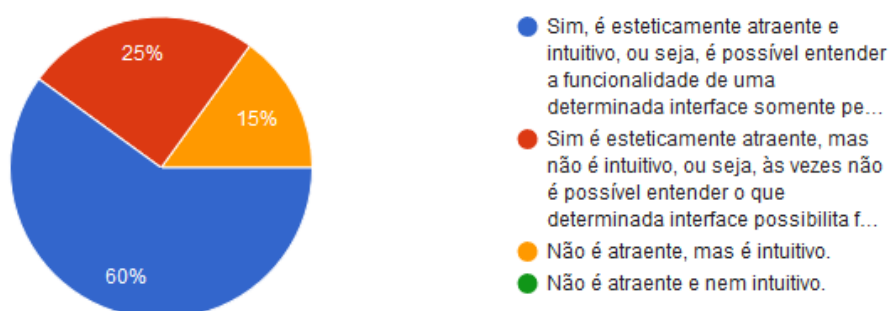


Figura 31. Resultado gráfico da segunda pergunta sobre o item usabilidade.

O Booszinga teve um bom aproveitamento nos resultados de usabilidade. A ferramenta se mostrou fácil de usar e de entender. Além de ter uma interface intuitiva, a *WebApp* precisa ser funcional. Este item será tratado a seguir.

## 7.2 Funcionalidade

Pressman (2011, p. 362) faz referência ao padrão ISO 9126 que define funcionalidade como: “O grau com que o software satisfaz às necessidades declaradas conforme indicado pelos seguintes subatributos: adequabilidade, exatidão, interoperabilidade, conformidade e segurança”. Em resumo, a aplicação deve entregar para o usuário o que foi prometido, e realizar isto de maneira adequada. Uma das características esperada pela *WebApp* é a capacidade de realizar a busca e a recuperação de dados de maneira correta.

As duas perguntas elaboradas para o item de funcionalidade foram:

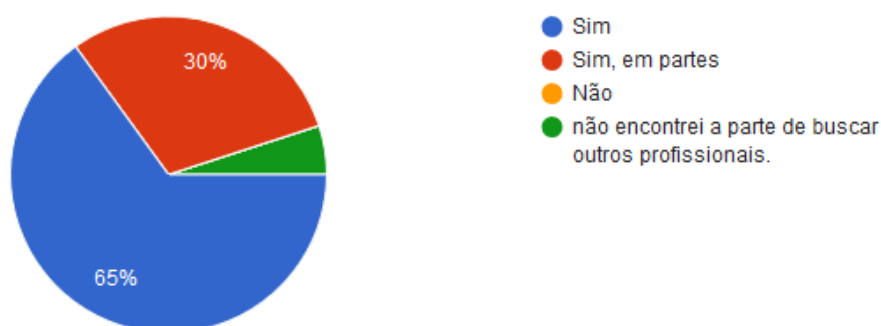
1. O Booszinga cumpre o que promete fazer, ou seja, fornece a interação entre Prestadores de Serviços e Consumidores?
  - a. Sim;
  - b. Sim, em partes;
  - c. Não;
  - d. Outros;
2. Em algum momento durante o uso do Booszinga ocorreu alguma falha inesperada?
  - a. Nenhuma vez;
  - b. 1 vez;
  - c. 2 vezes;
  - d. Entre 5 e 10 vezes;
  - e. Mais de 10 vezes;

O intuito dessas perguntas é ter o conhecimento por parte dos usuários, se a ferramenta é funcional, se a sua existência é útil. Uma aplicação existe para cumprir um objetivo, se esta não cumpre o que propõe, se torna uma ferramenta inútil. A Figura 32 mostra que para 65% dos usuários, o Booszinga cumpre o que promete, ou seja, a *WebApp* consegue promover a interação entre prestadores de serviço e consumidores. Para 35% dos usuários, a ferramenta consegue cumprir seu objetivo de certa forma; aqui o entrevistado

não tem total certeza disso, mas acredita que de algum modo a ferramenta cumpre o seu papel. Apenas um usuário não encontrou o buscador de prestadores de serviço.

### Funcionalidade: O Booszinga cumpre o que promete fazer, ou seja, fornece interação entre Prestadores de Serviços e Consumidores:

20 respostas



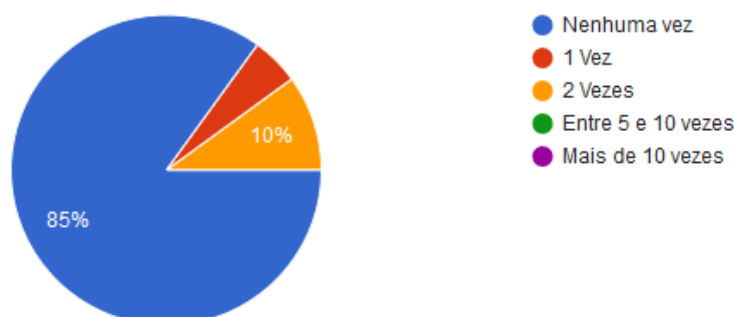
**Figura 32. Resultado gráfico da primeira pergunta sobre o item funcionalidade.**

A Figura 33 demonstra o *feedback* dos usuários em relação aos erros apresentados durante o uso da ferramenta. A primeira pergunta sobre funcionalidade é direta, ela tenta extrair dos usuários quantas vezes o sistema falhou enquanto estava em uso. O Booszinga não apresentou nenhum erro para 85% dos usuários. Para 5% dos usuários, o erro ocorreu apenas uma vez e 10% dos usuários tiveram duas ocorrências de erro.

Conforme os resultados apresentados, o Booszinga consegue atender as funcionalidades sem apresentar altos índices de erros. Para grande maioria dos avaliadores a ferramenta não apresentou erros, e isso se deve a baixa complexidade das funcionalidades. A *WebApp* é uma ferramenta que trabalha com cadastros e recuperação destes cadastros. Algumas falhas que podem apresentar são relacionadas ao envio de e-mail, ocorreram relatos por parte dos usuários que o e-mail de validação de cadastro não estava chegando. E isto pode ser o responsável pelas porcentagens dos erros que ocorreram.

## Funcionalidade: Em algum momento durante o uso do Booszinga ocorreu alguma falha inesperada?

20 respostas



**Figura 33. Resultado gráfico da segunda pergunta sobre o item de funcionalidade.**

O Booszinga se mostrou bastante funcional, com aprovação da maioria dos usuários que participaram da avaliação. Além de cumprir o objetivo proposto, a ferramenta raramente apresenta erros. Por ser funcional e não apresentar erros, a *WebApp* se torna confiável, e este é o item que será tratado a seguir.

### 7.3 Confiabilidade

Para Sommerville (2011), o software é confiável quando não apresenta falhas durante o seu uso, ou seja, funciona corretamente. Grady e Caswell (1987) *apud* Pressman (2011, p. 210) complementa essa informação, levantando os seguintes pontos: qual a frequência em que ocorrem as falhas no sistema?; qual a severidade destas falhas?; como o sistema se recupera dessas falhas? Fazendo as medições adequadas, é possível avaliar se o sistema é confiável ou não.

Para extrair a opinião dos usuários em relação à confiabilidade, foram feitas as seguintes perguntas:

1. Todos os links do Booszinga direcionam para páginas corretas?
  - a. Sim;

- b. Não;
2. O Booszinga consegue dar *feedbacks* e orientar o usuário durante o seu uso?
- a. Sim;
  - b. Às vezes;
  - c. Não;

O usuário precisa sentir confiança durante o uso da ferramenta. Deve-se evitar a existência de *links* quebrados (*links* para páginas inexistentes). Este tipo de comportamento de uma aplicação faz com que o usuário se irrite e abandone o uso da ferramenta. Outro fator que passa confiança para o usuário são os *feedbacks*. Caso ocorra algum erro, é desejável que a aplicação avise o usuário sobre esta ocorrência. A solução ideal é que a *WebApp* consiga se recuperar do erro, dando continuidade na transação iniciada pelo usuário.

A Figura 34 mostra que, para todos os usuários avaliados, o Booszinga é apresenta links que realmente funcionam e direcionam para páginas reais. Em relação aos *feedbacks*: podemos ver na Figura 35 que 60% dos usuários acreditam que a ferramenta consegue dar um retorno satisfatório durante o seu uso. Para 25% somente às vezes a ferramenta cumpre esse papel, enquanto que 10% acreditam que a ferramenta não dá nenhum tipo de auxílio quando ocorre algum erro. Isto se deve ao fato de que a ferramenta informa sobre o erro ocorrido, mas em algumas situações ela não mostra uma solução para a falha. Um exemplo disso é no momento da validação de cadastro por e-mail: a ferramenta informa que usuário não pode entrar no sistema porque não validou o seu cadastro, mas não oferece uma opção de reenvio de e-mail, então se o e-mail de validação de cadastro não chegou à caixa de entrada do usuário, ou foi perdido por algum motivo, o usuário não consegue entrar no sistema, lhe restando apenas fazer outro cadastro, ou entrar em contato com o suporte da ferramenta.

### Confiabilidade: Todos os links do Booszinga direcionam para páginas corretas?

20 respostas

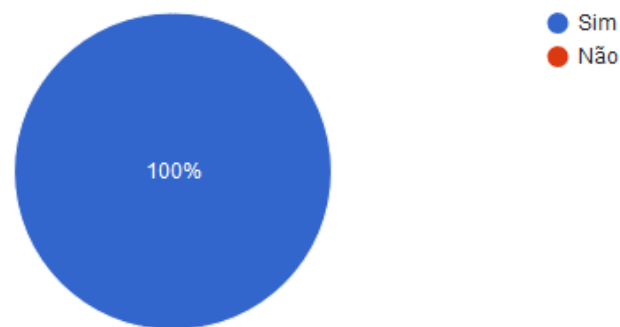


Figura 34. Resultado gráfico da primeira pergunta sobre o item de confiabilidade.

### Confiabilidade: O Booszinga consegue dar feedbacks e orientar adequadamente o usuário durante o seu uso?

20 respostas

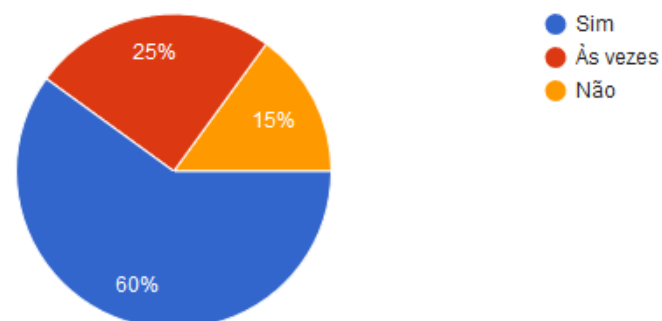


Figura 35. Resultado gráfico da segunda pergunta sobre o item de confiabilidade.

O Booszinga obteve um resultado positivo no quesito confiabilidade. Alguns pontos precisam melhorar em relação aos *feedbacks*, estas melhorias estão previstas para o próximo incremento da ferramenta. Além de ser

funcional e confiável, a ferramenta precisa ser eficiente. Este item é tratado a seguir.

#### 7.4 Eficiência

Atualmente os usuários estão cada vez mais exigentes em relação à eficiência no uso de um software. A cada dia mais pessoas têm acesso à internet de qualidade. Não cabe mais esperar minutos para carregar uma imagem ou alguma informação em uma página *web*. Sommerville (2011, p. 5) faz a seguinte colocação em relação à eficiência de um software: “O software não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. Portanto, eficiência inclui a capacidade de resposta, tempo de processamento, uso de memória, etc.”.

Em relação à eficiência foram elaboradas as seguintes perguntas:

1. O Booszinga consegue processar em poucos segundos as requisições feitas?
  - a. Sim, estou satisfeito com o tempo de resposta da aplicação;
  - b. Não, demora razoavelmente;
  - c. Não, demora um pouco;
2. As páginas do Booszinga são carregadas em tempo aceitável?
  - a. Sim;
  - b. Não;
  - c. Às vezes;

Estas perguntas têm o objetivo de extrair do usuário a sua opinião sobre a eficiência da ferramenta. Os resultados foram totalmente satisfatórios. Para ambas as perguntas os usuários foram unânimes nas respostas, apontando a ferramenta como totalmente eficiente. Todos os avaliados afirmaram que o Booszinga consegue dar retorno do que lhe foi proposto em tempo aceitável, conforme pode ser visto nas Figuras 36 e 37.

### Eficiência: O Booszinga consegue processar em poucos segundos as requisições feitas?

20 respostas

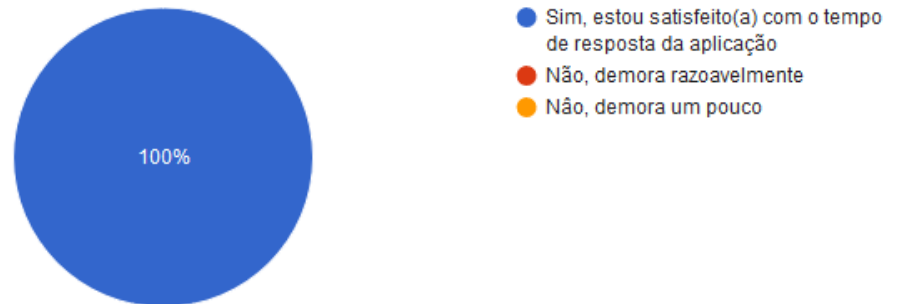


Figura 36. Resultado gráfico da primeira pergunta sobre o item de eficiência.

### Eficiência: As páginas do Booszinga são carregadas em tempo aceitável?

20 respostas

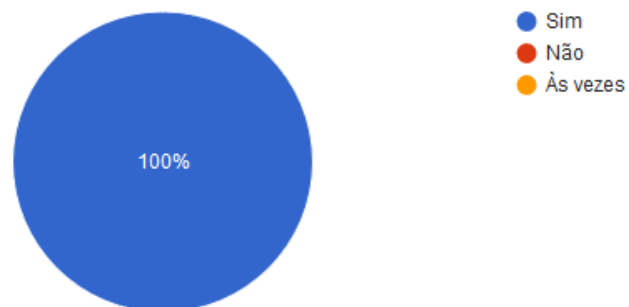


Figura 37. Resultado gráfico da segunda pergunta sobre o item de eficiência.

De modo geral, pode-se depreender da avaliação realizada que a ferramenta foi bem concebida e bem implementada, atendendo adequadamente aos objetivos propostos inicialmente. A avaliação foi importante para a reflexão sobre melhorias para o próximo incremento da *WebApp*.

Este Capítulo fecha este trabalho. Por meio da avaliação foi possível

fechar todo o ciclo em torno do processo de desenvolvimento. O Booszinga foi idealizado, concebido por meio das documentações de software, implementado, testado e por fim avaliado pelo usuário final.

## 8. CONCLUSÃO

Foram objetos de estudo deste trabalho: a concepção, a especificação, a implementação e a avaliação da *WebApp* Booszinga. Depois da concepção, fez-se a especificação de requisitos, a especificação de análise e a especificação do projeto. Com isto pronto, foi possível o desenvolvimento do sistema proposto. Para validar se a ferramenta realmente atingiu seu objetivo, foi realizada avaliação por uma amostra de seus potenciais usuários. Esta avaliação se mostrou positiva, validando o trabalho desenvolvido.

Com a elaboração deste trabalho de conclusão de curso foi possível pôr em prática grande parte do que foi aprendido na academia. O processo de desenvolvimento de um software é complexo. Em razão disso, e para tratar adequadamente a complexidade envolvida, o processo é organizado em etapas; estas devem ser cumpridas para que os objetivos sejam atingidos plenamente.

O desenvolvimento do trabalho contribuiu significativamente para o entendimento do processo de desenvolvimento de software. Apesar de ter sido elaborado por uma pessoa somente, observou-se a necessidade de uma equipe, com pessoas e funções bem definidas para realizar tarefa de tal magnitude. Uma equipe ideal seria formada por: Analista de Requisitos, Analista de Sistemas, Administrador de Banco de Dados, Programador, Analista de Testes e Analista em IHC. Então, com a conclusão do Booszinga, como lições aprendidas com o trabalho, foi possível assimilar conhecimentos da prática de cada um desses profissionais.

Como visto no Capítulo anterior, apesar de ter um resultado positivo nas avaliações, foi apontado que a ferramenta precisa de algumas melhorias. Estas melhorias constituirão trabalhos futuros, pois, mesmo após a defesa do TCC, novos incrementos da ferramenta serão desenvolvidos. Nesta linha, pretende-se também, para futuro próximo, que a ferramenta se torne uma aplicação móvel. Portanto, há interesse de levar a *WebApp* para dispositivos móveis, permitindo seu uso em aparelhos que utilizem sistemas operacionais Android e iOS.

## REFERÊNCIAS

BENTO, Evaldo Junior. **Desenvolvimento web com PHP e MySQL**. 1. ed. Editora Casa do Código, 2014.

BOOCH, Grady et al. **UML: Guia do usuário**. 6. ed. Rio de Janeiro: Elsevier, 2006.

CINTRA, Flavia Cristina. Marketing Digital: a era da tecnologia on-line. **Revista Investigação**, São Paulo, v. 10, n.1, p. 6-12, mai. 2010.

FURTADO, Alfredo Braga; COSTA JR., Júlio Valente da. **Prática de Análise e Projeto de Sistemas**. 1 ed. Belém: abfurtado.com.br, 2010.

GABARDO, Ademir Cristiano. **PHP e MVC com CodeIgniter**. 1. ed. São Paulo: Novatec, 2012.

GASPARIN, Gabriela. **Entenda como a crise de 2008 influenciou a vida dos brasileiros**. São Paulo, 2011. Disponível em <<http://g1.globo.com/economia/seu-dinheiro/noticia/2011/09/entenda-como- crise-de-2008-influenciou-vida-dos-brasileiros.html>>. Acesso em 06 de maio de 2017.

MELLO, Diego Teixeira de. **Portal “Quem Conserta”**. 2004, 58f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.

MILANI, André. **MySQL: Guia do Programador**. 1. ed. São Paulo: Novatec, 2006.

NIEDERAUER, Juliano. **Desenvolvendo Websites com PHP**. 2. ed. São Paulo: Novatec, 2011.

POHL, Klaus; RUPP, Chris. **Fundamentos da Engenharia de Requisitos**. 1. ed. São Paulo: T&M Teste de Software Ltda., 2012.

PRESSMAN, Roger S. **Uma Abordagem Profissional**. 7. ed. São Paulo: Pearson Books, 2011.

SABINO, Vanessa; KON, Fabio. 2009. **Licenças de Software Livre História e Características**. São Paulo, USP (Relatório Técnico RT-MAC-IME-USP 2009-1).

SANT'ANA, Jessica. **Crise econômica leva quatro em cada dez brasileiros a empreender em 2015**. Curitiba, 2016. Disponível em <<http://www.gazetadopovo.com.br/economia/empreender-pme/crise-economica-leva-quatro-em-cada-dez-brasileiros-a-empreender-em-2015-4wqj7ssfe3kpi2k1pxmn8ltq>>. Acesso em 06 de Maio de 2017.

SCHWABER, Ken; SUTHERLAN, Jeff. **Um guia definitivo para o Scrum: As regras do jogo**. 1. ed. 2013.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

**Usando Suporte Git no NetBeans IDE**. Disponível em <[https://netbeans.org/kb/docs/ide/git\\_pt\\_BR.html](https://netbeans.org/kb/docs/ide/git_pt_BR.html)>. Acesso em 20 de Junho de 2017.

## **APÊNDICE A – DOCUMENTO DE ESPECIFICAÇÃO DE REQUISITOS**

### **1. Introdução**

Este documento contém a especificação de requisitos para o sistema de busca online de empreendedores, denominado aqui de booszinga, com a utilização da técnica de modelagem de Casos de Uso.

O sistema abrange um buscador de empreendedores online, que vai desde o cadastro até a busca online de forma otimizada, ou seja, o retorno das requisições de buscas mediante palavras-chave será apresentado para o usuário no máximo até 5 segundos.

### **2. Ambiente do sistema**

Este sistema consiste de uma plataforma online de cadastro e busca de empreendedores. Para isso será desenvolvido um ambiente administrativo (virtual), no qual o empreendedor faz o cadastramento do seu perfil, do portfólio de suas atividades e de quaisquer informações que julgar importantes para divulgar seus produtos ou serviços. Desta forma, potenciais clientes desses produtos ou serviços oferecidos pelo profissional podem encontrá-lo por intermédio do ambiente de busca. Com a interação entre as partes, há a possibilidade de que negócios sejam fechados. Sempre que isto ocorrer (negócios forem concretizados), o Booszinga atinge seu objetivo.

Considera-se uma pessoa interessada em divulgar seus serviços e produtos de forma online e, para isso, ela deverá realizar todos os cadastros necessários. Neste documento, essa pessoa é identificada como Empreendedor, que, nesse caso, é o usuário que terá acesso ao painel administrativo do sistema.

O Empreendedor é uma pessoa interessada em divulgar seus serviços e produtos de maneira online, disponibilizando seu portfólio, seus contatos e informando sua experiência profissional e faixa de preços que pratica em sua atuação no mercado. Na maioria dos casos, ele poderá pertencer a várias categorias de serviços, ou seja, ele pode ser um programador, designer e mesmo fazer outros trabalhos que não sejam de natureza tecnológica.

Do outro lado, há o Cliente, que tem a necessidade de adquirir um determinado produto ou serviço e que necessita dessa informação em tempo hábil e de forma precisa e consistente. Ou seja, se ele procura um advogado na cidade de Belém que atende a uma determinada faixa de preço, ele deve encontrar essa informação de maneira simples, rápida e precisa.

### **3. Objetivos do produto**

Esse sistema objetiva oferecer uma ferramenta online que possibilite que empreendedores divulguem seus serviços e produtos e que estes possam ser encontrados por potenciais clientes na internet.

### **4. Benefícios do produto**

As seguintes funcionalidades serão consideradas no sistema a ser desenvolvido:

- Painel administrativo

- Cadastro e manutenção do perfil do empreendedor

- Cadastro e manutenção de categorias

- Cadastro e manutenção de portfólio

- Cadastro e manutenção das informações adicionais relacionadas às categorias

- Recuperação de senha e dados

- Visualização da página do empreendedor

- Relatórios

- Área de buscas

- Busca de empreendedores por palavras-chave

- Página com perfil do empreendedor

- Salvar informações de quantidade de visualizações

- Salvar palavras-chave pesquisadas para determinado resultado.

### **5. Modelos de Casos de Uso**

Neste sistema, temos os atores listados abaixo:

- **Ator Empreendedor:** usuário do sistema responsável pela alimentação de dados de perfil, portfólio, categorização e informações adicionais sobre seus produtos ou serviços que oferece.
- **Ator Cliente:** usuário do sistema responsável pelas buscas de empreendedores e requisições de respostas ágeis e corretas.

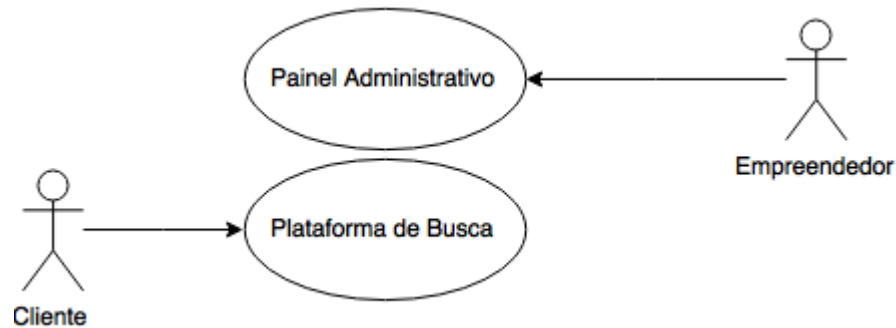


Figura 1.0 Diagrama de Casos de Uso de Contexto.

### 5.1 Subsistema Administrativo

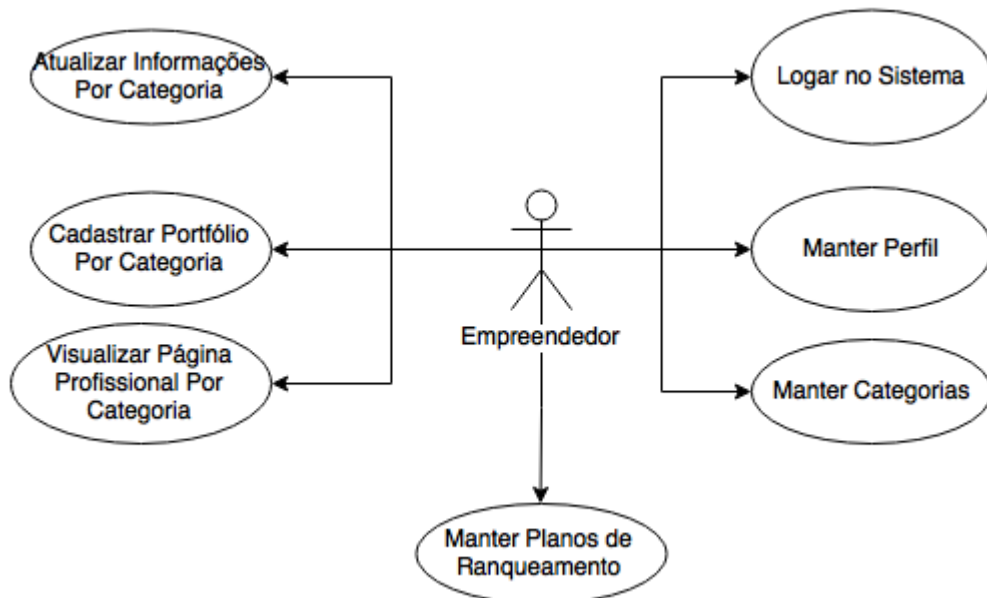


Figura 1.1 Diagrama de Casos de Uso do Subsistema Administrativo

#### • CASO DE USO 1: LOGAR NO SISTEMA

**Descrição:** O empreendedor deverá entrar com login e senha e o sistema deverá validar suas credenciais permitindo acesso ou não ao sistema.

**Ator primário:** Empreendedor

**Precondições:** O empreendedor já está cadastrado no sistema.

**Fluxo Principal:**

1. O empreendedor entra com as credenciais de login e senha pré cadastradas.
2. O sistema processa as informações e valida o login.
3. Caso o login seja válido o usuário entra no sistema administrativo.
4. Caso o login seja inválido o usuário não entra no sistema administrativo.

**Pós-Condições:** nenhuma

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Diário.

- **CASO DE USO 2: MANTER PERFIL**

**Descrição:** O empreendedor mantém o seu perfil (inclusão e atualização de informações pessoais, inclusão e atualização de informações profissionais relacionadas a uma determinada categoria).

**Ator primário:** Empreendedor

**Precondições:** O usuário deve estar logado no sistema.

**Fluxo Principal:**

1. O empreendedor atualiza seus dados pessoais e profissionais através de um formulário.
2. O empreendedor salva as informações.

**Pós-Condições:** O sistema deve mostrar uma mensagem sobre o sucesso ou fracasso da operação.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Eventual.

**Fluxo alternativo(3):** Desvincular da Categoria.

- a. Empreendedor escolhe a opção de se desvincular da categoria
- b. O sistema deve verificar se a categoria já tem informações atreladas como portfólios ou informações profissionais.

**Pós-Condições:** Informar o usuário sobre o sucesso ou fracasso dessa operação.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Eventual.

- **CASO DE USO 3: MANTER CATEGORIA**

**Descrição:** O empreendedor vincula-se a uma ou mais categorias.

**Ator primário:** Empreendedor

**Precondições:** O usuário deve estar logado no sistema.

**Fluxo Principal:**

1. O empreendedor vincula-se a uma ou mais categorias
2. O empreendedor desvincula-se de uma ou mais categorias.

**Pós-Condições:** Caso ele já tenha informações cadastradas naquela categoria, o sistema deve retornar uma mensagem de erro.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Eventual.

- **CASO DE USO 4: ATUALIZAR INFORMAÇÕES POR CATEGORIA**

**Descrição:** O empreendedor, para cada nova categoria cadastrada, deverá atualizar as informações referentes a esta.

**Ator primário:** Empreendedor

**Precondições:** O usuário deve estar logado no sistema.

**Fluxo Principal:**

1. O empreendedor insere os dados sobre seu perfil profissional (sobre seu trabalho, redes sociais, faixa de preços e tempo de experiência) relacionado a categoria que ele selecionou.
2. O usuário salva as informações.

**Pós-Condições:** O sistema deve retornar uma mensagem de sucesso ou fracasso para essa operação.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Eventual.

- **CASO DE USO 5: CADASTRAR PORTFÓLIO POR CATEGORIA**

**Descrição:** O empreendedor a cada nova categoria cadastrada, poderá cadastrar seu portfólio referentes a esta.

**Ator primário:** Empreendedor

**Precondições:** O usuário deve estar logado no sistema.

**Fluxo Principal:**

1. O empreendedor faz upload de até 5 imagens por vez.
2. O empreendedor cadastra títulos e descrições das imagens.
3. O empreendedor exclui e altera informações das imagens.
4. O empreendedor marcar uma imagem como destaque, para essa ser a primeira imagem na ordem.

**Pós-Condições:** O sistema deve retornar uma mensagem de sucesso ou fracasso para essa operação.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Mensal.

- **CASO DE USO 6: VISUALIZAR PÁGINA PROFISSIONAL POR CATEGORIA**

**Descrição:** O empreendedor visualiza a sua página na web referente a uma determinada categoria.

**Ator primário:** Empreendedor

**Precondições:** O usuário deve estar logado no sistema.

**Fluxo Principal:**

1. O empreendedor clica na opção de visualizar página referente a uma determinada categoria.
2. Um modelo da sua página com as informações que ele já cadastrou é apresentado.

**Pós-Condições:** nenhuma.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Diário.

- **CASO DE USO 7: MANTER PLANO DE RANQUEAMENTO**

**Descrição:** O empreendedor mantém o cadastro de planos.

**Ator primário:** Empreendedor

**Precondições:** O usuário deve estar *logado* no sistema.

**Fluxo Principal:**

1. O empreendedor escolhe um plano de ranqueamento.
2. O empreendedor escolhe 5 palavras-chave que se enquadram no seu perfil.
3. O empreendedor salva as informações.

**Pós-Condições:** O sistema deve retornar uma mensagem de sucesso ou fracasso para essa operação.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Eventual.

## 5.2 Subsistema Plataforma de Busca

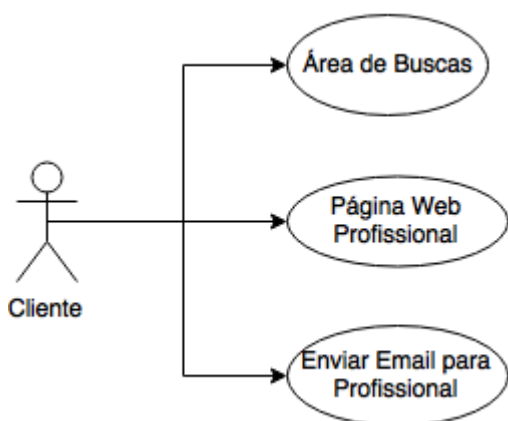


Figura 1.2 Diagrama de Casos de Uso do Subsistema Plataforma de Busca

### ● CASO DE USO 1: ÁREA DE BUSCAS

**Descrição:** O cliente pesquisa através de palavras chaves e encontra os serviços ou produtos de seu interesse.

**Ator primário:** Cliente

**Precondições:** nenhuma.

**Fluxo Principal:**

1. O cliente insere palavras chaves em um campo de busca único.
2. O sistema retorna o profissional que mais se adequa ao que o cliente procura, levando em consideração as seguintes regras de ranqueamento:
  - a. Classificação Orgânica:
    - i. Do cadastro mais antigo;
    - ii. Do cadastro com atualizações mais recentes.;
    - iii. Com perfil mais completo;
    - iv. Mais informações de portfólio;
    - v. Com maior número de visualizações de perfil;
  - b. Classificação Paga
    - i. Plano Premium
    - ii. Plano Ouro

- iii. Plano Prata
- iv. Plano Avulso

**Pós-Condições:** O sistema deve listar os profissionais baseado nas regras de ranqueamento.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Diário.

- **CASO DE USO 2: PÁGINA WEB PROFISSIONAL**

**Descrição:** O cliente escolhe um dos profissionais listados e entra na sua página web.

**Ator primário:** Cliente

**Precondições:** nenhuma.

**Fluxo Principal:**

1. O cliente clica no resumo do profissional do seu interesse
2. O sistema redireciona o cliente para a página do profissional escolhido
3. O sistema mostra a página do clientes com todos os dados cadastrados pelo mesmo

**Pós-Condições:** nenhuma

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Diário.

- **CASO DE USO 2: ENVIAR EMAIL PARA O PROFISSIONAL**

**Descrição:** O cliente entra em contato com o profissional através do formulário de email na página do cliente.

**Ator primário:** Cliente

**Precondições:** Empreendedor deve ter um email pré-cadastrado..

**Fluxo Principal:**

1. O cliente insere as suas informações (nome, email e telefone) e a mensagem que deseja enviar para o profissional no formulário de contato.
2. Ele clica no botão enviar.
3. A mensagem é enviada para o email pré-cadastrado do empreendedor.

**Pós-Condições:** O sistema deve retornar uma mensagem de sucesso ou fracasso para essa operação.

**Prioridade de Desenvolvimento:** 1.

**Frequência de uso:** Diário.

## **6. Requisitos não funcionais**

**6.1 Segurança:** O sistema deve prover facilidade para autenticação de todos os usuários, com a digitação de usuário e senha. As operações realizadas numa seção deverão registrar quem as executam.

**6.2 Desempenho:** Todas as operações realizadas deverão ser executadas em no máximo 1 segundo.

**6.3 Interface:** Deve apresentar boa usabilidade, já que os usuários farão uso diariamente.

**6.4 Portabilidade:** O sistema deve permitir ampla portabilidade, de modo a executar em ambientes operacionais diversos.

## **7. Restrições do software**

O software a ser desenvolvido deve rodar em todos os navegadores atualizados de qualquer tipo de dispositivo; a linguagem de programação escolhida para a implementação deve garantir código eficiente e compacto, o banco de dados (relacional) a ser utilizado deve ser livre.

## APÊNDICE B – DOCUMENTO DE ESPECIFICAÇÃO DE ANÁLISE

### 1. Introdução

Este documento contém a especificação da análise do BOOSZINGA (Sistema de Pesquisa de Prestadores de Serviços). Abrange o modelo de classes (diagrama de pacotes, de classes e dicionário de dados) e o modelo comportamental (diagrama de estados, de sequência, de atividade, de colaboração e a descrição das operações).

### 2. Modelo de Classes

Este modelo é útil para a identificação de classes (e seu agrupamento em pacotes), de relacionamentos, de atributos e de operações.

Nem sempre é possível identificar todas essas características nesse ponto da análise, sendo necessário recorrer ao modelo comportamental para complementar a especificação. Os resultados obtidos nessas duas fases, no entanto, encontram-se condensados nos diagramas apresentados adiante.

#### 2.1 Diagrama de Pacotes

Este diagrama objetiva garantir uma visualização de mais alto nível do sistema provendo um agrupamento lógico das classes.

Para que se alcance o objetivo proposto, deve-se partir da análise do problema e da solução detalhada nos casos de uso. Abaixo se encontra o diagrama de pacote booszinga baseado no modelo MVC (Model-View-Controller). MVC é um padrão ou arquitetura de desenvolvimento que particionou o processo de criação e manutenção de sistemas buscando a escalabilidade e eficiência da aplicação [SILVA, 2012]. As classes que estiverem dentro do pacote admin são referentes ao painel administrativo e o que estiver fora são referentes ao ambiente de buscas.

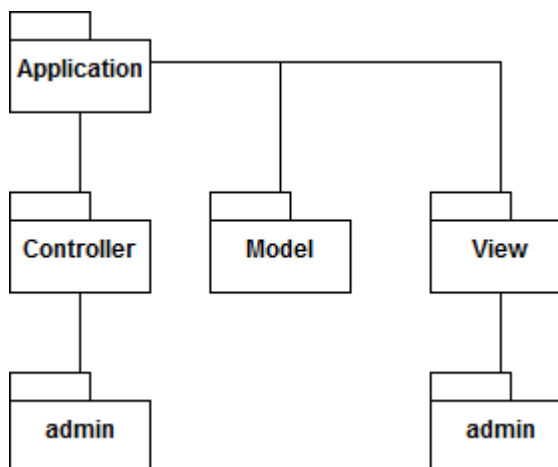


Figura 1.0 Diagrama de Pacotes do Booszinga.

## 2.2. Diagrama de Classes

Temos abaixo a Figura 2.0 que mostra o diagrama de classes referente ao pacote application/model, no padrão MVC. O modelo trabalha na manipulação dos dados internos de uma aplicação, e se comunica especialmente com o armazenamento de dados [SILVA, 2012]. A Figura 3.0 mostra o diagrama de classes referente ao pacote application/controller/admin, e a Figura 4.0 mostra o diagrama de classes referente ao pacote application/controller. No padrão MVC, a camada de Controle exerce funcionalidades que envolvem o comportamento da aplicação; controla os fluxos entre as camadas de Visão e Modelo, e gera a resposta ao usuário.

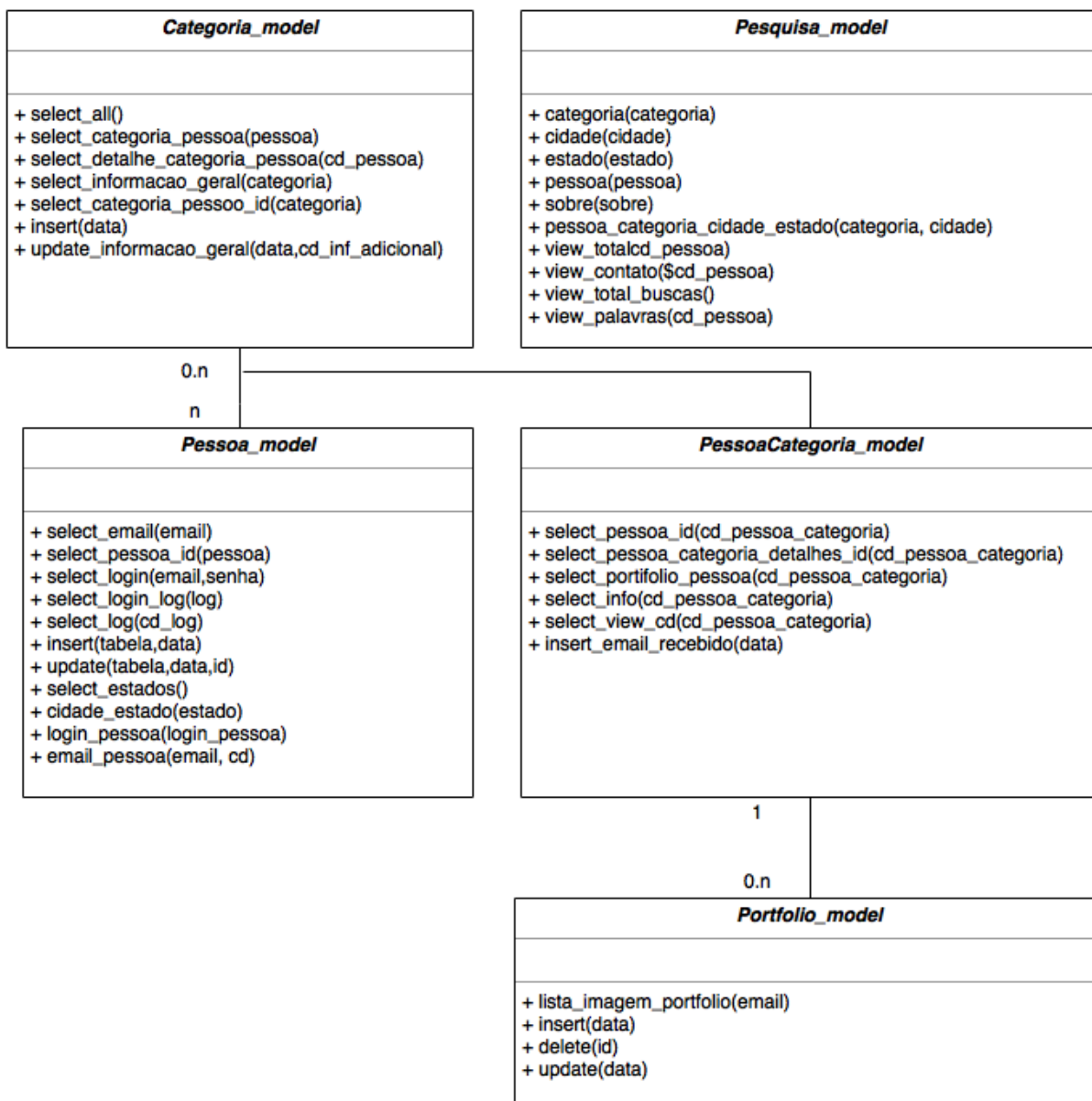


Figura 2.0 Diagrama de Classes do pacote application/model.

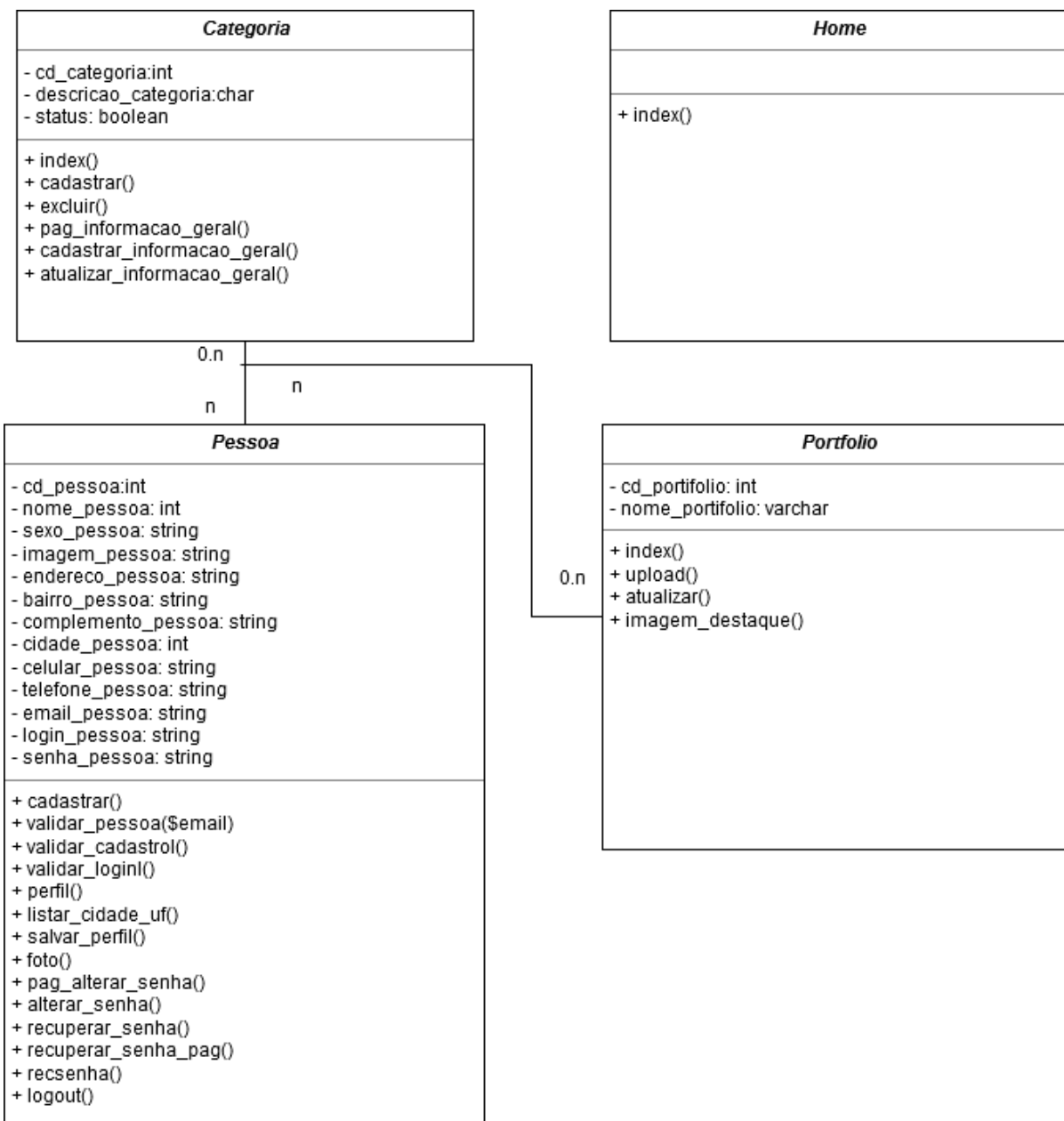


Figura 3.0 Diagrama de Classes do pacote application/controller/admin.

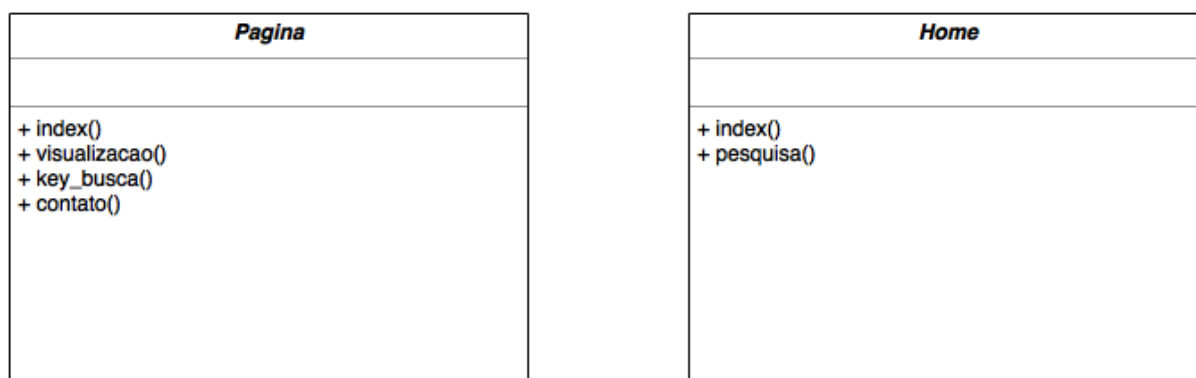


Figura 4.0 Diagrama de Classes do pacote application/controller.

## 2.3. Diagrama de Sequência

### 2.3.1. UC01: Logar no sistema

A Figura 5.0 detalha os caminhos que o ator Empreendedor terá que seguir para conseguir logar no sistema. A sequência dos passos inicia com o ator inserindo seus dados no formulário de login; a submissão desses dados chama o método `validar_login()` que está presente na classe `Pessoa` situada no pacote controller; esse método usa a função `select_login` que está presente na classe `Pessoa_model` que, por sua vez, está situada no pacote model. A função tem dois retornos, podendo ser nulo (ou seja, não achou o usuário e o sistema retorna para a tela de login), ou trará os dados do usuário fazendo com que ele tenha acesso ao painel administrativo.

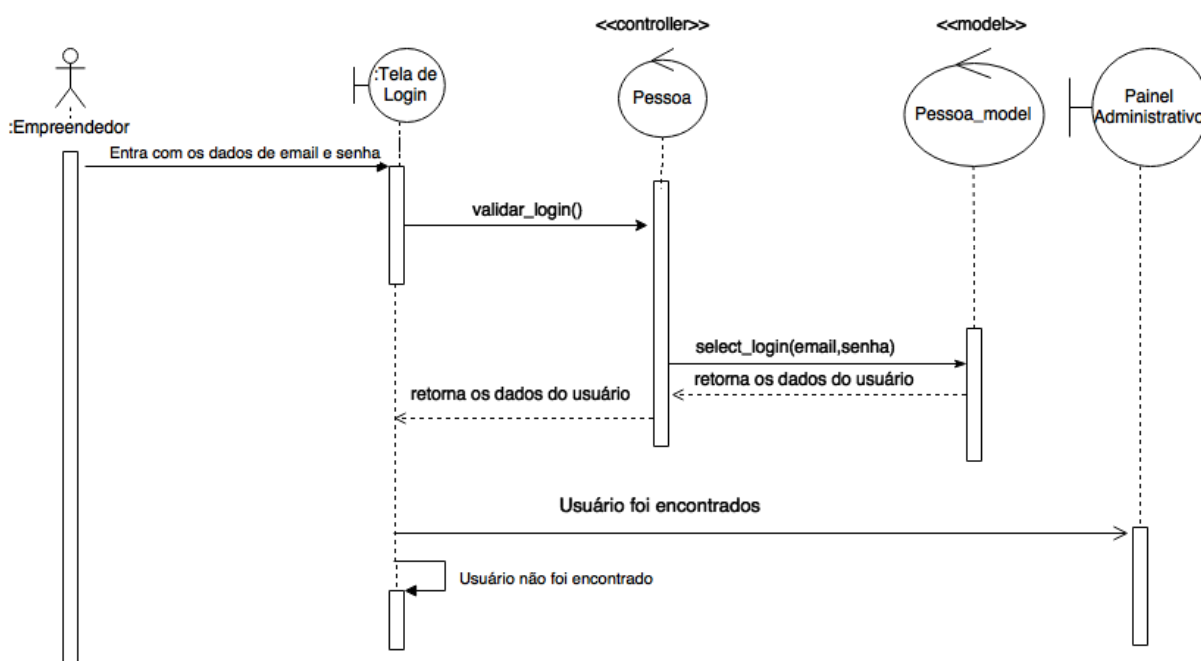


Figura 5.0 Diagrama de Sequência do funcionamento do UC01.

### 2.3.2. UC02: Manter perfil

A Figura 6.0 detalha os caminhos que o ator Empreendedor terá que

seguir para manter seu cadastro atualizado no sistema. A sequência dos passos inicia com o ator inserindo seus dados no formulário de cadastro que está disponível na tela de cadastro; a submissão desses dados chama o método `salvar_perfil` que está presente na classe `Pessoa` situada no pacote `controller`; esse método usa a função `update` que está presente na classe `Pessoa_model` que, por sua vez, está situada no pacote `model`. A função retorna uma mensagem de erro ou sucesso ao usuário.

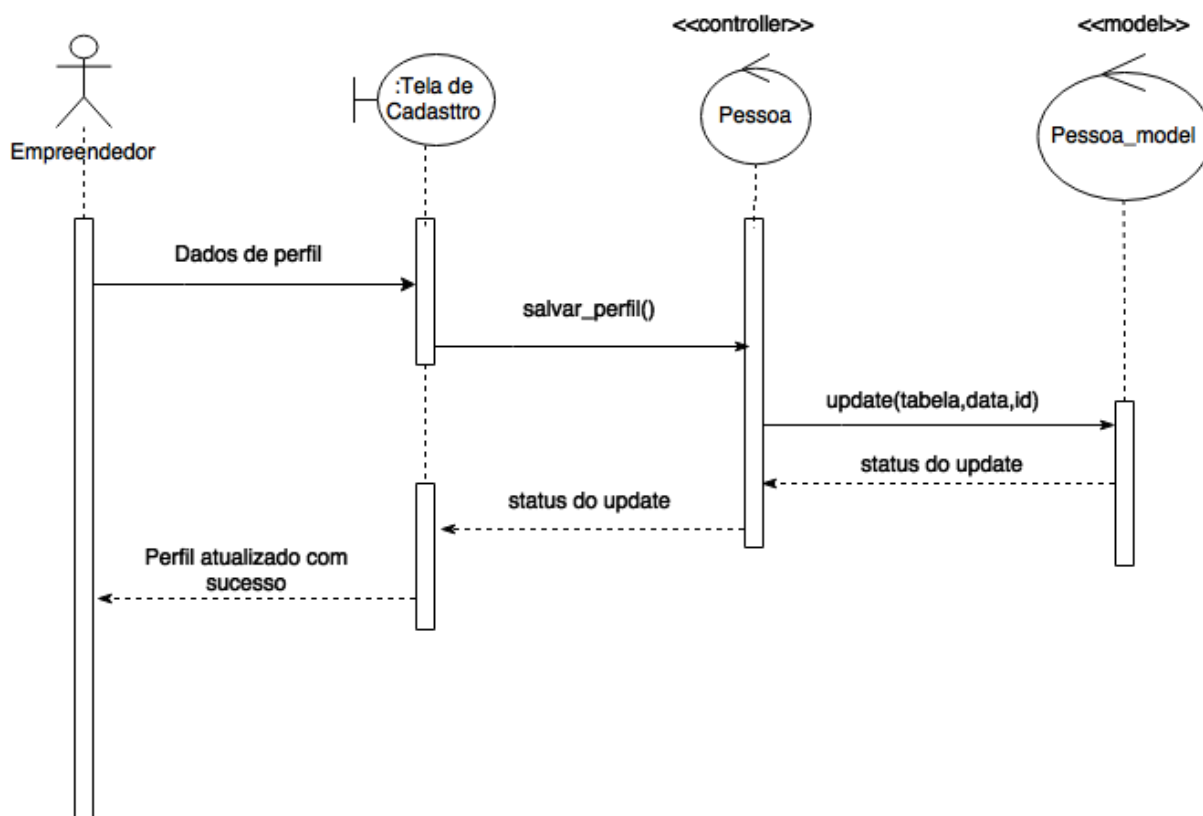


Figura 6.0 Diagrama de Sequência do funcionamento do UC02.

### 2.3.2. UC03: Manter categoria

A Figura 7.0 detalha os caminhos que o ator Empreendedor terá que seguir para manter as categorias das quais participa no sistema. A sequência dos passos inicia com o ator inserindo seus dados no formulário de cadastro de categoria que está disponível na tela de categoria; a submissão desses dados chama o método `cadastrar` que está presente na classe `Categoria` situada no pacote `controller`; esse método usa a função `insert` que está presente na classe `Categoria_model` que, por sua vez, está situada no pacote `model`. A função

retorna uma mensagem de erro ou sucesso ao usuário.

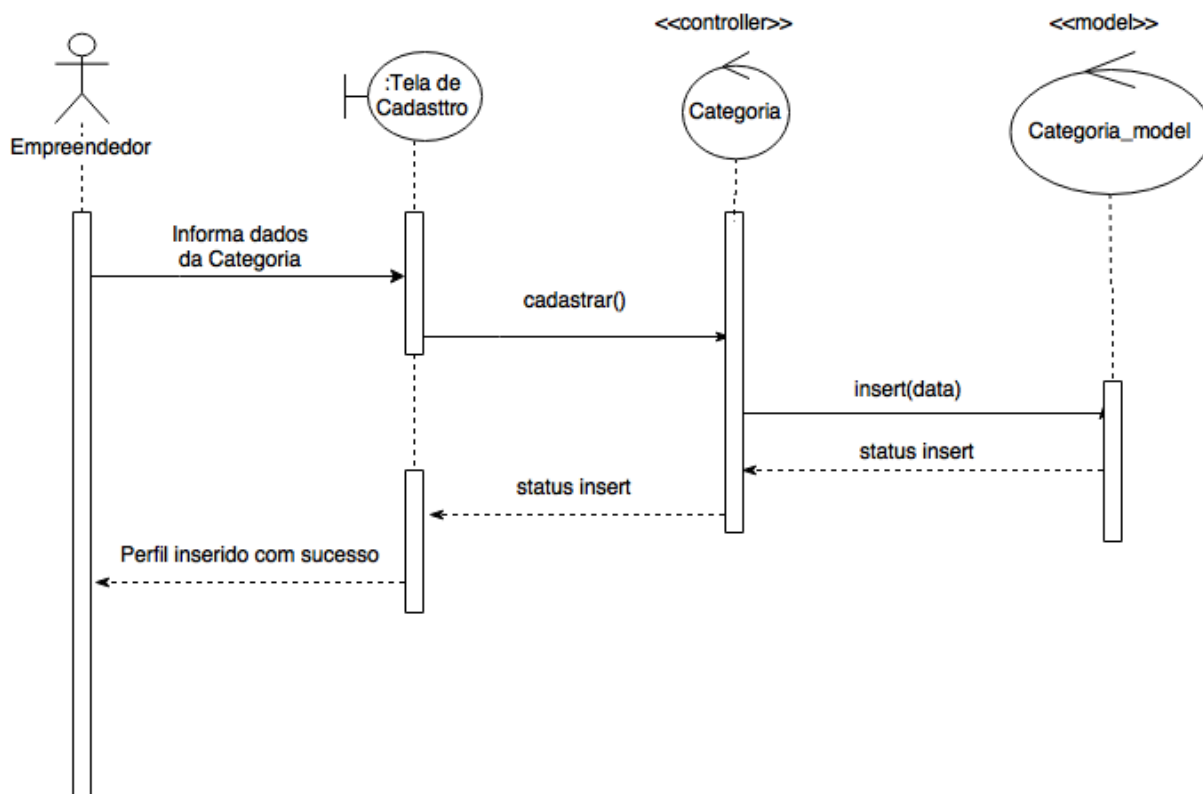


Figura 7.0 Diagrama de Sequência do funcionamento do UC03.

### 2.3.3. UC04: Cadastrar portfólio por categoria

A Figura 8.0 detalha os caminhos que o ator Empreendedor terá que seguir para cadastrar seu portfólio para uma determinada categoria. A sequência dos passos inicia com o ator podendo enviar até 5 imagens na página de cadastro de portfólio; a submissão desses dados chama o método upload que está presente na classe Portfolio situada no pacote controller; esse método usa a função insert está presente na classe Portfolio\_model que, por sua vez, está situada no pacote model. A função retorna uma mensagem de erro ou sucesso ao usuário.

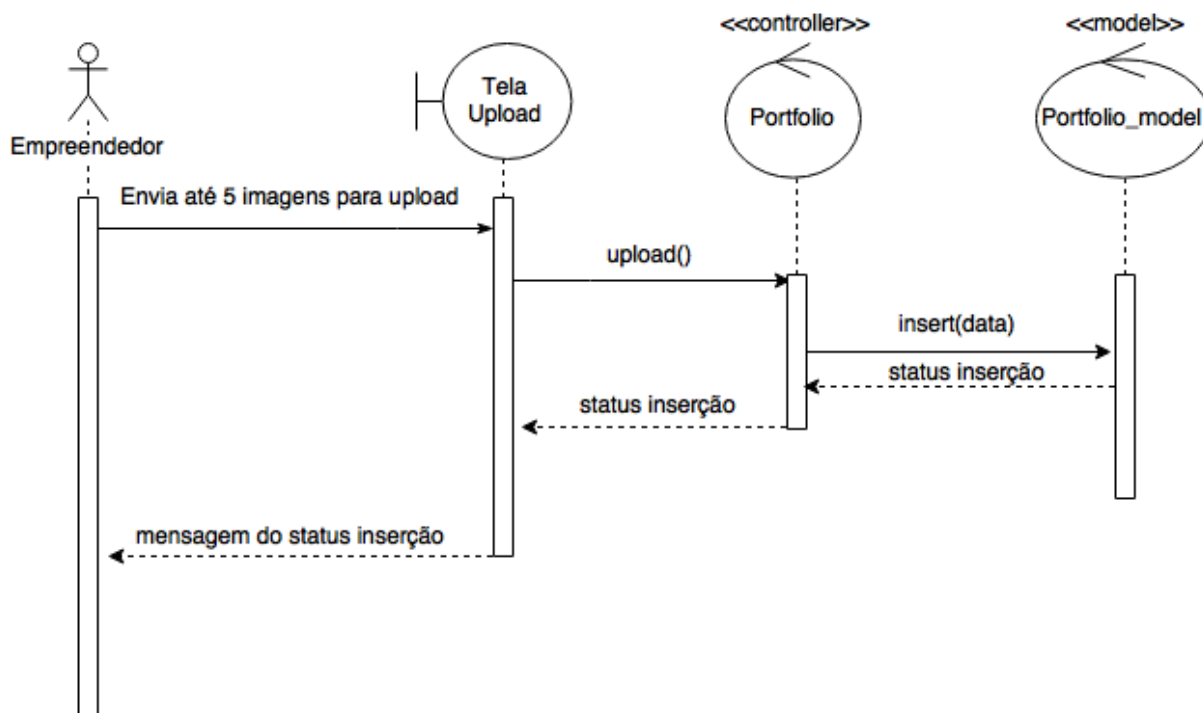


Figura 8.0 Diagrama de Sequência do funcionamento do UC04.

#### 2.3.4. UC05: Manter plano de ranqueamento

A Figura 9.0 detalha os caminhos que o ator Empreendedor terá que seguir para manter seu plano de ranqueamento atualizado no sistema. A seqüência dos passos inicia com o ator escolhendo qual plano de ranqueamento deseja aderir, esta opção está disponível na tela de planos; a submissão desses dados chama o método salvar que está presente na classe Plano situada no pacote controller; esse método usa a função insert está presente na classe Plano\_model que, por sua vez, está situada no pacote model. A função retorna uma mensagem de erro ou sucesso ao usuário.

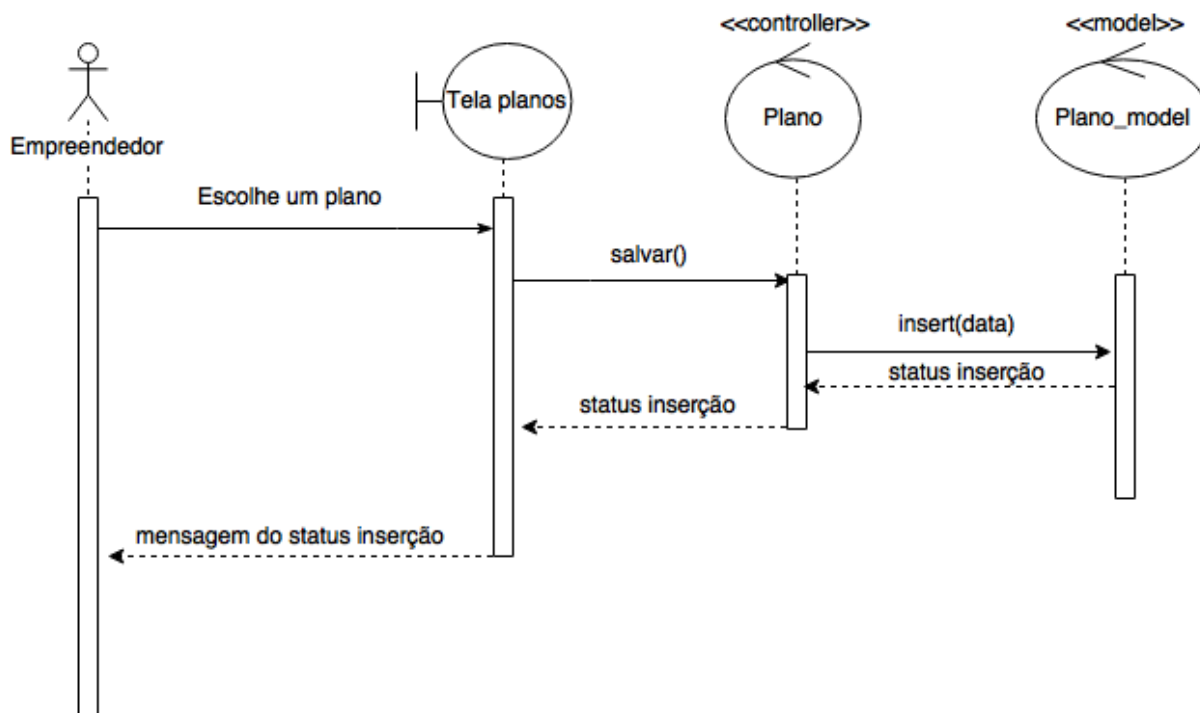


Figura 9.0 Diagrama de Sequência do funcionamento do UC05.

### 2.3.5. UC06: Área de buscas

A Figura 10.0 detalha os caminhos que o ator Cliente terá que seguir para realizar buscas por prestadores de serviços no sistema. A sequência dos passos inicia com o ator inserindo as palavras chaves no campo de buscas que está disponível na tela de buscas ou tela inicial da aplicação; a submissão desses dados chama o método **pesquisa** que está presente na classe **Home** situada no pacote *controller*; esse método usa as funções **categoria**, **estado**, **cidade e sobre** está presente na classe **Pesquisa\_model** que, por sua vez, está situada no pacote *model*. A função retorna uma lista com os dados encontrados.

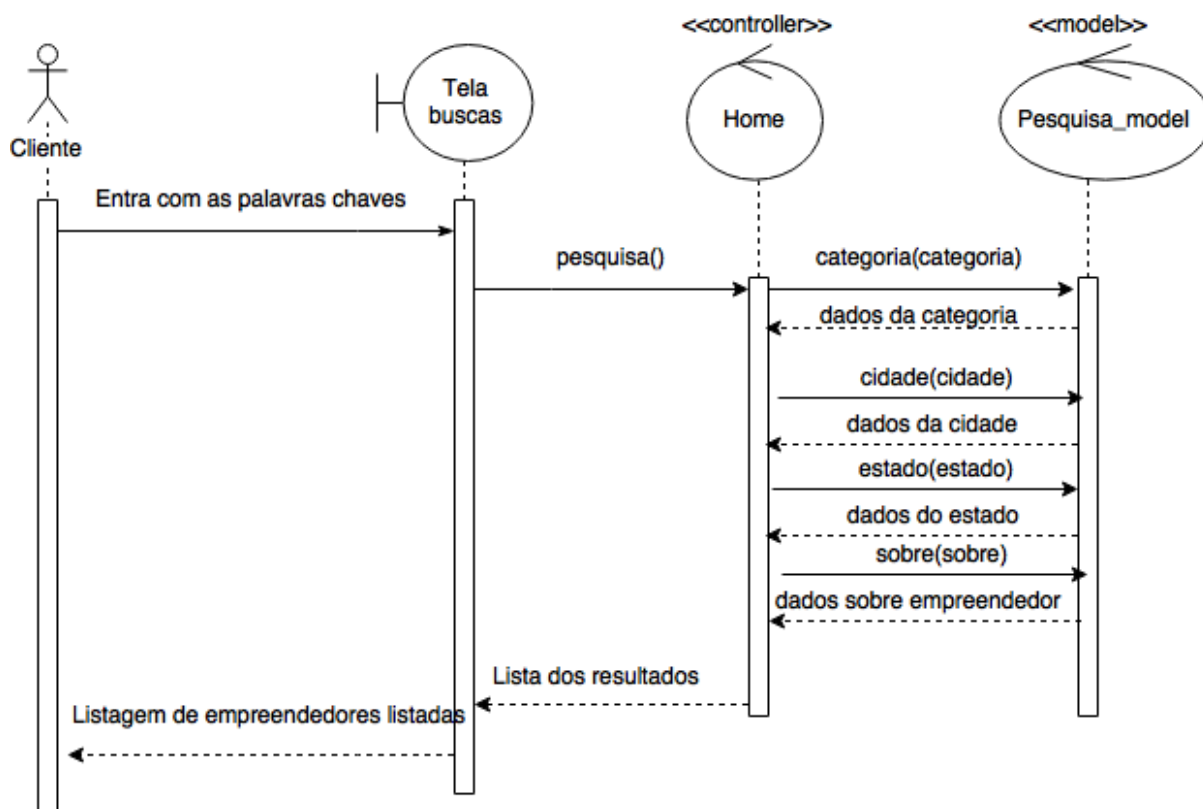


Figura 10.0 Diagrama de Sequência do funcionamento do UC06.

### 2.3.6. UC07: Enviar email para profissional

A Figura 11.0 detalha os caminhos que o ator Cliente terá que seguir para mandar um email para o prestador de serviço do seu interesse. A sequência dos passos inicia com o ator inserindo seus dados no formulário de e-mail que está disponível na tela de perfil do prestador de serviços; a submissão desses dados chama o método **contato** que está presente na classe **Pagina** situada no pacote *controller*. A função retorna uma mensagem de erro ou sucesso no envio do e-mail para usuário.

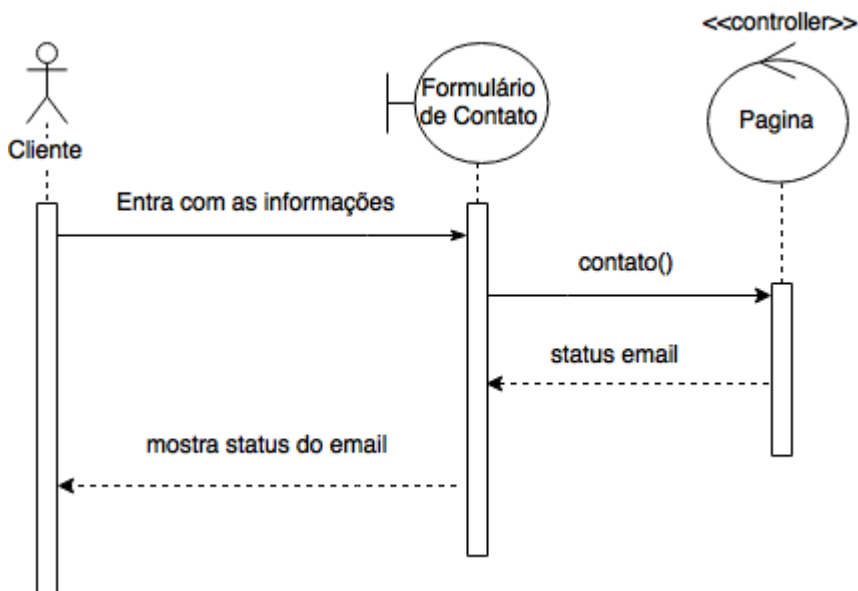


Figura 11.0 Diagrama de Sequência do funcionamento do UC07.

## 2.4. Dicionário de Dados

**CATEGORIA\_MODEL = {categoria\_model}**

**Categoria\_model** = \*acessa e registra os dados de categorias no banco de dados\*.

@cd\_categoria + nome\_categoria

**PESQUISA\_MODEL = {pesquisa\_model}**

**Pesquisa\_model** = \*acessa os dados relacionados a categorias, cidades, pessoas e portfólio no banco de dados\*.

**PESSOA\_MODEL = {pessoa\_model}**

**Pessoa\_model** = \*acessa e registra os dados dos usuários no banco de dados\*.

@cd\_usuario + nome\_usuario + senha\_usuario + email\_usuario + telefone\_usuario + endereco\_usuario + bairro\_usuario + cidade\_usuario + uf\_usuario + data\_cadastro

**PESSOA\_CATEGORIA\_MODEL = {pessoa\_categoria\_model}**

**Pessoa\_categoria\_model** = \*acessa e registra os dados das pessoas

vinculados a categorias no banco de dados\*.

@cd\_pessoa\_categoria + categoria + pessoa + informacao\_adicional +  
faixa\_preco

**PORTFOLIO\_MODEL = {portfolio\_model}**

**Portfolio\_model = \*acessa e registra os dados do portfólio no banco de dados\***

@cd\_portfolio + url\_imagem + titulo\_imagem + descricao\_imagem +  
data\_cadastro

**CATEGORIA\_MODEL = {categoria\_model}**

**Categoria\_model = \*acessa e registra os dados de categorias no banco de dados\*.**

@cd\_categoria + nome\_categoria

## **APÊNDICE C – DOCUMENTO DE ESPECIFICAÇÃO DE PROJETOS**

### **1. Introdução**

Este documento contém a Especificação de Projeto para o Sistema Booszinga. Nele será especificada a arquitetura do sistema, por meio de diagramas de pacotes e dos respectivos diagramas de classes para os componentes identificados. Para dar mais riqueza de detalhes ao projeto, contaremos também com os diagramas de componentes e de implantação.

É importante definir nesta fase a plataforma de implementação e o mecanismo de persistência de dados utilizados, para que o restante do projeto seja construído de forma realista, visando a implementação final. Neste caso, o sistema será desenvolvido utilizando a linguagem de programação PHP e banco de dados MySQL.

### **2. Plataforma computacional**

Por se tratar de uma aplicação web o dispositivo de acesso deve ter as configurações mínimas para conseguir executar um navegador web e ter acesso à internet.

### **3. Arquitetura do sistema**

A arquitetura do sistema já está definida no documento de Especificação de Análise, nos diagramas de classes, incluindo além das classes de entidade, as classes de fronteira e classes de controle.

### **4. Projeto de interação humano-computador**

Nessa parte são apresentadas as telas e os formulários que serão usados no sistema.



Figura 1. Página inicial.

Tela Página Inicial				
SEQ	Nome do Campo	Descrição	Formato	Tamanho
1	Busca	Campo para informar as palavras chaves da busca	Alfanumérico	200
2	Pesquisar	Botão para pesquisar	N/A	N/A
3	Login	Botão para direcionar para tela de Login	N/A	N/A
4	Criar Conta	Botão para direcionar para a tela de Cadastro de Usuário	N/A	N/A

Figura 2. Tela de login do sistema.

Tela de Login do Sistema				
SEQ	Nome do Campo	Descrição	Formato	Tamanho
1	Email	Campo para informar o Email do usuário do sistema	Alfanumérico	50
2	Senha	Campo para informar a Senha do usuário do sistema.	Alfanumérico/ Password	50
3	Entrar	Botão para submeter o formulário de Login.	N/A	N/A
4	Esqueceu Sua Senha	Link para abrir o popup de recuperação de senha.	N/A	N/A
5	É novo aqui? Cadastre - se	Link que direciona para página de cadastro de novo usuário.	N/A	N/A

**Nova conta**

Nome Completo

Email

Senha

Selecione uma categoria

Cadastrar

Figura 3. Tela de cadastro de nova conta do sistema.

<b>Tela de Cadastro de Novo Usuário</b>				
<b>SEQ</b>	<b>Nome do Campo</b>	<b>Descrição</b>	<b>Formato</b>	<b>Tamanho</b>
1	Nome Completo	Campo para informar o nome completo.	Caracteres	50
2	Email	Campo para informar o email do usuário.	Alfanumérico	50
3	Senha	Campo para informar a senha do usuário.	Alfanumérico/ Password	50
4	Categoria	Campo para selecionar a categoria do usuário).	N/A	N/A
5	Cadastrar	Botão para submeter o formulário de cadastro.	N/A	N/A

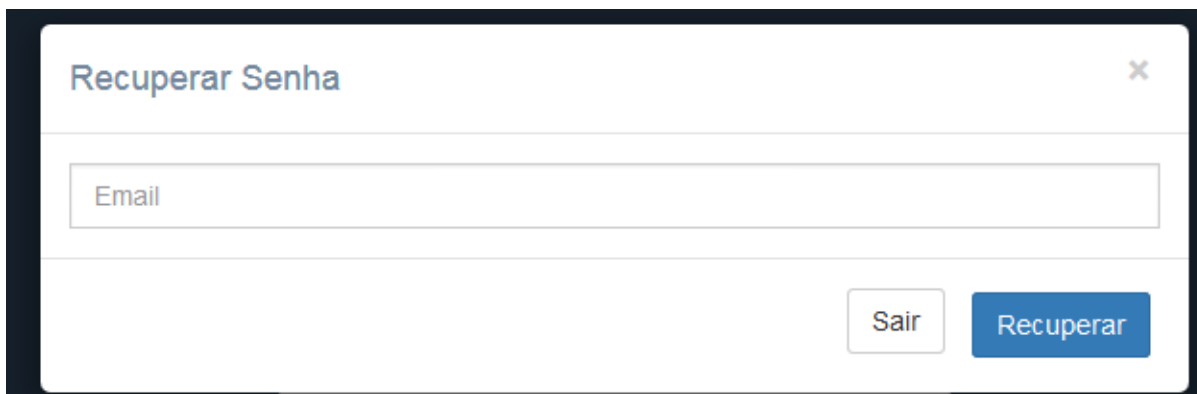


Figura 4. Popup Recuperar Senha.

Tela de Recuperar Senha				
SEQ	Nome do Campo	Descrição	Formato	Tamanho
1	Email	Campo para informar o email para recuperação de senha.	Alfanumérico	100
2	Sair	Botão para sair do popup.	N/A	N/A
3	Recuperar	Botão para submeter o formulário de recuperar senha.	N/A	N/A

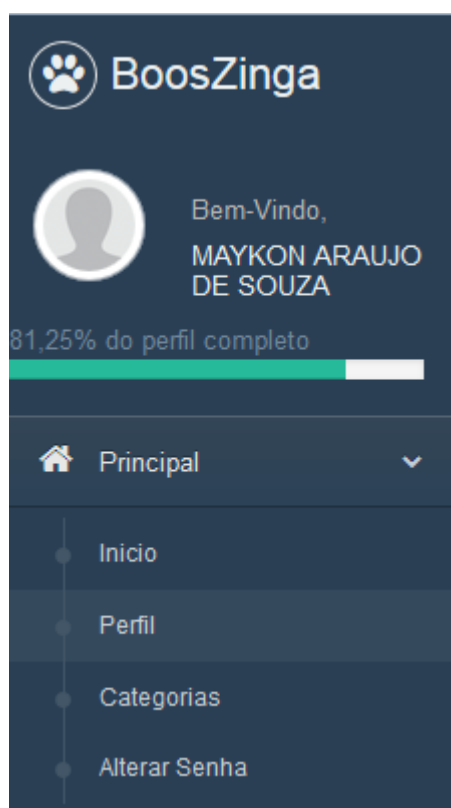


Figura 5. Menu lateral área administrativa.

Menu lateral área administrativa				
SEQ	Nome do Campo	Descrição	Formato	Tamanho
1	Início	Link para a página inicial da área administrativa.	N/A	N/A
2	Perfil	Link para a tela de perfil do usuário.	N/A	N/A
3	Categorias	Link para a tela de cadastro de categoria.	N/A	N/A
4	Alterar Senha	Link para a tela de alterar senha.	N/A	N/A

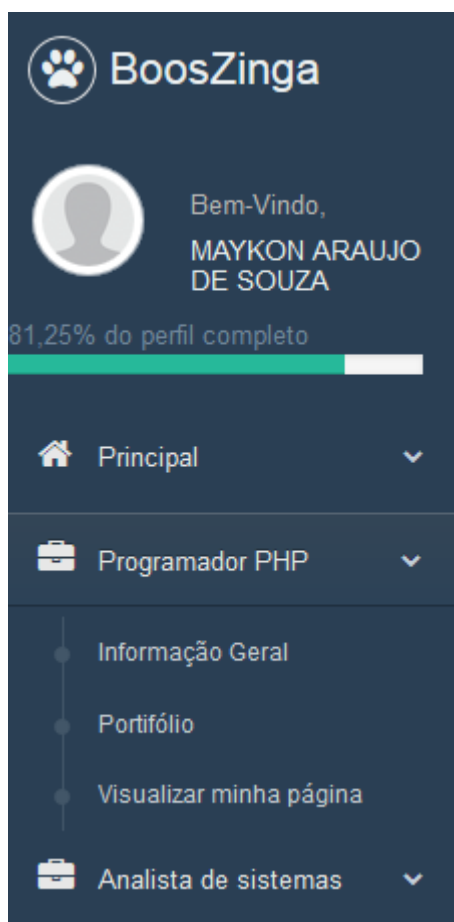


Figura 6. Menu lateral área administrativa com detalhes das categorias..

Menu lateral área administrativa com detalhes das categorias				
SEQ	Nome do Campo	Descrição	Formato	Tamanho

1	Informação Geral	Link para a página de cadastro de informação geral da categoria.	N/A	N/A
2	Portfólio	Link para a tela de cadastro de Portfólio.	N/A	N/A
3	Visualizar minha página	Link para o exemplo de página do usuário.	N/A	N/A

### Perfil do Usuário

Dados do Usuário



**MAYKON ARAUJO DE SOUZA**

📍 Belém ,Pará ,BR  
 💻 Programador PHP  
 🌐 <http://codacta.com>

Nome\*

Gênero \*  Masculino  Feminino

Endereço

Bairro

Complemento

Estado \*

Cidade \*

Celular

WhatsApp

Email \*

Login

Figura 7. Tela de cadastro de complementação do perfil do usuário.

Tela de Cadastro de complementação do perfil do usuário				
SEQ	Nome do Campo	Descrição	Formato	Tamanho
1	Nome	Campo para informar o nome completo.	Caracteres	50
2	Gênero	Campo para escolha do gênero do usuário.	N/A	N/A
3	Endereço	Campo para informar o endereço do usuário.	Alfanumérico	100
4	Bairro	Campo para informar a o bairro do usuário.	Alfanumérico	50

5	Complemento	Campo para informar o complemento do endereço do usuário.	Alfanumérico	100
6	Estado	Campo para escolher o estado do usuário.	N/A	N/A
7	Cidade	Campo para escolher a cidade do usuário baseado no estado escolhido.	N/A	N/A
8	Celular	Campo para informar o celular do usuário.	Alfanumérico	15
9	WhatsApp	Campo para informar o número do <i>WhatsApp</i> do usuário.	Alfanumérico	15
10	Email	Campo para informar o email do usuário.	Alfanumérico	50
11	Login	Campo para informar o login do usuário.	Alfanumérico	50

## Categorias Profissionais

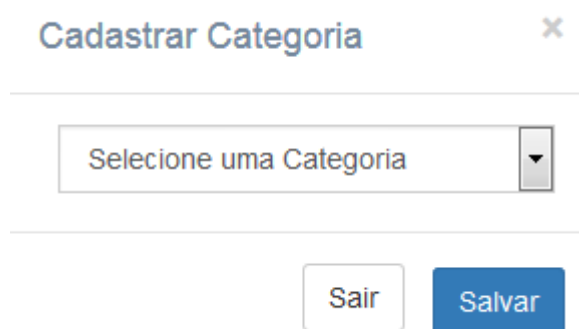
[+ Cadastrar](#)

### Categorias Cadastradas

- Programador PHP ✕
- Analista de sistemas ✕

Figura 8. Tela de visualização de categorias cadastradas.

Tela de visualização de categorias				
SEQ	Nome do Campo	Descrição	Formato	Tamanho
1	Cadastrar	Botão para abrir o popup de cadastro de categoria.	N/A	N/A
2	Excluir	Botão para excluir a categoria.	N/A	N/A



Cadastrar Categoria ×

---

Selecione uma Categoria ▼

---

Figura 9. Tela de cadastro de categorias.

Tela de cadastro de categorias				
SEQ	Nome do Campo	Descrição	Formato	Tamanho
1	Categoria	Selecionar uma categoria.	N/A	N/A
2	Sair	Botão para sair do popup.	N/A	N/A
3	Salvar	Botão para submeter o formulário de categorias.	N/A	N/A

## Alterar Senha

Nova Senha

Senha:   **Mostrar Senha**

Repetir Senha:

Figura 10. Tela de alterar senha.

Tela Alterar Senha				
SEQ	Nome do Campo	Descrição	Formato	Tamanho
1	Senha	Informar a nova senha.	Alfanumérico/ Password	50
2	Repetir Senha	Informar a nova senha novamente.	Alfanumérico/ Password	50
3	Salvar	Botão para submeter o formulário de recuperar senha.	N/A	N/A
4	Limpar	Botão para limpar o formulário de recuperar senha.	N/A	N/A
5	Mostrar senha	CheckBox para mostrar ou não a senha.	N/A	N/A

## Informações Gerais | Programador PHP

## Dados Cadastrados

Email Profissional:

Site Profissional:

Facebook:

Twitter:

Linkedin:

Tempo de Experiencia: \*

Fale sobre seu trabalho: \*

Faixa de Preço:

Figura 11. Tela de cadastro de informação por categoria.

Tela de Cadastro de informações gerais do da categoria do usuário				
SEQ	Nome do Campo	Descrição	Formato	Tamanho
1	Email Profissional	Campo para informar o email profissional.	Caracteres	50
2	Site Profissional	Campo para informar o link do site profissional.	Alfanumérico	100
3	Facebook	Campo para informar o link do facebook do usuário.	Alfanumérico	100
4	Twitter	Campo para informar o link do twitter do usuário.	Alfanumérico	100
5	Linkedin	Campo para informar o link do linkedin do usuário.	Alfanumérico	100
6	Tempo de Experiência	Campo para informar o tempo de experiência do usuário em determinada	Numérico	2

		categoria.		
7	Fale sobre seu trabalho	Campo para informar detalhes sobre o trabalho.	Alfanumérico	256
8	Faixa de preço	Campo para informar a faixa de preço que o usuário trabalha.	Alfanumérico	50



Figura 12. Tela de visualização de portfólio.

Tela de visualização de portfólio				
SEQ	Nome do Campo	Descrição	Formato	Tamanho
1	Cadastrar	Botão para abrir o popup de cadastro de portfólio.	N/A	N/A
2	Imagem destaque	Botão para tornar a imagem como capa do portfólio ou primeira imagem.	N/A	N/A
3	Editar	Botão para editar informações da imagem como título e descrição.	N/A	N/A
	Excluir	Botão para excluir a imagem do portfólio.	N/A	N/A

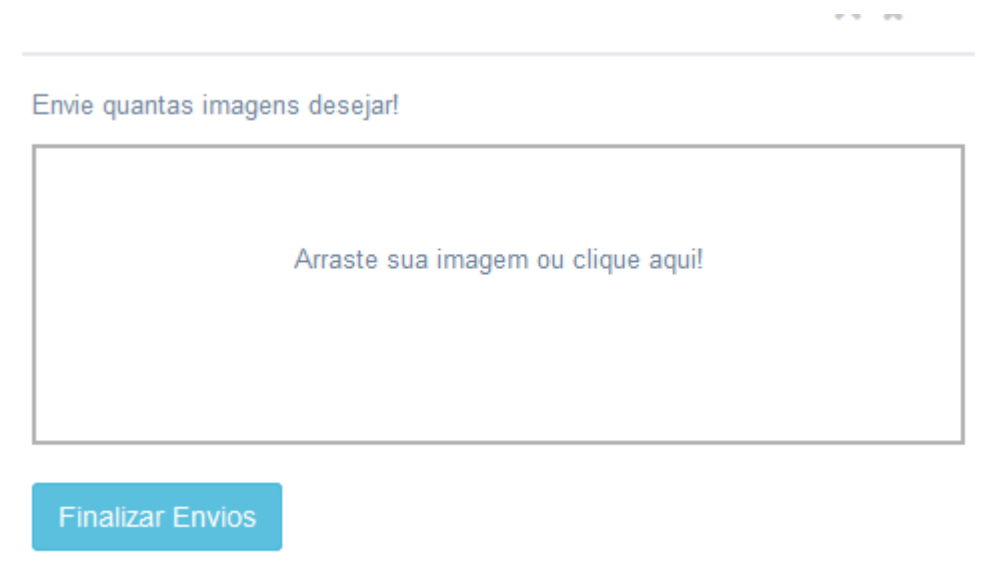


Figura 13. Tela de upload de imagens do portfólio.

Tela de upload de imagens do portfólio				
SEQ	Nome do Campo	Descrição	Formato	Tamanho
1	Imagem	Campo para arrastar ou selecionar a imagem ou imagens que deseja no portfólio.	N/A	N/A
2	Finalizar envios	Botão para fazer upload das imagens.	N/A	N/A

## 5. Projeto de banco de dados

A **Figura 14** apresenta o diagrama relacional do sistema *Booszinga*, usando uma extensão da UML para representar diagramas deste tipo.

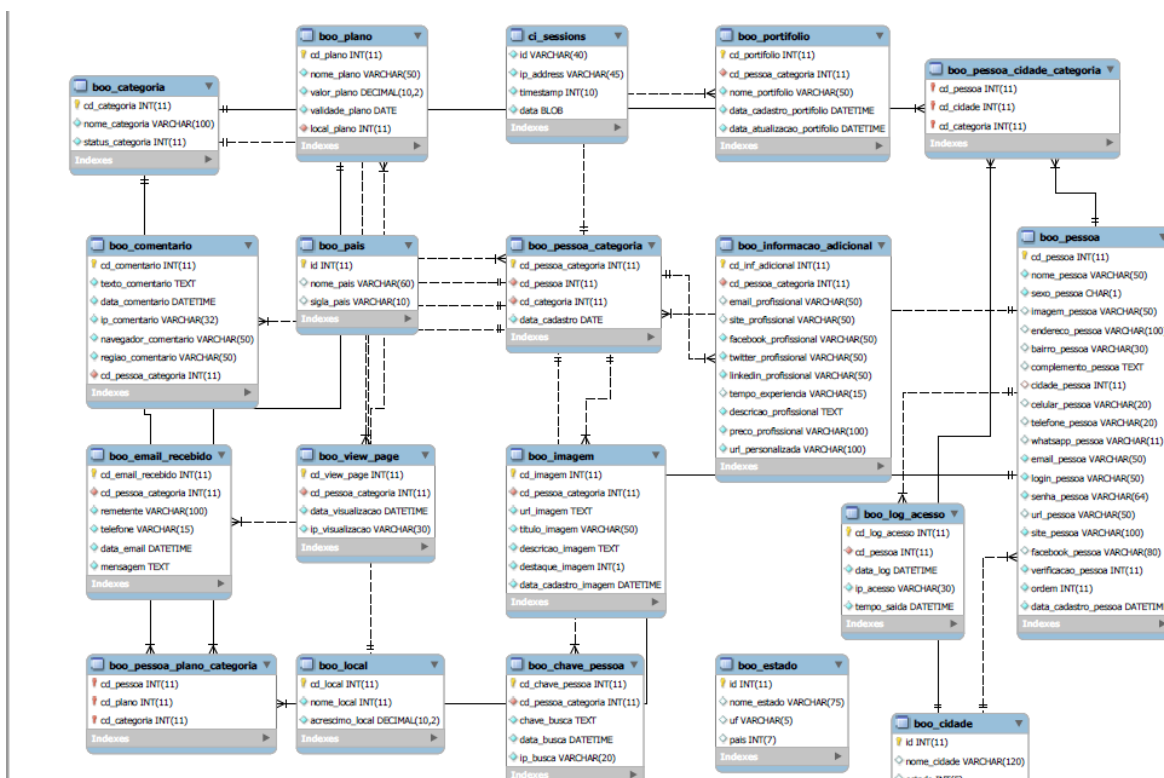


Figura 14. Diagrama de entidade relacional do Booszinga.

boo\_categoria(cd\_categoria, nome\_categoria, status\_categoria);

boo\_pessoa(cd\_pessoa, nome\_pessoa, sexo\_pessoa, imagem\_pessoa, endereco\_pessoa, bairro\_pessoa, complemento\_pessoa, cidade\_pessoa, celular\_pessoa, telefone\_pessoa, whatsapp\_pessoa, email\_pessoa, login\_pessoa, senha\_pessoa, url\_pessoa, site\_pessoa, facebook\_pessoa, verificacao\_pessoa, ordem, data\_cadastro\_pessoa)

cd\_pessoa chave primária

cidade\_pessoa chave estrangeira referenciando a chave primária cd\_cidade da tabela boo\_cidade;

boo\_pais(id, nome\_pais, sigla\_pais)

id chave primária;

boo\_estado(id, nome\_estado, uf, pais)

id chave primária

pais chave estrangeira referenciando a chave primária id da tabela boo\_pais;

boo\_cidade(id, nome\_cidade, estado)

id chave primária

estado chave estrangeira referenciando a chave primária id da tabela boo\_estado;

boo\_log\_acesso(cd\_log\_acesso, cd\_pessoa, data\_log, ip\_acesso, tempo\_saida)

cd\_log\_acesso chave primária

cd\_pessoa chave estrangeira referenciando a chave primária

cd\_pessoa da tabela boo\_pessoa;  
     boo\_informacao\_adicional(cd\_inf\_adicional, cd\_pessoa\_categoria,  
 email\_profissional, site\_profissional, facebook\_profissional, twitter\_profissional,  
 linkedin\_profissional, tempo\_experiencia, descricao\_profissional,  
 preco\_profissional, url\_personalizada)  
     cd\_inf\_adicional chave primária  
     cd\_pessoa\_categoria chave estrangeira referenciando a chave primária  
 cd\_pessoa\_categoria da tabela boo\_pessoa\_categoria;  
     boo\_portfolio(cd\_portfolio, cd\_pessoa\_categoria, nome\_portfolio,  
 data\_cadastro\_portfolio, data\_atualizacao\_portfolio)  
     cd\_portfolio chave primária  
     cd\_pessoa\_categoria chave estrangeira referenciando a chave primária  
 cd\_pessoa\_categoria da tabela boo\_pessoa\_categoria;  
     boo\_pessoa\_categoria(cd\_pessoa\_categoria, cd\_pessoa, cd\_categoria,  
 data\_cadastro)  
     cd\_pessoa\_categoria chave primária  
     cd\_categoria chave estrangeira referenciando a chave primária  
 cd\_categoria da tabela boo\_categoria  
     cd\_pessoa chave estrangeira referenciando a chave primária cd\_pessoa  
 da tabela boo\_pessoa;  
     boo\_imagem(cd\_imagem, cd\_pessoa\_categoria, url\_imagem,  
 titulo\_imagem, descricao\_imagem, destaque\_imagem, data\_cadastro\_imagem)  
     cd\_imagem chave primária  
     cd\_pessoa\_categoria chave estrangeira referenciando a chave primária  
 cd\_pessoa\_categoria da tabela boo\_pessoa\_categoria;  
     boo\_email\_recebido(cd\_email\_recebido, cd\_pessoa\_categoria,  
 remetente, telefone, data\_email, mensagem)  
     cd\_email\_recebido chave primária  
     cd\_pessoa\_categoria chave estrangeira referenciando a chave primária  
 cd\_pessoa\_categoria da tabela boo\_pessoa\_categoria;  
     boo\_comentario(cd\_comentario, texto\_comentario, data\_comentario,  
 ip\_comentario, navegador\_comentario, navegador\_comentario,  
 regioao\_comentario, cd\_pessoa\_categoria)  
     cd\_comentario chave primária  
     cd\_pessoa\_categoria chave estrangeira referenciando a chave primária  
 cd\_pessoa\_categoria da tabela boo\_pessoa\_categoria;  
     boo\_pessoa\_plano\_categoria(cd\_pessoa, cd\_plano, cd\_categoria)  
     cd\_pessoa chave primária e chave estrangeira referenciando a chave  
 primária cd\_pessoa da tabela boo\_pessoa  
     cd\_plano chave primária e chave estrangeira referenciando a chave  
 primária cd\_plano da tabela boo\_plano  
     cd\_categoria chave primária e chave estrangeira referenciando a chave  
 primária cd\_categoria da tabela boo\_categoria;  
     boo\_local(cd\_local, nome\_local, crescimento\_local)

cd\_local chave primária  
 boo\_chave\_pessoa(cd\_chave\_pessoa, cd\_pessoa\_categoria,  
 chave\_busca, data\_busca, ip\_busca\_  
 cd\_chave\_pessoa chave primária  
 cd\_pessoa\_categoria chave estrangeira referenciando a chave primária  
 cd\_pessoa\_categoria da tabela boo\_pessoa\_categoria;  
 boo\_view\_page(cd\_view\_page, cd\_pessoa\_categoria, data\_visualizacao,  
 ip\_visualizacao)  
 cd\_view\_page chave primária  
 cd\_pessoa\_categoria chave estrangeira referenciando a chave primária  
 cd\_pessoa\_categoria da tabela boo\_pessoa\_categoria;  
 boo\_plano(cd\_plano, nome\_plano, valor\_plano, validade\_plano,  
 local\_plano)  
 cd\_plano chave primária  
 ci\_session(id,ip\_address, timestamp, data)

## 6. Projeto de componentes

### ● Especificação do Componente HOME

1) FUNÇÃO: Executar a pesquisa dos prestadores de serviços baseados nas palavras-chaves informadas e retornar uma lista de resultados mais próxima possível do esperado pelo usuário.

#### 2) PSEUDOCÓDIGO DO PROGRAMA HOME

1. Início.
  2. Fazer inicializações
  3. Receber palavras-chave
- SE palavras-chaves não for nula ENTÃO
- executar pesquisar
- SENÃO
- exibir “Informe ao menos uma palavra-chave”
- FIM SE
4. Finalizações
  5. Sair

pesquisar

Início

inicializa variáveis

ENQUANTO houver palavras chaves, faça:

categoria ← pesquisar-categoria  
 cidade ← pesquisar-cidade  
 estado ← pesquisar-estado

```
sobre ← pesquisar-sobre
pessoa ← pesquisar-pessoa
```

FIM ENQUANTO

```
melhor_caso ← interseccao(categoria,
cidade,estado,sobre,pessoa)
caso_medio ← i nterseccao(categoria, cidade, estado)
caso_normal ← interseccao(categoria, cidade)
pior_caso ← interseccao(categoria, pessoa)

resultado ← merge(melhor_caso, caso_medio, caso_normal,
pior_caso)
```

/\*

Critério para ordenação

A ordenação será trabalhada com pesos

exemplo abaixo:

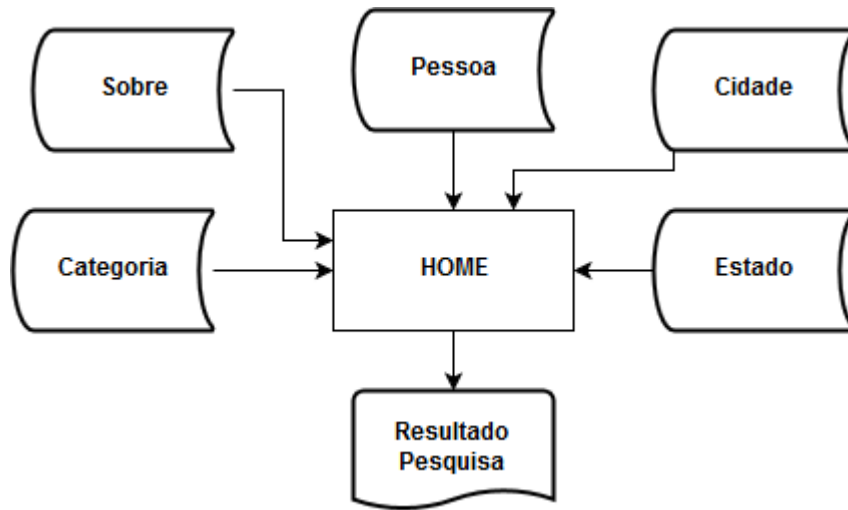
- 1 - Usuário mais antigos - 6
- 2 - Acesso recente no painel administrativo - 5
- 3 - % do perfil - 4
- 4 - Quantidade de visualizações de página -3
- 5 - Quantidade de fotos no portfólio - 2
- 6 - Quantidade de informações adicionais -1

fórmula usada  $OF = (o1*6) + (o2*5) + (o3*4) + (o4+3) + (o5+2) + (o6+1)$

\*\*\*o1 = Ordenação 1, o2 = Ordenação 2 sucessivamente.

\*/

## 6.1. Rotina Gráfica



## APÊNDICE D – DOCUMENTO DE APRESENTAÇÃO DO BOOSZINGA

### 1. Introdução

Este documento tem por objetivo a apresentação de uma ferramenta chamada Booszinga, nome criado usando a combinação da variação da palavra em inglês *Boss* (*chefe*) e o bordão Bazinga usado pelo personagem Sheldon Cooper (Jim Parsons) na série *The Big Bang Theory* ou Teoria do *Big Bang* (título no Brasil). O Booszinga é um aplicativo Web, que tem por objetivo promover a interação entre prestadores de serviços e consumidores, de modo que consumidores tenham seus interesses de consumo atendidos com a informação de quem pode atendê-los e que prestadores de serviços se apresentem ao mercado consumidor interessado nos seus serviços. Portanto, o Booszinga faz a intermediação entre consumidores e prestadores de serviços.

A aplicação está disponível no endereço <http://booszinga.com>.

Serão descritos neste documento as interfaces e as funcionalidades da aplicação.

### 2. Interfaces e funcionalidades

O Booszinga possui dois módulos: Administrativo e Busca. O módulo Administrativo é de uso do prestador de serviço. Este precisa se cadastrar na plataforma para ter acesso às funcionalidades do ambiente administrativo: cadastro de múltiplas categorias, cadastro de perfil, cadastro de portfólio e cadastro de informações profissionais. O módulo de Busca é de uso do consumidor. Neste módulo são realizadas as buscas de prestadores de serviços pelos consumidores. As buscas retornam um link para uma página interna do prestador de serviço selecionado. A seguir vamos entender melhor cada módulo e suas interfaces.

#### 2.1 Módulo Administrativo

O prestador de serviço é autenticado antes de entrar no módulo Administrativo, para isso ele precisa se cadastrar antes na plataforma. Como

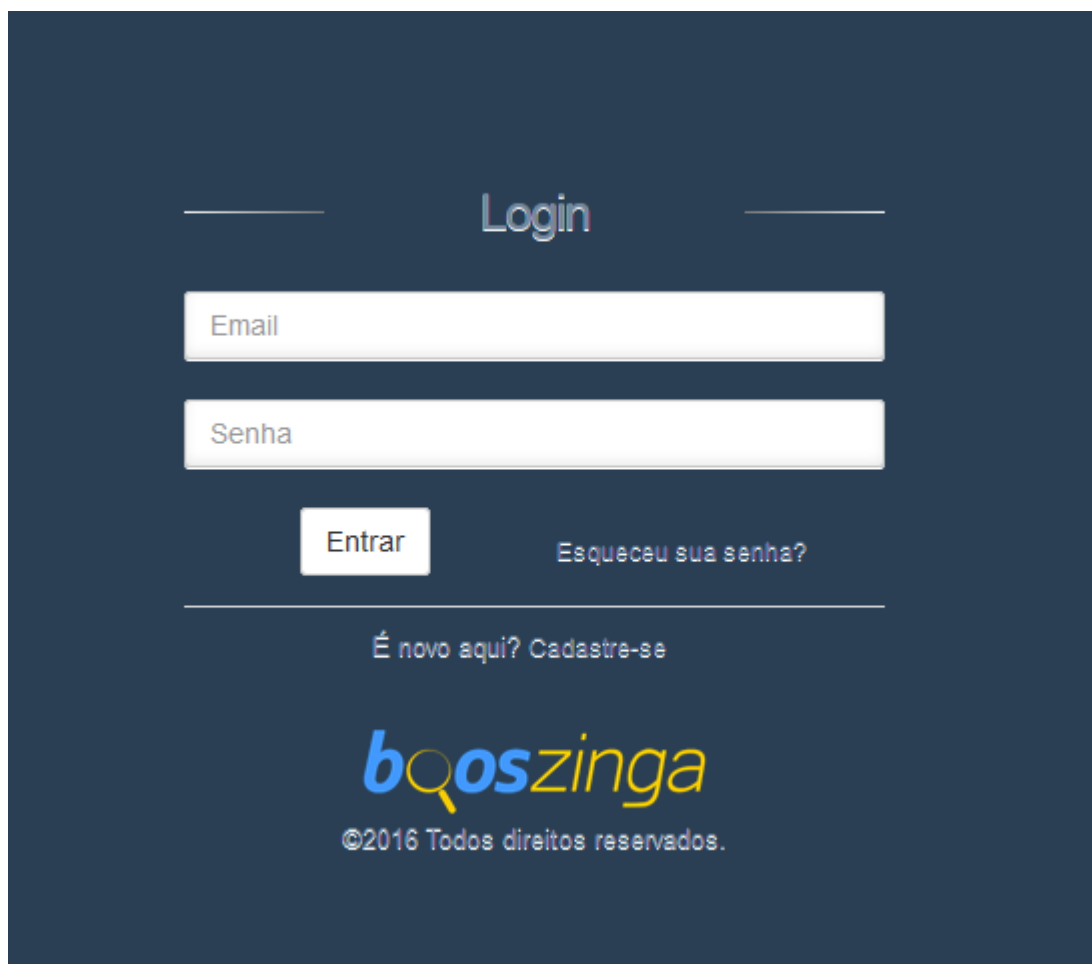
pode ser visto na Figura 1, a tela de cadastro conta com os campos necessários para uma identificação inicial do prestador de serviço: nome completo, email, senha e categoria, podendo esse cadastro ser atualizado posteriormente.



A imagem mostra a interface de usuário para criar uma nova conta no Booszinga. O título principal é "Criar uma nova conta". Abaixo dele, há quatro campos de entrada: "Nome Completo", "Email", "Senha" e "Selecione uma categoria" (um menu suspenso). Um botão "Cadastrar" está posicionado abaixo dos campos. Na base da tela, há o link "Tudo Certo ? Entrar", o logotipo "booszinga" e o texto "©2016 Todos direitos reservados".

**Figura 1. Tela de cadastro de prestador de serviço no Booszinga.**

Quando realiza o cadastro, o prestador de serviço recebe um email de confirmação, para evitar cadastros falsos realizados por robôs ou pessoas mal intencionadas. Após confirmar por email o seu cadastro, o prestador de serviços está apto a se autenticar na ferramenta e ter acesso ao painel administrativo. A Figura 2 mostra a tela de login do Booszinga, o prestador de serviço informa o e-mail e a senha que cadastrou, se suas credenciais estiverem corretas ele entra no sistema, senão, a entrada no sistema é negada, podendo o usuário recuperar a senha a qualquer momento.



**Figura 2. Tela de login do booszinga.**

O prestador de serviço autenticado tem acesso ao painel administrativo. Na Figura 3 é apresentada a tela de início com um relatório visual contendo os seguintes itens informacionais para o prestador de serviço: visualizações do perfil, pessoas que entraram em contato e palavras-chaves usadas para encontrar o perfil.



**Figura 3. Tela de Inicial do Painel Administrativo**

Como visto, o cadastro do prestador de serviço, que permite acesso a ferramenta, é básico e exige poucos campos. Na Figura 4 é apresentado o formulário de cadastro completo, onde o prestador de serviço pode atualizar seus dados: imagem de perfil, gênero, endereço, contatos e redes sociais.



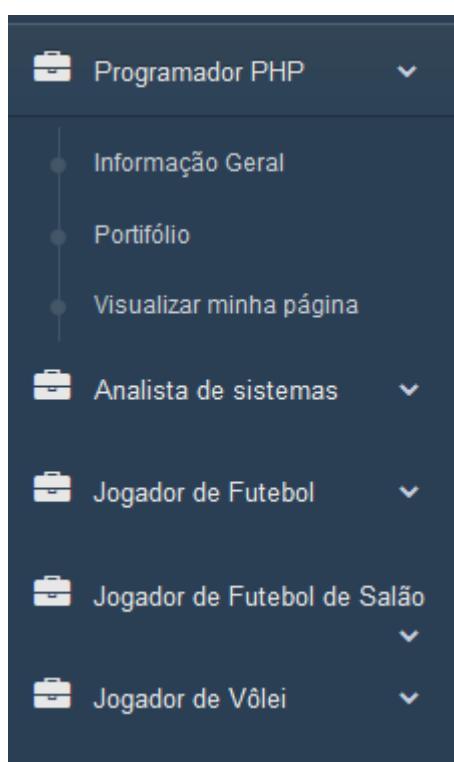
**Figura 4. Tela de atualização de perfil do prestador de serviço.**

O Booszinga permite que o prestador de serviço se cadastre em várias categorias: pintor, programador, professor, etc. Na Figura 5 é apresentada a tela de categorias vinculadas ao prestador de serviço.



**Figura 5. Tela de categorias vinculadas ao prestador de serviço.**

Cada categoria vinculada a um determinado prestador de serviço, corresponde a um usuário da aplicação. ou seja, um prestador de serviço X na categoria Y é um usuário e o prestador de serviço X na categoria Z é tratado como outro usuário do sistema. A Figura 6 apresenta um prestador de serviço vinculado a várias categorias, e em cada categoria o prestador de serviço pode cadastrar: Informação Geral, Portfólio e Visualizar página.



**Figura 6. Prestador de serviço vinculado a várias categorias.**

Na Figura 7 é apresentada a tela de cadastro de informação geral. Esta tela tem por objetivo o cadastro de informações profissionais do prestador de serviço em uma determinada categoria. Aqui pode ser informado contatos e redes sociais do profissional, tempo de experiência, descrição das suas experiências e faixa de preços praticados.

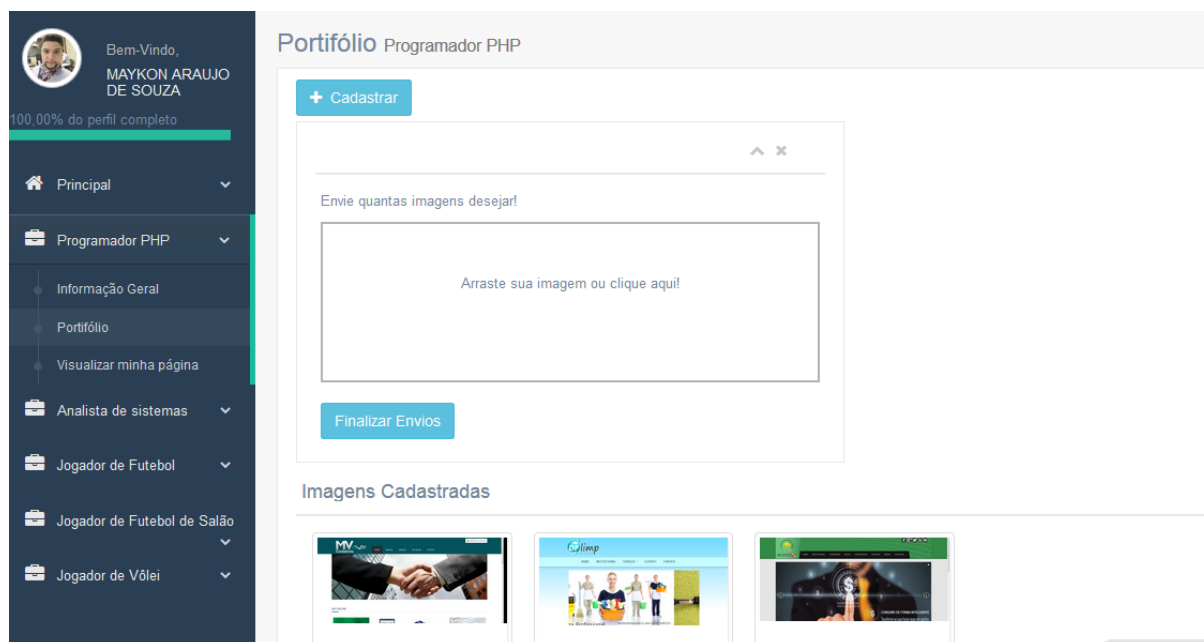


The screenshot displays the BoosZinga user interface. On the left is a dark sidebar with the user's profile information: 'BoosZinga', a profile picture, 'Bem-Vindo, MAYKON ARAUJO DE SOUZA', and a progress bar for '100,00% do perfil completo'. The main content area is titled 'Informações Gerais | Programador PHP'. Below this is a section 'Dados Cadastrados' with an orange warning banner: 'Olá Booszingueiro! Se você não informar nada nessa seção, as informações serão tiradas do seu Perfil!'. The form contains the following fields:

Email Profissional:	<input type="text"/>
Site Profissional:	<input type="text" value="http://codacta.com"/>
Facebook:	<input type="text" value="http://fb.com/maykonaraujo23"/>
Twitter:	<input type="text" value="http://twitter.com/exemplo"/>
LinkedIn:	<input type="text" value="http://linkedin.com/exemplo"/>
Tempo de Experiência: *	<input type="text" value="5 anos"/>
Fale sobre seu trabalho: *	<input type="text" value="Profissional com 5 anos de experiência na área da tecnologia da informação, atuando principalmente na área de Sistemas de Informação com ênfase em Desenvolvimento de Software e Banco de"/>

**Figura 7. Informação Geral**

O prestador de serviço pode cadastrar um portfólio para cada categoria a ele vinculado. A Figura 8 apresenta a tela de cadastro de portfólio, em que o usuário pode fazer uploads de imagens para formar o portfólio, adicionar título e descrição para essa imagem.



**Figura 8. Tela de cadastro de portfólio.**

Cada categoria vinculada ao prestador de serviço gera uma página web (Figura 9), ou seja, o prestador de serviço cadastrado como Programador PHP tem uma página web contendo: nome, foto do perfil, categoria, portfólio, informações gerais, formulário de contato, contatos e redes sociais.

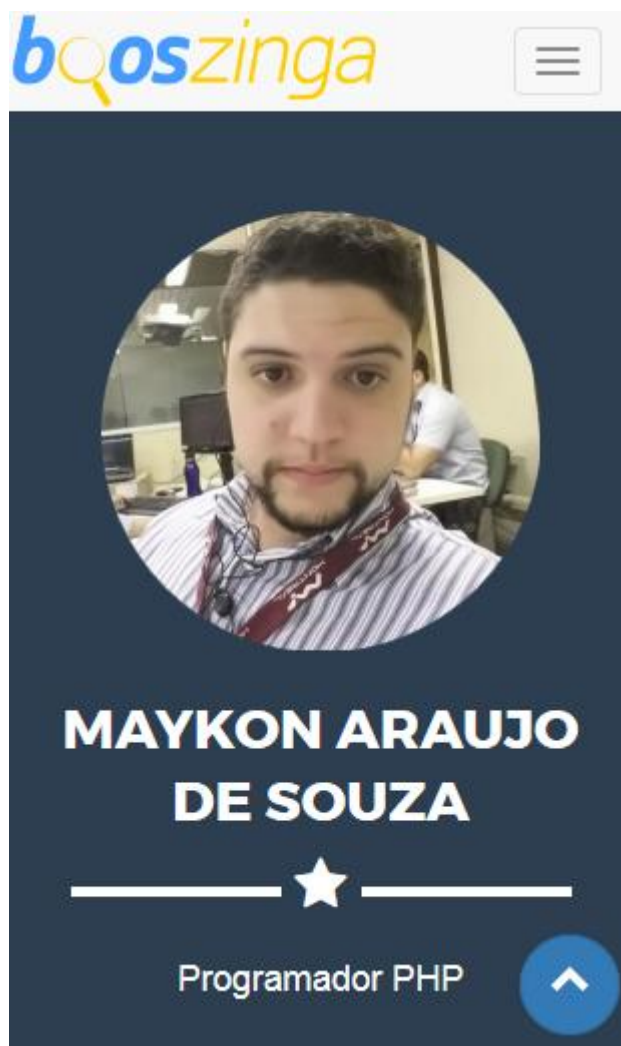


Figura 9. Página do prestador de serviço.

A página apresentada na Figura 9, ficará disponível para os consumidores. Este será o meio de interação entre prestador de serviço e consumidor, onde o consumidor poderá ter mais informações sobre o prestador de serviço e poderá entrar em contato pelos canais de comunicação por ele disponibilizado.

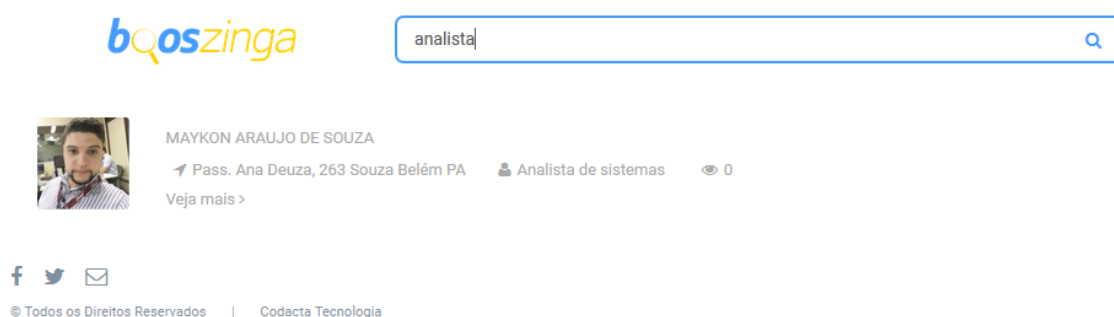
## 2.2 Módulo de Buscas

Este módulo é o responsável pela buscas dos consumidores na plataforma. Fazem parte deste módulo a página inicial da aplicação e a página do prestador de serviço demonstrada na Figura 9.



**Figura 10. Página inicial e de buscas do Booszinga.**

Na Figura 10 temos uma tela simples e objetiva, o usuário tem um campo de buscas, onde pode informar as palavras chaves que o ajudarão encontrar o prestador de serviço que melhor atenda sua demanda. Esta busca pode ser vista na Figura 11. Quando o consumidor seleciona um prestador de serviço, este é redirecionado para a página demonstrada na Figura 9.



**Figura 11. Resultado de pesquisa por palavra chave no Booszinga.**

Foram apresentadas neste documento todas as telas e descrições básicas de cada uma. O Booszinga através dos seus módulos e funcionalidades consegue cumprir o objetivo de promover a interação entre prestadores de serviços e consumidores.

Há interesse de aprimorar a ferramenta. Por isso, a equipe Booszinga solicita que sugestões e/ou críticas sejam enviadas para [contato@booszinga.com](mailto:contato@booszinga.com). Da mesma forma, encontra-se disponível formulário para avaliação da ferramenta, no endereço <http://booszinga.com/pagina/avaliacao>, sempre que o usuário desejar fazê-lo.

## APÊNDICE E – QUESTIONÁRIO DE VALIDAÇÃO DO BOOSZINGA

Este formulário tem por objetivo a avaliação da aplicação online, Booszinga, que está disponível no endereço <http://www.booszinga.com>. A pesquisa leva em consideração os itens de: Usabilidade, Funcionalidade, Confiabilidade e Eficiência.

\*Obrigatório

### 1. Endereço de e-mail \*

---

### 2. Usabilidade: Ao acessar o Booszinga, é possível perceber clara e rapidamente o objetivo da ferramenta? \*

*Marcar apenas uma oval.*

- Sim, completamente
- Sim, mas algumas coisas não ficaram claras
- Não

### 3. Usabilidade: O Booszinga possui estética atraente e intuitiva? \*

*Marcar apenas uma oval.*

- Sim, é esteticamente atraente e intuitivo, ou seja, é possível entender a funcionalidade de uma determinada interface somente pelo designer.
- Sim é esteticamente atraente, mas não é intuitivo, ou seja, às vezes não é possível entender o que determinada interface possibilita fazer.
- Não é atraente, mas é intuitivo.
- Não é atraente e nem intuitivo.

**4. Funcionalidade: O Booszinga cumpre o que promete fazer, ou seja, fornece interação entre Prestadores de Serviços e Consumidores: \***

*Marcar apenas uma oval.*

- Sim
- Sim, em partes
- Não
- Outro: \_\_\_\_\_

**5. Funcionalidade: Em algum momento durante o uso do Booszinga ocorreu alguma falha inesperada? \***

*Marcar apenas uma oval.*

- Nenhuma vez
- 1 Vez
- 2 Vezes
- Entre 5 e 10 vezes
- Mais de 10 vezes

**6. Confiabilidade: Todos os links do Booszinga direcionam para páginas corretas? \***

*Marcar apenas uma oval.*

- Sim
- Não

---

**7. Confiabilidade: O Booszinga consegue dar feedbacks e orientar adequadamente o usuário durante o seu uso? \***

*Marcar apenas uma oval.*

- Sim
- Às vezes
- Não

**8. Eficiência: O Booszinga consegue processar em poucos segundos as requisições feitas? \***

*Marcar apenas uma oval.*

- Sim, estou satisfeito(a) com o tempo de resposta da aplicação
- Não, demora razoavelmente
- Não, demora um pouco
- Outro: \_\_\_\_\_

**9. Eficiência: As páginas do Booszinga são carregadas em tempo aceitável? \***

*Marcar apenas uma oval.*

- Sim
- Não
- Às vezes
- Outro: \_\_\_\_\_