

UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL TECNOLÓGICA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

INGRID GIAROLA
PEDRO BELLI

BEM-SERVIDO - SISTEMA DE ANÚNCIOS DE SERVIÇOS

CURITIBA
2016

INGRID GIAROLA
PEDRO BELLI

BEM-SERVIDO - SISTEMA DE ANÚNCIOS DE SERVIÇOS

Trabalho de conclusão de curso para obtenção de grau do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná.

Orientador: Prof. Dr. Rafael Romualdo Wandresen

CURITIBA
2016

RESUMO

O *marketing* e gerenciamento de horários são algumas das preocupações constantes para um grupo crescente de empresas e profissionais que prestam serviços. Em um cenário onde grande parte do público de qualquer negócio está frequentemente antenado ao mundo *web*, profissionais de diversas áreas sentiram a necessidade de adaptação para atingir essas pessoas. Essa adaptação parece um pouco mais simples para quem comercializa produtos, uma vez que existem inúmeros *softwares* voltados para essa finalidade. Com base nessas constatações, propõe-se nesse trabalho o desenvolvimento de um *software* capaz de unir as principais funcionalidades que atendam a um profissional que vende serviços visando suprir as necessidades do cliente. O Bem-Servido foi implementado baseado nos principais requisitos levantados pela equipe em reuniões de *brainstorms* no início do projeto, aplicando-se a metodologia ágil *scrum*. Dentre os atributos da aplicação, destaca-se o agendamento *on-line* de horários para atendimento. A linguagem de programação estabelecida para o desenvolvimento foi JavaScript sobre o framework Node.js e o banco de dados utilizado o *Mysql*.

Palavras-Chave: Sistema. Agendamento. Serviços. Anúncio. Propaganda. Negócio. JavaScript. Node.

ABSTRACT

Marketing and schedule management are some of the constant concerns for a great number of companies and professionals that provide services. In a scenario where most of the business people are frequently connected to the web world, professionals of different areas needed to adapt in order to achieve this audience. Such adaptation seems to be simpler for the ones that sell products, once that there are several softwares available for this same purpose. Based on these findings, we are proposing the creation of a software that joins the most important functionalities that attend professional service sellers to meet client needs. The Bem-Servido was implemented based on the most important requirements collected by the team during brainstorm meetings in the beginning of the Project, applying agile methodology "scrum". Amongst application attributes, stand out the online schedule management. The program language chosen for development was JavaScript upon Node.js framework and *Mysql* DB.

Key-words: System. Schedule. Services. Announcement. Business. JavaScript. Node.

LISTA DE FIGURAS

Figura 1 – Representação do fluxo de informações do padrão MVC	19
Figura 2 – Diversas interfaces acessando a aplicação.	20
Figura 3 – Design Responsivo.	23
Figura 4 – Site do TRE-PR	27
Figura 5 – Busca de profissionais em destaque no site Bougue	28
Figura 6 - Site Mercado Livre	29
Figura 7 – Representação do quadro Kanban.....	42
Figura 8 – MVVM.....	44
Figura 9 – Exemplo de utilização da função como classe	46
Figura 10 – Código javascript.....	47
Figura 11 – Scrum: Fluxo das etapas de desenvolvimento do projeto	49
Figura 12 – Fluxo de acesso ao sistema	52
Figura 13 – Tela de início – Pesquisa.....	53
Figura 14 – Tela de cadastro de Profissional Primeiro Passo	54
Figura 15 – Tela de cadastro de Profissional Segundo Passo	55
Figura 16 – Tela de cadastro de Profissional Terceiro Passo.....	56
Figura 17 – Tela de cadastro de Profissional Último Passo.....	57
Figura 18 – Tela de cadastro de Profissional Mensagem Erro	58
Figura 19 – Tela de Cadastro - Cliente	59
Figura 20 – Tela de cadastro de Cliente Mensagem Erro	60
Figura 21 – Erro e-mail já cadastrado.....	61
Figura 22 – Tela Agenda Profissional	62
Figura 23 – Minha Agenda (Profissional)	63
Figura 24 – Tela Novo Agendamento.....	64
Figura 25 – Tela Minhas Habilidades.....	64
Figura 26 – Tela Meus Serviços	65
Figura 27 - Tela Meu Perfil (profissional).....	65
Figura 28 – Perfil Cliente.....	66
Figura 29 – Tela Meus Agendamentos.....	67

Figura 30 – Tela Meu Perfil.....	68
Figura 31 – Tela Agenda do Profissional.....	69
Figura 32 – Tela Emissão de Relatórios.....	70
Figura 33 – Tela Emissão de Relatório por Período.....	71
Figura 34 – Casos de Uso do Sistema	89
Figura 35 - DV1 – Tela Cadastro Profissional	91
Figura 36 - Tela área do profissional.....	93
Figura 37 - DV3 – Tela Fazer Login	95
Figura 38 – Tela Emissão de Relatórios.....	99
Figura 39 – Tela Agenda do Profissional.....	103
Figura 40 - DV6 – Tela Cadastro Cliente.....	106
Figura 41 – DV7 – Tela Área/Perfil do Cliente.....	108
Figura 42 – DV8 – Tela Realizar Agendamento.....	110
Figura 43 - Qualificações a fazer	113
Figura 44 - DV11 – Tela Principal - Pesquisa.....	115
Figura 45 - Tela opção de Logout.....	117
Figura 46 – Diagrama de Classes do Sistema	119
Figura 47 – Diagrama Entidade Relacionamento	121
Figura 48 - WBS – Parte 1	123
Figura 49 - WBS - Parte 2.....	124
Figura 50 - WBS - Parte 3.....	125
Figura 51 - WBS - Parte 4.....	126
Figura 52 - WBS - Parte 5.....	127
Figura 53 - WBS - Parte 6.....	128
Figura 54 - WBS - Parte 7.....	129
Figura 55 – Diagrama de Sequência Cadastro no Sistema	131
Figura 56 - Diagrama de Sequência Editar Cadastro.....	132
Figura 57 - Diagrama de Sequência Login no sistema	133
Figura 58 - Diagrama de sequência Pesquisa Serviço	134
Figura 59 - Diagrama de Sequência Agendamento.....	134
Figura 60 - Diagrama de sequência Erro conexão	135

Figura 61 - Diagrama de Arquitetura.....	136
--	-----

LISTA DE TABELAS

Tabela 1 – Tabela de Responsabilidades de Análise.....	32
Tabela 2 – Cronograma Geral.....	33
Tabela 3 – Cronograma de Documentação - Análise.....	34
Tabela 4 - SPRINT 0.....	35
Tabela 5 - SPRINT 1.....	35
Tabela 6 - SPRINT 2.....	36
Tabela 7 - SPRINT 3.....	36
Tabela 8 - SPRINT 4.....	37
Tabela 9 - SPRINT 5.....	37
Tabela 10 - SPRINT 6.....	38
Tabela 11 – Plano de Riscos.....	120
Tabela 12 – Plano de Comunicação.....	130

LISTA DE SIGLAS

API	- Application Programming Interface (Interface de Programação de Aplicativos)
ASP	- Active Server Pages
AWT	- Abstract Window Toolkit
C.E.	- Critérios Ergonômicos
CEO	- Chief Executive Officer (Diretor Executivo)
CRUD	- Create, Read, Update e Delete (operações utilizadas no banco de dados)
DER	- Diagrama Entidade Relacionamento
GB	- Gigabyte (Unidade de medida)
HTML	- HyperText Markup Language (Linguagem de Marcação de Hipertexto)
HTTP	- Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)
IHC	- Interação Homem-Computador
JIT	- Just in Time
JSP	- JavaServer Pages
LAMP	- Combinação de softwares livres
MVC	- Model View Controller
PDF	- Portable Document Format (Formato Portátil de Documento)
PHP	- Personal Home Page (Linguagem de programação)
POC	- Prova de conceito
SQL	- Structured Query Language (Linguagem de Consulta Estruturada)
RAM	- Random Access Memory (Memória de acesso aleatório)

- RF** – Requisito Funcional
- RNF** – Requisito Não-Funcional
- TCC** – Trabalho de Conclusão de Curso
- TI** – Tecnologia da Informação
- TER** – Tribunal Regional Eleitoral
- UC** – Use case
(Diagrama de caso de uso)
- UML** – Unified Modeling Language
(Linguagem de Modelagem Unificada)
- VRAM** – Vídeo RAM
- WBS** – Work breakdown structure
(EAP - Estrutura Analítica de Projeto)
- WWW** – World Wide Web

SUMÁRIO

1. INTRODUÇÃO	11
1.1 JUSTIFICATIVA	12
1.2 OBJETIVOS DO PROJETO	12
1.2.1 Objetivos Gerais	12
1.2.2 Objetivos Específicos.....	13
1.3 Características do Usuário	13
2. FUNDAMENTAÇÃO TEÓRICA.....	14
2.1 INTERNET E COMÉRCIO	14
2.2 ENGENHARIA DE SOFTWARE	16
2.2.1 Model View Controller (MVC)	18
2.2.2 API	20
2.3 DESIGN RESPONSIVO	22
2.4 ERGONOMIA.....	23
2.5 UNIFIED MODELING LANGUAGE (UML)	25
2.6 DEMANDAS ENCONTRADAS.....	26
2.7 SOFTWARES SEMELHANTES EXISTENTES	26
2.7.1 Agendamento de Biometria – Tribunal Regional Eleitoral	26
2.7.2 Bougue	27
2.7.3 Mercado Livre.....	28
3. METODOLOGIA.....	30
3.1 SCRUM	30
3.2 REQUISITOS	31
3.2.1 Requisitos Funcionais.....	31
3.2.2 Não Funcionais	31
3.2.3 Outros Requisitos	31
3.3 PLANO DE ATIVIDADES	31
3.3.1 Responsabilidades	32
3.3.2 Cronograma.....	32
3.4 PLANO DE RISCO	38

3.5 PLANO DE COMUNICAÇÃO	38
3.6 MATERIAIS.....	39
3.6.1 Hardwares	39
3.6.2 Softwares.....	39
3.7 DESENVOLVIMENTO DO PROJETO	49
3.7.1 Definição do Backlog	50
4. APRESENTAÇÃO DO SOFTWARE	51
4.1 Interfaces do sistema	51
4.1.1 Acesso ao sistema	51
4.1.1.1 Página Principal - Pesquisa	52
4.1.1.2 Usuário não cadastrado	53
4.1.1.3 Perfil do profissional.....	62
4.1.1.4 Perfil do Cliente.....	66
4.1.1.5 Agenda do Profissional.....	69
4.1.1.6 Tela de Emissão de Relatórios.....	70
4.2 CONSIDERAÇÕES FINAIS.....	72
4.3 DIFICULDADES	72
4.4 FUNCIONALIDADES NÃO CONTEMPLADAS.....	73
REFERÊNCIAS	75
APÊNDICE A – REQUISITOS FUNCIONAIS - INTERFACE DO PROFISSIONAL	79
APÊNDICE B – REQUISITOS FUNCIONAIS - INTERFACE DO CLIENTE.....	83
APÊNDICE C – REQUISITOS ORGANIZACIONAIS.....	87
APÊNDICE D – CASOS DE USO.....	89
APÊNDICE E – ESPECIFICAÇÕES DE CASOS DE USO	90
APÊNDICE F – DIAGRAMA DE CLASSES	119
APÊNDICE G – PLANO DE RISCOS	120
APÊNDICE H – DIAGRAMA ENTIDADE-RELACIONAMENTO (DER)	121
APÊNDICE I – WBS.....	122
APÊNDICE J – PLANO DE COMUNICAÇÃO	130
APÊNDICE L – DIAGRAMA DE ARQUITETURA.....	136

1. INTRODUÇÃO

Os avanços tecnológicos vêm proporcionando uma série de melhorias e praticidades para o dia a dia da humanidade. Atualmente, muitos serviços já são automatizados e é possível realizar uma infinidade de tarefas sem sair da comodidade do lar e, se antes os impactos desses avanços eram mal vistos pelas pessoas, hoje as crianças são incentivadas a utilizarem aparelhos tecnológicos antes mesmo de serem alfabetizadas.

Sandra Turchi (TURCHI, 2012) defende a ideia de que a inclusão digital, ocorrida nos últimos anos, fez crescer gradativamente o investimento das empresas em marketing e publicidade na internet, visando alcançar esse novo consumidor.

O propósito do presente projeto é unir em uma só aplicação a conexão entre cliente e profissional, agilizando a contratação de serviços e, ainda, criar uma possível solução para profissionais que ainda encontram uma barreira para se lançarem no mundo digital. Nomeado de Bem-Servido, o projeto consiste no desenvolvimento de um sistema web, utilizando a linguagem de programação JavaScript sobre a plataforma node, para que profissionais de diversos ramos e negócios se cadastrem e compartilhem sua agenda de horários, permitindo que potenciais clientes os encontrem e agendem um horário, sem que haja a necessidade de um prévio contato. Para a concretização do agendamento, o sistema exige que o usuário realize um prévio cadastro de suas informações básicas.

A praticidade e agilidade na contratação de serviços são a essência desse projeto, que foi estruturado nesse documento da seguinte forma:

O primeiro capítulo destina-se a apresentar os objetivos gerais e específicos, além de apresentar as características dos usuários para os quais se destina a aplicação desenvolvida.

No segundo capítulo são abordadas as informações teóricas, retratando as tecnologias utilizadas, tal quais as técnicas e pesquisas realizadas pela equipe. Seguido pelo terceiro capítulo, em que são identificadas as metodologias utilizadas.

O quarto capítulo apresenta o software desenvolvido, explicitando as funcionalidades do mesmo. Por fim, no capítulo quinto, são levantadas as considerações finais da equipe, onde também são referenciadas as funcionalidades

que precisaram ser excluídas do escopo e que, caso sejam desenvolvidas posteriormente, irão incorporar grande relevância ao sistema.

1.1 JUSTIFICATIVA

A necessidade de uma aplicação capaz de facilitar a contratação de serviços motivou o desenvolvimento desse projeto. O sistema de anúncio de serviços tem como fundamento potencializar a rapidez e eficiência de agendamento de horários para profissionais de diversas áreas, bem como para seus clientes. O intuito do sistema é conduzir o usuário a realizar um agendamento de forma intuitiva, bastando que este esteja autenticado. Por outro lado, o profissional estará apto a controlar sua agenda de horários sem nenhuma dificuldade.

A tendência é que cada vez mais sistemas *web* sejam lançados no mercado com o propósito de descomplicar a vida de seus usuários. A plataforma construída visa contribuir com a economia de recursos e tempo, considerando que os usuários estarão aptos a agendar horários com profissionais cadastrados no sistema, acessando-o via internet em qualquer lugar e horário.

1.2 OBJETIVOS DO PROJETO

Esta seção apresenta os objetivos do sistema, estabelecendo o que se espera conquistar ao final do desenvolvimento do projeto.

1.2.1 Objetivos Gerais

O objetivo geral desse trabalho é implementar um sistema responsivo utilizando a linguagem de programação *Javascript* junto o *framework Node*, que possibilite que o usuário anuncie seus serviços *on-line* e disponibilize sua agenda de horários. Por outro lado, o cliente poderá pesquisar o perfil do profissional filtrando a busca por especialidade e localidade. Ao visualizar o anúncio, o cliente é capaz de agendar um horário para atendimento, sem que seja necessário estabelecer um prévio contato. Não faz parte do escopo desse sistema o processo de pagamento ao profissional, já que este costuma ser realizado após a prestação do serviço.

1.2.2 Objetivos Específicos

Para atingir o objetivo geral deverão ser cumpridos os seguintes objetivos específicos:

- Definir e analisar requisitos do sistema;
- Definir tecnologias a serem utilizadas;
- Analisar se todos os requisitos são possíveis de implementar e a maneira como serão implementados realizando a prova de conceito (POC);
- Criar toda a documentação do *software* a ser desenvolvido;
- Desenvolver um modelo para uma base de dados que suporte o sistema;
- Implementar os módulos definidos no levantamento de requisitos.

1.3 CARACTERÍSTICAS DO USUÁRIO

O sistema é focado em atender às necessidades de usuários em busca de serviços que buscam agilidade no agendamento de atendimento. Atualmente não há uma plataforma onde profissionais possam divulgar seu atendimento e ainda possibilitar que o próprio cliente agende um horário. Diversas vezes para esse cliente agendar um horário com um profissional, necessita realizar diversas tentativas de ligações até que consiga entrar em contato com o profissional ou sua secretária, e ainda assim, possivelmente, não consegue agendar um horário cômodo para si.

Os interesses de microempresários ou profissionais autônomos são igualmente atendidos, já que é possível que eles se cadastrem no sistema e criem um anúncio disponibilizando sua agenda de horários. Podem se cadastrar no sistema, profissionais de diversos ramos, como *lava-cars*, cabeleireiros, podólogos, marido de aluguel, jardinagem, serviços domésticos, entre vários outros.

O cliente pode realizar uma busca para encontrar um anúncio que atenda melhor às suas expectativas e na localidade em que preferir e, assim agendar um horário.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo irá abordar as técnicas e pesquisas realizadas para o desenvolvimento do sistema, assim como traz as informações necessárias sobre as tecnologias e linguagens utilizadas para atingir o objetivo final do projeto. Por fim, o capítulo retrata alguns softwares relacionados ao tema, que comportam algum ponto similar à proposta do Bem-Servido.

2.1 INTERNET E COMÉRCIO

A internet surgiu em meio ao crescimento tecnológico, principalmente após a criação do conceito de World Wide Web (www), mas a sua disseminação entre os usuários ocorreu gradativamente, a partir do momento em que se alastraram os provedores que davam acesso ao serviço (NANNI; CANETE, 2010).

Existem muitos aplicativos (*softwares*) que permitem realizar uma série de atividades que rodam na internet, propiciando um conjunto de ferramentas de comunicação e possibilitando discussões à distância ao mesmo tempo em salas de bate-papo (*chats*) ou em outro lugar, chamado de fórum mediante um *mix* de som, texto escrito e imagem, adequando para um período de transição entre a banda estreita e larga, de maior velocidade (MORAN *et al*, 2007).

Verifica-se que por meio do desenvolvimento da Internet o ambiente digital passou a ser visto pelo setor corporativo como um terreno fértil a ser explorado e como uma forma de imprimir maior agilidade aos negócios (TURCHI, 2012).

De acordo ainda com Turchi (2012, p.16):

Na década de 1980, as grandes corporações, principalmente dos setores industrial e varejista, já realizavam transações comerciais pela via eletrônica, com o uso de tecnologias específicas, como a *Electronic Funds Transfer* (EFT ou TEF – Transferência Eletrônica de Fundos) para envio de documentos (ordens de compra e de pagamentos), visando agilizar as operações de logística e da cadeia de suprimentos. Esses serviços eram disponibilizados por provedores especializados, como a *Interchange* (atual GXS Brasil) e Proceda, entre outras. A partir do final dos anos 1990 e início de 2000, com a evolução da Internet e dos mecanismos de segurança, essas operações começaram a ser feitas também pela *web*. Mas o potencial oferecido pela Internet levou as empresas a pensarem de forma mais ampla e a incluir em seus planejamentos de marketing as estratégias de *e-business* – um conceito abrangente que se refere a uma fusão complexa dos processos internos, aplicações empresariais e estrutura organizacional com uso de tecnologia (*hardware e softwares*), de modo a criar um modelo de negócios de alto desempenho.

Esses foram os pilares para o surgimento do *e-commerce*, que se refere às operações de compra e venda de produtos e serviços por meio da internet e os recentemente oriundos deste: *m-commerce* (comércio por meio de aplicativos móveis), *t-commerce* (comércio por meio da televisão), *social-commerce* (comércio por meio de redes sociais) e o mais recente *F-commerce* (comércio por meio da rede social facebook).

No contexto comercial, o Bem-Servido, mesmo que não concretizando a venda do serviço, é uma fusão de *mobile commerce* e *social commerce*, que pratica as modalidades *B2C* e *C2C*, já que não é fundamental que o prestador de serviço seja pessoa jurídica.

Turchi define que o comércio eletrônico, em termos conceituais, compreende as modalidades (TURCHI, 2012, p.17):

- **B2C- Business to Consumer:** segmento dentro do comércio eletrônico que abrange qualquer transação em que uma companhia ou organização vende seus produtos ou serviços para consumidores que navegam pela Internet. Esse segmento se assemelha muito às lojas que fazem venda direta ao consumidor (varejo) através de catálogos e se apresenta tipicamente na web na forma de lojas virtuais. Também pertencem a essa categoria os leilões virtuais, que funcionam como os shoppings tradicionais, onde diferentes lojas vendem seus produtos e pagam taxa de condomínio;
- **B2B – Business to Business:** abrange as transações comerciais – compra e venda de produtos e *services* – entre empresas ou entidades. Como exemplos, temos as indústrias que vendem para os atacados, os atacados que vendem para pequenas lojas de varejo, e assim por diante;
- **B2G – Business to Govern:** define as atividades comerciais pela Internet entre empresas privadas e governamentais;
- **B2I – Business to Institutions:** atividades comerciais pela via eletrônica entre empresas e instituições (educacionais, associações etc.);
- **B2E – Business to Employee:** modelo de comércio eletrônico em que empresas vendem serviços ou produtos aos seus funcionários;

- **E-Procurement:** modalidade de comércio eletrônico utilizado pelas empresas para compra de suprimentos (como materiais de escritório, higiene e limpeza etc.);
- **C2C – Consumer to Consumer:** comércio entre consumidores, feitos de forma direta, por meio de sites apropriados, tais como Mercado Livre, ImóvelWeb, Webmotors, ou seja, são pessoas anunciando para vender para outras pessoas.

2.2 ENGENHARIA DE SOFTWARE

A engenharia de *software* é vista como um ramo da engenharia de sistemas, direcionada para o desenvolvimento de *software* de grande porte e complexos (FINKELSTEIN; KRAMER, 2004). Este ramo estuda, além de técnicas de desenvolvimento de *software*, administração de projetos de *software*, criando ferramentas, teorias e métodos que possam oferecer suporte na criação de sistemas computacionais (VALENTIM, SOUZA NETO, 2004).

Em relação a engenharia de *software*, busca-se enfatizar a sua importância no desenvolvimento de sistemas, abordando fatores como: métodos, ferramentas e procedimentos como elementos fundamentais. Na visão de Presman (1995, p.31):

- Métodos: visam proporcionar de forma detalhadas como fazer para melhorar a construção de *softwares*. Os métodos proporcionam uma série de atividades como: planejamento, estimativa, análise de requisitos de sistemas, codificação, teste, manutenção;
- Ferramentas: implementadas visando apoiar de maneira automatizada ou semiautomatizada os métodos. Atualmente já existem várias ferramentas bastante populares (*freeware* e proprietárias);
- Procedimentos: realizam a coesão entre as ferramentas e os métodos, assim provendo o desenvolvimento racional e oportuno dos sistemas computacionais.

Analisando os elementos descritos acima, a engenharia de *software* busca melhorar o desenvolvimento dos processos de produção, viabilizando assim, a qualidade do produto. De acordo com Sommerville (2004, p.36) os processos de

software são um conjunto de atividades e resultados associados que levam a produção de um produto de *software*.

Dentre as atividades fundamentais dos processos de *software*, citam-se (SOMMERVILLE, 2004):

- Especificação de *software*: responsável pela definição das funcionalidades do *software* e restrições em sua operação, nesse momento são definidos os requisitos e prioridades, assim como é realizada a documentação do sistema;
- Projeto e implementação de *software*: deve ser produzido o *software* de acordo com sua especificação;
- Validação de *software*: deve ser validado visando a garantir a conformidade dos requisitos solicitados pelo cliente;
- Evolução de *software*: necessária a evolução do *software* para melhor atender as necessidades mutáveis do cliente.

Analisando o contexto da engenharia de *software*, verificou-se que eram recorrentes diversos problemas no desenvolvimento de sistemas. Estes fatores contribuíram para o surgimento de uma ferramenta em busca de soluções de problemas, o *Design Patterns* ou Padrões de Projetos, nos quais aplicam-se os problemas clássicos na produção de *softwares* (VALENTIM, SOUZA NETO, 2004). *Design Patterns* são conceituados como modelos de organização de hierarquias de classes e protocolos, bem como da distribuição de responsabilidade entre classes e objetos, caracterizando construções de projeto orientado a objetos, por meio da reutilização (VALENTIM, SOUZA NETO, 2004). Sendo assim, a ferramenta *Design Patterns* torna-se um mecanismo eficiente reunindo as melhores práticas dos desenvolvedores de softwares para solucionar problemas, que possam ser repetidos e diversificar os projetos de *softwares*.

A engenharia de *software* busca as melhores soluções de problemas identificados durante o desenvolvimento de um *software*. Sendo assim, busca-se melhor viabilizar a construção de *softwares* econômicos, eficientes, confiáveis, de qualidade, e que possuem uma satisfatória funcionalidade.

O *Design Pattern* utilizado no desenvolvimento do projeto foi o *Model View Controller* (MVC), que será melhor detalhado no tópico a seguir.

2.2.1 Model View Controller (MVC)

O *Model View Controller* (MVC) é um padrão de arquitetura (*design pattern*) que pode ser utilizado para representar e entender a comunicação existente entre os componentes de uma aplicação seja para *web*, para *desktop* ou para dispositivos móveis. Os modelos de arquitetura que utilizam o padrão MVC definem claramente a separação de responsabilidades e a comunicação entre os componentes de uma aplicação *web* (TERUEL, 2012, p.5).

O padrão MVC é categorizado na literatura especializada como um *design pattern* no desenvolvimento de *software*, sem que haja muita discordância em relação a definição precisa desse padrão. O MVC viabiliza maior representatividade da comunicação entre os componentes que possam produzir o conteúdo de apresentação, a lógica do negócio e o processamento de requisições (TERUEL, 2012, p.7).

Por meio do MVC é simples dividir os trabalhos entre uma equipe de desenvolvimento. É possível definir três grupos de componentes principais, o **Model** (Modelo), o **View** (Apresentação) e o **Controller** (Controle), conforme descrito abaixo (TERUEL, 2012, p.6):

- *Model*: componente responsável pelo conhecimento do domínio de negócio. São representadas as classes que persistem e recuperam do banco de dados um tipo específico de objeto, classes de entidade e as classes que implementam as regras de negócio;
- *View*: Camada de interação com o usuário, responsável pela exibição dos dados;
- *Controller*: componente responsável por controlar o fluxo e o estado da entrada do usuário, fazem a comunicação entre as páginas e as classes que acessam o banco de dados.

Para melhor compreender o funcionamento do padrão utilizado no projeto, a Figura 1 retrata a sua representação estrutural:

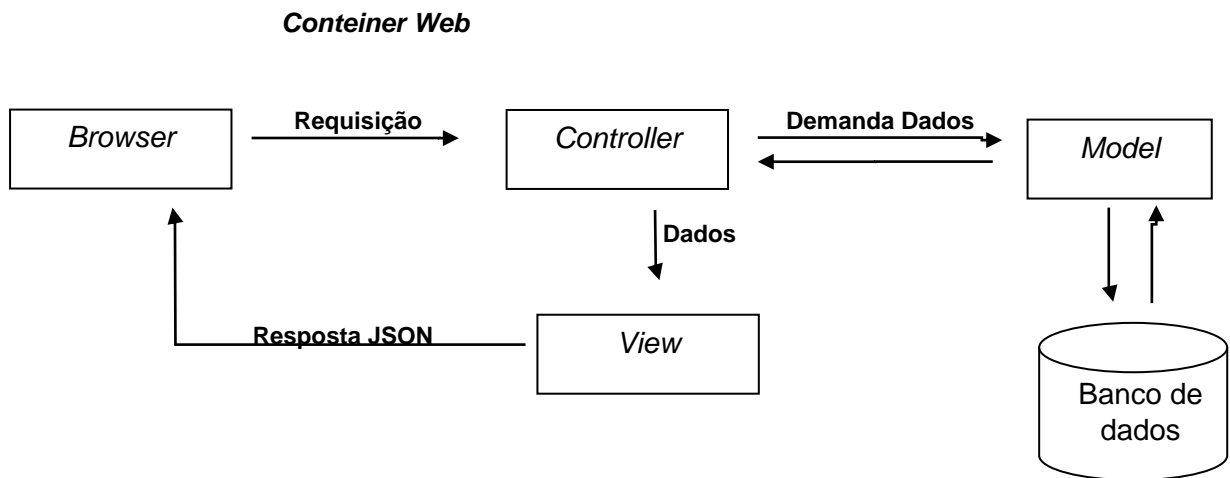


Figura 1 – Representação do fluxo de informações do padrão MVC

Fonte: Os autores

Dentre os objetivos do MVC, destacam-se:

- Desmembrar lógica de negócios (*Model*), interface do usuário (*View*) e fluxo da aplicação (*Control*);
- Assegurar que várias interfaces acessem uma mesma *Model*, sem que esta identifique com quantas e quais *views* está interagindo;

A Figura 2 na página seguinte demonstra diversas interfaces tendo acesso à aplicação:

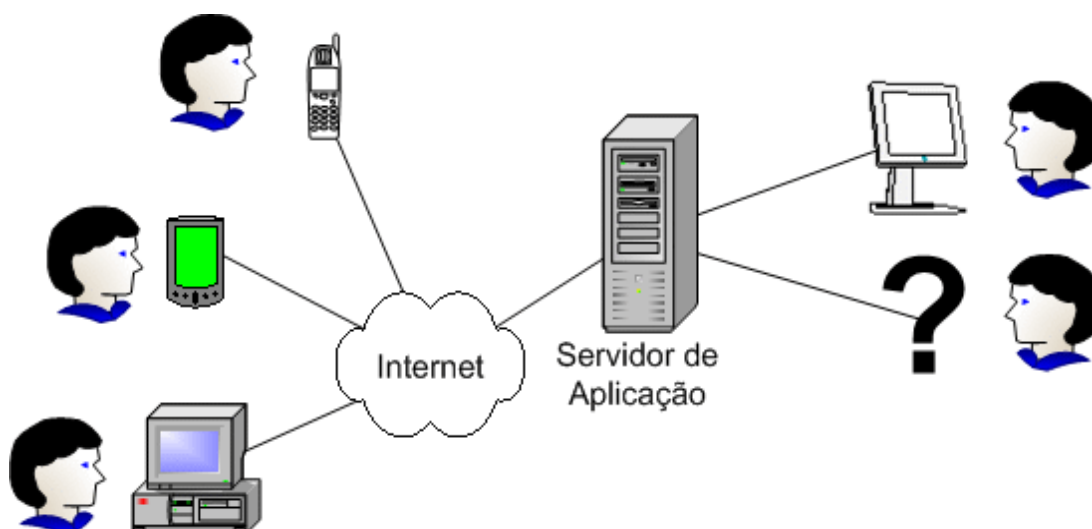


Figura 2 – Diversas interfaces acessando a aplicação.

Fonte: TERUAL, 2012.

2.2.2 API

A Interface de Programação de Aplicação (API) é constituída por um conjunto de rotinas e padrões implementados mediante um *software*, disponibilizadas para que diferentes softwares possam ser desenvolvidos associando recursos deste outro, sem ter que se preocupar com qual tecnologia foi usada ou como a implementação deste recurso/serviço foi feita (SOMERA, 2006).

Um ótimo exemplo é o Google APIs, enquanto é possível usar de seus serviços como o Maps ou o Calendar, também é possível, por meio da API disponibilizada por eles, criar eventos ou carregar mapas a partir de uma aplicação terceira sendo está escrita em C#, JavaScript, Java ou qualquer outra linguagem que possa fazer uma chamada HTTP (CANAL TECH, 2016).

No contexto de desenvolvimento Web vem-se abandonando o uso de APIs SOAP, pela sua maior complexidade, em favor da técnica REST. A Transferência de Estado Representacional (REST) é um estilo de arquitetura usado para definir o uso de Web Standards como HTTP e URIs devem ser feitos, este termo foi definido por Roy T. Fielding em sua tese de PhD. Para que uma API seja considerada REST ela deve ser ou possuir as seguintes características (FIELDING, 2000):

- Interface Uniforme: a interface de comunicação entre o cliente e o servidor deve ser uniforme e essa uniformização deve ser construída baseada nas

seguintes regras: A identificação de recursos deve ser feita em cima de URIs e as ações como criação, edição, etc. são feitas por métodos HTTP.

Abaixo são citados alguns exemplos de APIs que não seguem e que seguem os princípios do REST:

Exemplos de URL que não seguem os princípios do REST.

- /listarAgendamentos = URL para listar agendamentos;
- /criarAgendamento = URL para criar agendamentos;
- /editarAgendamento = URL para editar agendamentos;
- /apagarAgendamento = URL para apagar agendamentos;
- /obterProdutosPorVenda = URL para recuperar todos os produtos de uma venda.

Exemplos de URL que seguem os princípios do REST.

- /agendamentos = URL para listar agendamentos;
- /agendamentos/1 = URL para ações em cima dos recursos;
- /vendas/1/produtos = URL para recuperar todos os produtos de uma venda.

A primeira listagem pode parecer inicialmente mais completa, mas ao juntar as ações definidas pelos métodos HTTP (GET, POST, PUT e DELETE) a segunda listagem se faz muito mais clara e fácil de entender:

- Stateless: cada requisição deve ter todas as informações relevantes para o processamento da mesma, ou seja, o servidor tem como responsabilidade utilizar dos dados recebidos na requisição, realizar o processamento e retornar a resposta para o cliente sem um estado de comunicação entre requisições;
- Cacheável: deve-se indicar ao cliente, quando possível, a possibilidade do cacheamento dos resultados retornados;
- Cliente-servidor: se faz necessária uma definição clara do que faz parte do servidor e o que faz parte do cliente, o cliente não precisa saber como o servidor trata o armazenamento de dados por exemplo, só precisa consumir suas informações, sendo assim uma implementação não depende da outra.

- Sistema em camadas: possibilidade de divisão do servidor em camadas, a interface da API deve funcionar de forma que não seja possível que o cliente diferencie quem o está atendendo, por exemplo é possível fazer uma API em C# para processamento de dados em uma planilha XML e outra em Node.js para processamento de dados cadastrais;
- Código sob demanda: este último item permite a execução de códigos sob demanda por parte do cliente, ou seja, estender parte da lógica do cliente sendo por applets ou scripts. Assim sendo diferentes clientes podem ter comportamentos específicos mesmo utilizando o mesmo serviço. Este é o único item que não faz parte da arquitetura em si, sendo dessa forma opcional.

Apesar do REST incluir conceitos sobre o uso de recursos e uma interface uniforme, ela não diz como essa uniformização deve acontecer e nem a que métodos ela deve responder.

2.3 DESIGN RESPONSIVO

O *design* responsivo busca melhor adaptar o *layout* da aplicação à página, isto é, depende do tamanho da tela do dispositivo que será renderizado. Desta forma, eleva o poder de utilização de dispositivos móveis, sendo necessário melhor adaptação das aplicações e *web sites*, para que possam ser visualizados e utilizados nesse novo tipo de tecnologia, que a partir desta nova premissa, originou o *design* responsivo (FRAIN, 2012).

Para Silva (2014, p.35) o conceito de design responsivo ou layout responsivo “expande e contrai com a finalidade de se acomodar de maneira usável e acessível à área onde é visitado ou, mais genericamente, ao contexto onde é renderizado, seja um smartphone, um *tablet*, um leitor de tela, um mecanismo de busca”.

Entende-se que design responsivo como um princípio de desenvolvimento para Web, com a finalidade em melhor adaptar o layout das páginas a qualquer dispositivo, tela e resolução, com o propósito em garantir a boa experiência do

usuário, proporcionando maior navegação e leitura confortáveis sem comprometer o conteúdo.

A figura 3 apresenta o design responsivo, conforme abaixo:



Figura 3 – Design Responsivo.

Gerada a partir de <<http://nextecommerce.com.br/design-responsivo-nao-e-questao-de-opcao-2/>>.
Acesso em Maio de 2016.

2.4 ERGONOMIA

A ergonomia se mostra habilitada a enfrentar os problemas e propor soluções lógicas para o desenvolvimento de *softwares* interativos que podem ser adaptados aos usuários e adequados a suas tarefas. Historicamente, esta disciplina tem tido um papel importante tanto no desenvolvimento de sistemas, como no controle de centrais nucleares e do tráfego aéreo, onde o caráter crítico exige mínimos níveis de erros e uma máxima eficiência na interação com o operador (CYBIS et al, 2000).

Além da utilização de modelos de tarefas, no entanto, com o propósito em implementar sistemas que possam suportar de maneira efetiva e satisfatória as

tarefas do usuário, há alguns princípios ergonômicos para o processo de construção de sistemas. Uma vez que este processo não é algorítmico, a formulação destes princípios é o resultado de uma experiência coletiva acumulada pela comunidade de Ergonomia e de IHC, examinando as práticas bem (ou mal) sucedidas. Os princípios são tipicamente apresentados em forma de critérios, recomendações ou normas, e raramente são associados a algum método de concepção. São por isto, altamente dependentes de quem os interpreta e aplica. Bastien & Scapin (1995), desenvolveram, a partir do exame exaustivo de uma base de recomendações ergonômicas, o conjunto de critérios ergonômicos (C.E.) que é empregado nas intervenções de ergonomia praticadas pelo LabiUtil.

1. A condução se define no convite (presteza) do sistema, na legibilidade das informações e telas, no feedback imediato das ações do usuário e no agrupamento e distinção entre itens nas telas. Esse último sub-critério refere-se tanto aos formatos (agrupamento e distinção por formato) como a localização (agrupamento e distinção por localização) dos itens.
2. A carga de trabalho se define na brevidade das apresentações e entradas (concisão), no comprimento dos diálogos (ações mínimas) e na densidade informacional das telas como um todo.
3. O controle explícito se define no caráter explícito das ações do usuário (ações explícitas) e no controle que ele tem sobre os processamentos (controle do usuário).
4. A adaptabilidade refere-se tanto às possibilidades de personalização do sistema que são oferecidas ao usuário (flexibilidade) como ao fato da estrutura do sistema estar adaptada a usuários de diferentes níveis de experiência (consideração da experiência do usuário).
5. A gestão de erros refere-se tanto aos dispositivos de prevenção que possam ser definidos nas interfaces (proteção contra erros) como à qualidade das mensagens de erro fornecidas e às condições oferecidas para que o usuário recupere a normalidade do sistema ou da tarefa (correção dos erros).
6. A consistência se refere à homogeneidade e coerência das decisões de projeto quanto as apresentações e diálogos.
7. O significado dos códigos e denominações refere-se à relação conteúdo-expressão das unidades de significado das interfaces.

8. A compatibilidade se define no acordo que possa existir entre as características do sistema e as características, expectativas e anseios dos usuários e suas tarefas. Esse conjunto de critérios tem sido, durante os últimos anos, alvo de uma série de experimentos que visam validá-los em suas pretensões adicionais de organizar os conhecimentos do domínio da ergonomia, contribuir para a normalização e de aperfeiçoar o diagnóstico ergonômico. Assim, os autores consideram que atualmente os critérios são válidos, confiáveis e úteis para, em particular, aumentar a performance de ergonomistas avaliadores.

2.5 UNIFIED MODELING LANGUAGE (UML)

O termo UML refere-se a uma linguagem de modelagem que foi desenvolvida visando a criação de modelos abstratos de processos. Inicialmente, não há restrições quanto aos tipos de processos sendo modelados, desde que possam implementar processos do mundo real ou processos de desenvolvimento de software ou ainda detalhes internos do próprio *software*. Assim, pode ser possível utilizar o UML para descrever o mundo real, como a organização interna de uma empresa por exemplo, como os detalhes internos que descrevem um sistema de *software*. A descrição de um processo envolve a determinação de duas classes básicas de termos (GUDWIN, 2015, p.11):

- Os elementos estruturais que compõem o processo;
- O comportamento que esses elementos desenvolvem quando interagindo.

Assim, o UML tanto pode ser utilizado para a análise dos elementos ontológicos participantes de um processo como do comportamento destes elementos no processo (GUDWIN, 2015, p.11).

Particularmente no que tange a engenharia de *software*, a linguagem UML pode ser utilizada para modelar todas as etapas do processo de desenvolvimento de *software*, bem como produzir todos os artefatos de *software* necessários a documentação dessas etapas (GUDWIN, 2015, p.11).

A linguagem UML, por meio de seus diagramas, permite a definição e *design* de *threads* e processos, que permitem o desenvolvimento de sistemas distribuídos ou de programação concorrente. Da mesma maneira, permite a utilização dos

chamados *patterns* e a descrição de colaborações (esquemas de interação entre objetos que resultam em um comportamento do sistema) (GUDWIN, 2015, p.11).

2.6 DEMANDAS ENCONTRADAS

Durante as pesquisas para o desenvolvimento do projeto, foi notada uma lacuna que falta ser preenchida com relação ao anúncio de serviços na internet, onde além de visualizar propagandas seja possível também que o usuário agende um horário com o profissional anunciante.

Os softwares similares, que possuem a funcionalidade de agendamento de serviços costumam focar em determinadas funções e áreas. O Bem-Servido pretende abranger o máximo de profissões conceituadas em todos os municípios do país, em que se enquadre o agendamento de horários.

2.7 SOFTWARES SEMELHANTES EXISTENTES

Esta seção apresenta alguns projetos que compartilham de certas funcionalidades contempladas pelo Bem-Servido ou onde seja possível realizar agendamento de atendimento *on-line*. Os tópicos seguintes apresentam uma breve descrição sobre cada portal e uma imagem de suas páginas iniciais.


2.7.1 Agendamento de Biometria – Tribunal Regional Eleitoral

O TRE (Tribunal Regional Eleitoral) encontrou uma maneira para facilitar o atendimento aos cidadãos interessados em agendar um atendimento. O sistema de quase todo o país já possui a funcionalidade de agendamento *on-line*, o que proporciona ao usuário a experiência de agendar um horário sem que haja a necessidade de fazer uma ligação telefônica ou se locomover até o local.

O sistema de Agendamento de Atendimento ao Eleitor do TRE-PR (Tribunal Regional Eleitoral do Paraná) foi criado para que os paranaenses possam agendar um dia e horário através do site www.tre-pr.jus.br para realizar a gravação da biometria referente ao título eleitoral.

A Figura 4 demonstra o site do TER-PR, demonstrando o agendamento de atendimento ao eleitor:

Agendamento de atendimento ao Eleitor


Tribunal Regional Eleitoral do Paraná
Agendamento de Atendimento ao Eleitor

Confirmar Agendamento
Agendamento • Novo Agendamento • Confirmar Agendamento

IDENTIFICAÇÃO DO ELEITOR

Nome:	NOME SOBRENOME
Data de nascimento:	13/11/1990
Nome da mãe:	Não consta / Em branco
Nome do pai:	Não consta / Em branco
Telefone de contato:	<input style="width: 80%;" type="text"/> <small>Informe seu telefone, caso precisemos modificar seu agendamento.</small>

LOCAL DE ATENDIMENTO

Município: ▼

Figura 4 – Site do TRE-PR

Fonte: <http://www.tre-pr.jus.br/eleitor/agenda-biometria/agendamento-de-atendimento-ao-eleitor> (Acessado em Outubro de 2016)

2.7.2 Bougue

O Bougue é uma plataforma de busca de serviços focada em realizar orçamentos de reformas residenciais, apenas na cidade de São Paulo.

Lançado em Agosto de 2011 pelo empreendedor Fernando Canuto, o site surgiu com a finalidade de servir como uma ponte entre o prestador de serviços e o consumidor final. Inicialmente a ideia era abranger a contratação de outros tipos de serviços, mas o portal acabou focando em profissionais ligados à reformas e obras. Para 2012, a expectativa do empreendedor era de alcançar outras grandes capitais, como Rio de Janeiro, Brasília, Belo Horizonte, Porto Alegre e Fortaleza, o que não aconteceu.



Figura 5 – Busca de profissionais em destaque no site Bougue

Fonte: <http://www.bougue.com.br/profissionais/destaques/>
(Acessado em Outubro de 2016)

2.7.3 Mercado Livre

O Mercado Livre é a maior comunidade de compra e venda *on-line* de produtos da América Latina. A empresa oferece comércio eletrônico, onde é possível que usuários se cadastrem, anunciem, comprem, vendam ou troquem produtos *on-line*. O cofundador e CEO Marcos Galperin, em 1999, quando finalizava seu MBA na escola de negócios em Stanford, escreveu o plano de negócios do Mercado Livre e desde então reuniu um grupo de pessoas para auxiliá-lo a pôr sua ideia em prática. Atualmente o Mercado Livre atua em 14 países, contando com mais de 30 milhões de usuários cadastrados (MERCADO LIVRE, 2016).

A figura 5 na página seguinte apresenta o site de busca Mercado Livre:

mercado livre

Cadastre-se | Entre | Vender | Contato

Pague parcelado!

VISA 12 x

MasterCard 12 x

Hipercard 12 x

12 x

Diners Club INTERNACIONAL 12 x

elo 12 x

mercado pago 24 x

Ver mais

Encontre o que você está procurando!

Existem milhões de produtos anunciados, as melhores marcas e os menores preços.

Classificados

- Carros, Motos e Outros
- Imóveis
- Serviços

Marcas

- Lojas oficiais

Categorias

- Acessórios para Veículos
- Agro, Indústria e Comércio
- Animais
- Antiquidades
- Arte e Artesanato
- Bebês
- Brinquedos e Hobbies
- Calçados, Roupas e Bolsas
- Câmeras e Acessórios
- Casa, Móveis e Decoração
- Celulares e Telefones
- Coleções e Comics
- Eletrodomésticos
- Eletrônicos, Áudio e Vídeo
- Esportes e Fitness
- Filmes e Seriados
- Games
- Informática
- Instrumentos Musicais
- Joias e Relógios
- Livros
- Música
- Saúde e Beleza
- Mais Categorias

Figura 6 - Site Mercado Livre

Fonte: <http://www.mercadolivre.com.br/>
(Acessado em Outubro de 2016)

3. METODOLOGIA

Os próximos tópicos pretendem abordar a metodologia SCRUM utilizada para o desenvolvimento do *software* proposto, assim como os requisitos do sistema e a divisão dos *Sprints*.

3.1 SCRUM

Há poucos anos, as metodologias utilizadas baseavam-se em documentação extensa, prazos apertados, grandes equipes e, principalmente, desentendimento entre as partes.

O *Scrum* é uma ferramenta que apresenta agilidade para gestão e planejamento de projetos de software. O *Scrum* foi baseado no Manifesto Ágil, surgindo em 2001, nos EUA, no momento que alguns desenvolvedores de softwares obtiveram a percepção que determinados erros de processos, documentação e problemas, eram corriqueiros e semelhantes (BARROS *et al*, 2010).

O SCRUM utiliza-se de padrões de processo em cada iteração, cada padrão de processo define um conjunto de atividades de desenvolvimento (BARROS *et al*, 2010):

- *Pendência (product backlog)*: Uma lista de requisitos priorizados pelo cliente. Itens podem ser adicionados a pendência em qualquer momento, fazendo com que as modificações sejam introduzidas.
- *Sprints*: Unidade de trabalho de uma equipe Scrum. Em uma sprint, escolhem-se quais requisitos presentes na pendência a equipe vai desenvolver. Os requisitos escolhidos são então fixos, de forma que qualquer alteração na pendência não reflita na sprint atual. Dessa forma os membros da equipe trabalham de forma ágil, em curto prazo, e com estabilidade.
- *Reuniões Scrum (Daily Meeting)*: São reuniões de curta duração feitas diariamente pela equipe SCRUM. É baseada sempre em três perguntas:
 - O que você fez desde a última reunião da equipe?
 - Que obstáculos você está encontrando?
 - O que você planeja realizar até a próxima reunião da equipe?

3.2 REQUISITOS

Os requisitos formam o *product backlog* do projeto e foram definidos pela equipe durante as primeiras reuniões realizadas para as definições dos sprints.

3.2.1 Requisitos Funcionais

Os requisitos funcionais representam as funções que o *software* deve desempenhar e como deve se comportar. Para o levantamento destes, foi utilizada a técnica de *Brainstorming*, que consiste em realizar diversas reuniões entre a equipe de desenvolvedores para que as ideias sejam geradas e exploradas.

Os requisitos funcionais relacionados à interface do profissional foram especificados no apêndice A, já os requisitos funcionais relacionados ao cliente, podem ser visualizados no apêndice B.

3.2.2 Não Funcionais

Os requisitos não funcionais definem as propriedades do *software*. Para esse trabalho, foram especificados os requisitos organizacionais no apêndice C.

3.2.3 Outros Requisitos

Visando a melhor comunicação com o cliente final, o sistema será desenvolvido na versão responsiva, isto é, a tela irá se ajustar para *tablets*, *notebooks* ou *smartphones*.

O sistema deverá ser suportado pelos sistemas operacionais *Windows*, *IOS* e *Linux*.

3.3 PLANO DE ATIVIDADES

O plano das atividades a serem realizadas ao longo do projeto foi criado levando em conta que a equipe possuía quatro recursos de *software* e *hardware* no

início do desenvolvimento. O plano foi baseado na tabela de responsabilidades e no cronograma, planejado com base no WBS (Apêndice I) criado no início do desenvolvimento do projeto.

3.3.1 Responsabilidades

A tabela de responsabilidades lista as atividades executadas por cada membro da equipe durante o desenvolvimento do projeto.

Tabela 1 – Tabela de Responsabilidades de Análise

Atividades	Membros da equipe
Declaração do Escopo	Ingrid e Pedro
Plano de Risco	Ingrid
Plano de Comunicação	Ingrid
Plano de Atividades	Ingrid
WBS	Ingrid
POC	Pedro
Levantamento de Requisitos	Ingrid e Pedro
Modelagem de Casos de Uso	Ingrid
Definição da Metodologia	Ingrid e Pedro
Protótipos das Interfaces do Sistema	Ingrid e Pedro
Diagrama de Arquitetura	Pedro
Diagramas UML	Ingrid
Definição da Linguagem de Programação	Ingrid e Pedro
Desenvolvimento da metodologia	Ingrid e Pedro
Desenvolvimento do servidor Central	Pedro
Desenvolvimento dos Web Service	Pedro
Criação da Base de Dados	Ingrid
Desenvolvimento do Software	Pedro
Testes Funcionais	Ingrid e Pedro
Correção de Falhas	Ingrid e Pedro
Entrega do Software	Ingrid e Pedro
Apresentação do Projeto	Ingrid e Pedro
Correção da Documentação	Ingrid e Pedro

3.3.2 Cronograma

A elaboração de um cronograma dá-se por necessária para que haja um planejamento inicial de todas as atividades a serem realizadas para a fabricação do sistema dentro do tempo estimado para a entrega.

O cronograma foi elaborado com base no WBS, considerando-se que para a confecção do projeto foi utilizada a metodologia ágil *scrum* que, como citado previamente, é desenvolvida em ciclos, estes chamados de *sprints*. O prazo definido para realizar as entregas das *sprints* foi de duas a três semanas, iniciando na

segunda-feira e finalizando no domingo da semana seguinte. Foi previsto pela equipe um total de sete *sprints* até a finalização do projeto e o cronograma foi montado visando evidenciar o que seria desenvolvido em cada uma destas *sprints*, o tempo previsto para cada tarefa, assim como demonstrar as datas planejadas.

3.3.2.1 Cronograma do Projeto

Esta seção apresenta os cronogramas realizados, assim como data de início e término de cada atividade executada pela equipe. O quadro abaixo representa o cronograma de planejamento geral do TCC e, posteriormente, o cronograma organizado em *sprints*, iniciando a contagem destes em zero e finalizando em seis.

O *sprint* 0 foi focado apenas na pré análise inicial do projeto e os demais na análise e desenvolvimento do sistema.

Tabela 2 – Cronograma Geral

Nº	Tarefa	Trabalho (horas)	Início	Término	Predecessoras
	PROJETO				
-	Gerência de Projeto	64	Segunda-feira 16/05/16	Domingo 29/05/16	
-	Documentação	200	Segunda-feira 16/05/16	Domingo 27/11/2016	
-	Desenvolvimento		Domingo 21/08/16	Domingo 20/11/16	
1	Desenvolvimento da Metodologia	80			
2	Desenvolvimento do Projeto		Domingo 21/08/16	Domingo 20/11/16	
3	<i>Sprint</i> 0		Segunda-feira 16/05/16	Domingo 29/05/16	
4	<i>Sprint</i> 1		Segunda-feira 13/06	Domingo 21/08	3
5	<i>Sprint</i> 2		Segunda-feira 22/08	Domingo 11/09	4
6	<i>Sprint</i> 3		Segunda-feira 12/09	Domingo 02/10	5
7	<i>Sprint</i> 4		Segunda-feira 03/10	Domingo 30/10	6
8	<i>Sprint</i> 5		Segunda-feira 31/11	Domingo 20/11	7
9	<i>Sprint</i> 6		Segunda-feira 21/11	Quarta-feira 30/11	8
-	Teste	48	Segunda-feira 21/11/16	Quinta-feira 24/11/16	

30	Testes Funcionais	16	Segunda-feira 21/11/16	Quinta-feira 24/11/16	
31	Correção de Falhas	16	Segunda-feira 21/11/16	Quinta-feira 24/11/16	
-	Entrega	48	Sexta-feira 17/11/16	Sexta-feira 23/12/16	
32	Entrega do Software	1	Quinta-feira 01/12/16	Segunda-feira 05/12/16	Todas
33	Apresentação do Projeto	1	Quinta-feira 08/12/16	Quarta-feira 14/12/16	32
34	Correção da Documentação	48	Quarta-feira 14/12/16	Segunda-feira 29/12/16	33

Tabela 3 – Cronograma de Documentação - Análise

Nº	Tarefa	Trabalho (horas)	Início	Término	Predecessoras
	PROJETO				
-	<i>Sprint 0</i>		Segunda-feira 16/05/16	Domingo 29/05/16	
1	Termo de Abertura	8	Segunda-feira 16/05/16	Sexta-feira 13/05/16	
2	Declaração do Escopo	8	Terça-feira 17/05/16	Quarta-feira 18/05/16	
3	Plano de Comunicação	10	Quarta-feira 18/05/16	Quinta-feira 19/05/16	
4	Plano de Risco	10	Terça-feira 24/05/16	Quarta-feira 25/05/16	2;3
5	Plano de Atividades	10	Sexta-feira 27/05/16	Sexta-feira 27/15/16	4
6	WBS	12	Sábado 28/05/16	Domingo 29/05/16	5
-	<i>Sprint 1</i>		Segunda-feira 13/06/16	Domingo 21/08	
8	Requisitos Funcionais	50	Segunda-feira 13/06/16	Sexta-feira 17/06/16	6
9	Requisitos Não-Funcionais	50	Segunda-feira 13/06/16	Sexta 17/06/16	6
10	Definição da Metodologia	15	Quinta-feira 16/06/16	Sexta-feira 17/06/16	8
12	Diagrama de Arquitetura	10	Domingo 19/06/2016	Domingo 19/06/2016	10
13	Diagrama de Casos de Uso	10	Segunda-feira 20/06/16	Sexta-feira 24/06/16	8
14	Especificação dos Casos de Uso	15	Segunda-feira 20/06/16	Sexta-feira 24/06/16	13
15	Protótipo Interfaces	25	Domingo 14/08/16	Quarta-feira 17/08/16	13
16	Diagrama de Classes	6	Sexta-feira 13/08/16	Sexta-feira 13/08/16	10;13

18	Diagrama Entidade-Relacionamento	6	Quarta-feira 10/08/16	Sexta-feira 12/08/16	10;13
19	Definição da Linguagem de Programação	10	Quarta-feira 03/08/16	Sexta-feira 05/08/16	13
-	Sprint 6		Segunda-feira 21/11	Quarta-feira 30/11	
17	Diagrama de Sequência	25	Sexta-feira 18/11/16	Domingo 20/11/16	10;13
-	Análise	280	Segunda-feira 16/05/16	Domingo 20/11/2016	
-	Documentação	200	Domingo 11/09/16	Domingo 27/11/2016	

Tabela 4 - SPRINT 0

Tarefa	Trabalho (horas)	Início Planejado	Término Planejado
Termo de Abertura	8	Segunda-feira 16/05/16	Sexta-feira 13/05/16
Declaração do Escopo	8	Terça-feira 17/05/16	Quarta-feira 18/05/16
Plano de Comunicação	10	Quarta-feira 18/05/16	Quinta-feira 19/05/16
Plano de Risco	10	Terça-feira 24/05/16	Quarta-feira 25/05/16
Plano de Atividades	10	Sexta-feira 27/05/16	Sexta-feira 27/15/16
WBS	12	Sábado 28/05/16	Domingo 29/05/16

Tabela 5 - SPRINT 1

Tarefa	Trabalho (horas)	Início Planejado	Término Planejado
Requisitos Funcionais	50	Segunda-feira 13/06/16	Sexta-feira 17/06/16
Requisitos Não-Funcionais	50	Segunda-feira 13/06/16	Sexta 17/06/16
Definição da Metodologia	15	Quinta-feira 16/06/16	Sexta-feira 17/06/16
Diagrama de Arquitetura	10	Domingo 19/06/2016	Domingo 19/06/2016
Diagrama de Casos de Uso	10	Segunda-feira 20/06/16	Sexta-feira 24/06/16
Especificação dos Casos de Uso	15	Segunda-feira 20/06/16	Sexta-feira 24/06/16
Diagrama de Classes	4	Sexta-feira 13/08/16	Sexta-feira 13/08/16
Diagrama Entidade-Relacionamento	4	Quarta-feira 10/08/16	Sexta-feira 12/08/16

Definição da Linguagem de Programação	10	Quarta-feira 03/08/16	Sexta-feira 08/08/16
Criação da base de dados	20	Segunda-feira 15/08/2016	Segunda-feira 15/08/2016
POC	20	Segunda-feira 15/08/2016	Quinta-feira 18/08/2016
Protótipo Interfaces	25	Domingo 14/08/16	Quarta-feira 17/08/16
Consolidar Diagramas	5	Quinta-feira 18/08/2016	Quinta-feira 18/08/2016

Tabela 6 - SPRINT 2

Tarefa	Trabalho (horas)	Início Planejado	Término Planejado
Desenvolvimento do CRUD de Cliente	3	Segunda-feira 22/08/16	Segunda-feira 22/08/16
Desenvolvimento do CRUD de Habilidade	2	Segunda-feira 22/08/16	Segunda-feira 22/08/16
Desenvolvimento do CRUD de Serviço	4	Terça-feira 23/08/16	Terça-feira 23/08/16
Desenvolvimento do CRUD de Profissional	6	Terça-feira 23/08/16	Quarta-feira 24/08/16
Desenvolvimento do CRUD de Agendamento	12	Quinta-feira 25/08/16	Segunda-feira 29/08/16
Criar Perfil Profissional	8	Terça-feira 30/08/16	Quinta-feira 01/09/16
Documentação do TCC	35	Segunda-feira 22/08/16	Domingo 11/09/16
Revisão da Sprint	6	Domingo 11/09/16	Domingo 11/09/16

Tabela 7 - SPRINT 3

Tarefa	Trabalho (horas)	Início Planejado	Término Planejado
Criar Perfil Cliente	4	Segunda-feira 12/09	Segunda-feira 12/09
Pesquisa de Profissionais	10	Terça-feira 13/09	Sexta-feira 16/09
Desenvolvimento do CRUD de Horas de Trabalho	4	Sexta-feira 16/09	Sábado 17/09
Calendário de Agendamentos	8	Sábado 17/09	Segunda-feira 19/09
Tratamento de erros	6	Terça-feira 20/09	Quarta-feira 21/09

Autenticação	10	Quarta-feira 21/09	Sábado 24/09
Desenvolvimento e aplicação dos templates	20	Segunda-feira 12/09	Segunda-feira 24/09
Documentação do TCC	40	Segunda-feira 12/09	Domingo 02/10
Revisão	6	Domingo 02/10	Domingo 02/10

Tabela 8 - SPRINT 4

Tarefa	Trabalho (horas)	Início Planejado	Término Planejado
Implementar Integrações de autenticação	4	Terça-feira 18/10	Terça-feira 18/10
Complementar Agendamento	3	Quarta-feira 19/10	Quarta-feira 19/10
Complementar Busca	3	Quarta-feira 19/10	Quarta-feira 19/10
Tela interna de agendamento (profissional)	12	Quinta-feira 20/10	Domingo 23/10
Agendamento feito pelo profissional	6	Segunda-feira 24/10	Quarta-feira 27/10
Tela interna de agendamento (cliente)	8	Quarta-feira 27/10	Domingo 30/10/2016
Desenvolvimento e aplicação dos templates	20	Terça-feira 18/10	Domingo 30/10/2016
Documentação TCC	30	Segunda-feira 03/10	Domingo 30/10/2016
Revisão	6	Domingo 30/10/2016	Domingo 30/10/2016

Tabela 9 - SPRINT 5

Tarefa	Trabalho (horas)	Início Planejado	Término Planejado
Implementar bloqueio na agenda de profissional	10	Segunda-feira 31/11/2016	Quinta-feira 03/11/16
Implementar Qualificação	12	Sexta-feira 04/11/16	Segunda-feira 07/11/16
Implementar Relatórios	6	Terça-feira 08/11/16	Quarta-feira 09/11/16

Desenvolvimento da home	4	Quinta-feira 10/11/16	Quinta-feira 10/11/16
Desenvolvimento e aplicação dos templates	25	Segunda-feira 31/11/2016	Terça-feira 15/11/16
Diagrama de Sequência	25	Sexta-feira 18/11/16	Domingo 20/11/16
Documentação TCC	30	Segunda-feira 31/10/2016	Domingo 20/11/16
Revisão	6	Domingo 20/11/16	Domingo 20/11/16

Tabela 10 - SPRINT 6

Tarefa	Trabalho (horas)	Início Planejado	Término Planejado
Plano de Teste	8	Segunda-feira 21/11/16	Sábado 26/11/16
Testes Funcionais	8	Segunda-feira 21/11/16	Sábado 26/11/16
Ajustes UML	40	Segunda-feira 21/11/16	Quinta-feira 24/11/16
Ajustes e correções do desenvolvimento	24	Segunda-feira 21/11/16	Segunda-feira 28/11/16
Ajustes e correções da documentação	40	Segunda 21/11/16	Quarta-feira 30/11/16

3.4 PLANO DE RISCO

O plano de risco é uma maneira de abordar e planejar decisões e possíveis respostas sobre os riscos do projeto, definindo o tipo e o nível do risco. Esse passo é importante para o gerenciamento do projeto, pois o plano se torna um manual sobre como agir em caso de ocorrência de um dos riscos previstos.

O plano de risco se situa no Apêndice G.

3.5 PLANO DE COMUNICAÇÃO

O plano de comunicação é um planejamento com base na necessidade de informações e requisitos, para que as comunicações interna e externa do projeto sejam realizadas de maneira eficaz.

O plano de comunicação se situa no Apêndice J.

3.6 MATERIAIS

Os materiais utilizados para a confecção do projeto seguem descritos nesta seção.

3.6.1 *Hardwares*

Os recursos de *hardware* utilizados pela equipe para a execução do projeto, tanto para desenvolvimento quanto para documentação, foram dois notebooks pessoais, cada qual possuindo as seguintes configurações:

Proprietária: Ingrid Giarola

- Marca: *Acer*
- Sistema Operacional: *Windows 10*
- Memória RAM: 8GB
- Processador: Intel Core i5-5200U
- Espaço de armazenamento: 1000 GB
- Placa de vídeo: NVIDIA GeForce 920M with 2GB Dedicated VRAM

Proprietário: Pedro Belli

- Marca: *Dell*
- Sistema Operacional: *Windows 10*
- Memória RAM: 6GB
- Processador: Intel CORE i5
- Espaço de armazenamento: 1000GB
- Placa de vídeo: NVIDIA GeForce GT 625M e Intel HD GRAPHICS 4000

3.6.2 *Softwares*

Os recursos de *software* - padrões, técnicas, tecnologias e linguagens - utilizados pela equipe para a execução do projeto, tanto para desenvolvimento quanto para documentação seguem descritos nessa seção.

1. *GitHub*: Repositório *on-line* utilizado para armazenar os dados referentes a todas as versões geradas durante o projeto (item 4.6.2.1);
2. *Atom*: É um editor de texto, de código aberto, moderno e personalizável. Funciona junto ao *GitHub*;
3. *Astah Community*: ferramenta CASE completa para a criação de diagramas UML de análise e design, de código aberto;
4. Pacote *Microsoft Office 2010*: Editores de texto e de planilhas comerciais utilizados na documentação do projeto;
5. *MySQL Workbench*: Ferramenta de gerenciamento de banco de dados;
6. *WBSTool*: *Software on-line* gratuito para a criação do WBS que funciona em qualquer navegador desde que tenha o *plugin flash* instalado.
7. *Trello*: Ferramenta grátis para gerenciamento de projetos, baseada no sistema *Just in Time* (JIT) e adotando a metodologia *kanban* (item 4.6.2.2). A ferramenta é versátil e pode ser ajustada conforme as necessidades do usuário, sendo possível utilizá-la para gerenciar projetos individuais ou em equipe. O *Trello* pode ser acessado dos principais navegadores: Google Chrome, Mozilla Firefox, Safari e Internet Explorer; e o aplicativo também está disponível para download para todos os smartphones.

3.6.2.1 Github controle

Para melhor entender o conceito de *GitHub*, aborda-se o *Git* como um sistema de versões, que é um *software* criado para manter um registro das alterações realizadas em arquivos ao longo do tempo. Mais especificamente, o *Git* é um sistema de controle de versões distribuído, o que significa que todos que estiverem trabalhando em um projeto no *Git* terão uma cópia de todo o histórico do projeto, e não apenas do estado atual dos arquivos (BELL; BEER, 2015, p.13).

O *GitHub* também é uma rede social, que possui funcionalidades como feeds, *followers*, *wiki* e um gráfico que mostra como os desenvolvedores trabalham as versões de seus repositórios, dentre muitas outras funções.

O site permite que haja muita colaboração do usuário, tornando mais prático trabalhar com outras pessoas em um projeto. Isso é feito por meio da disponibilização de um local centralizado para compartilhar o repositório, uma interface web para visualizá-lo e recursos como *forking*, *pull requests*, *issues* e *wikis*, que permitem especificar, discutir e revisar alterações junto à sua equipe de maneira eficiente (BELL; BEER, 2015, p.13).

O Git foi empregado neste projeto pois é portátil e de fácil integração com o ambiente gerado para o desenvolvimento do projeto.

3.6.2.2 *Kanban e Just In Time*

A metodologia *Kanban* é uma maneira prática e inteligente de gerenciamento de projetos em Tecnologia da Informação (TI), ela tem como finalidade controlar o fluxo do desenvolvimento de um projeto, de maneira que organize as etapas do ciclo de produção – nomeadas de cartões – em quadros aos quais todos os integrantes da equipe têm acesso e controle, e um quadro se inicia apenas quando o anterior é finalizado.

O *Kanban* é um método que reduz o tempo de espera, diminuindo o estoque, melhorando a produtividade e interligando todas as operações em um fluxo uniforme ininterrupto (MOURA; R.A., 1996).

O Sistema JIT é um conjunto de técnicas de administração da produção, no qual são incluídos aspectos de administração de materiais, gestão de qualidade, arranjo físico, projeto do produto, organização do trabalho e gestão de recursos humanos. Ou seja, é um sistema, composto por práticas gerenciais, cuja característica principal é “puxar” a produção ao longo do processo (MOURA, 2007; CORRÊA; GIANESI, 2009).

A metodologia *Kanban* foi implementada, internamente, pela Toyota em seu sistema de produção por volta do ano de 1962, no Japão.

O cartão *Kanban* é um meio simples e visível que o sistema de Produção Toyota usa para encomendar componentes à medida que eles são necessários, significando que um nível mínimo de componentes é mantido na área de montagem. Assim que os estoques necessitam reabastecimento, o operador através de um

cartão *Kanban* assegura uma entrega *just-in-time* (Toyota Caetano Portugal, AS; Texto retirado do site da Toyota, 2016).

Atualmente há diversos *softwares kanban on-line* disponíveis para os usuários gerenciarem seus projetos, tais como:

- Trello (<http://trello.com>)
- KanbanFlow (<http://kanbanflow.com>)
- Kanban Tool (<http://kanbantool.com>)
- Backlogged (<http://backlogged.com.br>)

A imagem a seguir representa um quadro *Kanban*, no qual cada quadrado colorido simboliza um cartão, que estão dispostos em cada fase do ciclo do projeto:



Figura 7 – Representação do quadro *Kanban*

Fonte: Portal Sabesim, 2016.

3.6.2.3 Javascript

JavaScript foi implementada pela Netscape em parceria com a Sun Microsystems, com o propósito de fornecer um meio de adicionar interatividade a uma página web. A primeira versão, denominada JavaScript 1.0, sendo lançado em 1995 e implementada em março de 1996 pelo navegador Netscape Navigator 2.0 quando o mercado era dominado pela Netscape (SILVA, 2010).

A escolha do nome JavaScript causou muita especulação, já que várias pessoas pensaram que a Netscape estava tentando pegar carona no nome Java, uma palavra quente, na época. Infelizmente, a escolha do nome causou muita confusão, já que muitos supuseram que as duas linguagens estavam de alguma forma relacionadas. Na realidade, elas tinham muito pouco em comum (IHRIG, 2014).

A linguagem Javascript é leve, interpretada e baseada em objetos com funções de primeira classe, sendo mais conhecida como a linguagem de *script* para páginas *Web*, mas utilizada também em vários outros ambientes sem *browser*, como o *node.js*. É uma linguagem de *script* multi-paradigma, baseada em protótipo, é dinâmica e suporta estilos de programação orientado a objetos, imperativo e funcional. (Mozilla Developer Network, 2014)

Para Silva (2010, p.23) *JavaScript* é uma linguagem desenvolvida com a finalidade de “rodar no lado do cliente, isto é, a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no navegador do usuário. Isso é possível porque existe um interpretador JavaScript hospedado no navegador”.

Não há como fazer funcionar um formulário HTML com o uso de elementos HTML. A linguagem de marcação HTML destina-se a estruturar uma página web, e limita-se a criar os rótulos e campos de um formulário para serem preenchidos pelo usuário e nada mais. Com HTML, não é possível processar os dados nem mesmo enviá-los ao servidor ou a outra máquina qualquer. Para cumprir essas tarefas, é necessário utilizar um programa que consiga manipular e processar os dados. Entre as várias linguagens de programação destinadas a adicionar e processar dados em páginas web, destacam-se PHP, ASP, Java, Ruby, Python e ColdFusion, entre outras. As linguagens de programação como as citadas anteriormente foram desenvolvidas para rodar no lado do servidor, isto é, dependem de uma máquina remota onde estão hospedadas as funcionalidades capazes de interpretar e fazer funcionar os programas (SILVA, 2010, p.23).

3.6.2.4 KO - Knockout.JS

O *Knockout*, ou KO, é uma biblioteca para *Javascript* de fácil manutenção e que simplifica a criação de interfaces gráficas dinâmicas, dispondo de um *display* responsivo através da utilização do padrão MVVM (*Model-View-View-Model*).

3.6.2.5 MVVM

O padrão MVVM é baseado no modelo MVC, apresentado no tópico 2.2.1, e foi apresentado ao mundo dos desenvolvedores em 2005 no blog do arquiteto do *Windows Presentation Foundation* (WPF) e *Silverlight* na *Microsoft*, John Gossman (GOSSMAN, 2005).

O modelo é considerado uma evolução do padrão Model-View-Presenter (MVP) devido à divisão de responsabilidades e por não haver necessidade de uma referência direta para a *view*. A seguir, serão representadas as responsabilidades de cada esfera do MVVM (LUQUE *et al*, 2013):

- *Model*: Representa os dados e a lógica de negócio;
- *ViewModel*: Para cada *Model* há uma *ViewModel* associada por meio de um mecanismo de vinculação de dados (*binding*) e pode passar comandos à *Model*;
- *View*: Representa a interface gráfica, geralmente definida de maneira declarativa.

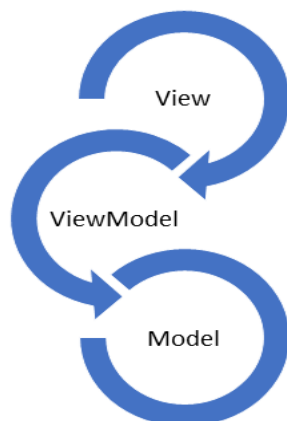


Figura 8 – MVVM

Fonte: Stradioti (2014).

3.6.2.6 Node.js

Em 2009, Ryan Dahl criou o Node.js, uma estrutura primariamente usada para criar servidores altamente escalonáveis para aplicações web. O Node.js, ou simplesmente, Node, é escrito em C++ e em *JavaScript*. Para impulsionar o Node, Dahl conectou a força do motor V8 de *JavaScript* do Google (V8 é o engenho dentro do Google Chrome, o navegador mais popular existente) (IHRIG, 2014).

Desta forma, utilizando o V8, os desenvolvedores podem escrever aplicativos completamente maduros – aplicativos que normalmente seriam escritos numa linguagem como o C ou o Java. Por meio da invenção do Node, o *JavaScript* tornou-se uma linguagem de confiança do lado servidor (IHRIG, 2014).

Node tem como objetivo construir aplicações de rede rápidas e escaláveis. Utiliza um modelo de I/O visando ao evento não bloqueante, proporcionando uma plataforma mais leve e eficiente, adequada para atender as aplicações em tempo real com troca intensa de dados mediante dispositivos distribuídos (ABERNETHY, MICHAEL; 2016).

Node apresenta um modelo de execução não convencional. Para entender como o Node é diferente, devemos compará-lo com o Apache, o popular servidor web da pilha de software Linux, Apache, MySQL e PHP (LAMP). Primeiro, o Apache processa apenas requisições HTTP, deixando que a lógica do aplicativo seja implementada numa linguagem tal como o PHP ou o Java. O Node remove uma camada da complexidade, combinando a lógica do servidor e do aplicativo em um só lugar. Alguns desenvolvedores criticaram esse modelo, por eliminar a tradicional separação de foco empregada na pilha LAMP. Porém, essa abordagem também enseja ao Node uma flexibilidade sem precedentes como servidor (IHRIG, 2014).

3.6.2.6.1 Orientação a objetos utilizando *JavaScript*

A orientação a objetos em *JavaScript* pode ser confusa e até impactante para desenvolvedores habituados com linguagens como Java, C#, Rails, entre outras. Uma vez que a linguagem oferece muita liberdade ao programador por não haverem restrições que o forcem a desenvolver seguindo um determinado padrão ou regras de desenvolvimento, isso exige que os desenvolvedores de um projeto em

JavaScript sejam mais atentos e disciplinados. Diferente da maioria das linguagens, *JavaScript* é orientada a objetos baseada em protótipos, a conotação de classes e função é a mesma, o que em outras linguagens se difere.

Um dos aspectos chave da linguagem é o uso de funções em inúmeros casos e não apenas utilizadas para receber argumentos, executar trechos de código e eventualmente retornar um valor. Um desses casos é a utilização da função como classe como no exemplo a seguir (Figura 9):

```
1 // define a classe Pessoa
2 var Pessoa = function(nome) {
3     this.nome = nome;
4
5     this.dizOi = function(){
6         console.log("Oi, meu nome é: " + this.nome);
7     };
8
9     this.dizTchau = function(){
10        console.log("Tchau");
11    };
12 }
13
14 var pessoa1 = new Pessoa('João');
15
16 pessoa1.dizOi(); // Oi, meu nome é: João
```

Figura 9 – Exemplo de utilização da função como classe

Fonte: Os autores (2016).

Este exemplo mostra a criação de uma classe Pessoa, a instanciação de um objeto do tipo pessoa, e a chamada a um método de instancia desse objeto.

```

18 // define a classe Cliente
19 var Cliente = function(nome) {
20     // Chama o método pai
21     Pessoa.call(this, nome);
22
23     this.dizerOi = function() {
24         console.log("Oi, meu nome é: " + this.nome + " e eu sou um cliente!");
25     }
26 };
27
28 // herda de Pessoa
29 Cliente.prototype = new Pessoa();
30
31 // corrige o ponteiro construtor, que aponta para Pessoa
32 Cliente.prototype.constructor = Cliente;
33
34 var cliente1 = new Cliente('Bob');
35 var cliente2 = new Cliente('Cleriston');
36
37 cliente1.dizerOi(); // Oi, meu nome é: Bob
38 cliente2.dizerOi(); // Oi, meu nome é: Cleriston
39 cliente1.dizTchau(); // Tchau
40
41 // checa a herança
42 console.log(cliente1 instanceof Pessoa); // true
43 console.log(cliente1 instanceof Cliente); // true
44

```

Figura 10 – Código javascript

Fonte: Os autores (2016).

A figura 10 ilustra um exemplo simples de herança em *JavaScript*, onde é criada a classe filha de *Pessoa* nomeada *Cliente*. Ao fim do exemplo é possível verificar que o objeto *cliente1* foi criado a partir dos construtores fornecidos, no caso *Pessoa* e *Cliente*.

3.6.2.7 Banco de Dados Relacional *MySQL*

O sistema gerenciador de banco de dados que será utilizado no desenvolvimento do projeto é o *MySQL*, que executa comandos na linguagem *SQL*.

O *MySQL* é o banco de dados *open source* mais popular do mundo, é ideal para pequenas ou grandes aplicações, possui alto desempenho, é escalonável e fornece um ótimo custo-benefício aos usuários. Sua escolha se justifica devido à familiaridade da equipe com a tecnologia, pois esse banco foi utilizado no

desenvolvimento de trabalhos anteriores e estudado nas disciplinas de Banco de Dados presentes na grade do curso.

3.6.2.7 *RequireJS*

RequireJS carrega arquivos e módulos de forma assíncrona e é muito usado tanto em clientes, como em servidores com JavaScript. Tendo em mente uma estrutura modular, o RequireJS faz com que os arquivos sejam carregados sob demanda, ou seja, apenas um arquivo será carregado quando realmente for necessário o uso dele.

3.6.2.8 *Framework*

Entende-se que um *framework* é definido como um *software* parcialmente completo projetado para ser instanciado. O *framework* define uma arquitetura para uma família de subsistemas e oferece os construtores básicos para criá-los. Também são explicitados os lugares ou pontos de extensão nos quais devem ser feitas adaptações do código para um funcionamento específico de certos módulos (BARRETO JUNIOR, 2006).

Utilizar *frameworks* conceituados que implementam padrões de arquitetura e de projeto no desenvolvimento de programas, aumenta significativamente a produtividade, além de tornar as aplicações robustas e confiáveis.

3.6.2.8.1 *Materialize*

Foi utilizado no desenvolvimento do projeto o Materialize, que é um *framework front-end* – toda a parte da apresentação visual de um sistema - responsivo baseado em *Material Design* do *Google* - um guia para redefinir o visual de aplicativos e *websites*, gerando um novo padrão.

3.7 DESENVOLVIMENTO DO PROJETO

Para o desenvolvimento do projeto foi utilizada a metodologia ágil *scrum* de maneira adaptada às necessidades da equipe.

Foi criado um fluxograma para representar as etapas realizadas durante o desenvolvimento baseado na metodologia, em seguida é apresentado um breve descritivo sobre as adaptações. A figura 11 apresenta as etapas de desenvolvimento do projeto.

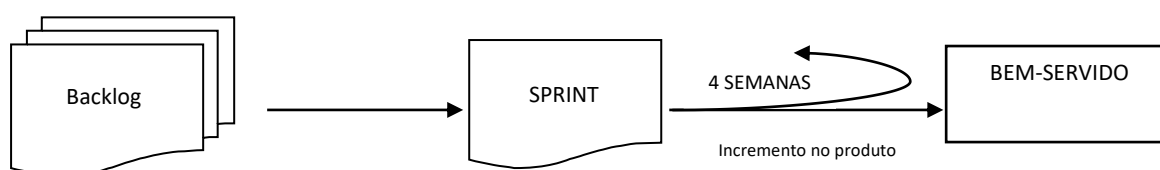


Figura 11 – Scrum: Fluxo das etapas de desenvolvimento do projeto

Fonte: Os autores (2016).

- Logo no início do projeto, todos os membros da equipe assumiram o papel de *Product Owner*.
- Foi realizada a definição dos requisitos funcionais e não-funcionais, compondo o *Backlog*.
- O progresso de cada *Sprint* foi monitorado utilizando a extensão do Trello: Elegantt, o qual gera e atualiza o *Release Burndown* automaticamente.
- Divisão do *Backlogs* em *Sprints*, com duração de duas semanas. Caso fosse necessário atrasar a *Sprint*, era acrescentado uma semana ao prazo. No decorrer do projeto, a duração de cada *Sprint* passou a ser de quatro semanas.
- O *scrum* prega que as equipes realizem reuniões presenciais diárias a cada dia de *Sprint* para que seja disseminado o andamento do projeto. Essa proposta se mostrou inviável para a equipe e as reuniões ocorreram, na realidade, aos domingos e quartas presencialmente, e sempre que necessário sendo realizada uma chamada através da tecnologia de comunicação *Hangouts*.

- O fim de cada *Sprint* se deu quando os itens de cada *Sprint Backlog* foram apresentados ao professor orientador nas reuniões semanais de sexta-feira.
- Uma *Sprint* só foi concretamente definida ao término da *Sprint* anterior.

3.7.1 Definição do *Backlog*

A partir da análise dos requisitos funcionais e não-funcionais, foi definido o *Backlog* do projeto. Deste ponto em diante, cada requisito foi dividido em partes menores e foram acrescentados os diagramas de documentação para ser realizada a divisão das *sprints*.

As *sprints* estão descritas no tópico 3.3.2 – Cronograma, nas tabelas de 4 a 10.

4. APRESENTAÇÃO DO SOFTWARE

O *software* proposto tem como finalidade atuar como um site de anúncio de serviços, onde um profissional anunciante pode disponibilizar uma breve descrição sobre suas especializações e agenda de horários.

Os usuários (clientes) visitam o sistema em busca de um profissional e podem filtrar a pesquisa por região e ramo e assim, ao se autenticar na plataforma, poderá agendar um horário cômodo com o prestador.

O *software* apresenta interfaces relacionadas ao perfil cliente e profissional, e cada um destes possui um *login* de acesso. Este capítulo irá fornecer uma visão mais ampla sobre o funcionamento do sistema.

4.1 Interfaces do sistema

Nos próximos tópicos serão introduzidas as interfaces do sistema, em que serão apresentadas as imagens de cada página web, e em seguida explicadas suas funcionalidades detalhadamente.

4.1.1 Acesso ao sistema

A utilização do sistema não depende do *login* do usuário, é viável que o usuário navegue pela página de busca e visualize os profissionais sem que seja cadastrado no sistema. Já para ter acesso à outras funcionalidades do sistema, é necessário que o usuário seja autenticado por meio da interface de *login*. O sistema verifica o perfil do usuário e, caso seja do tipo profissional, o usuário é direcionado para a interface contendo as funcionalidades de profissional, caso contrário, é redirecionado para a interface de cliente. Na próxima página, está representado o fluxo de acesso ao sistema Bem-Servido (figura 12).

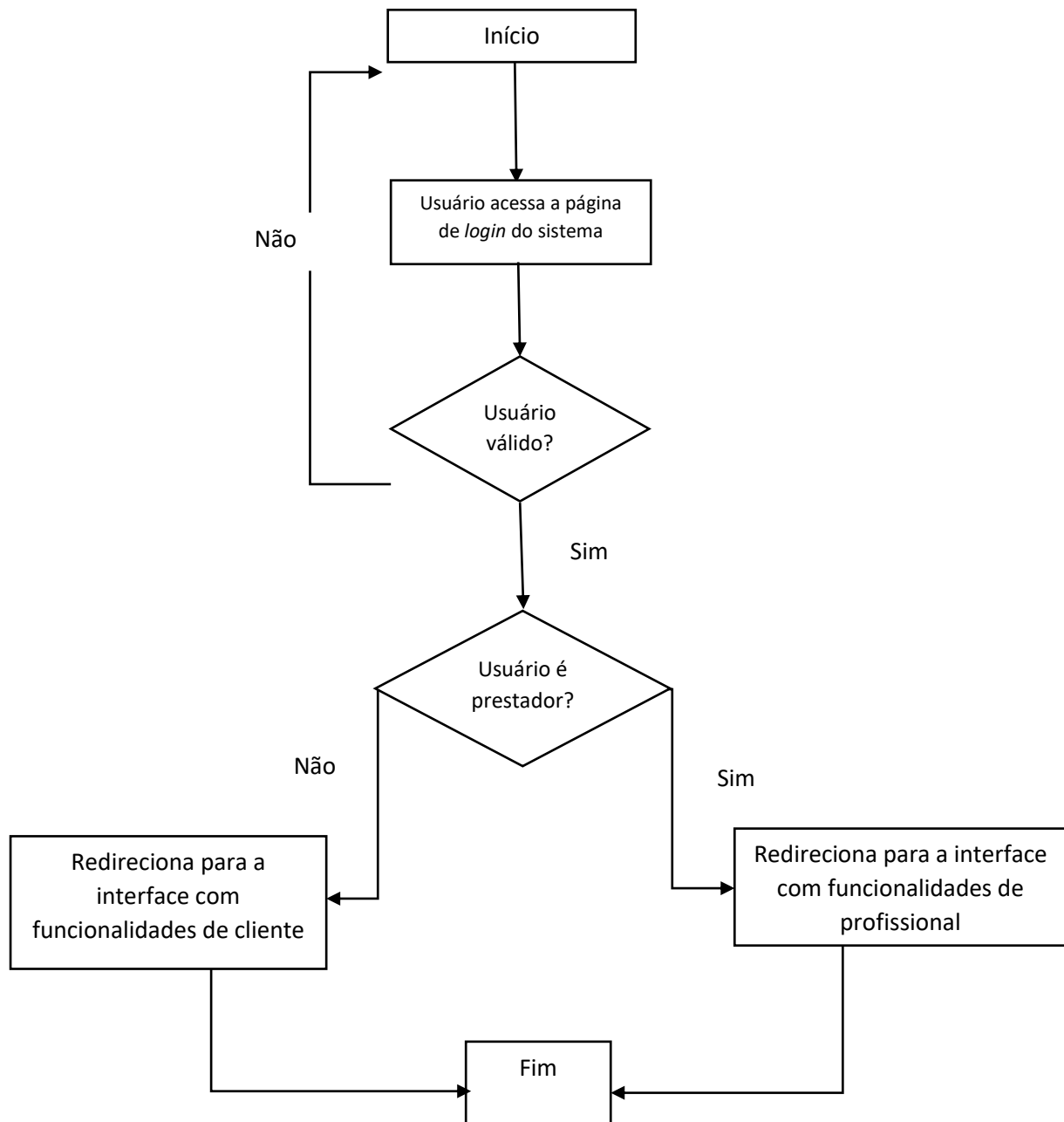


Figura 12 – Fluxo de acesso ao sistema

Fonte: Os autores (2016).

4.1.1 Página Principal - Pesquisa

Ao acessar o sistema, a tela com a qual os usuários se deparam é a página de pesquisa. A página é comum a todos que utilizam o sistema e apresenta no topo da página uma barra de pesquisa, onde, por meio desta, possibilita que seja

realizada uma busca filtrada por tipo de serviço, local e data. Abaixo, são exibidos os profissionais em destaque.

No cabeçalho da tela estão disponibilizadas as funcionalidades de cadastro básico e *login* no sistema, além da opção para os anunciantes criarem uma agenda profissional.

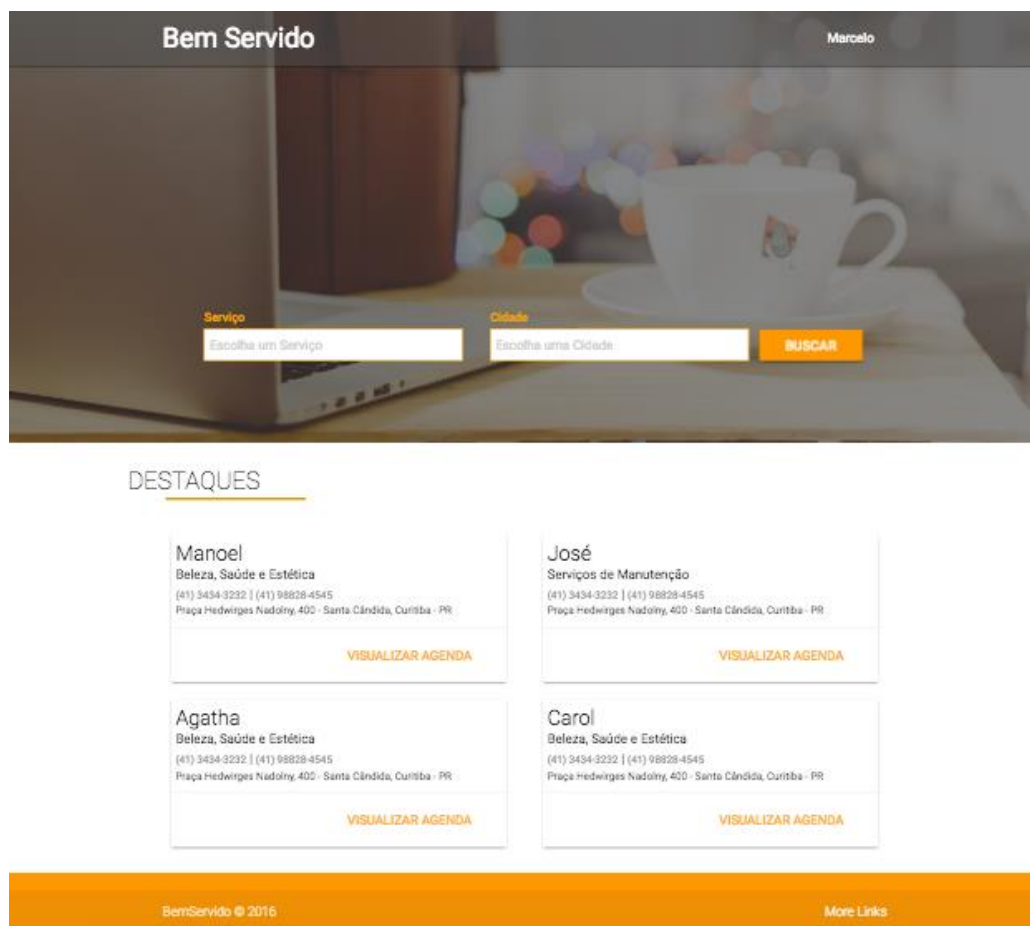


Figura 13 – Tela de início – Pesquisa

Após realizar a busca, o usuário deve selecionar o anúncio que deseja visitar e os dados serão carregados na tela, direcionando para a tela de “Anúncio do Profissional”, descrita no item 3.5 desta seção.

4.1.2 Usuário não cadastrado

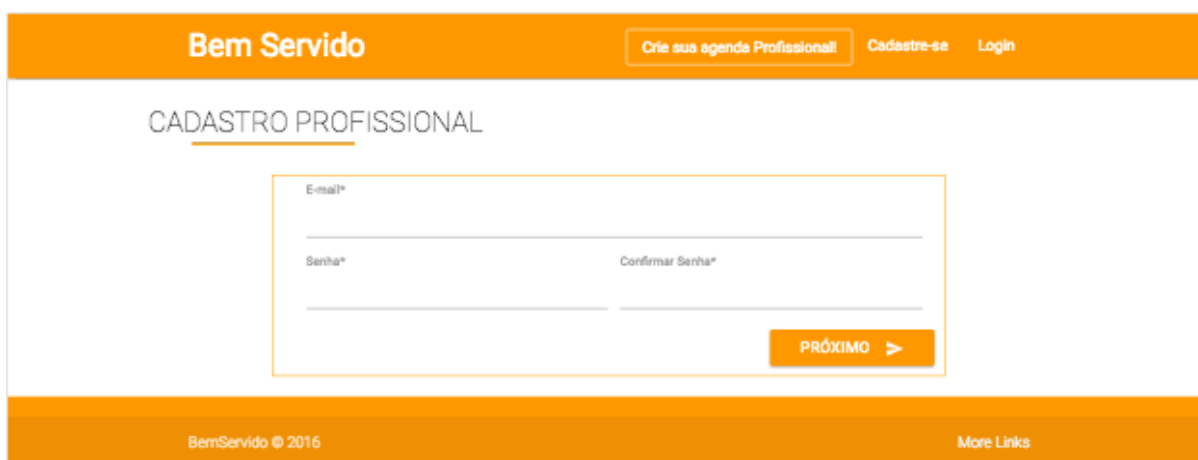
Caso o usuário não seja cadastrado no sistema, este deve acessar a interface de cadastro que contém as funcionalidades referentes ao registro de informações na

base de dados, selecionando a opção “Cadastro” no topo da tela principal do sistema. As interfaces de cadastro podem ser visualizadas abaixo.

4.1.2.1 Cadastro – Profissional – Crie sua agenda profissional

O cadastro de profissional ocorre em três passos, descritos a seguir:

Primeiro passo:



A imagem mostra a interface de usuário para o primeiro passo do cadastro profissional. No topo, há uma barra laranja com o texto "Bem Servido" à esquerda e, à direita, três botões: "Crie sua agenda Profissional!", "Cadastre-se" e "Login". Abaixo, o título "CADASTRO PROFISSIONAL" está centralizado. O formulário principal contém três campos de entrada: "E-mail*", "Senha*" e "Confirmar Senha*", todos com linhas de texto cinzas. Um botão laranja com o texto "PRÓXIMO" e uma seta para a direita está posicionado no canto inferior direito do formulário. Na base da página, uma barra laranja contém o texto "BemServido © 2016" à esquerda e "More Links" à direita.

Figura 14 – Tela de cadastro de Profissional Primeiro Passo

O primeiro passo para que o usuário realize o cadastro, é informar seu e-mail e definir uma senha de acesso, devendo clicar em “Próximo” para prosseguir com o cadastro.

Segundo passo:

The screenshot shows the 'CADASTRO PROFISSIONAL' form in the Bem Servido system. The form is titled 'CADASTRO PROFISSIONAL' and is part of a multi-step process. It contains several input fields for personal and contact information, including Name, CPF/CNPJ, Date of Birth, Sex, Telephone, Cellular, CEP, Address, City, and State. There are also navigation buttons for 'ANTERIOR' and 'PRÓXIMO'.

Nome Completo*

CPF/CNPJ* Data Nascimento* Sexo
Selecione uma opção ▼

Telefone Celular

CEP* Endereço

Bairro Numero* Complemento

Cidade Estado
Selecione uma opção ▼

< ANTERIOR PRÓXIMO >

BemServido © 2016 More Links

Figura 15 – Tela de cadastro de Profissional Segundo Passo

No segundo passo do cadastro, o profissional deve preencher suas informações básicas, não podendo deixar em branco os dados assinalados com asterisco (*). O profissional pode se cadastrar informando um CPF ou CNPJ, e deve preencher o cadastro com informações válidas, já que o sistema irá consistir as informações.

Terceiro passo:

Bem Servido [Crie sua agenda Profissional](#) [Cadastre-se](#) [Login](#)

CADASTRO PROFISSIONAL

Ramo de Serviço*
Beleza, Saúde e Estética

Cabelereiro

	Valor*	Duração (min)*
<input checked="" type="checkbox"/> Corte masculino	R\$ 25,00	20
<input type="checkbox"/> Corte feminino	R\$ 0,00	
<input type="checkbox"/> Escova	R\$ 0,00	

Manicure

[< ANTERIOR](#) [PRÓXIMO >](#)

BemServido © 2016 [More Links](#)

Figura 16 – Tela de cadastro de Profissional Terceiro Passo

Nesse passo, o usuário poderá cadastrar os seus dados no sistema para a disponibilização da sua agenda de horários, devendo selecionar o ramo de especialidade no qual oferece serviço e ademais, cadastrar os dias da semana e horários disponíveis para agendamento.

Bem Servido [Crie sua agenda Profissional!](#) [Cadastre-se](#) [Login](#)

CADASTRO PROFISSIONAL

	Horario Inicio*	Horario Fim*
<input type="checkbox"/> Domingo		
<input checked="" type="checkbox"/> Segunda	08:00	18:00
<input type="checkbox"/> Terça		
<input checked="" type="checkbox"/> Quarta	08:00	18:00
<input type="checkbox"/> Quinta		
<input checked="" type="checkbox"/> Sexta	08:00	18:00
<input type="checkbox"/> Sábado		

[< ANTERIOR](#) [CONCLUIR >](#)

BemServido © 2016 [More Links](#)

Figura 17 – Tela de cadastro de Profissional Último Passo

Na imagem acima, foi demonstrada a criação de uma agenda profissional.

Caso algum campo obrigatório não seja preenchido, ou seja, preenchido de forma incorreta ou inválida, o sistema exibe mensagem de erro, que pode ser visualizada na figura a seguir.

Bem Servido [Crie sua agenda Profissional!](#) [Cadastre-se](#) [Login](#)

CADASTRO PROFISSIONAL

Nome Completo*

CPF/CNPJ* Data Nascimento* Sexo
Selecione uma opção ▼

Telefone Celular

CEP*

Bairro

Cidade

Ocorreu um erro em seu cadastro!

- Os campos obrigatórios estão todos identificados(*), preencha para continuar com seu cadastro.

OK

BemServido © 2016 [More Links](#)

Figura 18 – Tela de cadastro de Profissional Mensagem Erro

4.1.2.2 Cadastro - Cliente – Cadastre-se

The image shows a web interface for client registration. At the top, there is an orange header with the text 'Bem Servido' on the left and three buttons: 'Crie sua agenda Profissional!', 'Cadastre-se', and 'Login'. Below the header, the title 'CADASTRO CLIENTES' is centered. The main content is a registration form with the following fields:

- Nome Completo* (text input)
- CPF* (text input)
- Data Nascimento* (text input)
- Sexo (dropdown menu with 'Selecione uma opção' and a downward arrow)
- Telefone (text input)
- Celular (text input)
- CEP* (text input)
- Endereço (text input)
- Bairro (text input)
- Número* (text input)
- Complemento (text input)
- Cidade (text input)
- Estado (dropdown menu with 'Selecione uma opção' and a downward arrow)
- E-mail* (text input)
- Senha* (text input)
- Confirmar Senha* (text input)

At the bottom left of the form, there is a link: 'Já é cadastrado | Login'. At the bottom right, there is an orange button labeled 'CONCLUIR'. The footer of the page is orange and contains 'BemServido © 2016' on the left and 'More Links' on the right.

Figura 19 – Tela de Cadastro - Cliente

Na figura acima, pode-se ver a interface de “Cadastro” de clientes. Nessa tela, o usuário poderá cadastrar os seus dados básicos no sistema para que tenha acesso às funcionalidades de cliente, não podendo deixar em branco os dados assinalados com asterisco (*).

Caso algum campo obrigatório não seja preenchido, ou seja, preenchido de forma incorreta ou inválida, o sistema também exibe mensagem de erro conforme figura 20.

Bem Servido [Crie sua agenda Profissional!](#) [Cadastre-se](#) [Login](#)

CADASTRO CLIENTES

Nome Completo*
LAYLA CRISTINE DE OLIVEIRA CODOGNO

CPF* 092.660.269-13 Data Nascimento* 29/11/2016 Sexo Feminino ▼

Telefone (41) 8437-3293 Celular (41) 8437-3293

CEP* 82640


Bairro Santa

Cidade Curitiba ▼

E-mail* layla.codogno@gmail.com

Senha* *** Confirmar Senha* ***

[Já é cadastrado | Login](#) [CONCLUIR](#)



Ocorreu um erro em seu cadastro!

- Um usuário com esse email já existe ou seu email não é válido, por favor verifique seus dados e tente novamente.

[OK](#)

BemServido © 2016 [More Links](#)

Figura 20 – Tela de cadastro de Cliente Mensagem Erro

Caso o usuário já possua um cadastro e tente se cadastrar, é exibida a mensagem de erro demonstrada na figura abaixo.

The image shows a registration form with a modal error message. The form fields are: CPF* (075.938.099-61), Data Nascimento* (01/11/2016), Sexo (dropdown: Selezione uma opção), Telefone, Celular, CEP* (82840-160), Bairro (Bairro Alto), Cidade (Curitiba), E-mail* (testedoído@...), Senha* (masked with dots), and Confirmar Senha* (masked with dots). A modal window with a red 'X' icon displays the error: "Ocorreu um erro em seu cadastro!" followed by "- Já existe um cliente cadastrado com este CPF!". An "OK" button is at the bottom of the modal. The form has a "CONCLUIR" button at the bottom right and a "Já é cadastrado | Login" link at the bottom left. The footer contains "BemServido © 2016" and "More Links".

Figura 21 – Erro e-mail já cadastrado

4.1.3 Perfil do profissional

The screenshot displays the 'Bem Servido' professional agenda interface. At the top, there is an orange header with the text 'Bem Servido'. Below the header, the page is titled 'MINHA AGENDA'. A date input field is set to '28/11/2016', and a 'NOVO AGENDAMENTO' button is visible. The main content area is a calendar grid titled 'AGENDAMENTO' showing a slot for 'Marcelo' from 09:40 to 10:10 for 'Unha'. The footer contains 'BemServido © 2016' and 'More Links'. A navigation menu on the right includes options like 'Meus agendamentos', 'Minhas habilidades', 'Meus serviços', 'Meus horários', 'Meu Perfil', and 'Logout'.

Figura 22 – Tela Agenda Profissional

Já autenticado no sistema, caso o usuário seja um profissional, o sistema o direciona para a página inicial representada na figura 20.

Nessa página, o usuário é apto a realizar as ações listadas a seguir:

- Filtrar agendamentos por dia
- Cadastrar um novo agendamento em sua agenda

Essas funcionalidades estão descritas no apêndice E – Especificações de Caso de uso.

As opções de menu apresentadas no topo da página, disponibilizam ao usuário visualizar seus dados cadastrais, explicitados nos tópicos 4.1.4.1 até 4.4.4.4.

4.1.3.1 Minha Agenda (profissional)

The screenshot displays the 'Minha Agenda (Professional)' interface. At the top, there is an orange header bar with the text 'Bem Servido' on the left and 'Agatha' on the right. Below this, the title 'MINHA AGENDA' is centered. Underneath the title, there is a 'Data' label and a text input field containing '28/11/2016'. To the right of the date field is a button labeled 'NOVO AGENDAMENTO +1'. The main content area is a calendar grid titled 'AGENDAMENTO' with a legend for 'HORÁRIO MARCADO'. The grid shows a vertical list of times from 08:00 to 16:30 in 30-minute increments. A single appointment is visible, represented by a brown bar spanning from 09:40 to 10:10. The appointment is labeled 'Marcelo' on the left and 'Unha' on the right. The footer of the interface is an orange bar containing 'BemServido © 2016' on the left and 'More Links' on the right.

Figura 23 – Minha Agenda (Profissional)

Nesta interface o profissional consegue visualizar seus agendamentos podendo filtra-los por dia. Clicando em novo agendamento, uma modal é apresentada na tela para a adição das informações e conclusão da ação conforme figura 24.

Bem Servido Agatha

MINHA AGENDA

Data: 28/11/2016

AGENDAMENTO

Cliente*

Telefone de Contato do Cliente

Data* 28/11/2016

Serviço*
Escolha um Serviço

Hora Início* Hora Fim 00:00

Valor: R\$ 0,00 Duração estimada: 0 minutos

CONFIRMAR

BemServido © 2016 More Links

Figura 24 – Tela Novo Agendamento

4.1.3.2 Minhas Habilidades

Tela em que o usuário pode editar ou excluir suas habilidades informadas no momento do cadastro.

Bem Servido Agatha

MINHAS HABILIDADES

NOVO

HABILIDADES		
Nome	Ações	
Cabelereiro	EDITAR	EXCLUIR
Manicure	EDITAR	EXCLUIR

BemServido © 2016 More Links

Figura 25 – Tela Minhas Habilidades

4.1.3.3 Meus Serviços

Tela em que o usuário pode editar ou excluir os serviços cadastrados.

The screenshot shows the 'Meus Serviços' page. At the top, there's a header with 'Bem Servido' and the user name 'Agatha'. Below the header, the page title is 'MEUS SERVIÇOS'. A 'NOVO' button is located in the top right corner. The main content is a table with the following data:

Nome	Valor	Duração	Ações	
Unha	R\$ 10,50	30 minutos	EDITAR	EXCLUIR
Unha + cutícula	R\$ 10,50	30 minutos	EDITAR	EXCLUIR
Corte feminino	R\$ 10,50	30 minutos	EDITAR	EXCLUIR

At the bottom of the page, there is a footer with 'BemServido © 2016' and a 'More Links' button.

Figura 26 – Tela Meus Serviços

4.1.3.4 Meu Perfil (profissional)

Tela em que o usuário pode editar os dados de sua conta e perfil.

The screenshot shows the 'Meu Perfil' page. At the top, there's a header with 'Bem Servido' and the user name 'Marcelo'. Below the header, the page title is 'MEU PERFIL'. The user name 'Marcelo' is displayed in a large font. The main content is divided into three sections, each with an 'EDITAR' button:

- MEUS DADOS:**
 - Nome: Marcelo
 - CPF: 075.938.099-61
 - Data Nascimento: 03/10/1975
 - Sexo: Masculino
 - Telefone: (41) 3434-3232
 - Celular: (41) 98828-4545
- ENDEREÇO:**
 - CEP: 82620-380
 - Endereço: Praça Hedwíges Nadolny, 843 - Santa Cândida, Curitiba - PR
- LOGIN:**
 - Email: teste1@gmail.com
 - Password: ****

At the bottom of the page, there is a footer with 'BemServido © 2016' and a 'More Links' button.

Figura 27 - Tela Meu Perfil (profissional)

4.1.4 Perfil do Cliente

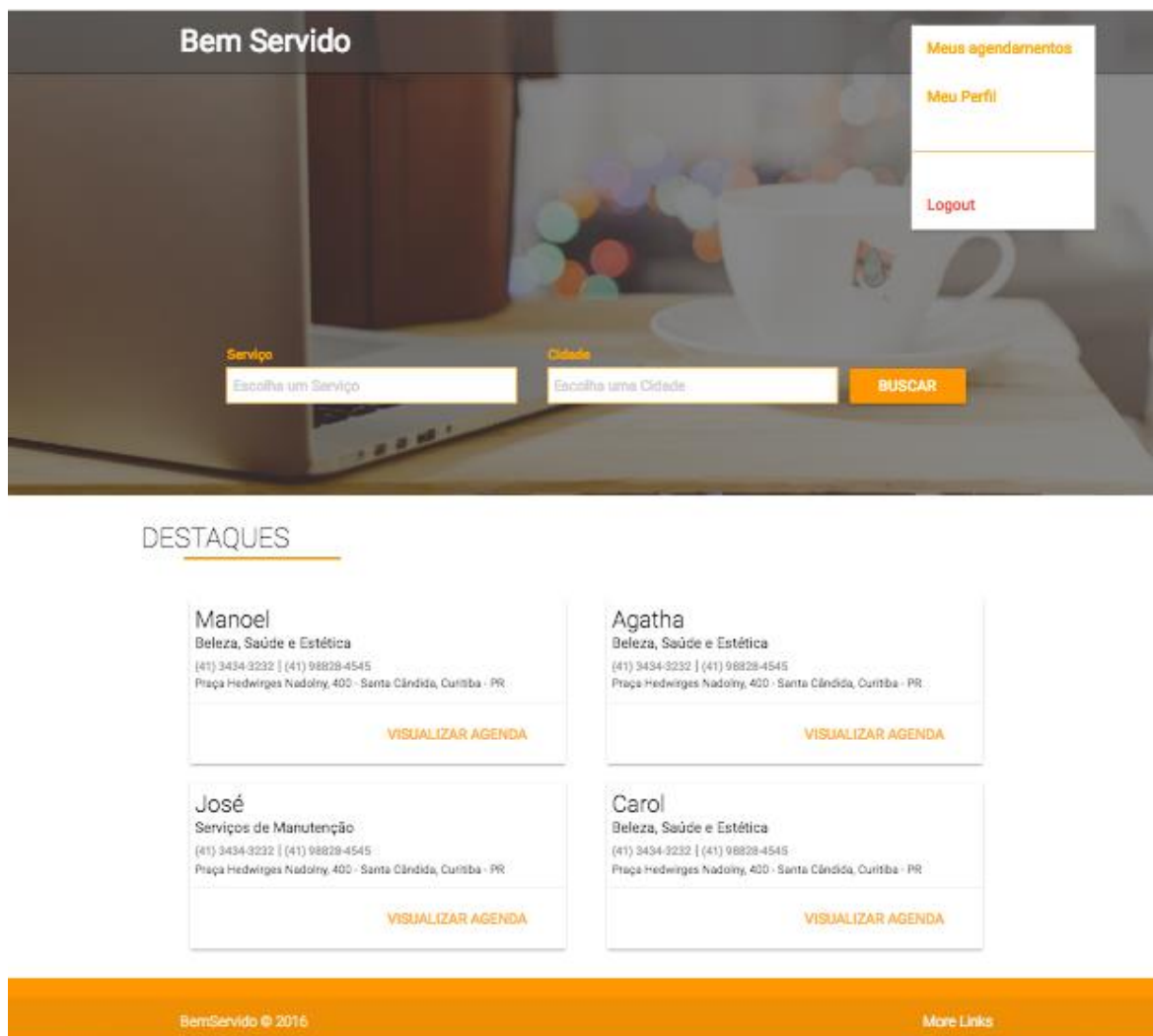


Figura 28 – Perfil Cliente

Após a autenticação do usuário com perfil de cliente, o sistema o direciona para a página principal representada no tópico 4.1.2, figura 10.

Autenticado no sistema, o usuário é apto a realizar as ações listadas a seguir:

- Meu Perfil (Edição dos dados cadastrais)
- Meus agendamentos

Essas funcionalidades estão descritas no apêndice E – Especificações de Caso de uso.

As opções de menu apresentadas no topo da página, disponibilizam ao usuário visualizar seus agendamentos e editar os dados cadastrais.

4.1.4.1 Meus Agendamentos (cliente)

The screenshot shows the 'Meus Agendamentos' page. At the top, there is a header with 'Bem Servido' and the user name 'Marcelo'. Below the header, the title 'MEUS AGENDAMENTOS' is displayed. There are two input fields for filtering: 'Data' with the value '28/11/2016' and 'Serviço' with the placeholder 'Escolha um Serviço'. A search button is located to the right of the service field. Below the search filters, a box titled 'AGENDAMENTOS' contains a table with one appointment entry. The table has two columns: 'Data' and 'Unha'. The appointment details are as follows:

Segunda, 28 de Novembro	Unha
Agatha Belaiza, Saúde e Estética Praça Hedwiges Nádolny, 400 - Santa Cândida, Curitiba - PR (41) 3434-3232 (41) 98828-4545	Serviço: Unha Horário: 09:40 - 10:10 Duração média: 30 minutos Valor: R\$ 10,00

At the bottom of the page, there is a footer with 'BemServido © 2016' and a 'More Links' button.

Figura 29 – Tela Meus Agendamentos

Essa tela possibilita que o usuário visualize todos os seus agendamentos, podendo filtrar as informações por data ou serviço.

As informações referentes ao agendamento são exibidas em um grid, contendo em seu topo a data e o serviço. Abaixo, os detalhes do profissional e horário marcado.

4.1.4.2 Meu Perfil (cliente)

Bem Servido Marcelo

MEU PERFIL

Marcelo

MEUS DADOS Nome: Marcelo CPF: 075.938.099-61 Data Nascimento: 03/10/1975 Sexo: Masculino Telefone: (41) 3434-3232 Celular: (41) 98828-4545 EDITAR	ENDEREÇO CEP: 82620-380 Endereço: Praça Hedwíges Nadolny, 843 - Santa Cândida, Curitiba - PR EDITAR	LOGIN Email: teste1@gmail.com Password: ***** EDITAR EMAIL EDITAR SENHA
---	--	---

BemServido © 2016 More Links

Figura 30 – Tela Meu Perfil

Caso o usuário queira alterar alguma informação referente ao cadastro, deve acionar a opção “Editar” no rodapé de cada grid de informações, onde deverá preencher novamente o campo a ser modificado.

4.1.5 Agenda do Profissional

Bem Servido Marcelo

AGENDA

Carol
Beleza, Saúde e Estética
Praça Hedwíges Nadolny, 450 - Santa Cândida, Curitiba - PR
(41) 3434-3232 | (41) 98828-4545

Data: AGENDAR SERVIÇO

SERVIÇOS

Corte masculino	R\$ 10,50 30 minutos
Corte feminino	R\$ 10,50 30 minutos
Escova	R\$ 10,50 30 minutos

HORÁRIOS

■ HORÁRIO INDESPONÍVEL

- 08:00
- 08:30
- 09:00
- 09:30
- 10:00
- 10:30
- 11:00
- 11:30
- 12:00
- 12:30
- 13:00
- 13:30
- 14:00
- 14:30
- 15:00
- 15:30
- 16:00
- 16:30

Figura 31 – Tela Agenda do Profissional

Ao acessar a tela de agenda, são exibidas todas as informações relacionadas ao profissional, assim como todos os serviços prestados e os horários que o mesmo disponibilizou em sua agenda.

4.1.6 Tela de Emissão de Relatórios

Bem Servido Crie sua agenda Profissional! Cadastre-se Login

RELATÓRIOS

Tipo Relatório
Selecione uma opção ▼

RELATÓRIO ANUAL

Data	Horario	Serviço	Cliente
28/11/2016	10:30	Corte de Cabelo Feminino	Suzana Vieira
28/11/2016	11:00	Corte de Cabelo Infantil	Suzana Vieira
28/11/2016	14:00	Tintura e aplicação de Máscaras	Joana Oliveira

BemServido © 2016 More Links

Figura 32 – Tela Emissão de Relatórios

Nessa tela, o profissional poderá emitir relatórios sobre os agendamentos realizados nos últimos 12 meses. Pode-se observar a tela de “Relatórios” na figura acima.

Para a emissão do relatório, pode-se filtrar as informações pelo período de tempo do qual deseja-se visualizar os dados.

The screenshot displays the 'Bem Servido' web application interface. At the top, there is an orange navigation bar with the text 'Bem Servido' on the left and three buttons: 'Crie sua agenda Profissional', 'Cadastre-se', and 'Login'. Below the navigation bar, the main content area is titled 'RELATÓRIOS'. Under this title, there is a section for 'Tipo Relatório' with a dropdown menu currently set to 'Mensal'. The main content area features a box titled 'RELATÓRIO MENSAL' containing a table with the following data:

Data	Horario	Serviço	Cliente
28/11/2016	10:30	Corte de Cabelo Feminino	Suzana Vieira
28/11/2016	11:00	Corte de Cabelo Infantil	Suzana Vieira
28/11/2016	14:00	Tintura e aplicação de Máscaras	Joana Oliveira

At the bottom of the page, there is an orange footer bar containing the text 'BemServido © 2016' on the left and 'More Links' on the right.

Figura 33 – Tela Emissão de Relatório por Período

4.2 CONSIDERAÇÕES FINAIS

O projeto foi implementado com o intuito de criar uma *rede* de serviços, tendo a finalidade básica de agilizar a contratação de serviços sem que haja a necessidade de um contato verbal. O Bem-Servido visa agilizar o agendamento de horário entre clientes e profissionais de diversos ramos e especialidades. Durante o projeto foi implementado um sistema web, que possibilita que sejam cadastrados profissionais e clientes, permitindo que profissionais realizem uma breve descrição sobre sua especialidade e disponibilize sua agenda de horários, para que qualquer usuário visualize seu anúncio e clientes cadastrados agendem um futuro atendimento. O profissional pode, ainda, emitir relatórios dentro do sistema.

A experiência na construção do software trouxe grande aprendizado para a equipe. Entretanto, o escopo idealizado para o projeto ficou amplo e o tempo calculado para o desenvolvimento excedeu o limite para que fosse finalizado no tempo de um semestre acadêmico, tornando necessário reprimir uma parte dos requisitos. Os requisitos retirados do escopo foram citados como funcionalidades não contempladas no tópico 4.4.

Embora tenham havido dificuldades durante o desenvolvimento, o Bem-Servido contribuiu para o aprendizado da equipe em gerenciamento de projeto e resultou em uma boa prática sobre as tecnologias utilizadas.

4.3 DIFICULDADES

A maior dificuldade enfrentada pelos membros da equipe foi o curto prazo para o aprendizado da tecnologia escolhida e a implementação de todos os requisitos definidos no escopo do projeto, além da conciliação de horários para as reuniões, vida acadêmica e profissional.

Durante o desenvolvimento do projeto, ocorreram divergências que resultaram na saída de dois integrantes da equipe, o que gerou atraso nas entregas e redistribuição das responsabilidades entre os que permaneceram no projeto, tornando necessária a alteração do cronograma e redefinição de prioridades no desenvolvimento.

A nova estruturação da equipe fez com que cada membro focasse em uma função: desenvolver o sistema ou documentá-lo, para que dessa forma as habilidades de cada um fossem melhor exploradas e aumentassem as chances da conclusão do escopo completo.

4.4 FUNCIONALIDADES NÃO CONTEMPLADAS

Algumas funcionalidades que poderão ser acrescentadas futuramente ou que foram retiradas do escopo inicial são:

- Implementação de um aplicativo para dispositivos móveis (RNF6)

O sistema poderá ser melhor explorado em aplicativos móveis e deve ser desenvolvido para as plataformas Android e IOS.

- Adicionar empresa como favorita (RF22)

Essa funcionalidade permitirá que o cliente possua uma lista de empresas favoritas para que a busca pelos profissionais que mais frequente seja agilizada.

- Adicionar amigos (R23)

A amizade será implementada para vincular clientes uns com os outros, e que exista maior interação entre os usuários.

- Sugerir empresa para amigos (RF24)

A funcionalidade de sugestão de empresas possibilitará que os clientes indiquem a seus amigos os seus profissionais favoritos.

- Troca de mensagens (RF25)

Os usuários poderão se comunicar por meio de mensagens para saber detalhes sobre o serviço ou tirar dúvidas sobre o agendamento.

- Perfil empresarial (RF5)

Empresas poderão criar um perfil e vincular os perfis de seus colaboradores, cadastrando as especialidades de cada um. Podendo incluir também um perfil intermediador, que não terá acesso a todas as funcionalidades, mas poderá agendar horários no perfil dos colaboradores e vincula-los ou cancelar o vínculo com a empresa.

- Vinculação de perfis (RF13)

Os perfis poderão ser vinculados, para facilitar a identificação de usuários que interagem entre si mais frequentemente.

- Integração com calendários (RF21)

Integração do sistema com Google *Calendar* e calendário do Outlook, de maneira que ao ser criado um agendamento, um evento será criado nos calendários citados.

REFERÊNCIAS

ABERNETHY, M. *Just what is Node.js?* **IBM developerWorks**, 2016.

BARRETO JUNIOR, C. G. **Agregando frameworks de infra-estrutura em uma arquitetura baseada em componentes**: um estudo de caso no ambiente aulanet. PUC-RJ. Rio de Janeiro, 2006. Disponível em: <http://www.maxwell.vrac.puc-rio.br/8623/8623_3.PDF> Acesso em: maio, 2016.

BARROS, P. V.; ALMEIDA, I.R.; D'Emery, R. A. SCRUM: Uma metodologia ágil para projetos WEB com pequenas equipes. UFRPE, Universidade Federal Rural de Pernambuco, 2010.

BASTIEN, C. & SCAPIN, D. L. Evaluating a user interface with ergonomic criteria. **International Journal of Human-Computer Interaction**, 7, 1995, 105-121.

BROD, C. **Scrum**: Guia prático para projetos ágeis. Editora Novatec, 2013

BELL, Peter; BEER, Brent. **Introdução ao GitHub**. Tradução Lucia A Kinoshita. São Paulo: Novatec, 2015.

CANAL TECH. **O que é API?** Disponível em:< <https://canaltech.com.br/o-que-e/software/o-que-e-api/>>. Acesso em: 15 nov. 2016.

CIBYS, W. A.; PIMENTA, M. S.; SILVEIRA, M. C.; GAMEZ, L. **Uma abordagem ergonômica para o desenvolvimento de sistemas interativos**. Universidade Federal de Santa Catarina- UFSC. Florianópolis, 2000.

FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. Dissertação de Doutorado - University of California, Irvine, 2000. Disponível em: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm. Acesso em: 15 nov. 2016.

FINKELSTEIN, A.; KRAMER J. **Software Engineering**: A Roadmap. In: CONFERENCE ON SOFTWARE ENGINEERING, 2000. Proceedings... [S. I.]: ACM Press, 2004.

FRAIN, B. **Responsive web design with HTML5 and CSS3**. Packt Publishing Ltd, 2012.

GUDWIN, Ricardo R. **Engenharia de software: uma visão prática**. 2 ed. Campinas: UNICAMP, 2015.

IHRIG, Colin J. **Pro Node.js para desenvolvedores**. Rio de Janeiro: Ciência Moderna, 2014.

KALIN, Martin. **Java Web Services: uma introdução rápida, prática e completa**. Rio de Janeiro: Alta Books, 2010.

LIPSMAN, Andrew. **Google Dominates Internet Landscape in India and Brazil**. Press-release publicado em 14 Set. 2009.

LUQUE, L.; VERISCIMO, E.; PEREIRA, G. **Entendendo e Estendendo o Framework MVVM Android Binding**: entenda o funcionamento do framework MVVM Android Binding, conheça suas principais funcionalidades e saiba como estendê-lo e adaptá-lo às suas necessidades. Disponível em: <http://www.univale.com.br/unisite/mundo-j/artigos/57_Androidbinding.pdf> Acesso em: maio 2016.

MERCADO LIVRE. **Sobre o Mercado Livre**. Disponível em: <<http://www.mercadolivre.com.br/institucional>>. Acesso em: 15 nov. 2016.

MOURA, R. A. **Kanban: A simplicidade do controle da produção**. São Paulo: IMAM, 1996.

MORAN, J. M., MASETTO, M. T. & BEHRENS, M. A. **Novas tecnologias e mediação pedagógica**. 6 ed. Campinas, SP: Papirus, 2003.

MOZILLA DEVELOPER NETWORK. **JavaScript**. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em maio de 2016.

NANNI, H. C.; CANETE, K. V. S. **A importância das redes sociais como vantagem competitiva nos negócios corporativos**. VII CONVIBRA Administração. São Paulo, 2010.

NETO, Guilherme. **Redes Sociais são responsáveis por 62% do tráfego na Internet**. Em matéria publicada dia 04/09/2009, no site Mundo do Marketing, disponível em: <<http://www.mundodomarketing.com.br/10,11077,redes-sociais-sao-responsaveis-por-62-do-trafego-nainternet-brasileira.htm>>. Acesso: maio 2016.

SABESIM. **Sistema Kanban**: aplicando o Kanban no dia a dia. Disponível em: <<http://www.sabesim.com.br/sistema-kanban-no-dia-a-dia/>>. Acesso em: maio 2016.

SILVA, R. L. **Tecnologia Wireless**. Ministério da Educação. Brasília: ME/SEED/DITEC, CETE, 2008.

SILVA, Maurício Samy. **JavaScript**: guia do programador. São Paulo: Novatec Editora, 2010.

SOMERA, Guilherme. **Treinamento profissional em Java**: aprenda a programar nesta poderosa linguagem. São Paulo: Digerati Books, 2006.

SOMMERVILLE, I. **Engenharia de software**. Rio de Janeiro: Prentice-Hall, 2004.

SOUZA, E. F. S.; GONÇALVES, P. A. S. **Um mecanismo de autenticação baseado em ECDH para redes IEEE 802.11**. Recife-PE: UFPE, Centro de Informática, 2008.

STRADIOTI, P. R. **KnockoutJS e ASP.NET MVC**. Fev, 2014. Disponível em: <<https://paulors.wordpress.com/2014/02/09/knockoutjs-e-asp-net-mvc/>>. Acesso em: maio 2016.

TERUEL, Evandro Carlos. **Arquitetura de sistemas para web com Java utilizando design patterns e frameworks**. Rio de Janeiro: Ciência Moderna, 2012.

TURCHI, Sandra R. **Estratégias de marketing digital e e-commerce**. São Paulo: Atlas, 2012.

VALENTIM, R. A. M.; SOUZA NETO, P. A. O impacto da utilização de Design Patterns nas métricas e estimativas de projetos de *software*: a utilização de padrões tem alguma influência nas estimativas? **Revista da FARN**. Natal/RN, Vol.4, n.1, jul. 2004.

VERÍSSIMO, F. **Segurança em redes sem fio**. Rio de Janeiro: COPPE – Instituto Alberto Luiz Coimbra, UFRJ, 2002.

GOSSMAN, J. Introduction to Model/View/ViewModel pattern for building WPF apps
Microsoft Blog. Out. 2005.

APÊNDICE A – REQUISITOS FUNCIONAIS - INTERFACE DO PROFISSIONAL

Interface: Profissional

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF1

Nome do Requisito: Cadastrar Profissional/Empresa

Prioridade: Alta

Descrição: O usuário poderá se cadastrar no sistema utilizando sua conta de e-mail e definindo uma senha de acesso.

Dependência: Não possui.

Histórico:

Interface: Profissional

Tipo de Requisito: Funcional (RF)

Código do Requisito: **Nome do Requisito:** Fazer *login*

Prioridade: Moderada

Descrição: O usuário poderá ter acesso às informações e funcionalidades do sistema, assim como alterar seus dados cadastrais e gerar relatórios.

Dependência: RF1

Histórico:

Interface: Profissional

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF3

Nome do Requisito: Alterar o Cadastro Profissional/Empresa

Prioridade: Alta

Descrição: Usuário deverá alterar a informação de cadastro desejada.

Dependência: RF1, RF2

Histórico:

Interface: Profissional

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF4

Nome do Requisito: Cadastrar Serviços

Prioridade: Alta

Descrição: O anunciante deverá cadastrar o tipo de serviço prestado e descrever brevemente sobre as atividades prestadas.

Dependência: RF1, RF2

Histórico:

Interface: Profissional

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF5

Nome do Requisito: Cadastrar Prestador de Serviço

Prioridade: Alta

Descrição: O anunciante deverá cadastrar os prestadores de serviço vinculados à empresa

Dependência: RF4

Histórico:

13/11/2016: O Requisito não será contemplado.

Interface: Profissional

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF6

Nome do Requisito: Gerar Relatório

Prioridade: Alta

Descrição: O profissional poderá gerar relatórios de horários agendados.

Dependência: RF2

Histórico:

Interface: Profissional

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF7

Nome do Requisito: Criar Agenda

Prioridade: Alta

Descrição: O profissional deverá informar em combo boxes quais dias da semana deseja agendar clientes, o intervalo entre os horários, horário de intervalo e os horários inicial e final a serem agendados.

Dependência: RF5

Histórico:

Interface: Profissional

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF8

Nome do Requisito: Alterar a agenda

Prioridade: Alta

Descrição: O profissional poderá alterar as configurações sobre a agenda.

Dependência: RF7

Histórico:

Interface: Profissional

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF10

Nome do Requisito: Agendar cliente

Prioridade: Média

Descrição: Para os casos em que o profissional não possui um sistema de agendamento e deseja agendar um cliente através do sistema, ele mesmo poderá agendar um cliente.

Dependência: RF7

Histórico:

Interface: Profissional

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF11

Nome do Requisito: Cancelar agendamento

Prioridade: Alta

Descrição: O profissional poderá cancelar o agendamento de um cliente.

Dependência: RF1, RF2

Histórico:

Interface: Empresa

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF12

Nome do Requisito: Fechar agenda

Prioridade: Média

Descrição: O profissional poderá fechar os agendamentos em um dia específico, nesse caso nenhum outro cliente poderá agendar horário no dia especificado.

Dependência: RF7

Histórico:

Interface: Profissional

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF13

Nome do Requisito: Solicitar Parceria

Prioridade: Alta

Descrição: Uma empresa envia solicitação de parceria à outra empresa, a qual deve aceitar ou recusar o pedido. Ao aceitar, uma empresa aparecerá no perfil da outra.

Dependência: RF1, RF2

Histórico:

13/11/2016: O Requisito não será contemplado.

APÊNDICE B – REQUISITOS FUNCIONAIS - INTERFACE DO CLIENTE

Interface: Cliente

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF14

Nome do Requisito: Cadastrar Cliente

Prioridade: Alta

Descrição: O usuário poderá se cadastrar no sistema utilizando sua conta no *gmail* ou *facebook*, ou se preferir, informando *e-mail*, senha e telefone para contato.

Dependência: Não possui.

Histórico:

Interface: Cliente

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF15

Nome do Requisito: Fazer *Login*

Prioridade: Alta

Descrição: Ao autenticar-se no sistema, o usuário poderá ter acesso às funcionalidades do sistema, como agendar horário com um prestador de serviço.

Dependência: RF14

Histórico:

Interface: Cliente

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF16

Nome do Requisito: Alterar Cadastro Cliente

Prioridade: Alta

Descrição: Usuário deverá informar qual dado do cadastro deseja alterar.

Dependência: RF15

Histórico:

Interface: Cliente

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF17

Nome do Requisito: Pesquisar Serviço ou Profissional/Empresa

Prioridade: Alta

Descrição: Usuário, mesmo não autenticado, poderá pesquisar uma empresa ou serviço, filtrando a busca por região e podendo ordenar por melhor avaliados.

Dependência: Não possui.

Histórico:

Interface: Cliente

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF18

Nome do Requisito: Solicitar Agendamento

Prioridade: Alta

Descrição: O usuário irá até o perfil do profissional escolhido, visualizará a agenda e poderá escolher o dia e horário desejados.

Dependência: RF15

Histórico:

Interface: Cliente

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF19

Nome do Requisito: Cancelar agendamento

Prioridade: Alta

Descrição: O cliente, na sua própria interface poderá cancelar um agendamento.

Dependência: RF15 e RF18 ou RF10

Histórico:

Interface: Cliente

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF20

Nome do Requisito: Avaliar serviço

Prioridade: Alta

Descrição: Após a data do agendamento, será possível avaliar o serviço – em escala de 1 a 5 - prestado pelo profissional com quem agendou um horário.

Dependência: RF15 e RF18 ou RF10

Histórico:

Interface: Cliente

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF21

Nome do Requisito: Integração com calendários

Prioridade: Média

Descrição: Integração do sistema com *Google Calendar* e calendário do *Outlook*, de maneira que ao ser criado um agendamento, um evento será criado nos calendários citados.

Dependência: Não há

Histórico:

13/12/2016: O requisito não será contemplado.

Interface: Cliente

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF22

Nome do Requisito: Adicionar empresa como favorita

Prioridade: Baixa

Descrição: O cliente será apto a adicionar uma empresa como favorita, para que a busca pelos profissionais que mais frequente seja agilizada.

Dependência: Não há

Histórico:

13/12/2016: O requisito não será contemplado.

Interface: Cliente

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF23

Nome do Requisito: Adicionar amigos

Prioridade: Baixa

Descrição: A amizade será implementada para vincular clientes uns com os outros, assim existirá maior interação entre os usuários.

Dependência: Não há

Histórico:

13/12/2016: O requisito não será contemplado.

Interface: Cliente

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF24

Nome do Requisito: Sugestão de empresas

Prioridade: Baixa

Descrição: A funcionalidade de sugestão de empresas possibilitará que os clientes indiquem profissionais a seus amigos.

Dependência: Não há

Histórico:

13/12/2016: O requisito não será contemplado.

Interface: Cliente e Profissional

Tipo de Requisito: Funcional (RF)

Código do Requisito: RF25

Nome do Requisito: Troca de mensagens

Prioridade: Baixa

Descrição: Os usuários poderão se comunicar por meio de mensagens.

Dependência: Não há

Histórico:

13/12/2016: O requisito não será contemplado.

APÊNDICE C – REQUISITOS ORGANIZACIONAIS

Tipo de Requisito: Não-Funcional (RNF)

Código do Requisito: RNF1

Nome do Requisito: Linguagem de Programação

Prioridade: Alta

Descrição: O sistema será desenvolvido na linguagem de programação *Javascript* utilizando o interpretador de código NODE.

Dependência: Não há.

Histórico:

Tipo de Requisito: Não-Funcional (RNF)

Código do Requisito: RNF2

Nome do Requisito: SGBD

Prioridade: Alta

Descrição: O SGBD utilizado será o *Mysql*.

Dependência: Não há.

Histórico:

Tipo de Requisito: Não-Funcional (RNF)

Código do Requisito: RNF3

Nome do Requisito: Utilizar padrão MVC

Prioridade: Alta

Descrição: O software será desenvolvido no padrão MVC (*Model-View-Controller*).

Dependência: Não há.

Histórico:

Tipo de Requisito: Não-Funcional (RNF)

Código do Requisito: RNF4

Nome do Requisito: Senha Criptografada

Prioridade: Alta

Descrição: A senha será criptografada para que apenas o usuário tenha acesso à informação.

Dependência: Não há.

Histórico:

Tipo de Requisito: Não-Funcional (RNF)

Código do Requisito: RNF5

Nome do Requisito: Validação dos campos

Prioridade: Alta

Descrição: As informações inseridas nos formulários serão validadas na interface da empresa e também do usuário.

Dependência: Não há.

Histórico:

Tipo de Requisito: Não-Funcional (RNF)

Código do Requisito: RNF6

Nome do Requisito: Desenvolvimento de aplicativos móveis

Prioridade: Média

Descrição: Será desenvolvido um aplicativo para as plataformas android e IOS.

Dependência: Não há.

Histórico:

12/11/2016: O requisito não será contemplado.

APÊNDICE D – CASOS DE USO

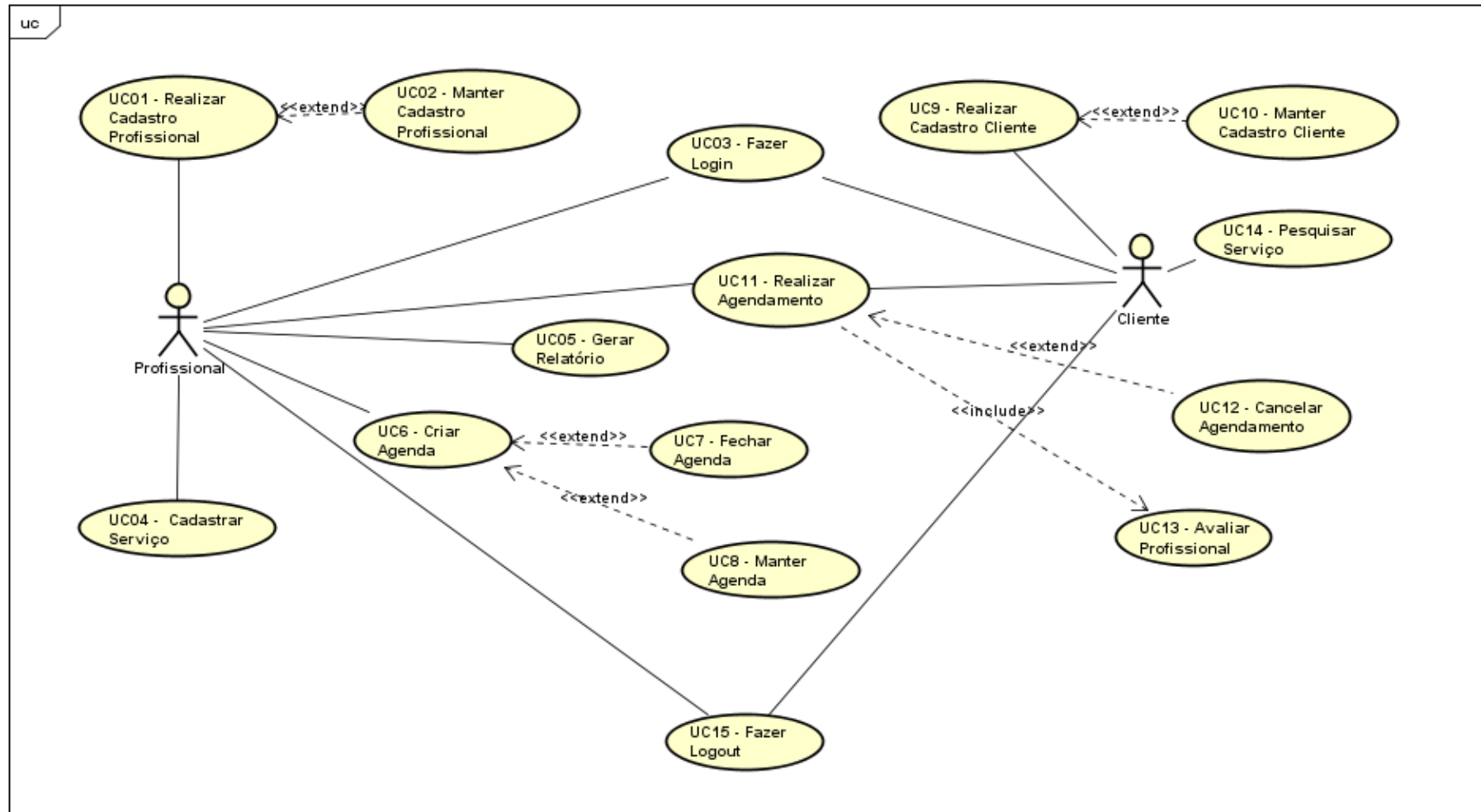


Figura 34 – Casos de Uso do Sistema

APÊNDICE E – ESPECIFICAÇÕES DE CASOS DE USO

Abaixo seguem as especificações de cada caso de uso (UC) representado no diagrama de caso de uso (Apêndice A).

Para cada UC, serão especificadas as pré-condições para que o caso se inicie, pós-condições que ocorrerão quando o caso for finalizado, o ator principal, e os fluxos principal, alternativo e de exceção. Eventualmente, será apresentada a tela desenvolvida para o caso e será descrita a regra de negócio quando houver.

UC01 – Realizar Cadastro Profissional

Descrição: Este caso de uso serve para que o profissional realize cadastro no sistema.

Data View:

DV1 – Tela Cadastro Profissional

Bem Servido

[Crie sua agenda Profissional](#)
[Agenda](#)
[Cadastro](#)
[Login](#)

CADASTRO PROFISSIONAL

Selecione uma opção ▼

PRÓXIMO ▼

Escolha seu Ramo ▼

PRÓXIMO ▼

Cabeleireiro
 Manicure

Segunda

Terça

Quarta

Terça

Quarta

Footer Content

You can use rows and columns here to organize your footer content.

Links

[Link 1](#)

[Link 2](#)

[Link 3](#)

[Link 4](#)

Figura 35 - DV1 – Tela Cadastro Profissional

Pré-condições

Não há.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Realizar cadastro do Profissional.

Ator Principal

Profissional

Fluxo de Eventos Principal

1. O sistema apresenta a “Tela Principal - Pesquisa” (DV9) para o usuário.
2. O usuário aciona o botão “Cadastre-se”
3. O sistema direciona o usuário para a “Página de Cadastro” (DV1)
4. O usuário preenche as informações solicitadas
5. O usuário informa senha vinculada ao perfil
6. O usuário clica no botão “Cadastrar” (E1) (E2)
7. O sistema exibe a mensagem de confirmação “Deseja confirmar o cadastro?”, apresentando as opções “sim/cancelar” (A1)
8. O usuário seleciona a opção “sim”
9. O sistema salva o cadastro e exibe a mensagem de ação realizada com sucesso
10. O caso de uso é encerrado

Fluxos Alternativos

A01. Usuário seleciona a opção “cancelar”

1. Sistema retorna ao passo 3 do fluxo principal

Fluxos de Exceção

E01. Usuário informa dados inválidos:

1. O usuário informa dados inválidos ou deixa de preencher um campo obrigatório
2. O sistema exibe uma mensagem informando o erro
3. O sistema retorna ao passo de chamada da exceção

E02. *Login* já existe no sistema:

1. O sistema exibe uma mensagem informando o erro
2. O sistema retorna ao passo de chamada da exceção

Regra de Negócio

R1. O e-mail de acesso ao sistema deve ser único, dessa forma, o sistema não deve permitir o cadastro de usuários que possuam *login*/CPF/CNPJ já cadastrados.

R2. O sistema deve gravar as senhas criptografadas.

UC02 – Manter Cadastro Profissional

Descrição: Este caso de uso serve para que o usuário mantenha o cadastro do profissional no sistema.

Data View:

DV2 – Tela Área do Profissional

The screenshot displays a user profile page for 'Marcelo' on the 'Bem Servido' website. The page is titled 'MEU PERFIL' and 'Marcelo'. It features three main sections for profile management:

- MEUS DADOS:**
 - Nome: Marcelo
 - CPF: 075.938.099-61
 - Data Nascimento: 03/10/1975
 - Sexo: Masculino
 - Telefone: (41) 3434-3232
 - Celular: (41) 98828-4545
 - Button: EDITAR
- ENDEREÇO:**
 - CEP: 82620-380
 - Endereço: Praça Hedwiges Nadolny, 843 - Santa Cândida, Curitiba - PR
 - Button: EDITAR
- LOGIN:**
 - Email: teste1@gmail.com
 - Password: *****
 - Buttons: EDITAR EMAIL, EDITAR SENHA

The footer contains 'BemServido © 2016' and 'More Links'.

Figura 36 - Tela área do profissional

Pré-condições

Para que o caso de uso se inicie é necessário que tenha sido executado:

1. UC01 – Realizar Cadastro Profissional

Pós-condições

Após a finalização normal do caso de uso em questão, o sistema deve:

1. Manter Cadastro.

Ator Principal

Profissional.

Fluxo de Eventos Principal

1. O sistema apresenta a tela “Área do Profissional” (DV2)
2. O usuário seleciona a opção “Editar Perfil”
3. O sistema apresenta a tela “Sobre” com as informações do usuário autenticado
4. O usuário altera os campos desejados e clica no botão “Salvar”
5. O sistema apresenta a mensagem “Deseja salvar informações?”, apresentando as opções “sim/cancelar” (A1)(E1)
6. O usuário seleciona a opção “sim”
7. O sistema salva o cadastro e exibe a mensagem de ação realizada com sucesso
8. O caso de uso é encerrado

Fluxos Alternativos

A1. Usuário seleciona a opção “Não”.

1. O Sistema retorna para o passo de chamada.

Fluxos de Exceção

E01. Usuário preenche informação inválida

1. O usuário informa dados inválidos ou deixa de preencher um campo obrigatório
2. O sistema exibe uma mensagem informando o problema
3. O sistema retorna para o passo de chamada da exceção

Regra de Negócio

Não há.

UC03 – Fazer Login

Descrição: Esse caso de uso serve para que o usuário se autentique no sistema.

Data View:

DV3 – Tela Fazer Login

The screenshot shows a web interface for a login page. At the top, there is an orange header bar with the text "Bem Servido" on the left and navigation links "Criar sua agenda Profissional!", "Agenda", "Cadastro", and "Login" on the right. Below the header, the word "LOGIN" is centered and underlined. The main content area contains a login form with two input fields: "E-mail" and "Senha". Below the "Senha" field is a blue link that says "Esqueci minha senha". To the right of the form is an orange "LOGIN" button. At the bottom, there is an orange footer bar. On the left side of the footer, it says "Footer Content" and "You can use rows and columns here to organize your footer content." On the right side, it says "Links" followed by a list: "Link 1", "Link 2", "Link 3", and "Link 4".

Figura 37 - DV3 – Tela Fazer Login

Pré-condições

Para que o caso de uso se inicie é necessário que:

1. UC01 – Realizar Cadastro Profissional
2. UC13 – Realizar Cadastro Cliente

Pós-condições

Após o fim normal do caso de uso em questão, o sistema deve:

1. Permitir que o usuário realize *login* no sistema.

Ator Principal

Todos

Fluxo de Eventos Principal

1. O sistema apresenta a “Tela Principal” (DV11)
2. O usuário seleciona a opção “*Login*”
3. O sistema apresenta a tela “Realizar *Login*” (DV3) com os campos “usuário” e “senha”
4. O usuário preenche as informações de usuário e senha
5. O usuário clica no botão “*Login*” (E01)
6. O sistema autentica o usuário
7. O sistema apresenta a tela do usuário (A1) (A2) (A3)
8. O caso de uso é encerrado

Fluxos Alternativos

A01. *Login* realizado pelo Profissional

1. O sistema autentica o usuário e apresenta a tela “Área do profissional” (DV2)
2. O caso de uso é encerrado.

A03. *Login* realizado pelo Cliente

1. O sistema autentica o usuário e apresenta a “tela principal” (DV9)
2. O caso de uso é encerrado.

Fluxos de Exceção

E01. Informação de usuário ou senha incorreta

1. O sistema exibe uma mensagem informando o problema
2. O sistema retorna para o passo de chamada da exceção

E02. Usuário não preenche campos obrigatórios

1. O sistema exibe uma mensagem informando o problema
2. O sistema retorna para o passo de chamada da exceção

Regra de Negócio

Não há.

UC04 – Cadastrar Serviço

Descrição: Este caso de uso serve para que o usuário possa cadastrar o ramo do serviço que presta, o que ocorre no momento do cadastro no sistema (DV1).

Pré-condições

Para que o caso de uso se inicie é necessário que tenha sido executado:

1. UC03 – Fazer *Login*

Pós-condições

Após o fim normal do caso de uso em questão, o sistema deve:

1. Permitir que o usuário visualize o serviço cadastrado.
2. Permitir que o usuário autenticado como faça alterações no serviço cadastrado.

Ator Principal

Profissional.

Fluxo de Eventos Principal

1. O sistema apresenta a tela “Área do Profissional” (DV2) para o usuário autenticado
2. O usuário clica na opção de “Serviços”
3. O sistema apresenta a tela “Serviços”
4. O usuário clica no botão “Cadastrar Novo”
5. O sistema habilita um campo para o usuário informar o serviço
6. O usuário informa o nome do serviço
 - 3.1. Ao passo que o usuário insere o nome do serviço, o sistema filtra os serviços já cadastrados com nomes semelhantes. (A01)
7. O usuário clica no botão “Salvar” (E01) (A02)
8. O sistema salva o cadastro e exibe a mensagem de ação realizada com sucesso
9. O caso de uso é encerrado

Fluxos Alternativos

A01. Serviço não cadastrado no sistema:

1. O usuário digita o serviço a ser cadastrado
2. O sistema retorna para o passo de chamada

A02. O usuário clica no botão “Cancelar”:

1. O sistema retorna ao passo 03 do fluxo principal

Fluxos de Exceção

E01. Usuário deixa de preencher campo obrigatório.

1. Mensagem informando a exceção é exibida

Regra de Negócio

R1. Quando o serviço não existir no cadastro do banco, o usuário pode cadastrar um novo.

UC05 – Gerar Relatório

Descrição: Este caso de uso serve para que o usuário possa gerar relatórios.

Data View:

DV4 – Tela Gerar Relatório

Bem Servido [Crie sua agenda Profissional!](#) [Cadastre-se](#) [Login](#)

RELATÓRIOS

Tipo Relatório
Selecione uma opção ▼

RELATÓRIO ANUAL

Data	Horario	Serviço	Cliente
28/11/2016	10:30	Corte de Cabelo Feminino	Suzana Vieira
28/11/2016	11:00	Corte de Cabelo Infantil	Suzana Vieira
28/11/2016	14:00	Tintura e aplicação de Máscaras	Joana Oliveira

BemServido © 2016 [More Links](#)

Figura 38 – Tela Emissão de Relatórios

Pré-condições

Para que o caso de uso se inicie é necessário que tenha sido executado:

1. UC03 – Fazer Login

Pós-condições

Após o fim normal do caso de uso em questão, o sistema deve:

1. Gerar relatórios

Ator Principal

Profissional

Fluxo de Eventos Principal

1. O sistema apresenta a tela “Área do Profissional” (DV2) para o usuário autenticado
2. O usuário clica na opção relatórios
3. O sistema apresenta a tela “Relatórios” (DV4)
4. O usuário seleciona a opção de relatório desejada
5. O sistema apresenta o relatório na tela e o botão “Exportar para PDF”
6. O usuário clica no botão “Exportar para PDF”
7. O sistema abre a janela para o usuário escolher o diretório em que deseja salvar
8. O usuário clica em “salvar” (E01)
9. O sistema gera o relatório no formato PDF
10. O caso de uso é encerrado

Fluxos Alternativos

Não há.

Fluxos de Exceção

E01. Erro na rede

1. O sistema apresenta a mensagem “Relatório não pode ser gerado”
2. O sistema retorna para o passo de exceção

Regra de Negócio

R1. O período máximo para consulta é de 12 meses.

UC06 – Criar Agenda

Descrição: Este caso de uso serve para que o usuário crie a agenda profissional, que ocorre no momento do cadastro no sistema (DV1).

Pré-condições

Para que o caso de uso se inicie é necessário que tenha sido executado:

1. UC03 – Fazer *Login*

Pós-condições

Após o fim normal do caso de uso em questão, o sistema deve:

1. Criar agenda

Ator Principal

Profissional

Fluxo de Eventos Principal

1. O sistema apresenta a tela “Agenda” (DV5)
2. O usuário clica no botão “Editar Agenda”
3. O sistema apresenta a tela “Editar Agenda” com os campos para criação da agenda
4. O usuário preenche todas as informações e clica no botão “salvar” (E01)
5. O sistema apresenta a mensagem de confirmação, com as opções “sim/cancelar”
6. O usuário seleciona a opção “sim” (A01)
7. O sistema salva o cadastro e apresenta a mensagem “agenda salva”
8. O caso de uso se encerra

Fluxos Alternativos

A01. O usuário seleciona a opção “cancelar”

1. O sistema retorna para o passo 3

Fluxos de Exceção

E01. Campo obrigatório não preenchido

1. O sistema exibe a mensagem de que é necessário preencher todos os campos obrigatórios, marcados com * (asterisco)
2. O sistema retorna para o passo de chamada

Regra de Negócio

Não há.

UC07 – Fechar Agenda

Descrição: Esse caso de uso serve para que o usuário possa fechar a agenda, dessa maneira não será possível agendar um horário na data e horário em questão.

Data View:

DV5 – Tela Agenda

The screenshot displays the 'Tela Agenda' interface. At the top, there is an orange header with 'Bem Servido' on the left and 'Marcelo' on the right. Below the header, the word 'AGENDA' is underlined. The main content area features the name 'Carol' in large font, followed by her business details: 'Beleza, Saúde e Estética', 'Praça Hedwiges Nadoirny, 450 - Santa Cândida, Curitiba - PR', and phone numbers '(41) 3434-3232 | (41) 98828-4545'. A date input field shows '27/11/2016' and an orange button labeled 'AGENDAR SERVIÇO'. Below this, a large white box with an orange border contains two sections: 'SERVIÇOS' and 'HORÁRIOS'. The 'SERVIÇOS' section lists three services: 'Corte masculino' (R\$ 10,50 | 30 minutos), 'Corte feminino' (R\$ 10,50 | 30 minutos), and 'Escova' (R\$ 10,50 | 30 minutos). The 'HORÁRIOS' section shows a vertical list of time slots from 08:00 to 16:30, with a legend for 'HORÁRIO INDEPONÍVEL'.

Figura 39 – Tela Agenda do Profissional

Pré-condições

Para que o caso de uso se inicie é necessário que tenha sido executado:

1. UC06 – Criar Agenda

Pós-condições

Após o fim normal do caso de uso em questão, o sistema deve:

1. Fechar agenda

Ator Principal

Profissional

Fluxo de Eventos Principal

1. O sistema apresenta a tela “Agenda” (DV5)
2. O usuário clica no botão “Fechar Agenda”
3. O sistema salva a alteração e exibe a mensagem “Agenda fechada com sucesso”
4. O caso de uso se encerra

Fluxos Alternativos

Não há.

Fluxos de Exceção

Não há.

Regra de Negócio

R1. Uma agenda pode ser fechada a qualquer momento.

UC08 – Manter Agenda

Descrição: Este caso de uso serve para o usuário manter a agenda, sendo capaz de editar ou excluir as informações a qualquer momento (DV5).

Pré-condições

Para que o caso de uso se inicie é necessário que tenha sido executado:

1. UC06 – Criar Agenda

Pós-condições

Após o fim normal do caso de uso em questão, o sistema deve:

1. Manter a agenda

Ator Principal

Profissional

Fluxo de Eventos Principal

1. O sistema apresenta a tela “Agenda” (DV5)
2. O usuário clica no botão “Editar Agenda” (A01)
3. O sistema apresenta a tela “Editar Agenda” com os campos para criação da agenda
4. O usuário preenche todas as informações e clica no botão “salvar” (E01)
5. O sistema exibe a mensagem “deseja salvar alterações?”, com as opções “sim/cancelar”
6. O usuário seleciona a opção “sim” (A02)
7. O sistema salva o cadastro e apresenta a mensagem “agenda salva”
8. O caso de uso se encerra

Fluxos Alternativos

A01. O usuário clica no botão “Excluir Agenda”

1. O sistema exibe a mensagem “deseja excluir agenda?”, apresentando as opções “sim/cancelar”
2. O usuário seleciona a opção “sim” (A03)
3. A agenda é excluída
4. O caso de uso se encerra

A02. O usuário seleciona a opção “cancelar”

1. O sistema retorna para o passo 3 do fluxo principal

A03. O usuário seleciona a opção “cancelar”

1. O sistema retorna para o passo 1 do fluxo principal

Fluxos de Exceção

E01. Campo obrigatório não preenchido

1. O sistema exibe a mensagem “É necessário preencher todos os campos com *”
2. O sistema retorna para o passo de chamada

Regra de Negócio

R1. O usuário pode editar a agenda a qualquer momento.

UC009 – Realizar Cadastro Cliente

Descrição: Este caso de uso serve para que o cliente realize cadastro no sistema.

Data View:

DV6 – Tela Cadastro Cliente

Bem Servido

Crie sua agenda Profissional | Agenda | Cadastro | Login

LOGIN

Nome Completo

E-mail

Data Nascimento

Sexo

CPF

Telefone

Endereço

Já é cadastrado | Login

CADASTRO

Footer Content
You can use rows and columns here to organize your footer content.

Links
Link 1
Link 2
Link 3
Link 4

Figura 40 - DV6 – Tela Cadastro Cliente

Pré-condições

Não há.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Cadastrar cliente no banco de dados.

Ator Principal

Cliente

Fluxo de Eventos Principal

1. O sistema apresenta a “Tela Principal” (DV9) para o usuário.
2. O usuário aciona o botão “Cadastre-se”
3. O sistema direciona o usuário para a “Página de Cadastro” (DV6)
4. O usuário preenche as informações solicitadas
5. O usuário clica no botão “Cadastrar” (E1) (E2)
6. O sistema exibe a mensagem de confirmação “Deseja confirmar o cadastro?”, apresentando as opções “sim/cancelar” (A1)
7. O usuário seleciona a opção “sim”
8. O sistema salva o cadastro e exibe a mensagem de ação realizada com sucesso
9. O caso de uso é encerrado

Fluxos Alternativos

A01. Usuário seleciona a opção “cancelar”

1. Sistema retorna ao passo 3 do fluxo principal

Fluxos de Exceção

E01. Usuário informa dados inválidos:

1. O usuário informa dados inválidos ou deixa de preencher um campo obrigatório
2. O sistema exibe uma mensagem informando o erro
3. O sistema retorna ao passo de chamada da exceção

E02. *Login* já existe no sistema:

1. O sistema exibe uma mensagem informando o erro
2. O sistema retorna ao passo de chamada da exceção

Regra de Negócio

R1. O e-mail de acesso ao sistema deve ser único sendo assim, o sistema não deve permitir o cadastro de usuários que possuam *login*/CPF/CNPJ já cadastrados.

R2. O sistema deve gravar as senhas criptografadas.

UC10 – Manter Cadastro Cliente

Descrição: Este caso de uso serve para que o usuário mantenha o seu cadastro no sistema, sendo capaz de visualizar e editar as informações a qualquer momento.

Data View:

DV7 – Tela Área do Cliente

The screenshot displays the 'Meu Perfil' (My Profile) page for a user named Marcelo. The page is structured as follows:

- Header:** Orange bar with 'Bem Servido' on the left and 'Marcelo' on the right.
- Section:** 'MEU PERFIL' with a sub-section for 'Marcelo'.
- Profile Data:**
 - MEUS DADOS:**
 - Nome: Marcelo
 - CPF: 075.938.099-61
 - Data Nascimento: 03/10/1975
 - Sexo: Masculino
 - Telefone: (41) 3434-3232
 - Celular: (41) 98828-4545
 - Button: EDITAR
 - ENDEREÇO:**
 - CEP: 82620-380
 - Endereço: Praça Hedwíges Nadolny, 843 - Santa Cândida, Curitiba - PR
 - Button: EDITAR
 - LOGIN:**
 - Email: teste1@gmail.com
 - Password: *****
 - Buttons: EDITAR EMAIL, EDITAR SENHA
- Footer:** Orange bar with 'BemServido © 2016' on the left and 'More Links' on the right.

Figura 41 – DV7 – Tela Área/Perfil do Cliente

Pré-condições

Para que o caso de uso se inicie é necessário que tenha sido executado:

1. UC09 – Realizar Cadastro Cliente

Pós-condições

Após a finalização normal do caso de uso em questão, o sistema deve:

1. Manter Cadastro.

Ator Principal

Cliente

Fluxo de Eventos Principal

1. O sistema apresenta a tela “Área do Cliente” (DV7)
2. O usuário seleciona a opção “Editar Perfil”
3. O sistema apresenta a tela “Sobre” com as informações do usuário autenticado
4. O usuário altera os campos desejados e clica no botão “Salvar”
5. O sistema apresenta a mensagem “Deseja salvar informações?”, apresentando as opções “sim/cancelar” (A1)(E1)
6. O usuário seleciona a opção “sim”
7. O sistema salva o cadastro e exibe a mensagem de ação realizada com sucesso
8. O caso de uso é encerrado

Fluxos Alternativos

A1. Usuário seleciona a opção “Não”.

1. O Sistema retorna para o passo de chamada.

Fluxos de Exceção

E01. Usuário preenche informação inválida

1. O usuário informa dados inválidos ou deixa de preencher um campo obrigatório
2. O sistema exibe uma mensagem informando o problema
4. O sistema retorna para o passo de chamada da exceção

Regra de Negócio

Não há.

UC11 – Realizar Agendamento

Descrição: Este caso de uso serve para que o cliente defina uma data e horário disponível na agenda do profissional para ser atendido.

Data View:

DV8 – Tela Realizar Agendamento

The screenshot displays a web interface for scheduling appointments. The main header is orange and contains the text "Bem Servido" and the name "Marcelo". Below the header, the word "AGENDA" is visible. A modal window titled "AGENDAMENTO" is centered on the screen. This modal contains the following fields and information:

- Data*:** 27/11/2016
- Serviço*:** Escolha um Serviço (with a dropdown arrow)
- Hora Início*:** (empty field)
- Hora Fim:** 00:00
- Valor:** R\$ 0,00
- Duração estimada:** 0 minutos
- CONFIRMAR** button at the bottom right.

In the background, a calendar grid is partially visible, showing dates from 27/11/2016 to 28/11/2016. A list of services is also visible, including "Corte" and "Escova".

Figura 42 – DV8 – Tela Realizar Agendamento

Pré-condições

Para que o caso de uso se inicie é necessário que tenha sido executado:

1. UC03 – Realizar Login.

Pós-condições

Após a finalização normal do caso de uso em questão, o sistema deve:

1. Notificar o usuário de que o agendamento foi solicitado.

Ator Principal

Cliente.

Fluxo de Eventos Principal

1. O sistema apresenta a página do profissional, exibindo a agenda profissional do mesmo, com as datas e horários disponíveis para agendamento.
2. O usuário clica no botão “agendar horário” (A01)
3. O usuário escolhe uma data e horário
4. O usuário clica no botão “Ok”
5. O sistema exibe mensagem “Confirmar horário?”, apresentando as opções “sim/cancelar”
6. O usuário seleciona a opção “sim” (A02) (E01)
7. O sistema cadastra as informações de agendamento na base
8. O caso de uso é encerrado

Fluxos Alternativos

A01. Usuário não autenticado

1. Agenda não será exibida para o cliente
2. Sistema exibe mensagem solicitando que o cliente realize autenticação
3. O sistema retorna ao passo de chamada

A02. Usuário seleciona a opção “cancelar”

1. Sistema retorna ao passo 3 do fluxo principal

Fluxos de Exceção

Não há.

Regra de Negócio

R1. Apenas clientes autenticados no sistema podem agendar horário com um prestador.

UC12 – Cancelar Agendamento

Descrição: Este caso de uso serve para que o usuário possa cancelar um horário agendado (DV5).

Pré-condições

Para que o caso de uso se inicie é necessário que tenha sido executado:

1. UC11 – Realizar Agendamento

Pós-condições

Após o fim normal do caso de uso em questão, o sistema deve:

1. Cancelar agendamento

Ator Principal

Todos

Fluxo de Eventos Principal

1. O sistema apresenta a tela “Agenda” (DV5)
2. O usuário clica no botão “Cancelar horário” ao lado do agendamento desejado
3. O sistema exibe mensagem “Confirmar horário?”, apresentando as opções “sim/cancelar”
4. O usuário seleciona a opção “sim” (A01)

5. O sistema salva as informações na base
6. O caso de uso é encerrado

Fluxos Alternativos

A01. Usuário seleciona a opção “cancelar”

1. Sistema retorna ao passo 3 do fluxo principal

Fluxos de Exceção

Não há.

Regra de Negócio

R1. Um horário agendado pode ser cancelado a qualquer momento.

UC13 – Avaliar Profissional

Descrição: Este caso de uso serve para que o cliente avalie um profissional pelo serviço agendado através do sistema.

Data View

DV12 – Qualificar Profissional

Bem Servido			Leandro
<u>QUALIFICAÇÕES</u>			
Qualifique os serviços			
SERVIÇOS FINALIZADOS			
Data	Profissional	Serviço	
27/11/2016	Agatha	Beleza, Saúde e Estética	AVALIAR ★
27/11/2016	Carol	Beleza, Saúde e Estética	AVALIAR ★
BemServido © 2016			More Links

Figura 43 - Qualificações a fazer

Pré-condições

Para que o caso de uso se inicie é necessário que tenha sido executado:

1. UC03 – Fazer Login
2. UC16 – Realizar Agendamento

Pós-condições

Após a finalização normal do caso de uso em questão, o sistema deve:

1. Permitir que o usuário avalie o atendimento do prestador com quem agendou um horário pelo sistema.

Ator Principal

Cliente

Fluxo de Eventos Principal

1. O sistema apresenta a tela de agenda
2. O usuário clica na opção “Avaliar”
3. O sistema exibe as empresas disponíveis para avaliação
4. O usuário avalia de 1 a 5
5. O usuário clica no botão “salvar”
6. O sistema salva
7. O caso de uso é encerrado

Fluxos Alternativos

Não há.

Fluxos de Exceção

Não há.

Regra de Negócio

R1. O cliente pode avaliar um atendimento desde que tenha agendado um horário com o prestador através do sistema “Bem Servido”.

UC14 – Pesquisar Serviço

Descrição: Este caso de uso serve para que usuários possam buscar serviços no sistema

Data View:

DV11 – Tela Principal - Pesquisa

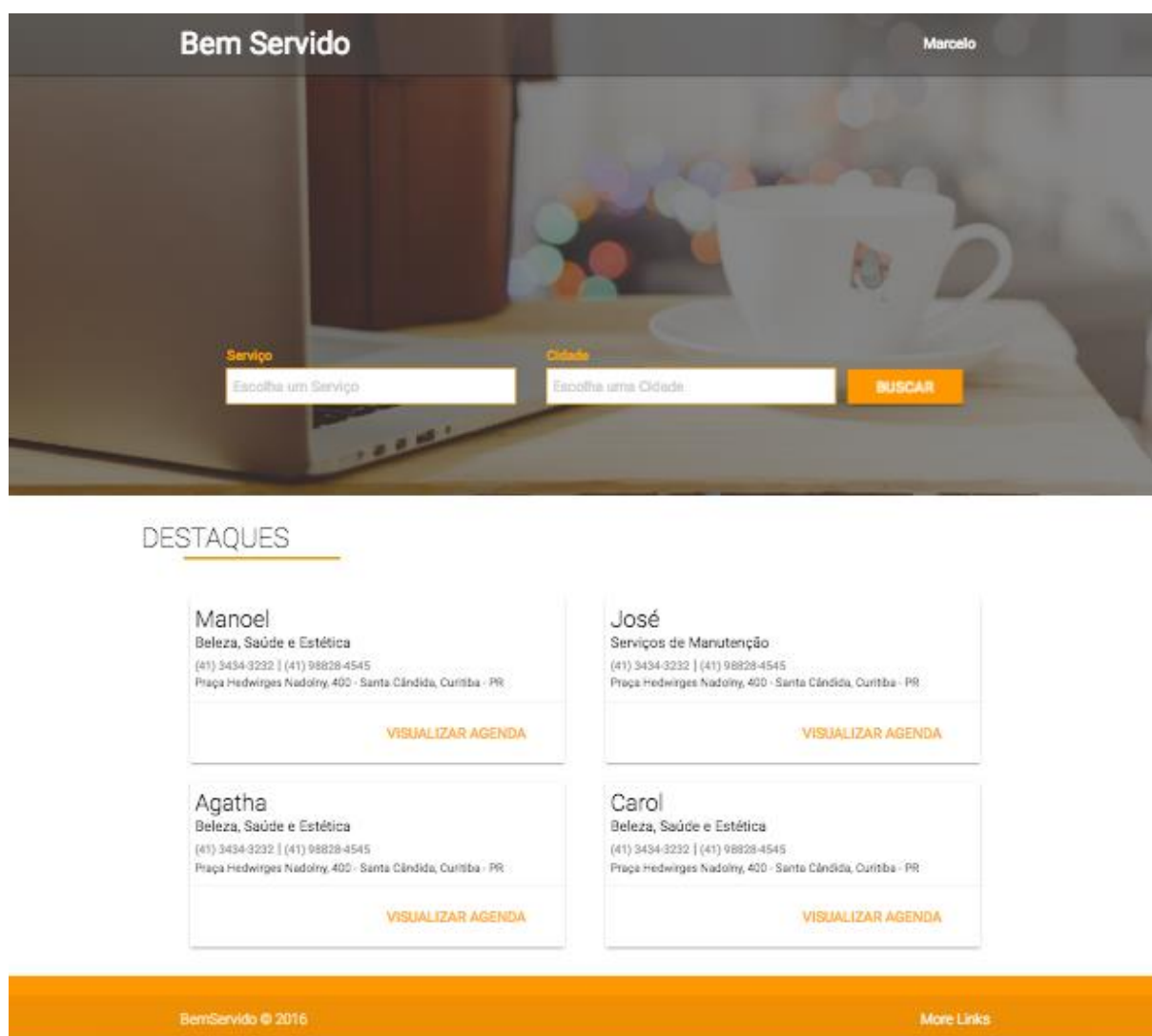


Figura 44 - DV11 – Tela Principal - Pesquisa

Pré-condições

Não há.

Pós-condições

Após o fim normal do caso de uso em questão, o sistema deve:

1. Permitir que o usuário visualize os resultados da pesquisa
2. Permitir que o usuário abra a página do resultado desejado

Ator Principal

Todos

Fluxo de Eventos Principal

1. O sistema apresenta a “tela principal” (DV9) para o usuário
2. O usuário insere no campo de busca o serviço e/ou localidade desejados
 - 2.1 Ao passo que o usuário insere o nome do serviço ou localidade, o sistema filtra resultados com nomes semelhantes.
3. O usuário clica no botão “pesquisar” (E01)
4. O sistema exibe os resultados da pesquisa
5. O caso de uso é encerrado

Fluxos Alternativos

Não há.

Fluxos de Exceção

E01. Não há resultados para a busca

1. Mensagem informando a exceção é exibida
2. O sistema retorna para o passo de chamada da exceção

Regra de Negócio

Não há.

UC15 – Logout

Descrição: Este caso de uso serve para que o usuário saia do sistema.

Data View

DV12 - Logout

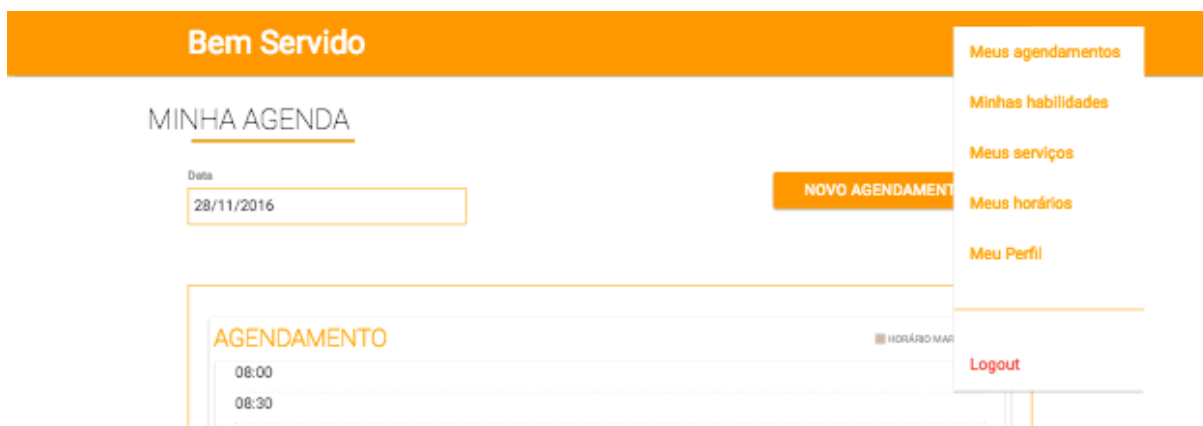


Figura 45 - Tela opção de Logout

Pré-condições

Para que o caso de uso se inicie é necessário que tenha sido executado:

3. UC03 – Fazer Login

Pós-condições

Após a finalização normal do caso de uso em questão, o sistema deve:

1. Desautenticar o e-mail do usuário.

Ator Principal

Cliente, Profissional

Fluxo de Eventos Principal

1. O usuário clica na opção de Menu no topo da página
2. O usuário seleciona a opção "Logout"

3. O sistema desloga o e-mail autenticado, recarregando a página principal
4. O caso de uso é encerrado

Fluxos Alternativos

Não há.

Fluxos de Exceção

Não há.

Regra de Negócio

Não há.

APÊNDICE F – DIAGRAMA DE CLASSES

A imagem a seguir representa o diagrama de classes do projeto, que retrata o relacionamento entre as classes que servem de modelo para objetos.

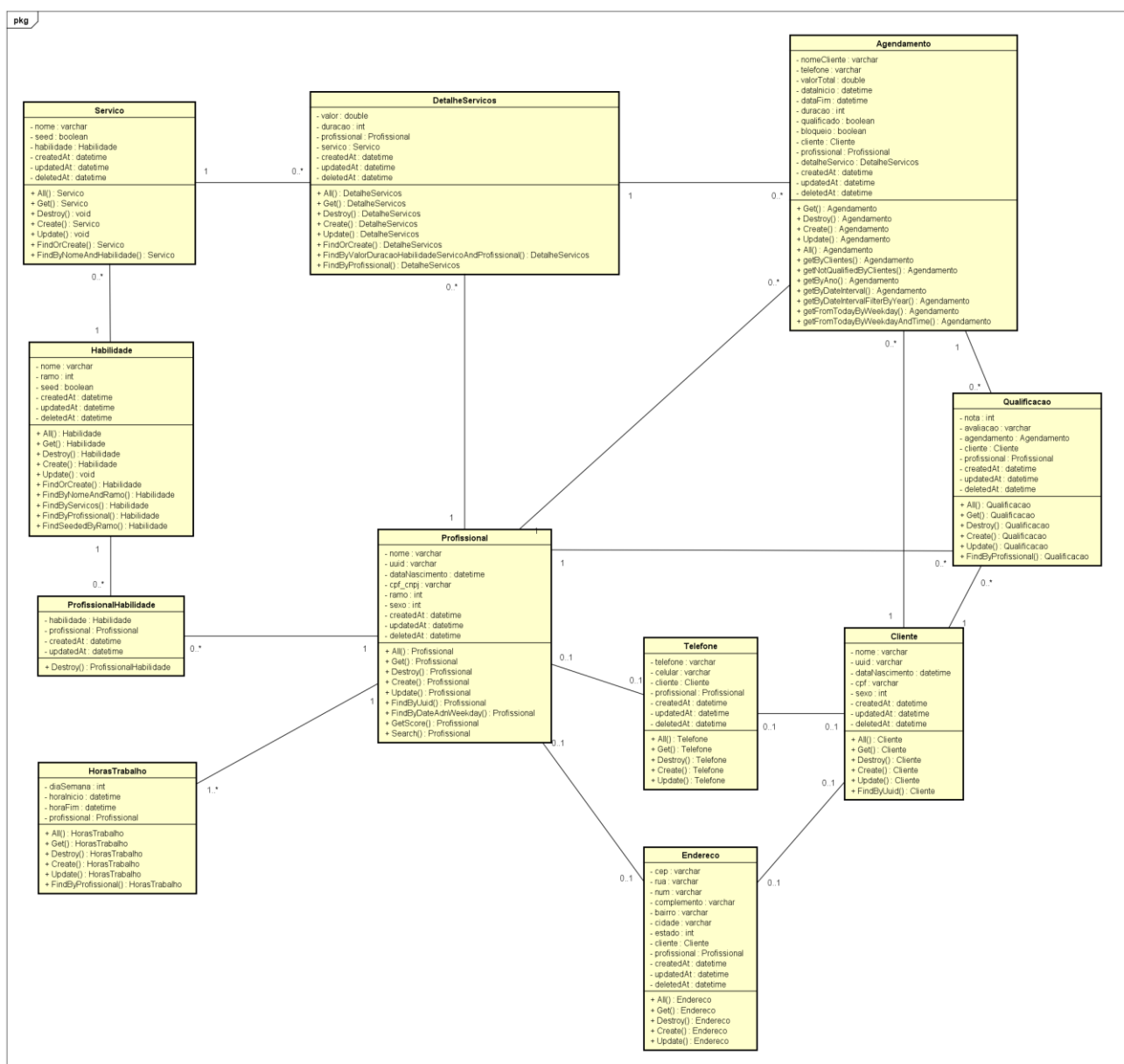


Figura 46 – Diagrama de Classes do Sistema

APÊNDICE G – PLANO DE RISCOS

O quadro a seguir demonstra o plano de riscos realizado pela equipe.

Tabela 11 – Plano de Riscos

PLANO DE RISCOS						
ID	Condição	Consequência	Ação	Monito- ramento	Probabi- lidade	Impacto
1	Alteração nos requisitos	Atraso no cronograma do projeto e retrabalho	Avaliar a real necessidade de alteração dos requisitos	Todos	Moderada	Moderado
2	Desfalque na Equipe durante reunião	Necessidade de utilizar o <i>hangouts</i> em caso de dúvidas	Prosseguir com a reunião	Todos	Moderada	Baixo
3	Falta de conhecimento técnico	Atraso na entrega do projeto	Necessidade de passagem de conhecimento e estudo em livros, apostilas, internet	Todos	Moderada	Alto
4	Saída do orçamento	Atraso na entrega do projeto	Alteração nos requisitos	Todos	Baixa	Alto
5	Saída de recurso	Atraso na entrega do projeto	Redistribuir o trabalho, reajustar o cronograma	Todos	Moderada	Moderada
6	Atraso no desenvolvimento	Atraso na entrega da Sprint	Reajustar o cronograma	Todos	Moderada	Alto

APÊNDICE H – DIAGRAMA ENTIDADE-RELACIONAMENTO (DER)

O diagrama a seguir é o DER, representando a estrutura do banco de dados e descrevendo os processos do negócio. O diagrama apresenta as entidades, que são as próprias tabelas do banco, ligadas por relacionamentos que expressam as dependências entre si.

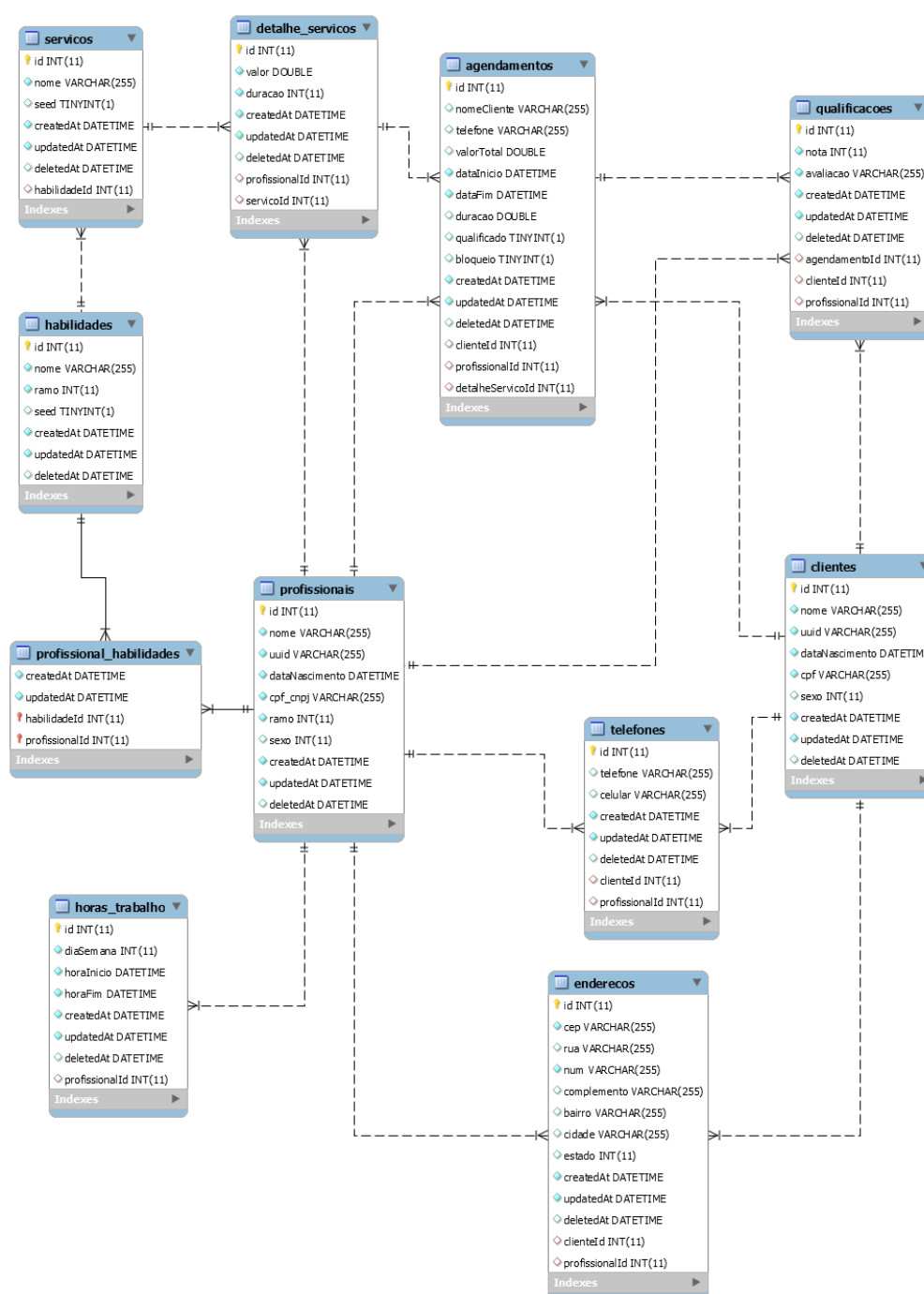


Figura 47 – Diagrama Entidade Relacionamento

APÊNDICE I – WBS

WBS é a sigla para *Work breakdown Structure*, ou Estrutura Analítica de Projetos (EAP) em português. É uma maneira de decompor hierarquicamente a estrutura do projeto em partes menores e flexíveis, facilitando o gerenciamento para a entrega final.

Nesse projeto, os itens do segundo nível foram moldados em *sprints*. Exceto o primeiro e último item, que retratam pré-projeto e documentação. Na prática, a documentação foi sendo elaborada paralelamente aos *sprints*, contudo, foi optado pela equipe representá-la como uma fase do projeto.

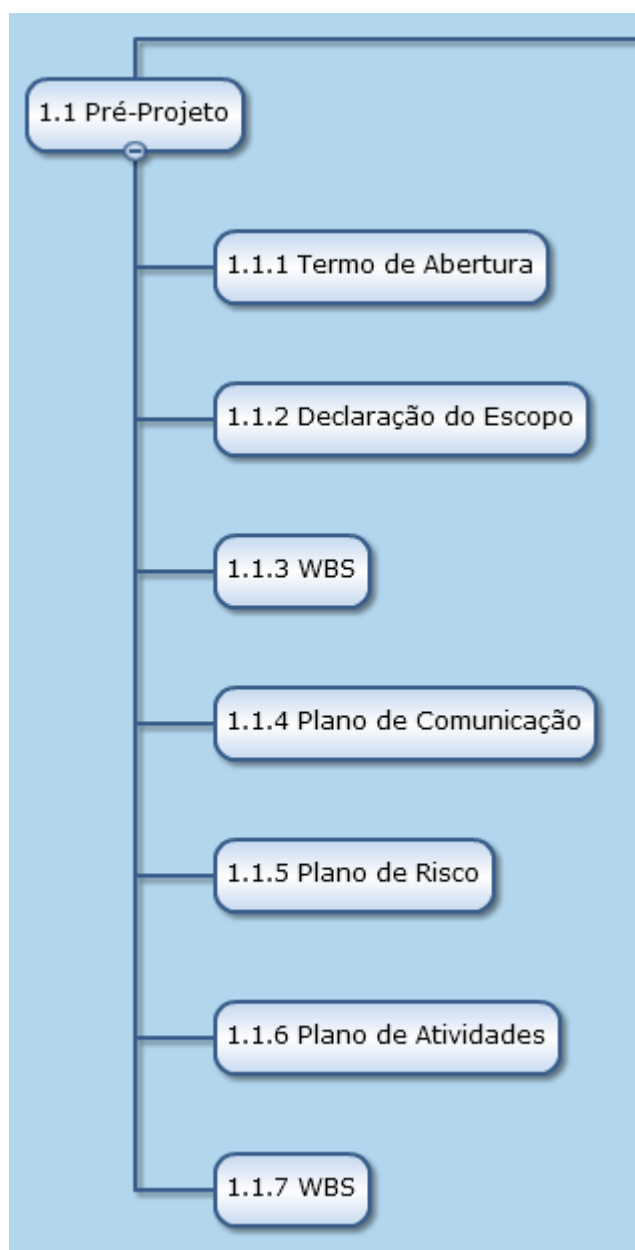


Figura 48 - WBS – Parte 1

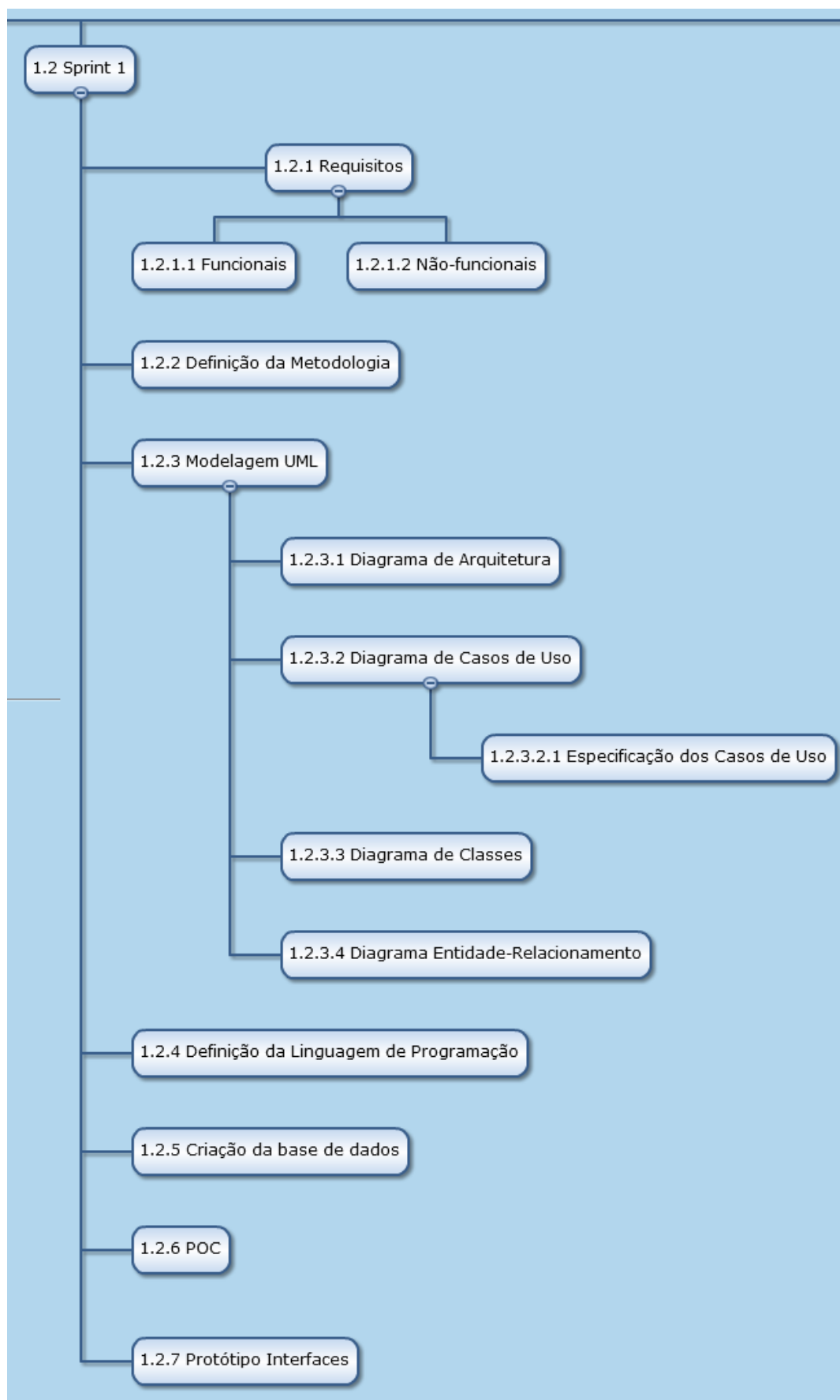


Figura 49 - WBS - Parte 2



Figura 50 - WBS - Parte 3

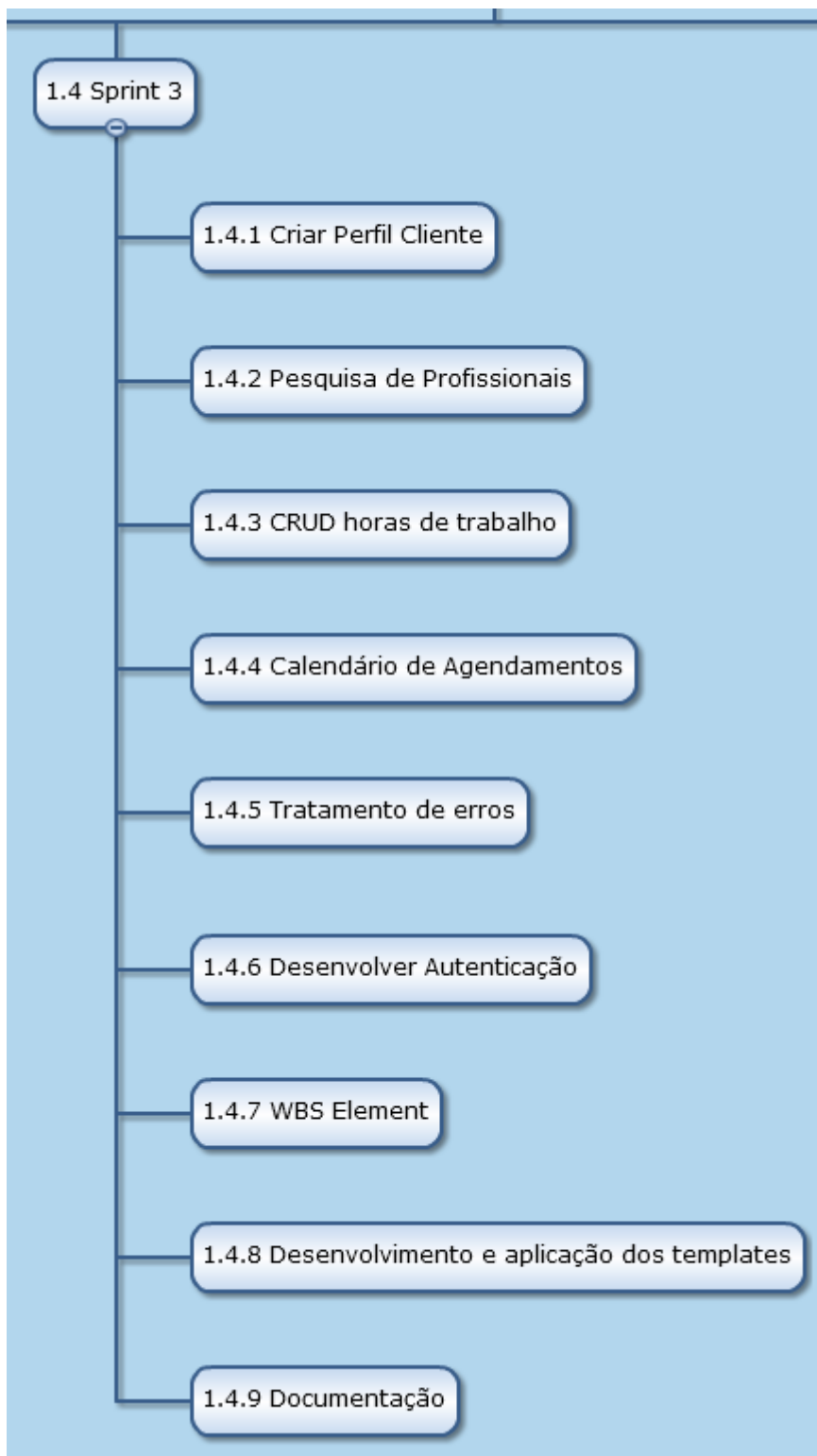


Figura 51 - WBS - Parte 4

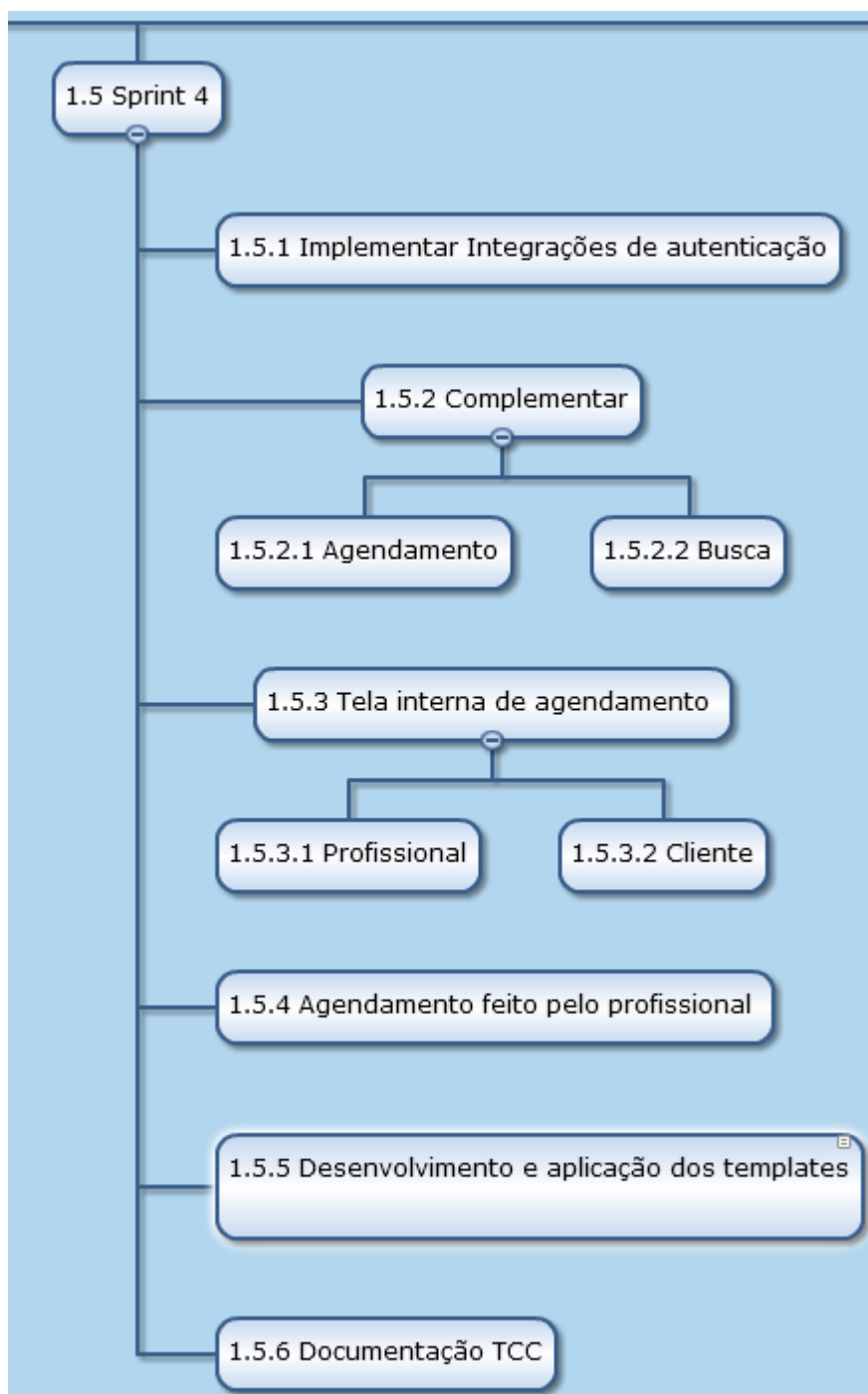


Figura 52 - WBS - Parte 5

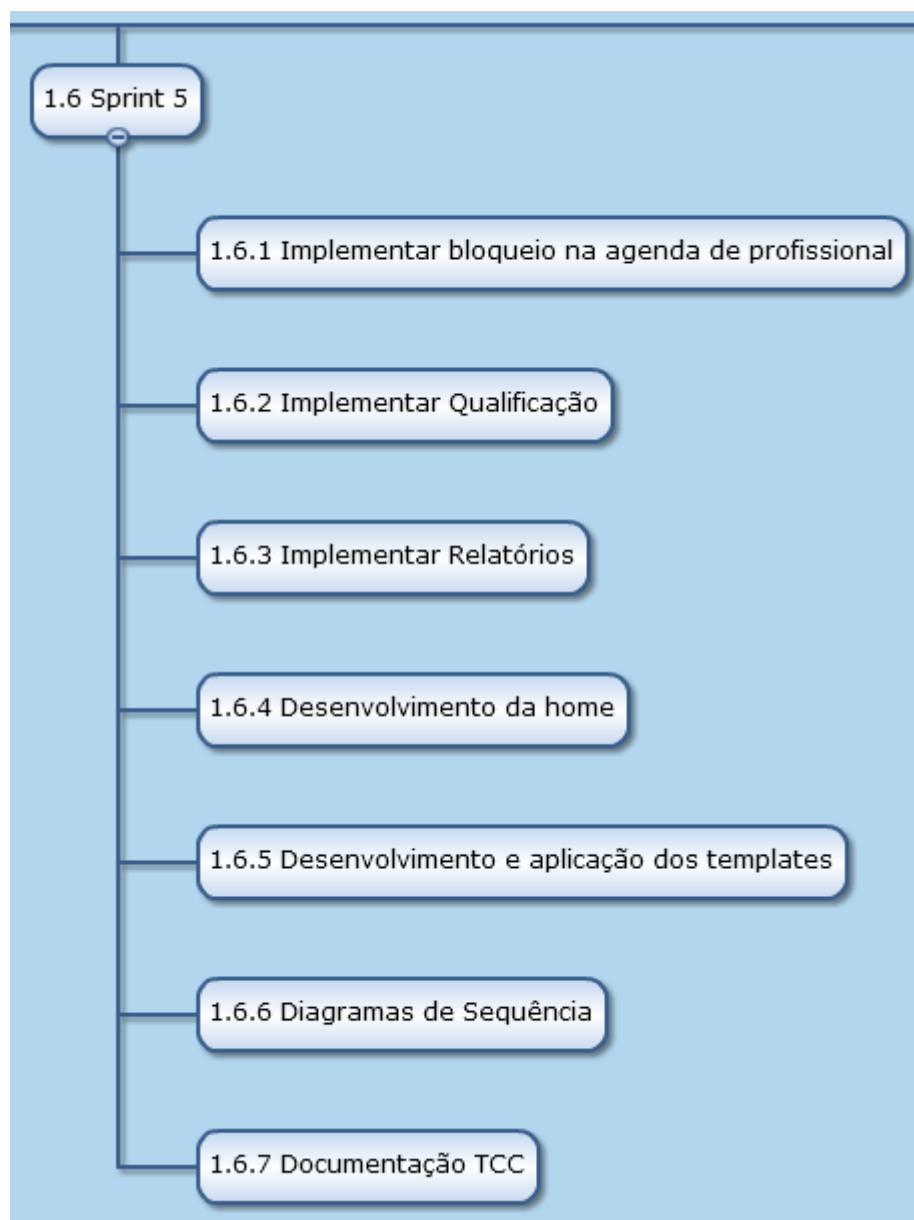


Figura 53 - WBS - Parte 6

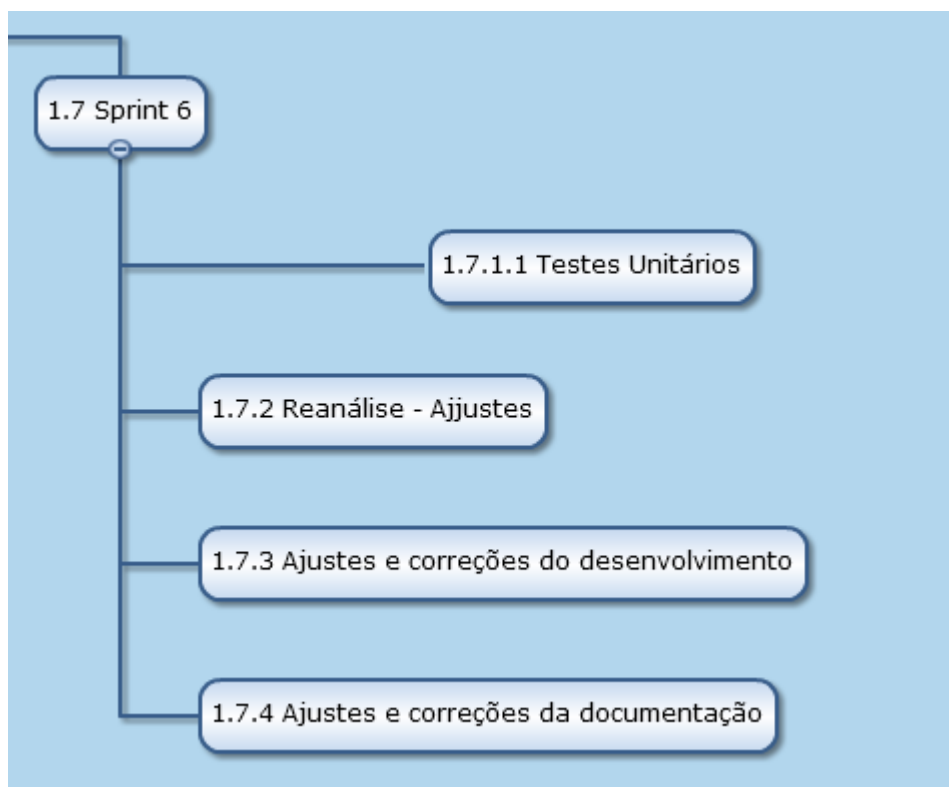


Figura 54 - WBS - Parte 7

APÊNDICE J – PLANO DE COMUNICAÇÃO

Tabela 12 – Plano de Comunicação

Grupo de Interessados	Foco	O que o grupo necessita saber?	Método	Quando
Informações Internas ao Projeto				
Equipe do projeto	Desenvolvimento em conjunto	Compartilhar conhecimentos durante o desenvolvimento e realização de testes	Reuniões	Aos Domingos
Equipe do projeto	Definição dos Sprints	Como iniciar o próximo sprint	Reuniões	Às Quartas-feiras
Equipe do projeto	Definição dos Diagramas	Como seguir com os diagramas	Reuniões	Às Quartas-feiras
Equipe do projeto	Dúvidas durante desenvolvimento individual do projeto	Sanar dúvidas	Hangouts	Quando necessário
Informações Externas ao Projeto				
Equipe do projeto junto ao professor orientador	Informações sobre o progresso do projeto	Sobre o andamento do projeto	Reuniões	Às Sextas-feiras

APÊNDICE K – DIAGRAMAS DE SEQUÊNCIA

O diagrama de seqüência demonstra a troca de mensagens entre os objetos do sistema no decorrer do tempo para a realização de uma operação.

Cadastro no Sistema

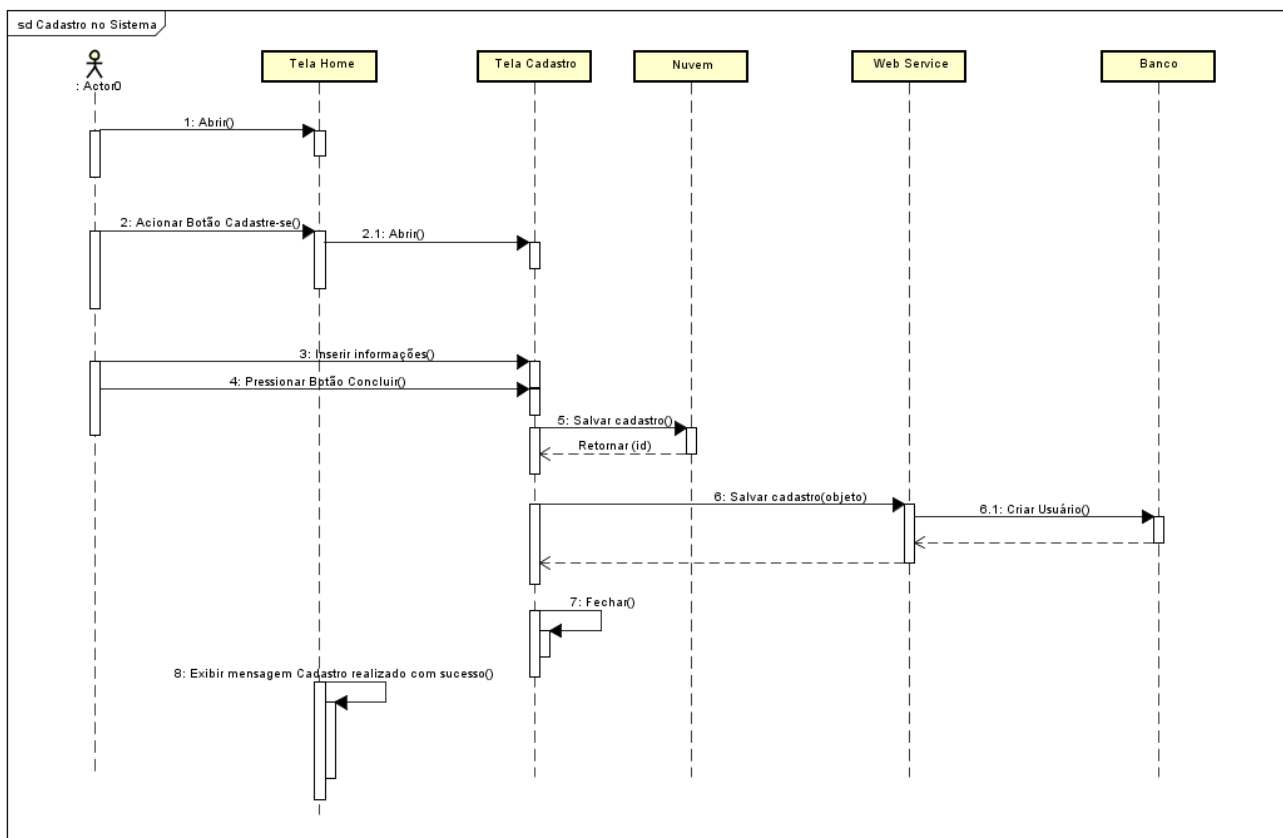


Figura 55 – Diagrama de Sequência Cadastro no Sistema

Editar Cadastro

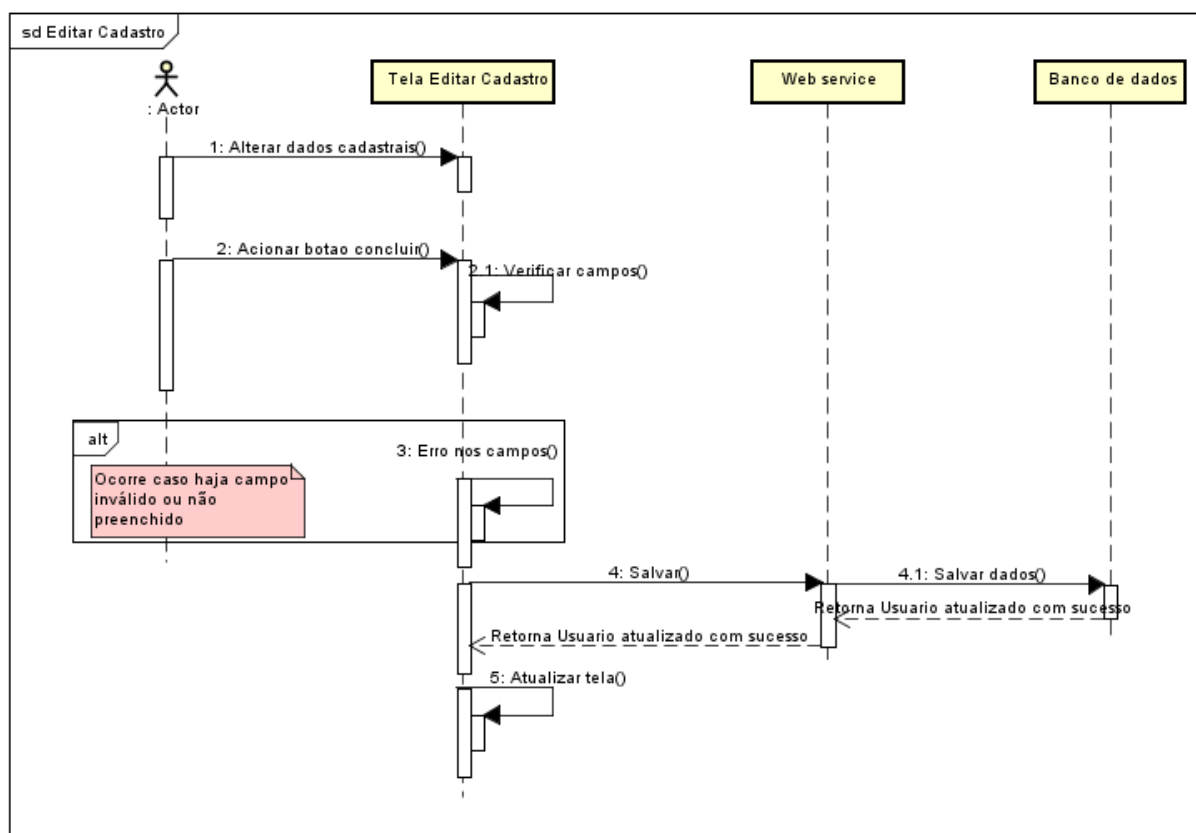


Figura 56 - Diagrama de Sequência Editar Cadastro

Login

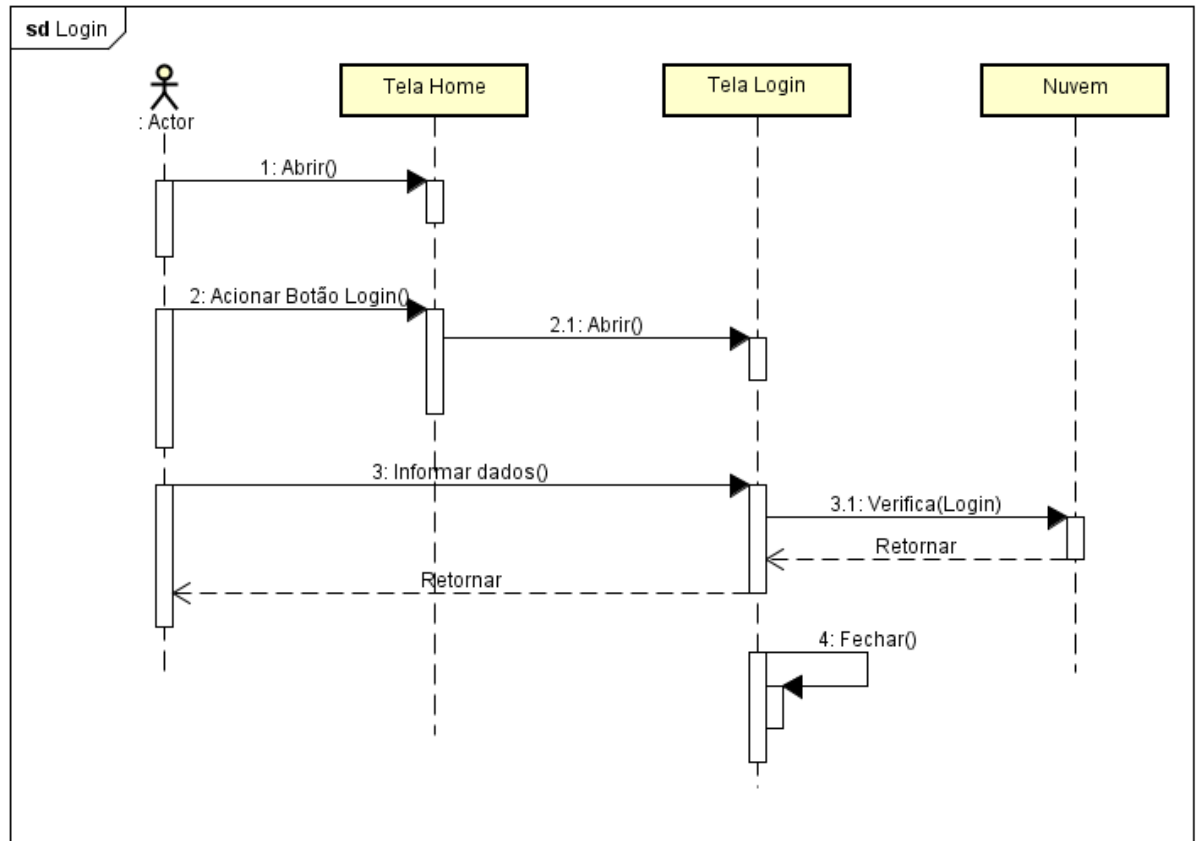


Figura 57 - Diagrama de Sequência Login no sistema

Pesquisar Serviço

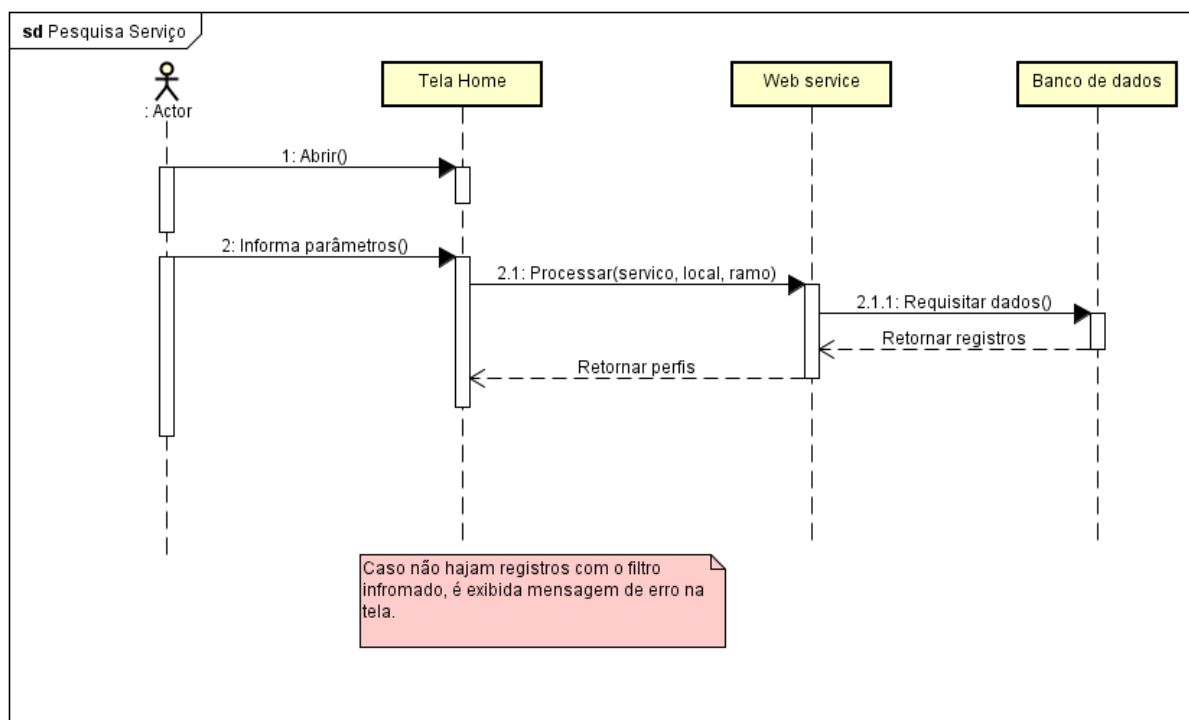


Figura 58 - Diagrama de sequência Pesquisa Serviço

Agendar Horário

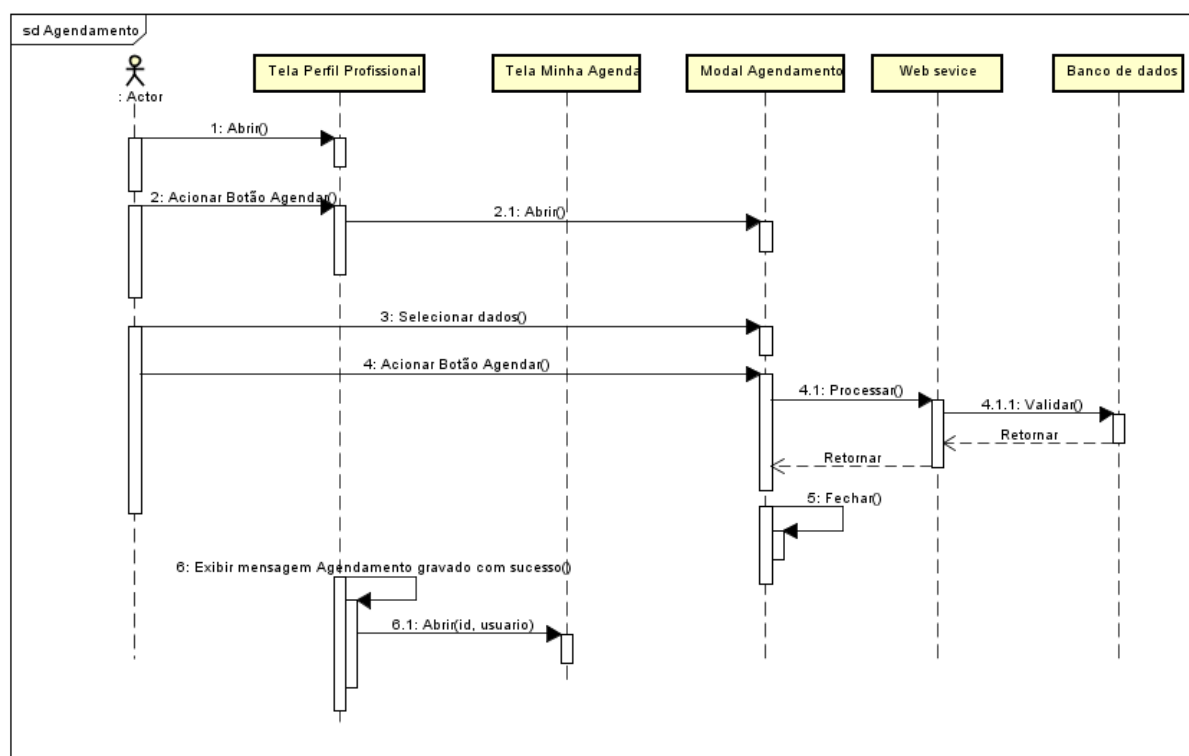


Figura 59 - Diagrama de Sequência Agendamento

Erro de conexão

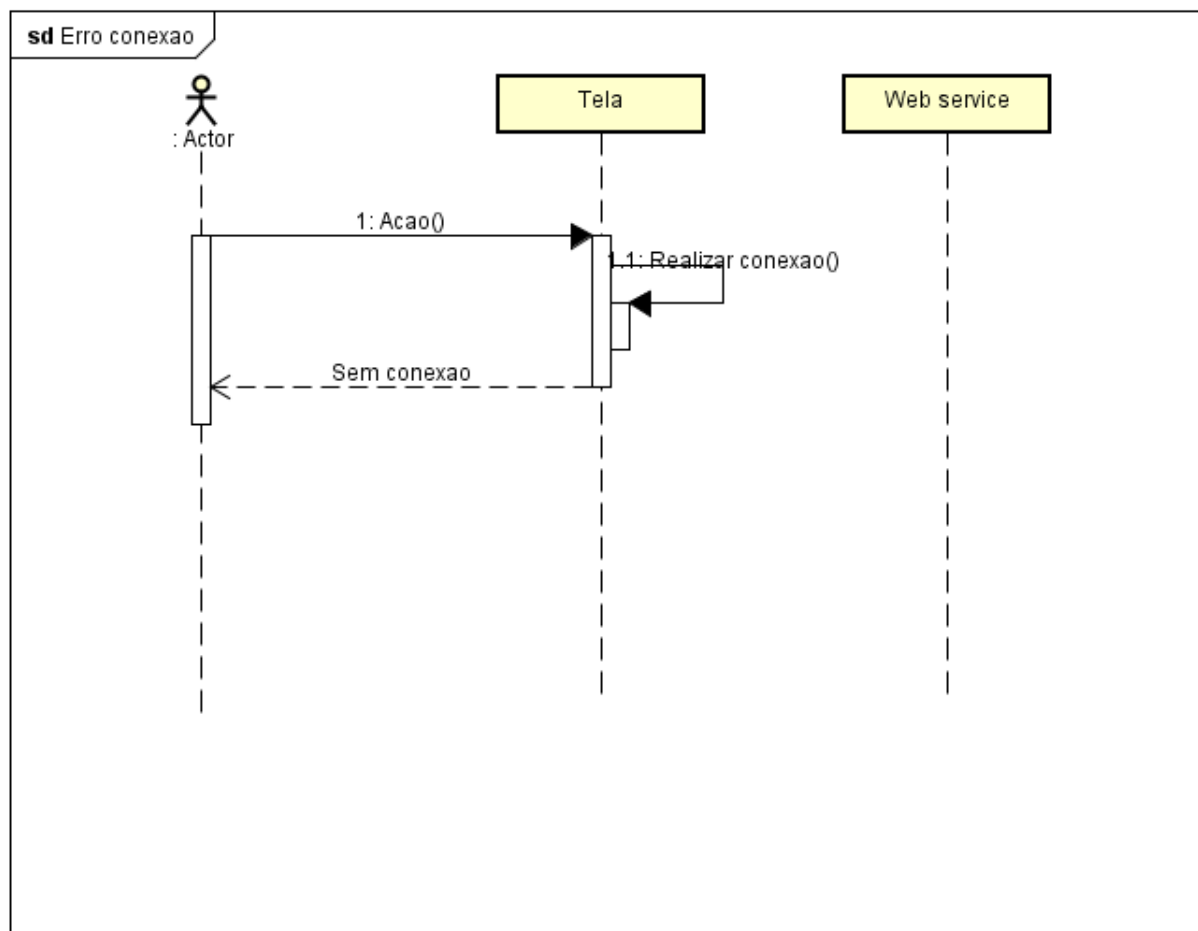
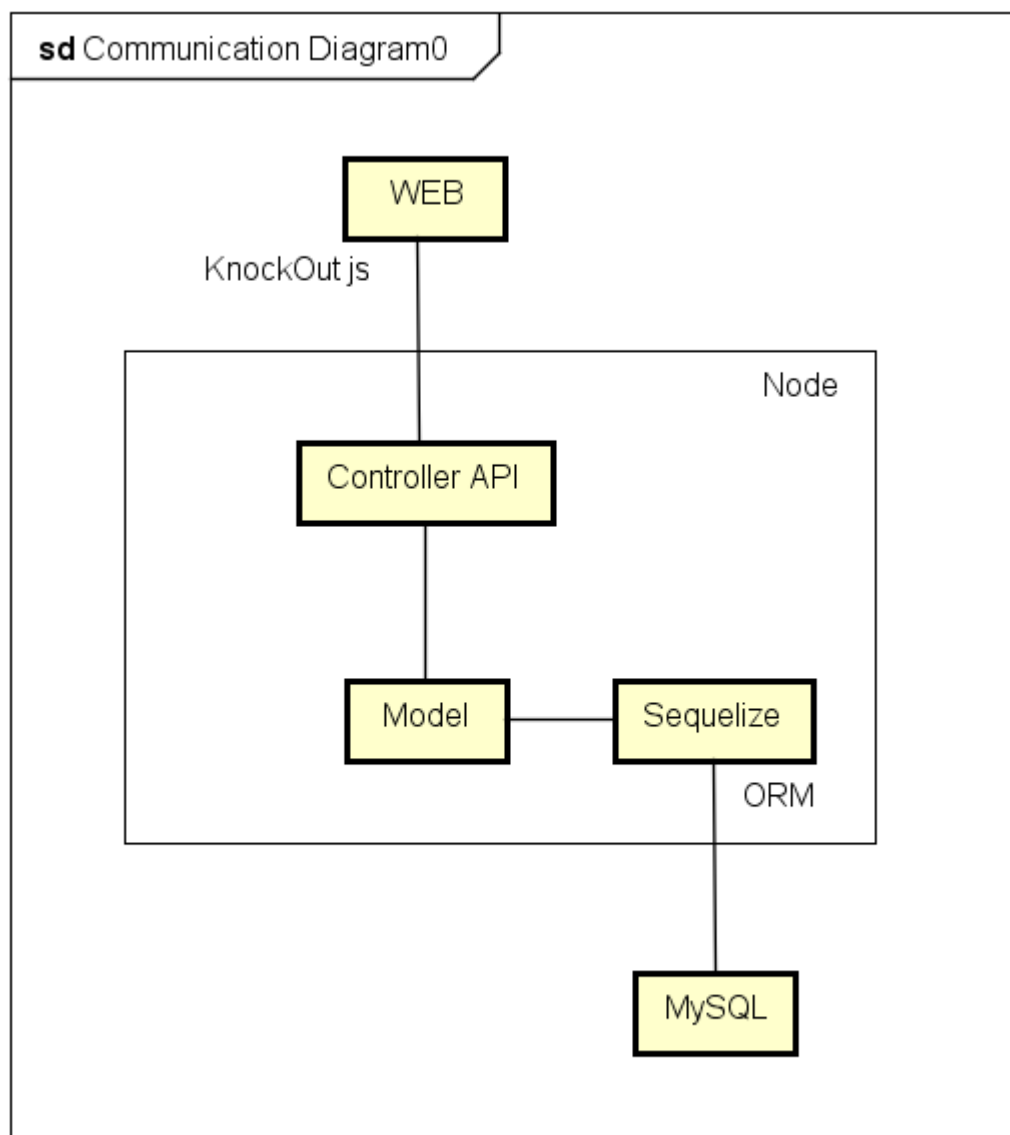


Figura 60 - Diagrama de sequência Erro conexão

APÊNDICE L – DIAGRAMA DE ARQUITETURA



powered by Astah

Figura 61 - Diagrama de Arquitetura