

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

**SISTEMA GERENCIADOR DE UMA CENTRAL DE
SERVIÇOS**

JUAN PAULO DA ROCHA

BLUMENAU
2015

2015/2-09

JUAN PAULO DA ROCHA

**SISTEMA GERENCIADOR DE UMA CENTRAL DE
SERVIÇOS**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Marcel Hugo, Mestre - Orientador

**BLUMENAU
2015**

2015/2-09

SISTEMA GERENCIADOR DE UMA CENTRAL DE SERVIÇOS

Por

JUAN PAULO DA ROCHA

Trabalho de Conclusão de Curso aprovado
para obtenção dos créditos na disciplina de
Trabalho de Conclusão de Curso II pela banca
examinadora formada por:

Presidente: _____
Prof. Marcel Hugo, Mestre – Orientador, FURB

Membro: _____
Prof. Matheus Carvalho Viana, Doutor – FURB

Membro: _____
Prof. Aurélio Faustino Hoppe, Mestre – FURB

Blumenau, 09 de dezembro de 2015

Dedico este trabalho aos meus familiares e aos meus amigos, que sempre me apoiaram e me incentivaram na busca pelos meus objetivos.

AGRADECIMENTOS

A Deus, por tudo que sempre me proporcionou e todo seu amor por mim.

Aos meus pais, José Hercílio da Rocha e Juraci de Souza Rocha, que me deram suporte, apoio e um ambiente propício em todos os momentos que precisei, me incentivando para que alcançasse meus objetivos.

Aos meus amigos Dhyon Ross Conhaque Hilleshein, Jean Carlo Martins, Leonardo Ribeiro, Marco Aurélio Rutzen, Paulo Ricardo de Mello, Rômulo Cesar Campos Menon e todos os demais, que estiveram do meu lado e entenderam todas as vezes que abdiquei de estar com eles para focar no meu trabalho.

Ao meu orientador, professor Marcel Hugo, que sempre foi muito prestativo com ideias e ajudas para o desenvolvimento deste trabalho, acreditando em mim e contribuindo muito para que o objetivo final fosse alcançado com êxito.

Aos meus colegas de trabalho, que me ajudaram com suas experiências profissionais e pessoais.

Aos professores do Departamento de Sistemas e Computação da Universidade Regional de Blumenau, que contribuíram com seus ensinamentos e experiências durante todo o curso.

O maior erro que você pode cometer, é o de ficar o tempo todo com medo de cometer algum.

Elbert Hubbard

RESUMO

Este trabalho apresenta o desenvolvimento de uma ferramenta para auxiliar prestadoras de serviço no gerenciamento de seus agendamentos, essa ferramenta é dividida em dois módulos: *web* e *mobile*. O módulo *web* funciona como uma ferramenta administrativa para as prestadoras de serviço poderem gerenciar seus agendamentos, enquanto o módulo *mobile* é voltado para os clientes poderem realizar agendamentos de serviços diversos como principal funcionalidade. Dessa forma, o administrador de uma prestadora de serviços pode gerenciar melhor seus agendamentos de horários para com seus clientes, assim como pessoas interessadas em contratar serviços de prestadoras podem ter uma maneira simplificada de realizar agendamentos para esses serviços. Ambos os módulos da aplicação, *web* e *mobile*, foram desenvolvidos utilizando a linguagem de programação Java sendo a versão *mobile* desenvolvida para a plataforma Android. O módulo *web* do sistema utiliza o banco de dados MySQL. Para que a sincronização da aplicação *mobile* com o banco de dados ocorresse, foi desenvolvido um *web service*, também na linguagem Java e com o auxílio da biblioteca Gson, para gerenciamento dos objetos transmitidos entre Android e banco de dados. O módulo *mobile* não possui banco de dados local. Os resultados obtidos com o uso da ferramenta permitem às prestadoras não utilizar papel para gerenciar seus agendamentos e aos clientes possuir uma forma objetiva para realizar tais agendamentos, demonstrando que a utilização do sistema é positiva.

Palavras-chave: Agendamento. Central de serviços. Android. Dispositivo móvel.

ABSTRACT

This work presents the development of a tool to assist services providers to manage their schedules, this tool is divided in two versions: web and mobile. The web version works as an administrative tool for providers can manage their schedule, while the mobile version is turned to customers can perform scheduling of various services as the main feature. At this way, the services provider administrator can better manage his time schedulers to his customers, where the web version works as an administrative tool for providers and the mobile version is geared for customers. Both application modules, web and mobile, were developed using Java programming language and the mobile version was developed for the Android platform. The system's web module uses the MySQL database. For the synchronization between mobile application and database, a web service has been developed also in Java, with the help of Gson library to manage the sent objects between Android and database. The mobile module doesn't have local database. The results obtained from the use of the tool enable providers not to use paper to manage their schedules and customers have an objective way to make such appointments, demonstrating that the use of the system is positive.

Key-words: Scheduling. Service Center. Android. Mobile Device.

LISTA DE FIGURAS

Figura 1 – Tela de pesquisa por horários disponíveis.....	17
Figura 2 – Tela de agendamento.....	17
Figura 3 – Interfaces do aplicativo	19
Figura 4 – Tela de pesquisa por voo, versão web.....	20
Figura 5 – Pesquisa por hotéis, versão para dispositivo <i>mobile</i>	21
Figura 6 – Modelo de casos de uso módulo <i>mobile</i>	26
Figura 7 – Modelo de casos de uso módulo web.....	26
Figura 8 – Diagrama de atividades do processo de realização de agendamento na aplicação web.....	28
Figura 9 – Diagrama de atividades do processo de realização de agendamento no módulo <i>mobile</i>	29
Figura 10 – Diagrama de classes do módulo web	30
Figura 11 – Diagrama de classes da aplicação <i>mobile</i>	31
Figura 12 – Modelo de entidade e relacionamento do sistema	32
Figura 13 – Tela de <i>login</i> do módulo web	42
Figura 14 – Tela de cadastro de prestadora de serviços.....	42
Figura 15 – Tela de serviços prestados pela prestadora.....	43
Figura 16 – Menu do módulo web.....	44
Figura 17 – Tela de manter horário de funcionamento da prestadora	44
Figura 18 – Tela de manter funcionário.....	45
Figura 19 – Tela de manter horário de trabalho do funcionário.....	46
Figura 20 – Tela de manter perfil de clientes	46
Figura 21 – Tela de agendamentos	47
Figura 22 – Tela de cadastro de cliente <i>mobile</i>	48
Figura 23 – Tela de pesquisa por serviço prestado.....	48
Figura 24 – Tela de informações da prestadora de serviços	49
Figura 25 – Tela de agendar horário para o cliente	50

LISTA DE QUADROS

Quadro 1 – Comparativo dos trabalhos correlatos	22
Quadro 2 – Requisitos funcionais do módulo <i>mobile</i>	24
Quadro 3 – Requisitos funcionais da aplicação web	25
Quadro 4 – Requisitos não funcionais do módulo <i>mobile</i>	25
Quadro 5 – Requisitos não funcionais do módulo web	25
Quadro 6 – XML gerenciador do Hibernate	34
Quadro 7 – Classe responsável por gerenciar as conexões ao banco de dados.....	36
Quadro 8 – Classe responsável pelo mapeamento da entidade <i>jpr001_prestadora</i>	37
Quadro 9 – Método responsável por pesquisar empresas pelo nome	39
Quadro 10 – Classe da aplicação <i>mobile</i> responsável pelas requisições de autenticação do cliente.....	40
Quadro 11 – Classe do <i>web service</i> responsável por gerenciar as requisições do cliente	41
Quadro 12 – Comparativo entre a ferramenta EasyPoint e os correlatos	51
Quadro 13 – Caso de uso 01 – Agendar horário	56
Quadro 14 – Caso de uso 02 – Manter perfil.....	57
Quadro 15 – Caso de uso 03 – Realizar login no aplicativo	57
Quadro 16 – Caso de uso 04 – Manter cliente.....	58
Quadro 17 – Caso de uso 05 – Procurar por prestadora de serviço específica.....	58
Quadro 18 – Caso de uso 06 – Procurar por um tipo de serviço prestado.....	59
Quadro 19 – Caso de uso 07 – Realizar feedback sobre o serviço utilizado	59
Quadro 20 – Caso de uso 08 – Visualizar Agendamentos.....	59
Quadro 21 – Caso de uso 09 – Manter horários de atendimento.....	60
Quadro 22 – Caso de uso 10 – Manter funcionários.....	60
Quadro 23 – Caso de uso 11 – Realizar login.....	61
Quadro 24 – Caso de uso 12 – Visualizar perfil de uma prestadora de serviços.....	61
Quadro 25 – Caso de uso 13 – Manter agenda.....	62
Quadro 26 – Caso de uso 14 – Visualizar agenda.....	62

Quadro 27 – Tabela jpr001_prestadora	63
Quadro 28 – Tabela jpr002_servico_prestadora.....	63
Quadro 29 – Tabela jpr003_profissional.....	64
Quadro 30 – Tabela jpr004_cliente	64
Quadro 31 – Tabela jpr005_feedback.....	64
Quadro 32 – Tabela jpr006_uf.....	64
Quadro 33 – Tabela jpr007_cidade.....	65
Quadro 34 - Tabela jpr008_endereco.....	65
Quadro 35 – Tabela jpr009_horario_prestadora.....	65
Quadro 36 – Tabela jpr10_horario_profissional.....	66
Quadro 37 – Tabela jpr011_agendamento	66
Quadro 38 – Tabela jpr012_profissional_servico	66
Quadro 39 – Tabela jpr013_servico_disponivel.....	67

LISTA DE ABREVIATURAS E SIGLAS

ADT – *Android Developer Tools*

CEP – Código de Endereçamento Postal

CNPJ – Cadastro Nacional de Pessoa Jurídica

CPF – Cadastro de Pessoa Física

EA – Enterprise Architect

IDE – *Integrated Development Environment*

JSON – *JavaScript Object Notation*

MER – Modelo de Entidade e Relacionamento

REST – *Representational State Transfer*

SGBD – Sistema de Gerenciamento de Banco de Dados

SQL – *Structured Query Language*

UF – Unidade Federativa

URL – *Uniform Resource Locator*

XML – eXtensible Markup Language

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 OBJETIVOS	14
1.2 ESTRUTURA.....	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 PRESTAÇÃO DE SERVIÇOS	15
2.2 TRABALHOS CORRELATOS	16
2.2.1 Sistema para agendamento de serviços de beleza on-line.....	16
2.2.2 Nos Trinks	18
2.2.3 Kayak	19
2.2.4 Análise comparativa	21
3 DESENVOLVIMENTO DO SOFTWARE.....	23
3.1 LEVANTAMENTO DE INFORMAÇÕES	23
3.2 ESPECIFICAÇÃO.....	24
3.2.1 Requisitos funcionais.....	24
3.2.2 Requisitos não funcionais	25
3.2.3 Modelos de casos de uso	25
3.2.4 Diagramas de atividades	27
3.2.5 Diagramas de classes	29
3.2.6 Modelo de entidade e relacionamento (MER)	31
3.3 IMPLEMENTAÇÃO	33
3.3.1 Técnicas e ferramentas utilizadas	33
3.3.1.1 Ferramentas	33
3.3.1.2 Persistência dos dados	34
3.3.1.3 Comunicação com o servidor.....	39
3.3.2 Operacionalidade da implementação	41
3.3.2.1 Módulo web	41
3.3.2.2 Módulo <i>mobile</i>	47
3.4 RESULTADOS E DISCUSSÕES	50
3.4.1 Comparativo com trabalhos correlatos	50
4 CONCLUSÕES.....	53
4.1 EXTENSÕES	53

1 INTRODUÇÃO

A tecnologia está presente na vida das pessoas. Mesmo havendo problemas na questão de privacidade, a tecnologia proporciona mais comodidade, permite o acesso mais fácil à informação, aumenta o bem estar pessoal além de auxiliar em diversas tarefas diárias (PUCCI, 2010). Ou seja, a tecnologia tem como objetivo tornar a vida das pessoas melhor.

Dispositivos móveis são uma das tecnologias que servem para auxiliar as pessoas nos seus afazeres diários, dando praticidade e comodidade às mesmas. São inúmeros aplicativos criados atualmente para as mais diversas atividades. A plataforma de desenvolvimento *mobile* que possui o sistema operacional Android é uma das mais difundidas no mercado, contando com 1.43 milhões de aplicativos disponíveis. Não bastando isso, registrou um crescimento de 40% na quantidade de aplicativos para a plataforma no ano de 2014 (POZZEBON, 2015).

Atualmente, empresas que utilizam Android no seu ambiente corporativo possuem diferenciais que auxiliam na troca de informações internas, por exemplo, proporcionando mais segurança na transição das mesmas. Além das funcionalidades existentes para empresas, a *Google*, que atualmente desenvolve a plataforma, planeja simplificar ainda mais a gerência de aparelhos com Android em empresas, assim como adicionar mais recursos (ZARATIN, 2014).

Além do uso interno, empresas que utilizam da tecnologia de dispositivos móveis para poderem manter um contato maior com seus clientes podem vir a ter uma elevação tanto organizacional quanto de lucros. Ter um contato maior com a clientela permite entender melhor quais as necessidades e os desejos da mesma. Segundo Ferreira (2012), “uma empresa que não se comunica com rapidez pode estar em desvantagem caso sua concorrente faça bom uso dos aplicativos que facilitam a tomada de decisão e tornam os processos mais ágeis”.

Existem aplicativos disponibilizados com o objetivo de auxiliar empresários na gestão de seus negócios, como por exemplo o Clear, um aplicativo organizador de tarefas que possui uma interface que facilita na hora de adicionar uma tarefa. Há também aplicativos que já são conhecidos de muitos usuários *desktop*, como é o caso do Dropbox, que é uma pasta virtual para armazenamento de dados, onde o usuário pode salvar arquivos pelo aplicativo, acessá-los e modificá-los a qualquer momento. (FERREIRA, 2012).

Dessa forma o gerenciamento de atividades, sejam elas genéricas ou específicas, pode ser auxiliado com a utilização de aplicativos para dispositivos móveis, observando que o sistema operacional Android, segundo Hamann (2014), “é o sistema operacional mais utilizado em portáteis. Hoje existem mais de 1 bilhão de aparelhos ativos [...]”. Tendo como

base um caso de gerenciamento mais restrito, como o de prestadoras de serviço com seus agendamentos de horários, o uso de um aplicativo que auxiliasse na realização dos agendamentos e na gerência dos mesmos poderia vir a ser útil. Afinal, agendamentos realizados manualmente em papel pelo prestador do serviço ocasionalmente podem gerar extravio de informações e outros tipos de erros ou equívocos. Com base nesses dados, foi desenvolvida uma ferramenta que auxilie prestadoras de serviços a gerenciar seus agendamentos de horários para com seus clientes.

1.1 OBJETIVOS

O objetivo geral do trabalho é disponibilizar um aplicativo *mobile*, para o sistema operacional Android, com uma parte administrativa em ambiente web, que realize o gerenciamento dos agendamentos de horários para prestadoras de serviços.

Os objetivos específicos do trabalho proposto são:

- a) desenvolver um módulo web para o prestador gerenciar informações de serviços e horários da empresa;
- b) construir *web services* que permitam a integração dos módulos web e *mobile*;
- c) desenvolver módulo *mobile* para clientes permitindo acesso aos horários de atendimento.

1.2 ESTRUTURA

A estrutura do trabalho é composta por quatro capítulos e está descrita a seguir.

No primeiro capítulo tem-se a introdução ao tema principal deste trabalho aliada à apresentação da justificativa e dos objetivos.

No segundo capítulo é apresentada a fundamentação teórica sobre prestação de serviços e trabalhos correlatos.

No terceiro capítulo é descrito o desenvolvimento do software, sendo apresentados os tópicos sobre levantamento de informações, os requisitos funcionais e não funcionais levantados, a modelagem de casos de uso, os diagramas de atividades, os diagramas de classes e, por fim, o modelo de entidade e relacionamento (MER) do sistema. Na sequência tem-se a apresentação das ferramentas e técnicas utilizadas na implementação, assim como a operacionalidade do sistema. Finalizando o capítulo são apresentados os resultados obtidos e uma discussão sobre o trabalho.

Conclusões e sugestões para trabalhos futuros são apresentadas no Capítulo 4.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta assuntos abordados no desenvolvimento do trabalho. A Seção 2.1 explica conceitos sobre prestação de serviços e a Seção 2.2 apresenta os trabalhos correlatos.

2.1 PRESTAÇÃO DE SERVIÇOS

Prestação de serviço, segundo UGF (2012), consiste na “[...] execução de um trabalho contratado por terceiros (empresa/comunidade), que pode ser estendida para consultorias e assessorias. Caracterizada por um processo de produção e uso simultâneo que pode não consistir necessariamente na posse de um bem”.

O principal ativo de uma empresa são seus clientes. Empresas que estabelecem uma relação com o cliente baseadas em um serviço prestado com louvor tornam-se mais competitivas no mercado. A utilização das tecnologias para fornecer isto é comum atualmente, porém a rápida evolução tecnológica faz com que hoje o que é um diferencial, em breve, torne-se algo básico (LABBATE, 2015).

O motivo dessa situação é que a evolução tecnológica supre as necessidades e expectativas dos clientes, tornando-os cada vez mais exigentes. Segundo Labbate (2015), “dentro de alguns anos, o serviço ao cliente estará tão mudado que será algo inerente ao próprio produto. Produtos físicos terão embutida a tecnologia ‘sempre online’, que transformará as oportunidades de serviços em importante fonte de valor agregado”. Tendo isso em vista, num futuro próximo, as empresas estarão cada vez mais conectadas com seus clientes, proporcionando uma experiência unificada e simplificada (LABBATE, 2015).

SENAC (2001, p. 7) destaca que “as empresas prestadoras de serviços e os profissionais precisam estar atentos a essa revolução nas relações de trabalho, para que possam criar estratégias que garantam seu espaço nesse novo mercado, cada vez mais competitivo”. Continuando, SENAC (2001, p. 37) também afirma que “o profissional hoje não tem como fugir das novas tecnologias, ele vai precisar quebrar tabus e aprender a lidar com computadores, com a internet, com todas as novidades do mundo moderno que estejam relacionadas à sua profissão e ao universo do trabalho”.

Além da utilização de tecnologias as empresas devem ter como premissas buscar a satisfação dos seus clientes. Um serviço ao alcance do cliente, aliado a uma boa prestação faz com que crie uma fidelidade do cliente com a empresa em questão. Para que haja a satisfação da clientela a empresa que presta serviços necessita manter o foco no que seus clientes

necessitam e desejam, podendo dessa forma proporcionar um serviço que possa contemplar as expectativas dos seus possíveis clientes (SENAC, 2001, p. 14).

Com uma gama elevada de dispositivos conectados todo instante à Internet, novas tecnologias entram como formas de diferencial no mercado. É o caso de equipamentos *wearables*, como acessórios diários que possuem dispositivos, medindo taxa de glicose, permitindo o acesso à Internet, armazenando informações bancárias, além de mecanismos com interface melhorada, uso de reconhecimento vocal e retina. Caberá então às empresas se adaptarem a essas tecnologias no seu processo de negócios, podendo assim transformar toda essa informação em um relacionamento para com seus clientes reais e pretendidos (LABBATE, 2015).

2.2 TRABALHOS CORRELATOS

Nessa seção são descritos os trabalhos que se assemelham ao referente trabalho. A Subseção 2.2.1 apresenta uma proposta de conclusão do curso de Sistemas de Informação na Universidade Regional de Blumenau por Cezerino (2011). A Subseção 2.2.2 traz explicações sobre o aplicativo *mobile* Nos Trinks (MCLEAN, 2013). Na Subseção 2.2.3 têm-se uma apresentação sobre o aplicativo *mobile* Kayak, criado em 2004 (ROCHA, 2015). Por fim, a Subseção 2.2.4 apresenta uma análise comparativa entre os trabalhos correlatos e este trabalho em questão.

2.2.1 Sistema para agendamento de serviços de beleza on-line

O trabalho de Cezerino (2011) foi desenvolver um sistema que permita a consulta por profissionais de beleza, possibilitando a busca pelo serviço, pelo horário desejado bem como o agendamento *on-line* do mesmo. A principal ideia do sistema é automatizar o processo de agendamentos e disponibilizar a agenda dos profissionais na Internet. Dessa forma tornando possível ao cliente o agendamento de serviços a qualquer hora do dia, sendo em horário comercial ou não.

A Figura 1 apresenta a tela de buscas por horários disponíveis. É possível passar filtros para a pesquisa, tais como o serviço desejado, o profissional que realiza o serviço, cidade e bairro, além da data desejada.

Figura 1 – Tela de pesquisa por horários disponíveis

Opção	Profissional	Data	Início	Final	Estabelecimento
Reservar	Cida Cezerino	16/12/2011	08:00	10:00	Cabelos e Cia- Victor Konder
Reservar	Cida Cezerino	16/12/2011	10:00	12:00	Cabelos e Cia- Victor Konder
Reservar	Cida Cezerino	16/12/2011	13:00	15:00	Cabelos e Cia- Victor Konder
Reservar	Cida Cezerino	16/12/2011	15:00	17:00	Cabelos e Cia- Victor Konder
Reservar	Cida Cezerino	16/12/2011	17:00	19:00	Cabelos e Cia- Victor Konder

Fonte: Cezerino (2011, p.41).

Na Figura 2, pode-se observar a tela de agenda contendo os horários disponíveis e já agendados do serviço, separados por cada profissional. A Figura 2 mostra também a possibilidade de após uma busca, realizar o agendamento para o serviço desejado.

Figura 2 – Tela de agendamento

Fonte: Cezerino (2011, p.43).

A utilização dessa ferramenta proporciona uma melhora no gerenciamento do ambiente em que a mesma se encontra. Torna a organização e controle dos horários agendados mais satisfatória, além da realização do agendamento estar disponível através do

site, assim como a eliminação das ligações telefônicas e da agenda de papel (CEZERINO, 2011).

2.2.2 Nos Trink's

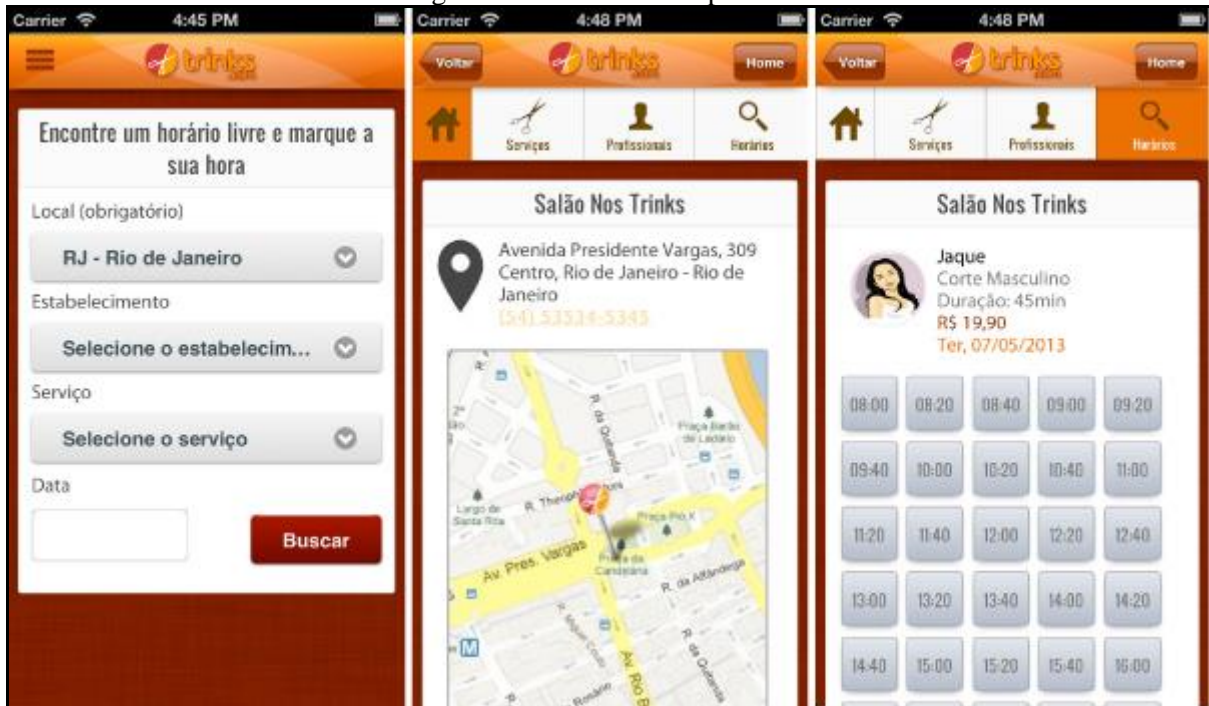
O aplicativo Nos Trink's foi criado com o objetivo de auxiliar o usuário na hora de realizar algum agendamento. O aplicativo está disponível para a plataforma web, assim como para as plataformas de dispositivos móveis Android e iOS, sendo gratuito para *download* (MCLEAN, 2013).

A aplicação foi criada com o intuito de proporcionar maior comodidade à vida de quem procura por algum serviço de estética. Esse fato se dá, pois a pessoa que utiliza o aplicativo não precisa despende mais seu tempo com ligações telefônicas, esperar o atendimento e esperar a procura por horários vagos (MCLEAN, 2013).

A utilização do aplicativo é simples e intuitiva. Obrigatoriamente, o usuário seleciona sua localização e escolhe o estabelecimento, serviço desejado e data de preferência para realizar sua pesquisa. Após efetuada uma busca, o aplicativo exibe todos os horários disponíveis que obedeçam os filtros passados. e então, basta apenas escolher um e realizar o agendamento. O aplicativo também fornece todas as informações do estabelecimento que se deseja agendar horário (MCLEAN, 2013).

A Figura 3 exibe três interfaces do aplicativo Nos Trink's. A interface da esquerda é a tela de busca por um horário disponível. A interface central é a responsável por exibir as informações do salão selecionado. Por fim, a interface da direita é a responsável por exibir os horários disponíveis para agendamento, exibindo o nome do profissional, o tipo de serviço, o tempo que o serviço leva para ser realizado, o valor cobrado pelo serviço e a data que foi selecionada.

Figura 3 – Interfaces do aplicativo



Fonte: Sampaio (2013).

O aplicativo também disponibiliza a funcionalidade de envio de e-mail como lembrete para o cliente. No dia anterior à data em que agendou um horário, o cliente recebe um e-mail para lembrar o mesmo de comparecer ao salão (MCLEAN, 2013).

2.2.3 Kayak

Kayak é um aplicativo disponível para as plataformas móveis Android, iOS e Windows Phone, contando também com versão web. Possui o objetivo de simplificar a vida de quem procura realizar alguma viagem, permitindo o usuário buscar e reservar voos e hotéis. Seus usuários podem pesquisar pelos melhores preços para qualquer destino, enquanto o aplicativo se encarrega de comparar hotéis, companhias aéreas e agências de turismo (ROCHA, 2015).

Uma das funcionalidades da ferramenta é possibilitar ao usuário a pesquisa pelos voos que se adequem aos seus desejos. Tanto na versão *mobile* quanto na web, o usuário pode escolher localização, datas, se a viagem será de ida e volta e quantas pessoas farão parte da viagem (ROCHA, 2015), como apresentado na Figura 4.

Figura 4 – Tela de pesquisa por voo, versão web

KAYAK HOTÉIS VOOS CARROS Entrar

Voos

Ida e volta Só ida Várias cidades

SAO Curitiba, Brasil - Afonso Pena 1 adulto, Econômica

Incluir próximos Incluir próximos Datas flexíveis ▾

Compare com o KAYAK

CVC Agarre Submarino Viagens Expedia Encontre Sua Viagem

Compare centenas de sites de viagem ao mesmo tempo.
Encontre rapidamente **as melhores promoções.**

Alertas de preço
Seja o primeiro a saber quando os preços mudarem.

App KAYAK
O melhor app de viagens do mundo.

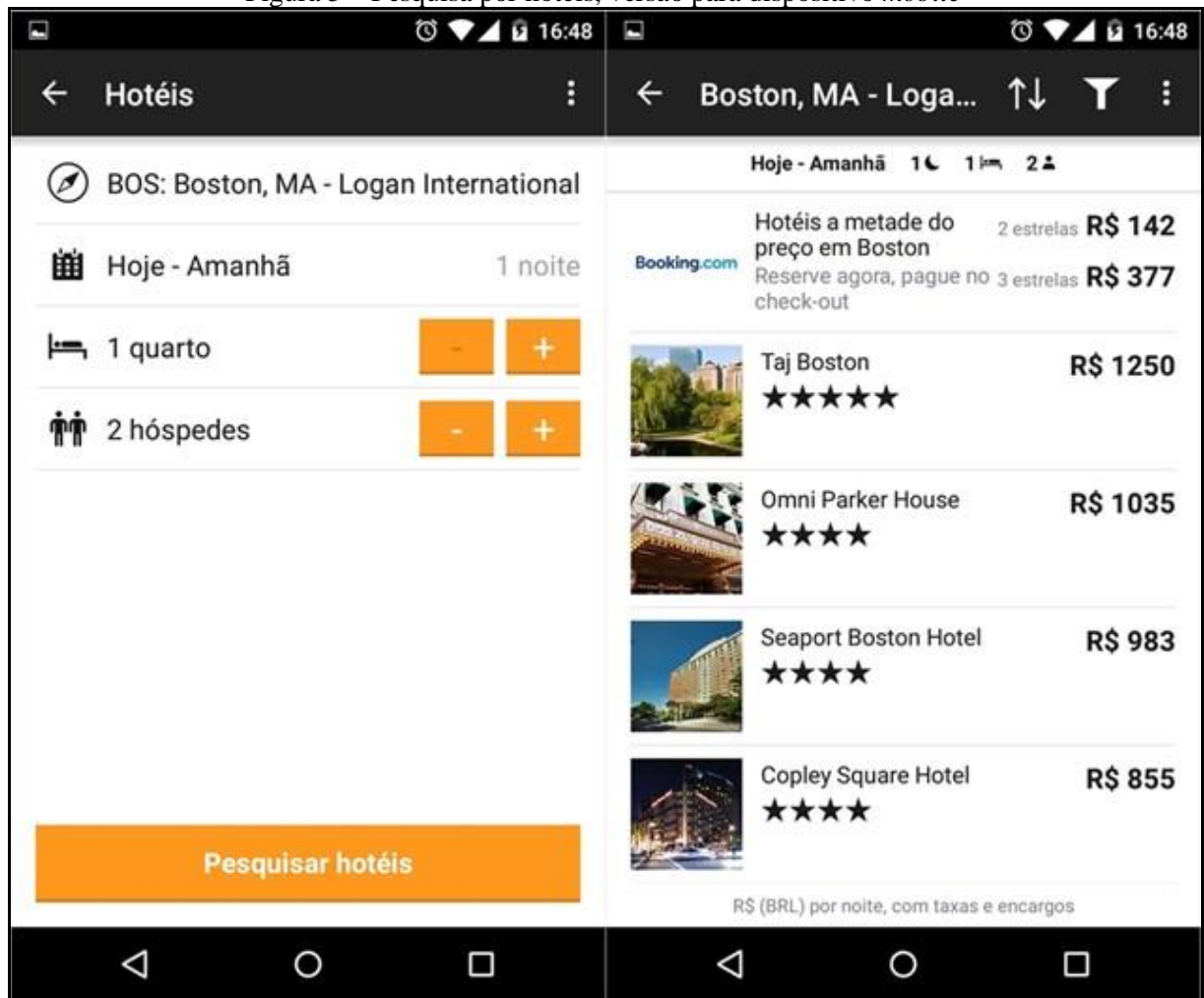
Explore
Saiba para onde viajar e por quanto.

R\$ 230
R\$ 152

Fonte: Rocha (2015).

Da mesma forma que acontece com a pesquisa por voos, o aplicativo fornece a possibilidade de uma pesquisa por hotéis, que retorna informações sobre locais onde se hospedar, taxas cobradas e preços. O aplicativo também intermedia o aluguel de veículos (ROCHA, 2015). A Figura 5 exemplifica uma pesquisa por hotéis, exibindo os resultados encontrados para possível reserva.

O Kayak já é bem difundido por todo o mundo, contando com a disponibilização de seus serviços em 31 países, com 18 idiomas oferecidos. Além das funcionalidades citadas e de outras presentes, há a promessa de mais aprimoramentos para a aplicação, focando sempre na satisfação e comodidade do cliente (ROCHA, 2015).

Figura 5 – Pesquisa por hotéis, versão para dispositivo *mobile*

Fonte: Rocha (2015).

2.2.4 Análise comparativa

Esta subseção traz uma breve comparação realizada entre cada um dos trabalhos correlatos apresentados nas Subseções 2.2.1, 2.2.2 e 2.2.3. Características similares e divergentes, plataformas para quais foram desenvolvidas e objetividade serão alguns dos pontos abordados.

O Quadro 1 exibe um comparativo sobre os três trabalhos. Pode-se observar que todas as ferramentas são disponibilizadas para o ambiente web, proporcionando mais comodidade aos clientes pelo acesso poder ser feito remotamente. Os aplicativos Nos Trinks e Kayak ainda são disponibilizados para dispositivos móveis, onde Nos Trinks possui sua versão para os sistemas operacionais Android e iOS e, Kayak possui sua versão para os sistemas operacionais Android, iOS e Windows Phone.

Quadro 1 – Comparativo dos trabalhos correlatos

Trabalho	Objetivo Principal	Plataformas
Cezerino (2011)	Agendamento no salão de beleza	Web
Trinks (2013)	Agendamento no salão de beleza	Web e <i>mobile</i> (Android e iOS)
Kayak (2004)	Reserva de voos e hotéis	Web e <i>mobile</i> (Android e iOS, Windows Phone)

Os três trabalhos apresentados tratam de prestação de serviço de alguma forma. A proposta de conclusão de curso de Cezerino (2011) tem objetivo principal de permitir usuários agendarem horários em salões de beleza. Muito similar encontra-se o aplicativo Nos Trinks, que também possibilita o usuário marcar um horário em um salão de beleza. O aplicativo Kayak se diferencia um pouco dos demais por possuir uma gama maior de funcionalidades, onde o seu uso permite que o usuário reserve voos e hotéis.

Cada qual com suas particularidades, como já relatadas nas Subseções 2.2.1 a 2.2.3, os trabalhos abordados possuem diferenças entre si. O sistema proposto por Cezerino (2011) foi desenvolvido apenas para plataforma web e funciona em apenas um salão, ou seja, ele não exibe uma relação de salões de beleza que disponibilizem serviços. Dessa forma o cliente pode realizar o agendamento em algum serviço do salão em questão.

O aplicativo Nos Trinks se diferencia por permitir que o cliente realize uma busca por salões também, não apenas exibindo serviços e demais características de um onde o sistema está hospedado, como o caso do trabalho anterior. O cliente pode escolher então para seu agendamento um salão do seu gosto e assim selecionar o serviço e demais opções. O aplicativo foi desenvolvido para ambiente web e também para as plataformas móveis Android e iOS.

O aplicativo Kayak também utiliza o conceito de exibir companhias aéreas e hotéis para seus clientes realizarem reservas. Da mesma forma que no aplicativo Nos Trinks o cliente tem uma relação de estabelecimentos, no Kayak o cliente possui uma relação de companhias aéreas, hotéis, locadoras de veículos e agências de viagens para realizar uma reserva que mais lhe agrade. O aplicativo Kayak foi desenvolvido para plataforma web, assim como para dispositivos móveis que utilizem sistemas operacionais Android, iOS ou Windows Phone.

3 DESENVOLVIMENTO DO SOFTWARE

Estão descritos nessa seção os aspectos técnicos do desenvolvimento do sistema, tais como levantamento de informações e sua especificação, que é composta pelos requisitos funcionais e não funcionais, os diagramas de atividades, os diagramas de classe e o MER elaborados. Seguindo a sequência, são apresentadas as técnicas e ferramentas utilizadas na implementação, assim como a operacionalidade do sistema. Por fim, são apresentados os resultados obtidos.

3.1 LEVANTAMENTO DE INFORMAÇÕES

O sistema em questão foi desenvolvido com o intuito de auxiliar prestadoras de serviços no gerenciamento de seus agendamentos de horários para com seus clientes. A aplicação web possibilita um gerenciamento das prestadoras de serviços por parte de seus administradores, como gerenciar seu perfil, funcionários, serviços que presta e horários de atendimentos, além de permitir a realização de agendamentos aos clientes. A aplicação *mobile* possui seu foco maior no cliente, o qual mesmo pode realizar pesquisas e efetuar agendamentos para si. Além disso, o administrador da prestadora pode também utilizar o aplicativo *mobile*, tendo em vista que há a possibilidade de visualizar a agenda no mesmo.

O módulo web permite ao administrador da prestadora de serviços manter o cadastro da mesma, definindo suas informações próprias, os serviços que presta, os funcionários que trabalham na prestadora e seus horários de funcionamento. O administrador da prestadora pode também visualizar seus agendamentos, assim como realizar um novo agendamento se um cliente assim desejar. Caso algum cliente queira efetuar um agendamento em uma empresa ele precisa ter seu cadastro no sistema. Se não possuir, o administrador poderá cadastrar o cliente pelo módulo web.

A aplicação *mobile* possui funcionalidades mais voltadas para o uso do cliente, o qual pode criar e manter seu perfil próprio. Existe a possibilidade do cliente realizar pesquisas por uma prestadora específica, ou seja, a partir do nome da mesma, ou por um serviço prestado, visualizando o perfil da prestadora de serviços desejada. O cliente pode realizar um agendamento por um serviço que deseje, visualizar seus agendamentos em aberto e os que já foram realizados, além de poder realizar um *feedback* sobre o agendamento realizado.

No aplicativo *mobile* há também a funcionalidade para o administrador da prestadora de visualizar seus agendamentos a serem prestados. Além disso, o administrador da

prestadora pode realizar as mesmas funções que o cliente, podendo agendar algum horário, visualizar agendamentos, manter perfil e realizar *feedback*.

3.2 ESPECIFICAÇÃO

Nesta seção são apresentados os requisitos funcionais e não funcionais do sistema web assim como da aplicação *mobile*, o diagrama de casos de uso, diagramas de atividades e diagramas de classes. Para a criação do diagrama de casos de uso, diagramas de atividades e diagramas de classes foi utilizada a ferramenta Enterprise Architect (EA).

3.2.1 Requisitos funcionais

Nesta subseção são apresentadas as principais funcionalidades do módulo *mobile* e web do sistema. O Quadro 2 apresenta os requisitos funcionais definidos para o módulo *mobile*, considerando sua rastreabilidade com os casos de uso, ou seja, vinculação com os casos de uso associados.

Quadro 2 – Requisitos funcionais do módulo *mobile*

Requisitos Funcionais	Caso de Uso
RF01: A aplicação deve permitir o cliente se cadastrar e manter seu perfil próprio.	UC02
RF02: A aplicação deve permitir o cliente realizar <i>login</i> no aplicativo.	UC03
RF03: A aplicação deve permitir o cliente procurar por uma prestadora de serviços específica.	UC05
RF04: A aplicação deve permitir o cliente procurar por um tipo de serviço prestado.	UC06
RF05: A aplicação deve permitir o cliente agendar um horário.	UC01
RF06: A aplicação deve permitir o cliente visualizar seus agendamentos de serviço em aberto e já realizados.	UC08
RF07: A aplicação deve permitir o administrador visualizar sua agenda.	UC14
RF08: A aplicação deve permitir o cliente realizar um <i>feedback</i> sobre o serviço utilizado.	UC07
RF09: A aplicação deve permitir o cliente visualizar o perfil de uma prestadora de serviços.	UC12

O Quadro 3 apresenta os requisitos funcionais definidos para o módulo web e sua rastreabilidade com os casos de uso.

Quadro 3 – Requisitos funcionais da aplicação web

Requisitos Funcionais	Caso de Uso
RF10: O sistema deve permitir o administrador se cadastrar e manter seu perfil próprio.	UC02
RF11: O sistema deve permitir o administrador realizar <i>login</i> no sistema web.	UC11
RF12: O sistema deve permitir o administrador manter os horários de funcionamento da sua empresa.	UC09
RF13: O sistema deve permitir o administrador manter seus profissionais.	UC10
RF14: O sistema deve permitir o administrador manter sua agenda.	UC13
RF15: O sistema deve permitir o administrador cadastrar um cliente.	UC04

3.2.2 Requisitos não funcionais

O Quadro 4 lista os requisitos não funcionais, considerando o módulo *mobile*, definidos para o sistema. O Quadro 5 lista os requisitos não funcionais, considerando o módulo web, definidos para o sistema.

Quadro 4 – Requisitos não funcionais do módulo *mobile*

Requisitos Não Funcionais
RNF01: A aplicação deverá ser desenvolvido na linguagem Java para a plataforma Android.
RNF02: A aplicação deverá suportar versões do Android compatíveis com a versão 5.0.1 ou superior.
RNF03: A aplicação deverá fazer a troca de informações com o banco de dado através de um <i>web service</i> .

Quadro 5 – Requisitos não funcionais do módulo web

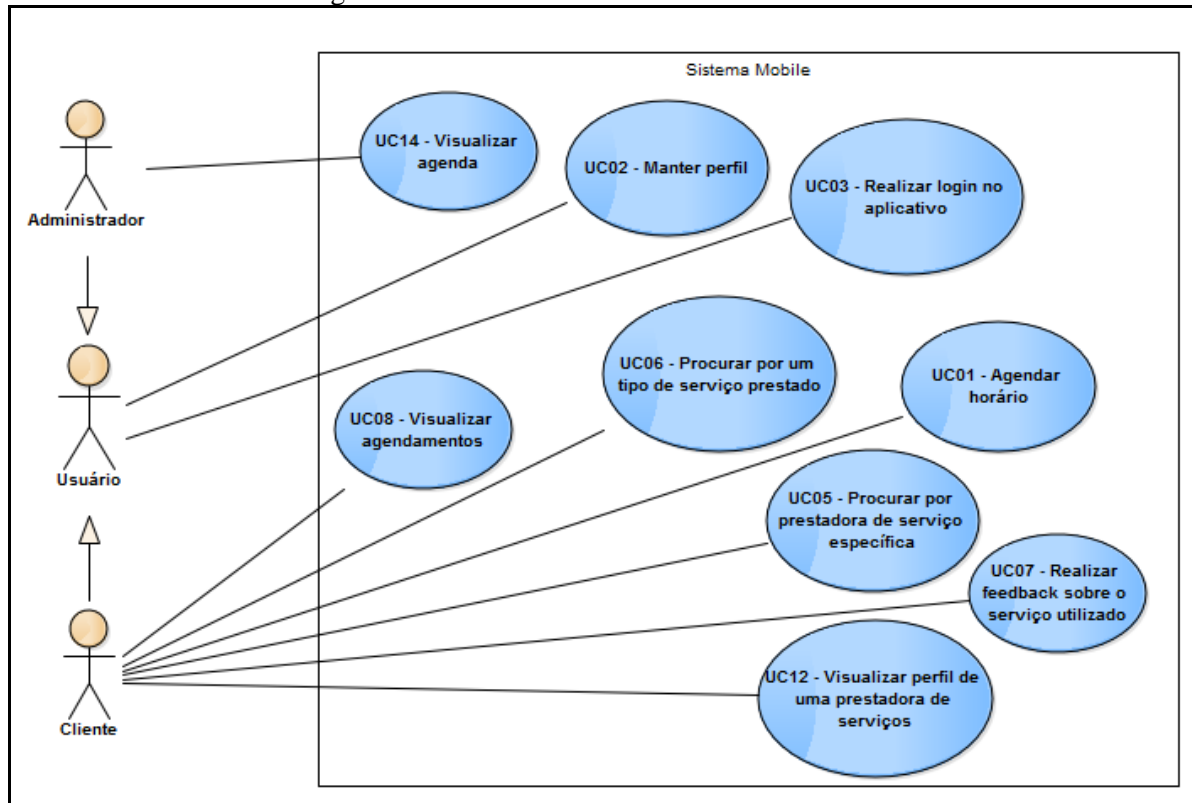
Requisitos Não Funcionais
RNF04: O sistema deverá ser acessível via Google Chrome (versão 46.0.2490.80 m ou superior) e Mozilla Firefox (versão 42.0 ou superior).
RNF05: O sistema deverá ser desenvolvido na linguagem Java
RNF06: O sistema deverá fazer a troca de informações com o banco de dados através de um <i>web service</i> .
RNF07: O sistema deverá utilizar uma base de dados MySQL (versão 5.6).

3.2.3 Modelos de casos de uso

Esta subseção apresenta a o modelo de casos de uso do sistema proposto, separados entre os módulos *mobile* e web. Os respectivos detalhamentos encontram-se descritos no Apêndice A.

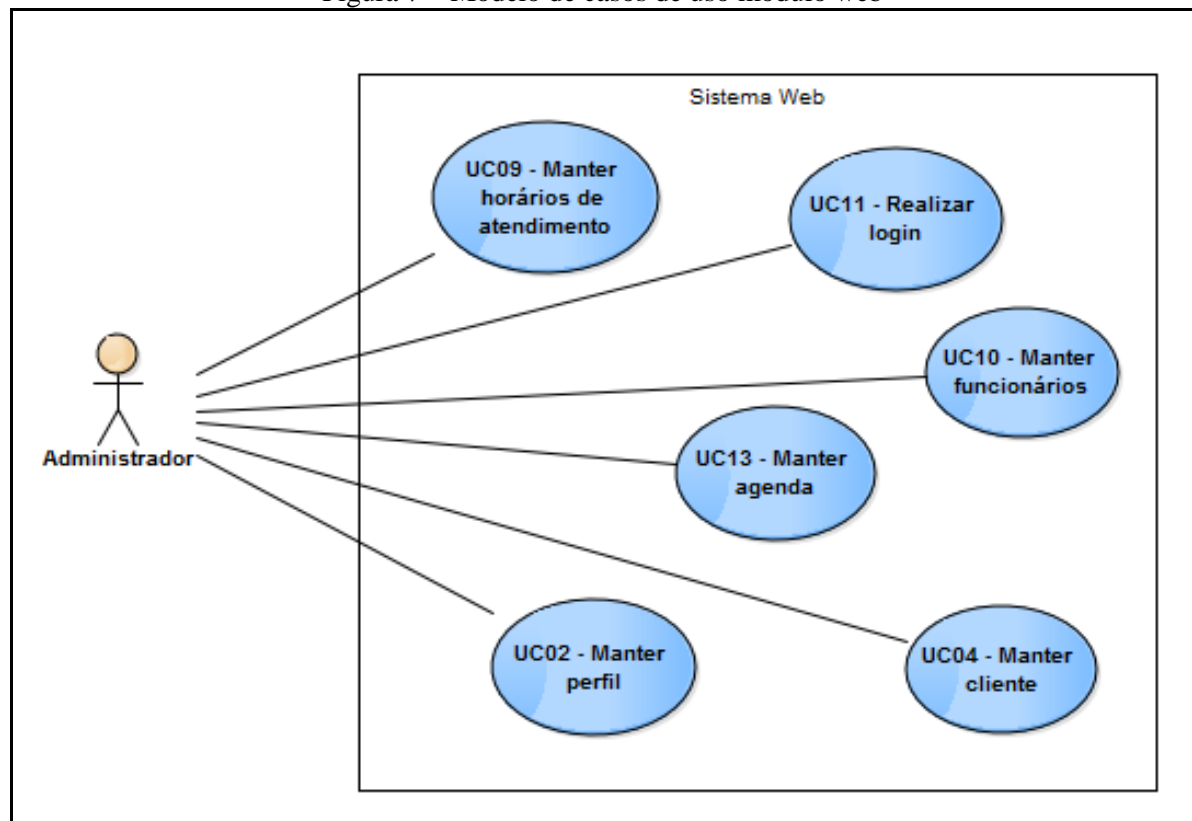
A Figura 6 retrata o modelo de casos de uso do módulo *mobile*, contendo os atores Administrador e Cliente.

Figura 6 – Modelo de casos de uso módulo *mobile*



A Figura 7 apresenta o modelo de casos de uso do módulo web, contendo o ator Administrador.

Figura 7 – Modelo de casos de uso módulo web



Uma breve descrição dos casos de uso do sistema é apresentada a seguir:

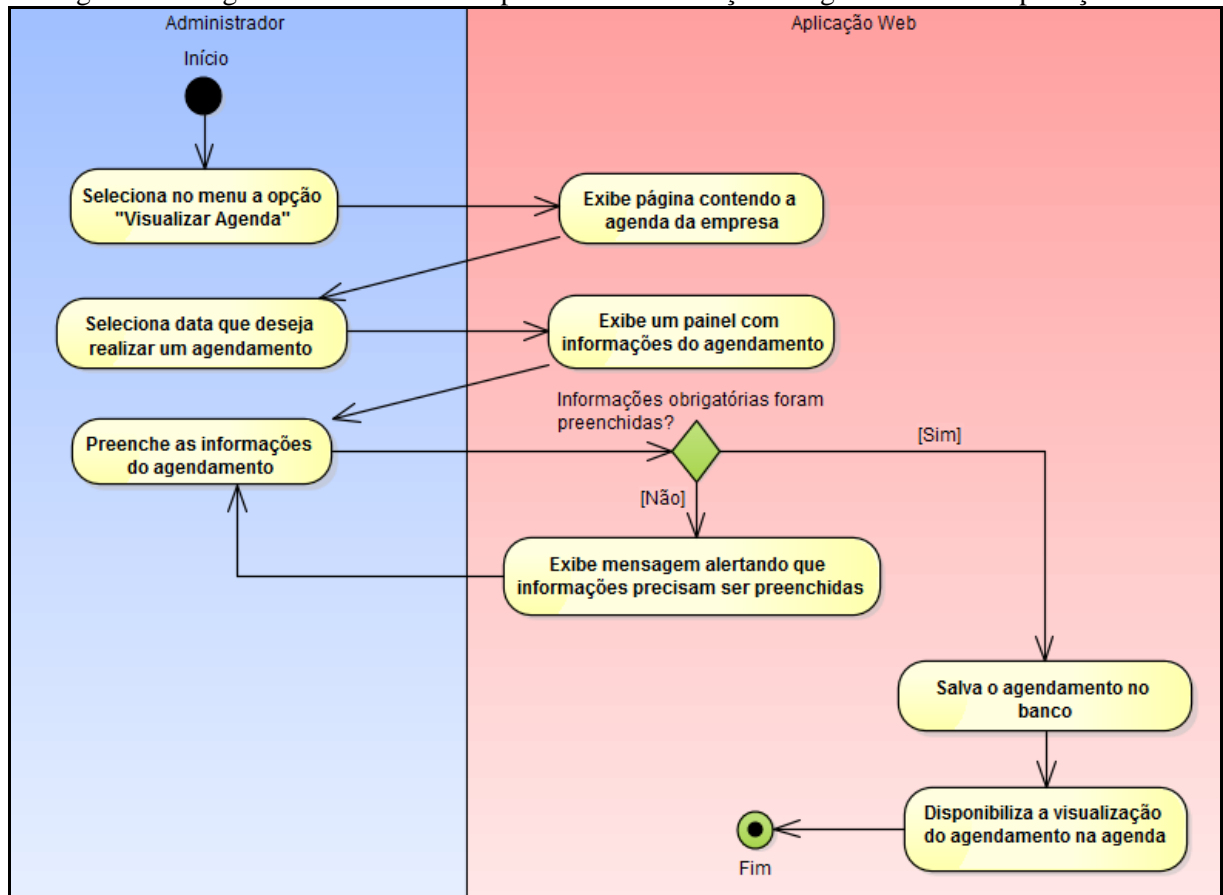
- a) UC01: permite que o cliente realize um agendamento de horário de um serviço;
- b) UC02: permite que cliente e administrador mantenham seu perfil;
- c) UC03: permite que cliente e administrador possam autenticar-se no aplicativo;
- d) UC04: permite que o administrador mantenha o perfil de algum cliente no módulo web;
- e) UC05: permite que o cliente realize uma pesquisa por prestadoras de serviço a partir do nome;
- f) UC06: permite que o cliente realize uma pesquisa por prestadoras de serviço a partir de um serviço prestado;
- g) UC07: permite que o cliente realize um comentário de um serviço que usufruiu;
- h) UC08: permite que o cliente visualize seus agendamentos que já foram finalizados e os que estão em aberto;
- i) UC09: permite que o administrador mantenha os horários de funcionamento da prestadora de serviços;
- j) UC10: permite que o administrador mantenha o cadastro dos funcionários da prestadora de serviços;
- k) UC11: permite que o administrador possa autenticar-se no módulo web;
- l) UC12: permite que o cliente visualize o perfil de uma prestadora de serviços e suas informações gerais;
- m) UC13: permite que o administrador mantenha sua agenda, realizando o gerenciamento de seus agendamentos para com seus clientes;
- n) UC14: permite que o administrador visualize seus agendamentos em aberto para o dia em questão.

3.2.4 Diagramas de atividades

Nesta subseção são apresentados os diagramas de atividade necessários para a compreensão das funcionalidades principais do módulo web e *mobile* do sistema. A Figura 8 exhibe as atividades executadas pelo administrador da prestadora de serviços e a aplicação web para que seja realizado o processo de agendamento de um horário a um cliente. Após autenticação no sistema e perfil estando com todas as informações necessárias, o administrador seleciona no menu a opção de visualizar a agenda. A partir da tela de agenda o administrador seleciona a data que deseja realizar um agendamento. Após a data ser selecionada, preenche os dados necessários para que o agendamento seja efetivado. Caso as

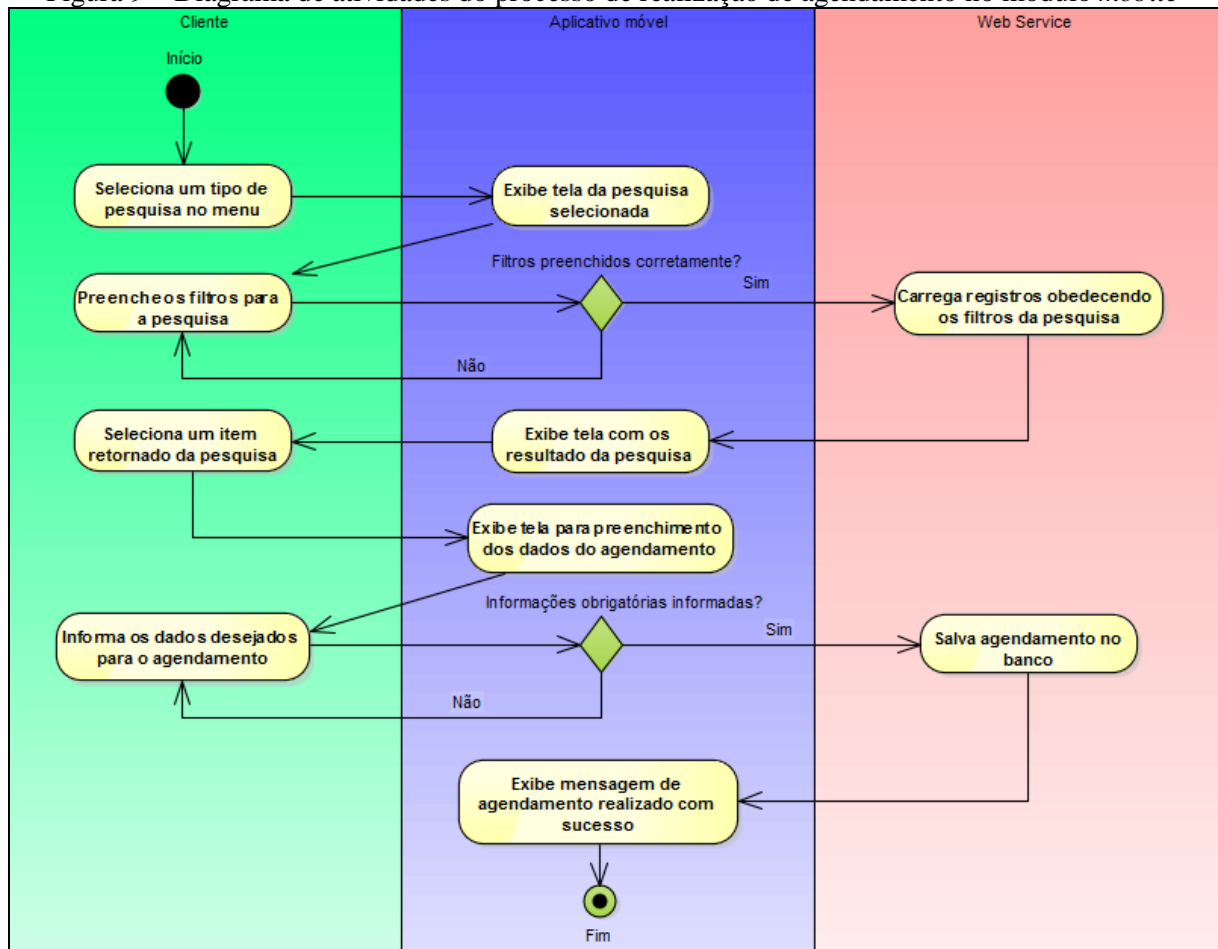
informações obrigatórias não sejam preenchidas, o aplicativo emite uma mensagem alertando ao administrador sobre o preenchimento de tais informações. Após ser gravado pela aplicação o agendamento pode ser visualizado na agenda da prestadora.

Figura 8 – Diagrama de atividades do processo de realização de agendamento na aplicação web



A Figura 9 apresenta as atividades executadas pelo cliente, o módulo *mobile* e o *web service* no processo de realização de agendamento. Estando autenticado no sistema, o cliente seleciona a opção de realizar uma busca. Com os filtros selecionados, o sistema carrega junto ao *web service* os registros encontrados que atendam à pesquisa. O cliente então seleciona um desses registros e preenche as informações necessárias para se realizar o agendamento. Caso alguma informação obrigatória para a realização do agendamento não tenha sido preenchida, o cliente é alertado e deve preenchê-la. Tendo todos os dados necessários, a aplicação juntamente com o *web service* grava o agendamento na base de dados. Por fim, é exibida uma mensagem informando que o agendamento foi realizado com sucesso.

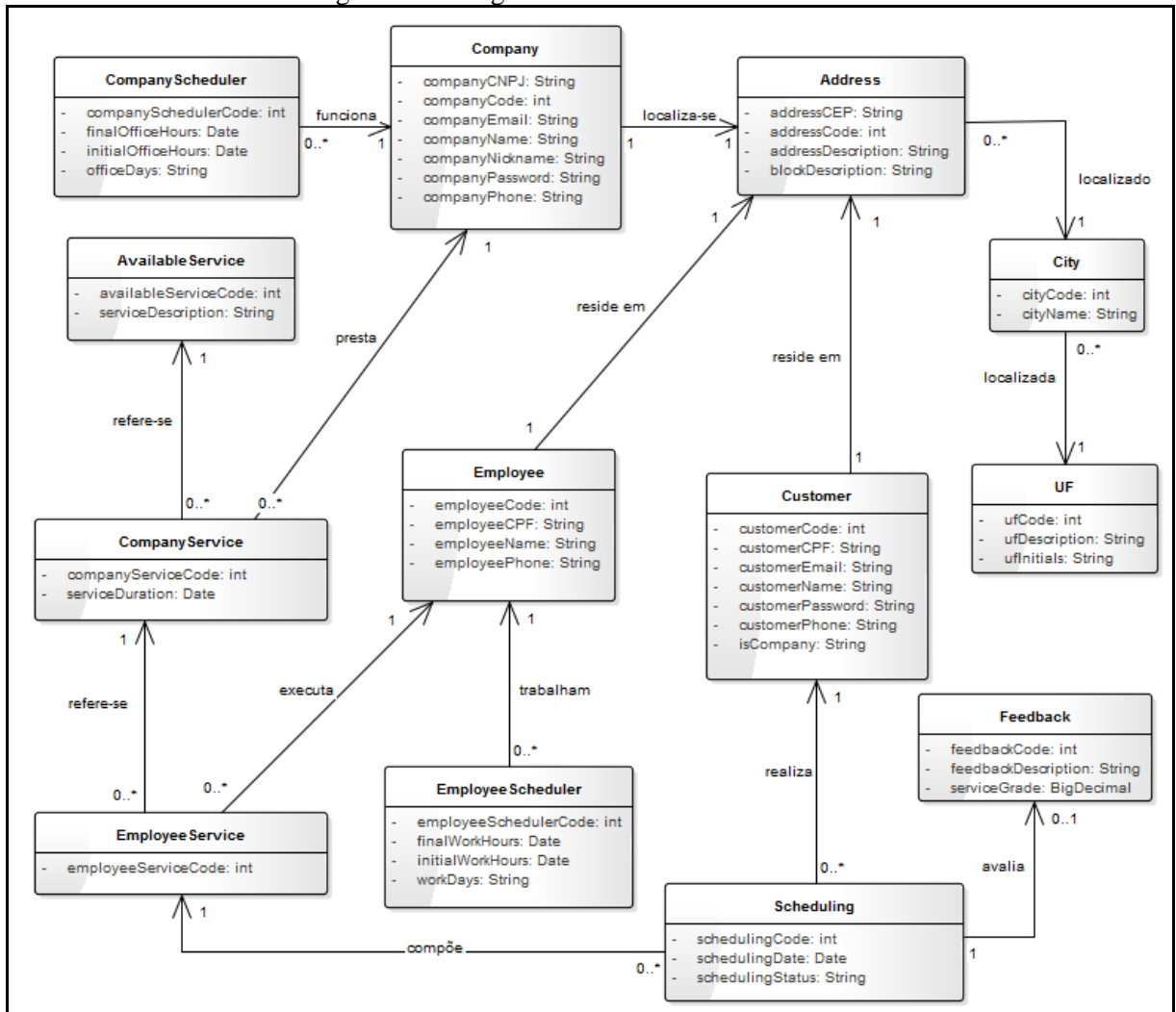
Figura 9 – Diagrama de atividades do processo de realização de agendamento no módulo *mobile*



3.2.5 Diagramas de classes

Esta subseção apresenta os diagramas de classes dos módulos web e *mobile* do sistema. A Figura 10 exibe o diagrama de classes referente ao módulo web. Em sequência uma descrição sucinta sobre as classes que compõe o diagrama.

Figura 10 – Diagrama de classes do módulo web



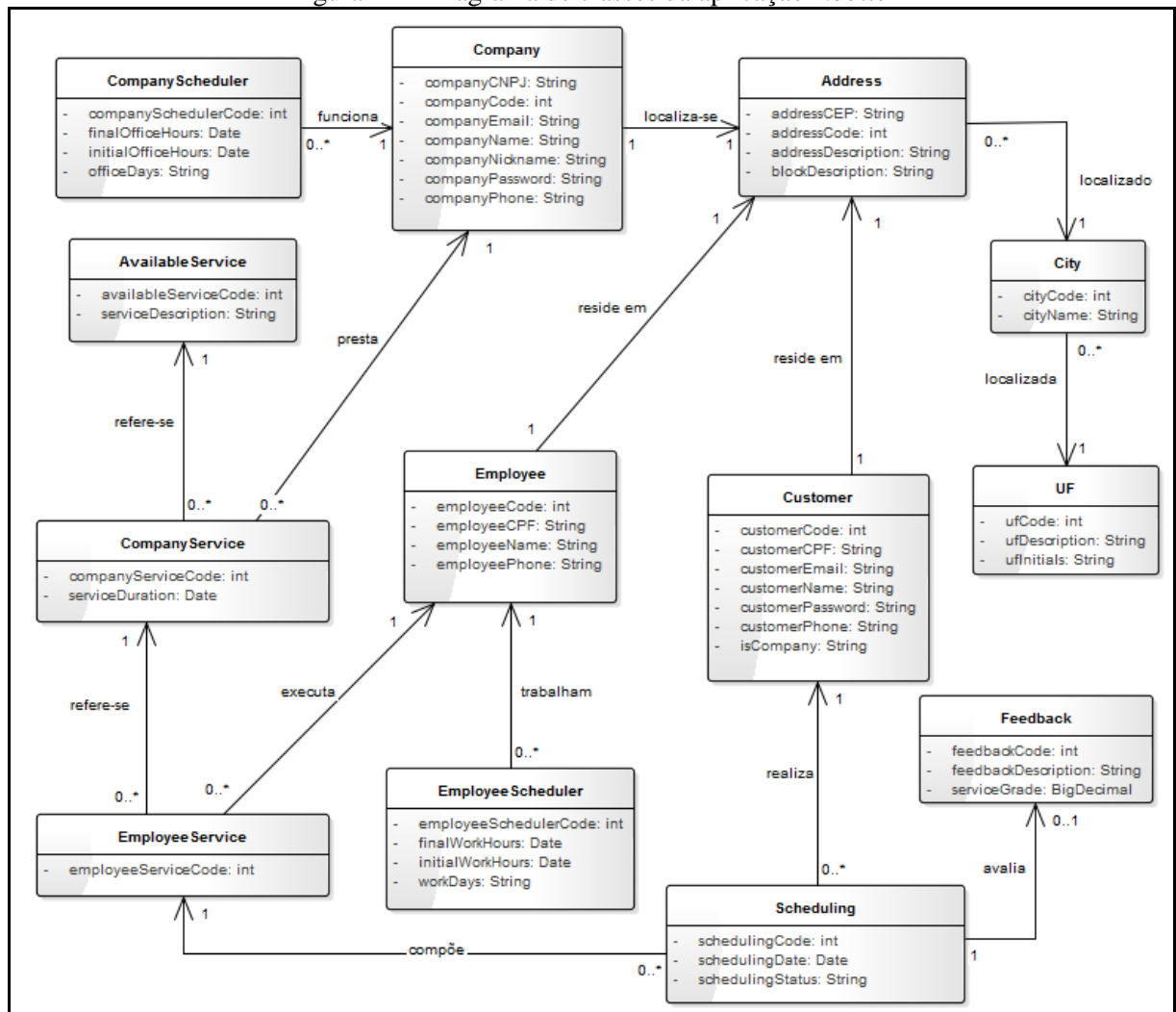
A classe `Company` refere-se às prestadoras de serviços do sistema e a classe `CompanyScheduler` armazena as informações sobre os horários de funcionamento de cada prestadora. A classe `CompanyService` é utilizada para gerenciar os serviços disponibilizados por uma prestadora de serviços. Os serviços disponíveis são referidos na classe `AvailableService`. A classe `Employee` refere-se aos funcionários de uma prestadora. A classe `EmployeeScheduler` é utilizada para se referir aos horários de trabalho dos funcionários. A classe `EmployeeService` refere-se a quais funcionários prestam os serviços oferecidos pela empresa. A classe `Customer` realiza o armazenamento das informações de um cliente que utiliza o sistema.

A classe `Address` é utilizada para definir o endereço de uma empresa, funcionário e cliente. A classe `City` faz referência à localização dos endereços contido na classe `Address`. Por fim, a classe `UF` armazena a localidade das cidades em uma unidade federativa.

A classe *Scheduling* refere-se aos agendamentos realizados no sistema. A classe *Feedback* é utilizada para armazenamento das informações de um *feedback* realizado sobre um agendamento de horário, comentários e notas sobre um serviço que foi prestado a um cliente.

A Figura 11 exibe o diagrama de classes da aplicação *mobile*. As classes presentes no aplicativos são as mesmas utilizadas no módulo web, com a diferença que a comunicação das classes do aplicativo com o banco de dados é feita com o auxílio do *web service*.

Figura 11 – Diagrama de classes da aplicação *mobile*



3.2.6 Modelo de entidade e relacionamento (MER)

A Figura 12 apresenta o MER do sistema. O dicionário de dados encontra-se no Apêndice B.

agendamento de horário;

- f) `jpr006_uf`: entidade que armazena as unidades federativas (UF) cadastradas no sistema;
- g) `jpr007_cidade`: entidade que armazena as cidades cadastradas no sistema;
- h) `jpr008_endereco`: entidade que armazena informações sobre endereço;
- i) `jpr009_horario_prestadora`: entidade que armazena os horários e dias de funcionamento das prestadoras cadastradas no sistema;
- j) `jpr010_horario_profissional`: entidade que armazena os horários e dias de trabalho de cada profissional cadastrado no sistema;
- k) `jpr011_agendamento`: entidade que armazena os agendamentos realizados no sistema;
- l) `jpr012_profissional_servico`: entidade intermediária que armazena o relacionamento entre profissionais e serviços prestados pela empresa;
- m) `jpr013_servico_disponivel`: entidade que armazena os serviços disponíveis para as prestadoras.

3.3 IMPLEMENTAÇÃO

Esta seção apresenta as técnicas e ferramentas utilizadas no desenvolvimento do trabalho, além das operacionalidades do sistema em si.

3.3.1 Técnicas e ferramentas utilizadas

A seguir são apresentadas as técnicas e ferramentas para o desenvolvimento do módulo web do sistema, assim como do módulo *mobile* e do *web service*.

3.3.1.1 Ferramentas

Foi utilizado o *Integrated Development Environment* (IDE) Eclipse para o desenvolvimento do módulo web do sistema, com a linguagem de programação Java. Para a construção das interfaces ao usuário do módulo foi utilizado o *framework* Primefaces. O Sistema de Gerenciamento de Banco de Dados (SGBD) utilizado para persistência dos dados foi o MySQL. Para mapeamento das entidades da base de dados foi utilizado o *framework* Hibernate.

O IDE Eclipse também foi utilizado para o desenvolvimento da aplicação *mobile*, com a adição do *plug-in Android Developer Tools* (ADT). Para realizar a comunicação com o *web service*, permitindo a troca de dados com o SGBD MySQL, foi utilizada a biblioteca Gson.

Para desenvolvimento do *web service* o IDE utilizado foi, também, o Eclipse, com a linguagem de programação Java juntamente com o *framework* Jersey e a biblioteca Gson para comunicação e troca de dados com a aplicação *mobile*. Para persistência de dados, da mesma forma que é feito no módulo web, foi utilizado o SGBD MySQL em conjunto com o *framework* Hibernate.

3.3.1.2 Persistência dos dados

Para realizar o armazenamento de dados do sistema foram criadas classes para mapear cada entidade da base de dados, utilizando o *framework* Hibernate, tendo como objetivo facilitar o mapeamento dos atributos da base de dados e a camada de modelo do aplicativo. O Quadro 6 exibe o arquivo *eXtensible Markup Language* (XML) responsável por configurar o *framework* Hibernate, contendo as propriedades utilizadas para realizar a configuração do próprio *framework*.

Quadro 6 – XML gerenciador do Hibernate

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
  <session-factory>
    <!-- Configurações da conexão com a base de dados -->
    <property name="connection.driver_class"> com.mysql.jdbc.Driver
  </property>
    <property name="connection.url">jdbc:mysql://localhost:3306/desenv
  </property>
    <property name="connection.username">admin</property>
    <property name="connection.password">admin</property>
    <!-- Dialeto SQL -->
    <property name="dialect">org.hibernate.dialect.MySQL5InnoDBDialect
  </property>
    <!-- Gerenciamento da sessão -->
    <property name="current_session_context_class">thread</property>
    <!-- Desabilita a cache -->
    <property name="cache.provider_class">
org.hibernate.cache.internal.NoCacheProvider </property>
    <!-- Exibe os comandos SQL -->
    <property name="show_sql">>true</property>
    <!-- Verifica se deve criar as tabelas ou verificar elas -->
    <property name="hbm2ddl.auto">validate</property>

    <!-- Mapeamento das classes -->
    <mapping class="br.com.project.domain.UF"/>
```

A seguir uma explicação sobre as propriedades do arquivo de configuração do *framework* Hibernate:

- a) `property name="connection.driver_class"`: configura o driver de conexão com o banco de dados;

- b) `property name="connection.url"`: configura a *Uniform Resource Locator* (URL) de conexão com o banco de dados;
- c) `property name="connection.username"`: configura o usuário de conexão com o banco de dados;
- d) `property name="connection.password"`: configura a senha para conexão com o banco de dados;
- e) `property name="dialect"`: configura o dialeto utilizado para criação dos comandos *Structured Query Language* (SQL);
- f) `property name="current_session_context_class"`: configura qual o contexto em que as sessões serão gerenciadas;
- g) `property name="cache.provider_class"`: configura o tratamento de cache pela sessão;
- h) `property name="show_sql"`: configura a exibição das queries SQL no console da aplicação;
- i) `property name="hbm2ddl.auto"`: configuração para o *framework* Hibernate criar, atualizar ou validar as entidades do banco de dados conforme o mapeamento realizado na aplicação;
- j) `mapping`: mapeamento das classes da aplicação com as entidades do banco de dados.

O Quadro 7 apresenta a classe utilitária `HibernateUtil` responsável por gerenciar as conexões do *framework* Hibernate com objetivo de obter as conexões com o banco de dados. Como o Hibernate trabalha com sessões, é necessária a criação de uma `SessionFactory` para o funcionamento. O método `buildSessionFactory()` carrega as configurações do *framework* definidas anteriormente, informa que o servidor está configurado com o Hibernate e realiza a criação da sessão, tornando possível o acesso ao banco de dados pelo sistema.

Quadro 7 – Classe responsável por gerenciar as conexões ao banco de dados

```

public class HibernateUtil {
    private static final SessionFactory sessionFactory =
buildSessionFactory();

    private static SessionFactory buildSessionFactory() {
        try {
            //Cria um SessionFactoring a partir do arquivo de configuração do
Hibernate
            Configuration config = new Configuration();
            config.configure();

            ServiceRegistry service = new
StandardServiceRegistryBuilder().applySettings(config.getProperties()).build();

            SessionFactory sessionFactory =
config.buildSessionFactory(service);

            return sessionFactory;
        }
        catch (Throwable ex) {
            // Exibe mensagem de erro
            System.err.println("Criação da SessionFactory falhou" + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}

```

O Quadro 8 apresenta a classe `Company` que realiza o mapeamento da entidade relacional da base de dados `jpr001_prestadora`. Anotações do *framework* Hibernate são utilizadas para realizar as configurações de mapeamento das entidades do banco de dados.

Quadro 8 – Classe responsável pelo mapeamento da entidade jpr001_prestadora

```

@Entity
@Table(name = "jpr001_prestadora")
@NamedQueries({
    @NamedQuery(name = "Company.SelectCompanyById", query = "SELECT
company FROM Company company WHERE company.companyCode = :companyCode"),
    @NamedQuery(name = "Company.SelectCompanyByLogin", query = "SELECT
company FROM Company company WHERE company.companyEmail = :companyEmail
AND company.companyPassword = :companyPassword"),
    @NamedQuery(name = "Company.SelectExistentCompany", query =
"SELECT company FROM Company company WHERE company.companyEmail =
:companyEmail"),
    @NamedQuery(name = "Company.SelectCompanyByName", query = "SELECT
company FROM Company company WHERE company.companyName LIKE
:companyName")
})
public class Company {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "cd_prestadora")
    private int companyCode;

    @Column(name = "ds_razao_social", length = 100, nullable = false)
    private String companyName;

    @Column(name = "ds_apelido", length = 50)
    private String companyNickname;

    @Column(name = "ds_cnpj", length = 20, nullable = false, unique =
true)
    private String companyCNPJ;

    @Column(name = "ds_telefone", length = 15, nullable = false)
    private String companyPhone;

    @Column(name = "ds_email", length = 50, nullable = false)
    private String companyEmail;

    @Column(name = "ds_senha", length = 30)
    private String companyPassword;

    @ManyToOne(fetch = FetchType.EAGER)
    @JoinColumn(name = "cd_endereco", referencedColumnName =
"cd_endereco", nullable = false)
    private Address address;
}

```

A seguir uma explicação sobre as anotações do framework Hibernate na classe jpr001_prestadora:

- a) **Entity:** identifica essa classe como sendo mapeamento de uma entidade do banco de dados;
- b) **Table:** identifica as propriedades da entidade do banco de dados que está sendo mapeada;
- c) **NamedQueries:** sessão onde encontram-se as consultas SQL utilizadas pelo sistema;

- d) `NamedQuery`: consulta SQL utilizada pelo sistema, contendo as informações de como a consulta é identificada e a consulta em si;
- e) `Id`: identificação de qual será a chave primária da entidade;
- f) `GeneratedValue`: define qual será o tipo de geração dos valores da chave primária da entidade, no caso a propriedade `Strategy` define que a geração será automática;
- g) `Column`: define as especificações da coluna em questão da entidade do banco de dados. Tais como nome da coluna, tamanho dos dados que serão inseridos na coluna, se a coluna pode receber valores nulos e se os dados na coluna devem ser únicos;
- h) `ManyToOne`: identifica o tipo de normalização utilizada para relacionamento entre as entidades, nesse caso, de vários para um.
- i) `JoinColumn`: define as especificações da chave estrangeira da entidade na qual está ocorrendo o relacionamento, como o nome da coluna que é a chave estrangeira, a coluna que está sendo referenciada na outra entidade e se pode receber valores nulos.

O Quadro 9 exibe o método `selectCompanyByName (String companyName)`, responsável por fazer uma pesquisa por prestadoras de serviço a partir do nome da mesma, utilizando uma `NamedQuery` do *framework* Hibernate, dentre outras técnicas do mesmo *framework*, como a criação da sessão e a execução da consulta.

Quadro 9 – Método responsável por pesquisar empresas pelo nome

```

/* Método responsável por buscar prestadoras a partir do seu nome */
@SuppressWarnings("unchecked")
public List<Company> selectCompanyByName(String companyName) {
    Session session =
HibernateUtil.getSessionFactory().openSession();
    List<Company> companyList = null;

    try {
        Query query =
session.getNamedQuery("Company.SelectCompanyByName");
        query.setString("companyName", "%" + companyName +
"%");

        companyList = query.list();

    } catch (Exception ex) {
        throw ex;
    } finally {
        session.close();
    }

    return companyList;
}

```

O relacionamento do *web service* com o banco de dados segue a mesma linha do que ocorre no módulo web, utilizando o *framework* Hibernate, evitando repetição de código já que as classes são similares.

3.3.1.3 Comunicação com o servidor

Com a finalidade de realizar a sincronização entre a aplicação *mobile* e a base de dados utilizou-se um *web service* baseado na arquitetura *Representational State Transfer* (REST) em conjunto do *framework* Jersey, com o auxílio da biblioteca Gson, com o objetivo de facilitar o processo de serialização e desserialização dos objetos. Classes que mapeiam as entidades do banco de dados foram criadas para que haja a comunicação entre a aplicação e a própria base.

A classe responsável por realizar a autenticação do cliente no aplicativo *mobile* é apresentada no Quadro 10. O construtor da classe recebe como parâmetro os métodos a serem executados quando houver uma resposta do servidor, caso ocorra um erro ele também é passado como parâmetro. O método `authenticate` envia uma requisição ao *web service* com o objetivo de autenticar o cliente no aplicativo, verificando seu e-mail e senha para saber se o mesmo possui cadastro. Para enviar os parâmetros `email` e `password` na requisição, é utilizado um *JavaScript Object Notation* (JSON) já serializado com o auxílio da biblioteca Gson. A requisição é criada passando como parâmetro o tipo da requisição, o método a ser executado no servidor caso ocorra resposta ou algum erro de conexão, em conjunto com o objeto serializado e o endereço do *web service*.

Para o aplicativo *mobile* conseguir acessar o *web service* é utilizado um endereço, formado pelo nome do servidor com as anotações `@Path` presentes na classe gerenciadora da requisição solicitada. A chamada de um método pela aplicação é realizada utilizando a URL `http://[ip_servidor]:[porta]/[nome_web_service]/[path_da_classe]/[path_do_metodo]`, onde `ip_servidor` deve ser substituído pelo IP do servidor, `porta` pelo número da porta do servidor, `nome_web_service` pelo nome do *web service* utilizado no sistema, `path_da_classe` pelo caminho da classe onde o método se localiza e `path_do_metodo` pelo caminho do método desejado.

Quadro 10 – Classe da aplicação *mobile* responsável pelas requisições de autenticação do cliente

```
public class AuthenticatorWS {

    private static final String TAG_AUTHENTICATE_REQUEST =
"TAG_AUTHENTICATE_REQUEST";
    private Listener<JSONObject> answerListener;
    private ErrorListener errorListener;

    public AuthenticatorWS(Response.Listener<JSONObject> answerListener,
Response.ErrorListener errorListener) {
        this.answerListener = answerListener;
        this.errorListener = errorListener;
    }

    public void authenticate(String email, String password) throws
JSONException {
        JSONObject obj = new JSONObject();
        obj.put("email", email);
        obj.put("password", password);

        JsonObjectRequest jsonObjReq = new
JsonObjectRequest(Method.POST,
"http://192.168.0.101:8090/WebService/cliente/autenticar", obj,
answerListener, errorListener);

        // Adding request to request queue
        ApplicationController.getInstance().addToRequestQueue(jsonObjReq,
TAG_AUTHENTICATE_REQUEST);
    }
}
```

O Quadro 11 apresenta uma parte da classe `CustomerResource` do *web service*, que é responsável por receber, processar e responder requisições realizadas pelo aplicativo *mobile*. Os métodos contidos nessa classe são os responsáveis por autenticar um cliente no aplicativo, cadastrar um novo cliente e atualizar o perfil de algum. Em todos os métodos o processo de desserialização funciona da mesma forma: a requisição chega em forma de JSON e os objetos são desserializados pela biblioteca Gson.

O método `insert`, responsável por cadastrar um novo cliente na base de dados, realiza a desserialização do cliente, de seu endereço e da cidade do mesmo, que são passados como parâmetro serializados como JSON. Após isso é utilizada a classe de acesso ao banco de

dados `CustomerDAO` para verificar se o e-mail que o cliente está tentando utilizar já não está sendo utilizado; caso não esteja, o cliente é inserido na base. Por fim esse novo cliente cadastrado é serializado como JSON, com a utilização da biblioteca Gson, convertido para `String` e retornado à aplicação *mobile*.

Quadro 11 – Classe do *web service* responsável por gerenciar as requisições do cliente

```
@Path("/cliente")
public class CustomerResource {

    @POST
    @Path("/salvar")
    @Consumes("application/json")
    @Produces("application/json")
    public String insert(String json) throws NoContentException,
    JSONException {
        JSONObject jsonObject = new JSONObject(json);
        try {
            City city = new
Gson().fromJson(jsonObject.getString("city"), City.class);
            Address address = new
Gson().fromJson(jsonObject.getString("address"), Address.class);
            Customer customer = new
Gson().fromJson(jsonObject.getString("customer"), Customer.class);

            CustomerDAO dao = new CustomerDAO();

            //verifica se o email do cliente já está cadastrado
            if
(dao.selectExistentCustomer(customer.getCustomerEmail()) != null)
                return new JSONObject().put("erro", "E-mail já
cadastrado!").toString();

            dao.insertCustomer(customer, address, city);

            return new JSONObject().put("customer", new
Gson().toJson(customer)).toString();
        } catch (JSONException jException) {
            throw new NoContentException("Erro de conexão!");
        }
    }
}
```

3.3.2 Operacionalidade da implementação

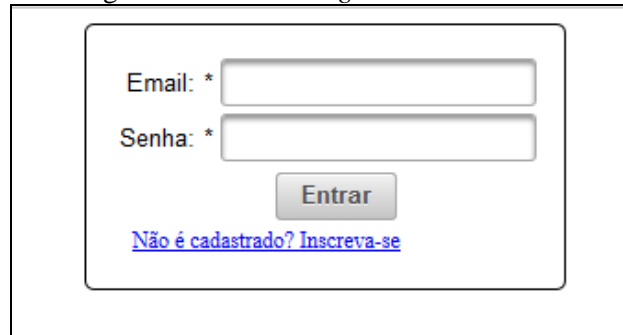
Esta subseção apresenta as operacionalidades do sistema, separados pelo módulo web com as funcionalidades do administrador da prestadora de serviço e pelo módulo *mobile*, contendo as funcionalidades destinadas a um cliente, assim como para o administrador da prestadora.

3.3.2.1 Módulo web

Para utilizar o módulo web do sistema, o administrador de uma prestadora necessita estar autenticado no sistema. A Figura 13 exibe a tela de autenticação do sistema, onde o

administrador insere suas credenciais. Caso não possua cadastro próprio, ele pode utilizar a opção Não é cadastrado? Inscreva-se, responsável por direcionar à página de cadastro de prestadora.

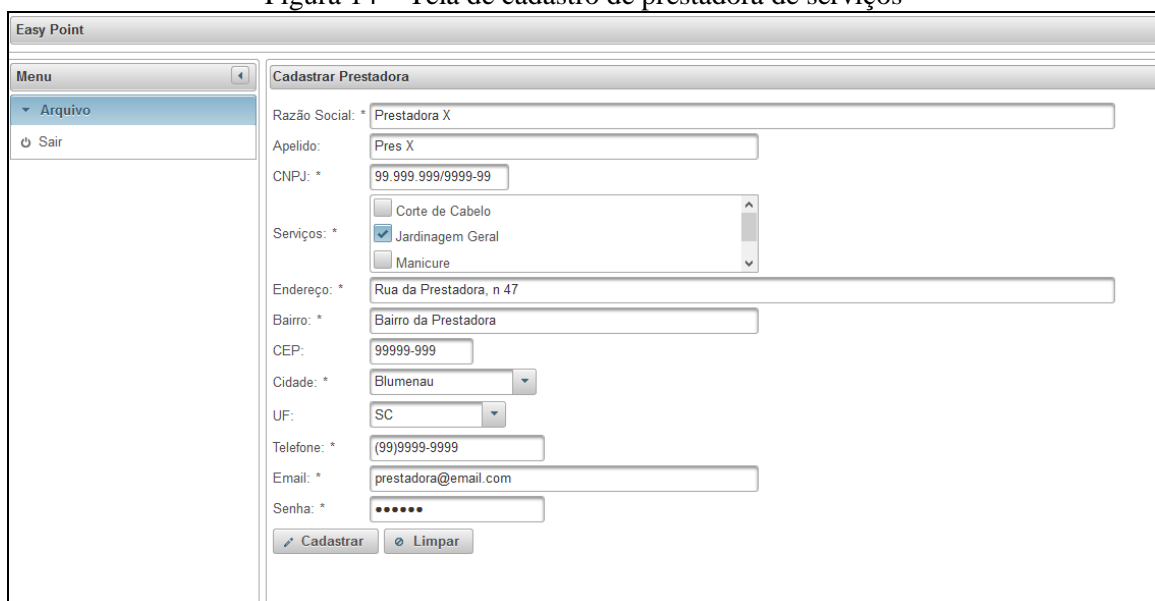
Figura 13 – Tela de *login* do módulo web



A tela de login do módulo web apresenta um formulário com dois campos de entrada: 'Email: *' e 'Senha: *'. Abaixo dos campos, há um botão 'Entrar' e um link azul que diz 'Não é cadastrado? Inscreva-se'.

A Figura 14 apresenta a tela responsável por cadastrar uma nova prestadora no sistema, onde os campos razão social, Cadastro Nacional da Pessoa Jurídica (CNPJ), serviços prestados, endereço, bairro, Código de Endereçamento Postal (CEP), cidade, Unidade Federativa (UF), telefone, e-mail e senha devem ser preenchidos para que o cadastramento seja realizado com sucesso. Após efetuar o cadastro, a página é redirecionada para a página de serviços prestados, onde o administrador pode definir a duração de cada serviço. Caso o administrador queira alterar algum dado do perfil da empresa ele poderá acessar a tela de alteração de informações, através da opção contida no menu principal. Essa tela é similar à tela de cadastramento de empresa, com a diferença que os dados da empresa já são carregados para as possíveis alterações.

Figura 14 – Tela de cadastro de prestadora de serviços



A tela de cadastro de prestadora de serviços, intitulada 'Easy Point' e 'Cadastrar Prestadora', contém um menu lateral com 'Arquivo' e 'Sair'. O formulário principal possui os seguintes campos:

- Razão Social: * Prestadora X
- Apelido: Pres X
- CNPJ: * 99.999.999/9999-99
- Serviços: * (checkboxes para Corte de Cabelo, Jardinagem Geral (selecionado), Manicure)
- Endereço: * Rua da Prestadora, n 47
- Bairro: * Bairro da Prestadora
- CEP: 99999-999
- Cidade: * Blumenau
- UF: SC
- Telefone: * (99)9999-9999
- Email: * prestadora@email.com
- Senha: * (mascarada com pontos)

Na base do formulário, há dois botões: 'Cadastrar' e 'Limpar'.

A Figura 15 exibe a tela de serviços prestados pela prestadora. Nela é possível que o administrador defina a duração que cada serviço prestado leva para ser executado.

Figura 15 – Tela de serviços prestados pela prestadora

The screenshot shows a web application interface for managing services. The main form is titled 'Serviços Prestados' and includes a 'Descrição' field containing 'Jardinagem Geral' and a 'Duração' field containing '01:30'. A 'Horário' (Time) modal window is open, allowing the user to adjust the 'Tempo' (Time) to '01:30', with 'Hora' and 'Minuto' sliders. The modal includes 'Salvar' (Save) and 'Cancelar' (Cancel) buttons. Below the form, a table displays a single entry with the number '4' and the description 'Jardinagem Geral'. At the bottom, there are navigation arrows and a page indicator showing '1'.

Após o cadastramento da empresa ou após se autenticar no sistema, o administrador visualiza o menu navegacional do módulo web, que é dividido nas seções agenda, dados da empresa, funcionários, serviços, clientes e arquivo, como apresentado na Figura 16. A seção Agenda contém a opção Visualizar Agenda, que permite ao administrador visualizar seus agendamentos e também, adicionar ou alterar esses agendamentos. A seção Dados da Empresa apresenta as opções Alterar Informações, em que o administrador pode alterar os dados cadastrais da prestadora e horário de funcionamento, em que o administrador define os horários em que a prestadora opera. A seção Funcionários serve para gerenciar os funcionários da prestadora, podendo manter o cadastro dos funcionários, na opção cadastro de funcionário e, definir as cargas horárias dos mesmos, na opção Horário do Funcionário. A seção Serviços traz a opção de gerenciar os serviços prestados. Na seção Clientes, o administrador tem a possibilidade de manter o cadastro dos clientes. Por fim, na seção Arquivo, o administrador pode navegar até a página inicial ou sair do sistema.

Figura 16 – Menu do módulo web



Para definir o horário de funcionamento da prestadora em conjunto com os dias da semana que a mesma presta atendimento, o administrador seleciona a opção *Horário de Funcionamento* do menu, que direciona a página para a tela que contém os dados referentes ao horário de funcionamento da empresa, como apresentado na Figura 17. É necessário informar o horário inicial e final de cada período de funcionamento, assim como os dias da semana em que esses horários são aplicados. Após cadastrar um período de funcionamento o mesmo é exibido no *grid* abaixo do formulário. O administrador pode alterar alguma informação de algum horário de funcionamento selecionando um item do *grid* e alterando as informações desejadas no formulário. Ao clicar no item do *grid* ele é carregado no próprio formulário.

Figura 17 – Tela de manter horário de funcionamento da prestadora

Horário Inicial ↕	Horário Final ↕
08:00	12:00
13:00	18:00
10:00	12:00
13:30	17:00

A Figura 18 exibe a tela onde o administrador pode cadastrar funcionários para a prestadora de serviços. Para que ocorra o cadastramento com sucesso os campos de nome, Cadastro de Pessoa Física (CPF), funções que o funcionário exerce, telefone, endereço, bairro, CEP, cidade e UF devem ser preenchidos. Após cadastrar um funcionário com sucesso o mesmo será exibido no *grid* abaixo dos campos do formulário de cadastro. O administrador pode também alterar dados do funcionário selecionando o mesmo no *grid*, fazendo com que os dados sejam carregados no formulário possibilitando a edição.

Figura 18 – Tela de manter funcionário

Código	Nome	CPF	Telefone
2	João	564.645.645-64	(99)9999-9999
3	Mauro	794.689.797-97	(55)4789-7444

De forma similar ao processo de definir os horários de atendimento da prestadora, ocorre o processo de manter os horários de trabalho de cada funcionário. A Figura 19 apresenta a tela onde o administrador pode manter os horários de cada funcionário, selecionando o funcionário desejado, depois definindo seu horário inicial de período de trabalho e o horário final do período, assim como os dias que esse horário se aplica.

Figura 19 – Tela de manter horário de trabalho do funcionário

Horário do Funcionário

Funcionário: *

Horário Inicial: *

Horário Final: *

Dias de Funcionamento: * Seg Ter Qua Qui Sex Sab Dom

Funcionário	Horário Inicial	Horário Final
João	08:00	12:00
João	13:00	18:00
Mauro	10:00	12:00
Mauro	13:30	17:00

A tela de cadastro de cliente, contida no módulo web e apresentada na Figura 20, pode ser utilizada pelo administrador para manter o perfil de algum cliente desejado. Ao cadastrar um cliente não significa que ele é cliente apenas da prestadora na qual o administrador o inseriu, pois esse perfil do cliente pode ser visto por todas as prestadoras que utilizam o sistema. Dessa forma o mesmo cliente de uma empresa que presta serviços de jardinagem, por exemplo, pode realizar agendamentos em um salão de beleza utilizando o mesmo perfil. Ao cadastrar um cliente é necessário o preenchimento dos campos de nome, CPF, telefone, endereço, bairro, CEP, cidade, UF, e-mail e senha. Caso o administrador deseje alterar dados de um cliente, é necessário selecionar o registro desse cliente no *grid*.

Figura 20 – Tela de manter perfil de clientes

Cadastro de Cliente

Nome: *

CPF: *

Telefone: *

Endereço: *

Bairro: *

CEP:

Cidade: *

UF:

Email: *

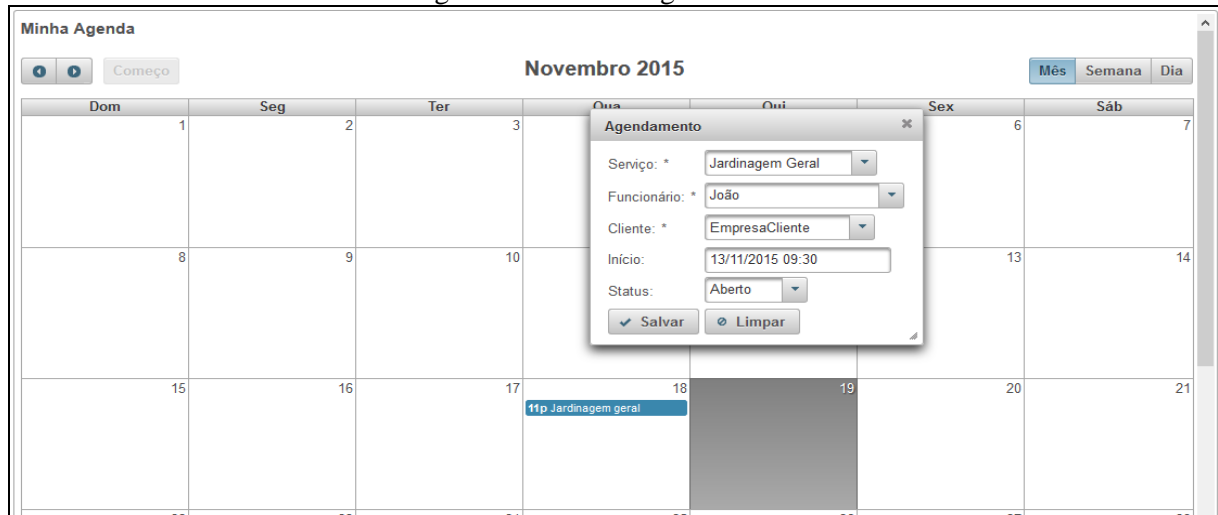
Senha:

Código	Nome	CPF	Telefone	Email
1	Cliente 1	644.543.164-58	(44)6549-7464	cliente@email.com
2	TesteCliente1	944.998.899-89	(84)9784-5646	sim
3	EmpresaCliente	12.369.789/6412-31	(47)5464-6646	empresacliente@email.com
4	ECliente	78.945.647/9413-13	(47)3338-9898	ecliente@email.com

A tela na qual o administrador pode gerenciar os agendamentos da prestadora é apresentada na Figura 21. Nela o administrador pode observar os agendamentos já realizados, assim como adicionar novos. Para adicionar um novo horário de agendamento, é necessário clicar sobre uma data que desejar e preencher os campos necessários. Qualquer alteração que

o administrador julgar necessária em algum agendamento basta clicar sobre o mesmo. Ao ser exibida a janela de informações sobre o agendamento selecionado, o administrador pode alterar as informações que desejar.

Figura 21 – Tela de agendamentos



3.3.2.2 Módulo *mobile*

O módulo *mobile* do sistema pode ser utilizado tanto pelo cliente quanto pelo administrador de uma prestadora. A seguir são apresentadas as principais funcionalidades do módulo *mobile*.

Para um cliente poder autenticar-se no aplicativo móvel ele precisa possuir seu perfil próprio, Caso não possua pode criar um através da opção *Não é cadastrado?*, presente na tela de *login*. A Figura 22 apresenta a tela de cadastramento de um cliente, onde deve-se preencher os campos de nome, CPF, endereço, bairro, cidade, Unidade Federativa (UF), telefone, e-mail e senha para que o cadastramento seja realizado com sucesso. Após o cadastramento ser efetuado, o cliente é redirecionado para a tela principal do aplicativo. Caso o cliente queira alterar algum dado do seu perfil, é necessário escolher a opção *Meu Perfil*, contida no menu principal do aplicativo. A tela de edição de cadastro é similar à tela de cadastramento de cliente, contendo os mesmos campos, porém os campos já estão preenchidos com as informações do cliente.

Figura 22 – Tela de cadastro de cliente *mobile*

Cadastrar-se

000.000.000-00

Telefone

(99)9999-9999

Endereço

Rua Exemplo, n 205

Bairro

Bairro

Cidade

Blumenau

UF

SC

Email

email@email.com

Senha

.....

Cadastrar

O cliente possui a funcionalidade de realizar pesquisas no aplicativo, tanto por serviço disponibilizado quanto por uma prestadora de serviços específica. A Figura 23 apresenta a tela de pesquisa por um serviço disponibilizado por prestadoras de serviço, onde o cliente deve selecionar o serviço por qual deseja realizar a busca. Após realizada pesquisa, o aplicativo apresenta uma lista com as prestadoras que se encaixam com o filtro aplicado, exibindo nome e telefone da prestadora, tornando possível ao cliente selecionar uma dessas prestadoras para visualizar mais informações sobre a mesma.

Figura 23 – Tela de pesquisa por serviço prestado

Pesquisa por Serviço

Corte de Cabelo

Pesquisar

Salão de Beleza S.A.
(47)9999-9999

Barbearia Dois Irmãos
(47)8888-8888

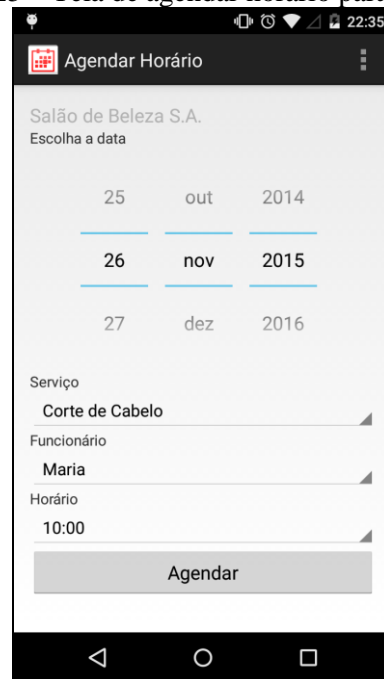
Após realizar uma pesquisa o cliente pode visualizar as informações da prestadora de serviços selecionando a mesma. Ao selecionar uma prestadora, o aplicativo exibe uma tela contendo as informações sobre nome, apelido da prestadora, telefone, e-mail e os serviços que oferece, além da opção de realizar um agendamento de horário. A Figura 24 apresenta a tela de informações da prestadora de serviços.

Figura 24 – Tela de informações da prestadora de serviços



Ao selecionar a opção *Agendar Horário*, presente na tela de informações da prestadora, o cliente é redirecionado para a tela de agendamento de horário. Nessa tela, o cliente pode agendar horário de um serviço da prestadora pesquisada. Para que o agendamento ocorra com sucesso é necessário escolher o serviço, o funcionário da prestadora que realiza o serviço desejado, a data que deseja realizar o agendamento e um horário da lista de horários disponíveis, como apresentado na Figura 25.

Figura 25 – Tela de agendar horário para o cliente



3.4 RESULTADOS E DISCUSSÕES

A aplicação *mobile* permite que o cliente realize pesquisas por prestadoras de serviço, assim como por serviços que sejam do seu interesse. Após realizadas buscas, o cliente pode visualizar o perfil da prestadora que desejar, visualizar os agendamentos que realizou e realizar novos agendamentos. O aplicativo ainda permite que o prestador possa visualizar sua agenda.

O módulo web do sistema oferece à prestadora recursos para gerenciar seus agendamentos, como gerenciamento dos serviços que presta, dos horários de atendimento e de seus funcionários. Possibilita também, o agendamento de um horário para um cliente, caso haja solicitação feita pelo mesmo. Ainda há o recurso de manter o cadastro de um novo cliente, que pode ser visualizado por todas as prestadoras, ou seja, a relação de clientes no sistema não é restrita apenas à prestadora que o inseriu no sistema.

Tendo em vista a falta de experiência no desenvolvimento da plataforma Android, algumas funcionalidades prometidas ficaram comprometidas. Como o caso da realização de algum comentário para um serviço prestado por parte do cliente, que não teve seu desenvolvimento concluído.

3.4.1 Comparativo com trabalhos correlatos

O Quadro 12 apresenta um comparativo entre algumas funcionalidades e características do sistema desenvolvido e dos trabalhos correlatos apresentados anteriormente:

o sistema desenvolvido por Cezerino (2011) e os aplicativos Nos Trinks (MCLEAN, 2013) e Kayak (ROCHA, 2015).

Quadro 12 – Comparativo entre a ferramenta EasyPoint e os correlatos

Funcionalidade / Característica	EasyPoint	Cezerino (2011)	Nos Trinks (MCLEAN, 2013)	Kayak (ROCHA, 2015)
Plataformas abordadas	Web e <i>mobile</i> (Android)	Web	Web e <i>mobile</i> (Android e Windows Phone)	Web e <i>mobile</i> (Android, iOS e Windows Phone)
Pesquisa por localidade	Não	Não	Sim	Sim
Pesquisa por empresas/prestadoras	Sim	Não	Em parte	Não
Cliente com acesso web	Não	Sim	Sim	Sim
Envio de alerta pela ferramenta ao cliente	Não	Não	Sim	Sim

Como base de informações para comparação observa-se que todos possuem versões para plataformas móveis e web, exceto o sistema desenvolvido por Cezerino (2011). A ferramenta EasyPoint desenvolvida neste trabalho opera apenas com a plataforma móvel Android, enquanto os aplicativos possuem um conjunto maior de plataformas.

A ferramenta EasyPoint possibilita a pesquisa por alguma prestadora específica caso assim o cliente desejar, porém não permite uma pesquisa pela localização destas prestadoras. Como o trabalho de Cezerino (2011) é focado em uma empresa só, dessa forma não há possibilidade de pesquisar por outras empresas, o sistema é interno para empresa. O aplicativo Nos Trinks (MCLEAN, 2013) permite que o cliente realize uma pesquisa por salão de beleza, entretanto ele é obrigado a escolher uma localização para poder visualizar apenas os salões da região selecionada. O aplicativo Kayak (ROCHA, 2015) não permite que os clientes pesquisem por hotéis, companhias aéreas ou automóveis específicos, apenas por locais onde estão esses hotéis, partida e destino de voos ou aluguel de carros.

O foco da versão web da ferramenta aqui desenvolvida é para administração da prestadora de serviços, como seus funcionários, horários e agendamentos, não possuindo funcionalidades para os clientes. Todos os outros sistemas possuem funcionalidades para seus clientes na plataforma web.

A ferramenta EasyPoint e a desenvolvida por Cezerino (2011) não possuem a funcionalidade de envio de e-mails como alguma forma de alerta a seus clientes. O aplicativo Nos Trinks (MCLEAN, 2013) fornece a funcionalidade de enviar um e-mail ao cliente um dia antes do horário agendado, como forma de reforçar e lembrar o compromisso. A ferramenta Kayak (ROCHA, 2015) oferece a possibilidade de envio de lembretes ao cliente sobre

informações dos voos, caso ocorram imprevistos, e a opção de compartilhar atualizações de status da viagem. Tudo isso por e-mail.

4 CONCLUSÕES

Neste trabalho foi apresentado o desenvolvimento de um sistema que auxiliasse prestadoras de serviços a gerenciar seus agendamentos, assim como facilitar o acesso de clientes a esses agendamentos. Além de um aplicativo *mobile* com uma visão mais voltada ao cliente, um módulo para plataforma web foi desenvolvido, onde o prestador pode realizar agendamentos para um cliente, além de outras finalidades. O objetivo principal foi alcançado, tendo os requisitos principais atendidos.

Com o aplicativo *mobile*, o cliente pode realizar agendamentos de horário por serviços que desejar, assim como manter um gerenciamento sobre os seus agendamentos. A aplicação ainda permite que o administrador de uma prestadora de serviços visualize sua agenda, mantendo assim o controle sobre seus horários, facilitando a organização. O administrador possui um controle organizacional maior na versão web, na qual pode gerenciar seus funcionários, horários de funcionamento da prestadora, serviços prestados e seus horários de agendamento.

A versão *mobile* do sistema foi desenvolvida utilizando a linguagem Java para a plataforma Android, com o auxílio da biblioteca Gson. A versão web também foi desenvolvida na linguagem Java, utilizando os *frameworks* Primefaces, para criação de *layouts* e, Hibernate, para mapeamento do banco de dados utilizado, no caso o MySQL. Da mesma forma que as versões web e *mobile* do sistema, o *web service* que realiza a comunicação entre o Android e o banco de dados MySQL foi criado utilizando a linguagem Java, em conjunto da biblioteca Gson e dos *frameworks* Jersey e Hibernate.

As tecnologias e ferramentas se mostraram satisfatórias e apropriadas para o desenvolvimento do sistema. Diante dos resultados obtidos durante o desenvolvimento, avalia-se de forma positiva o uso do sistema, pois proporciona ao prestador de serviços a facilidade ao gerenciar seus agendamentos de horário, assim como o processo para clientes realizarem agendamentos.

4.1 EXTENSÕES

Existem funcionalidades que podem ser agregadas ao desenvolvimento do sistema como forma de torná-lo mais completo. A seguir, algumas sugestões são listadas:

- a) disponibilizar na versão *mobile* mais funcionalidades para o administrador de uma prestadora de serviços;
- b) possibilitar um contato entre prestadora de serviço e cliente pelo próprio sistema, como por exemplo, um *chat* interno;

- c) permitir que o cliente mantenha uma lista de prestadoras e serviços favoritos;
- d) disponibilizar aos clientes o acesso a seus agendamentos sem necessitar conexão com a Internet;
- e) oferecer uma pesquisa por localização de prestadoras de serviço ao cliente;
- f) desenvolver uma interface web também para o acesso de clientes;
- g) disponibilizar o módulo *mobile* em mais plataformas, como Windows Phone e iOS.

REFERÊNCIAS

- CEZERINO, Aparecida. **Sistema para agendamento de serviços de beleza on-line**. 2011. 66 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- FERREIRA, Afonso. **Conheça 15 aplicativos de celular que ajudam na gestão dos negócios**. São Paulo, 2012. Disponível em: <<http://economia.uol.com.br/ultimas-noticias/redacao/2012/03/28/conheca-15-aplicativos-de-celular-que-ajudam-na-gestao-dos-negocios.jhtm>>. Acesso em: 15 set. 2015.
- HAMANN, Renan. **iOS, Android e Windows Phone: números dos gigantes comparados [infográfico]**. [S.l.], 2014. Disponível em <<http://www.tecmundo.com.br/sistema-operacional/60596-ios-android-windows-phone-numeros-gigantes-comparados-infografico.htm>>. Acesso em: 15 set. 2015.
- LABBATE, Alexsandro. **Como será a prestação de serviços ao cliente em 20 anos?**. [S.l.], 2015. Disponível em: <<http://www.mobiletime.com.br/08/10/2015/como-sera-a-prestacao-de-servicos-ao-cliente-em-20-anos-421165/news.aspx>>. Acesso em 19 out. 2015.
- MCLEAN, Viviane. **Trinks: Aplicativo para marcar horário no salão de beleza**. [S.l.], 2013. Disponível em: <<http://www.girlstuff.com.br/2013/10/trinks-aplicativo-para-marcar-horario-no-salao-de-beleza/>>. Acesso em: 20 set. 2015.
- POZZEBON, Rafaela. **Google Play ultrapassa App Store em número de aplicativos**. [S.l.], 2015. Disponível em: <<http://www.oficinadanet.com.br/post/13963-google-play-ultrapassa-app-store-em-numero-de-aplicativos/>>. Acesso em: 19 set. 2015.
- PUCCI, Claudio R. S. **Tecnologia mudou a vida das pessoas**. [S.l.], 2010. Disponível em: <http://aecambui.com.br/?menu=07&id_reg=53&id_reg2=2101>. Acesso em: 19 set. 2015.
- ROCHA, Leonardo. **Kayak: o aplicativo que é o melhor amigo de quem quer viajar com economia**. [S.l.], 2015. Disponível em: <<http://www.tecmundo.com.br/apps/78800-kayak-aplicativo-o-melhor-amigo-quer-viajar-economia.htm>>. Acesso em: 20 set. 2015.
- SAMPAIO, Gogoia. **Beleza nos trinks**. [S.l.], 2013. Disponível em: <<http://www.brconfidencial.com/beleza-nos-trinks/>>. Acesso em: 07 abr. 2015.
- SENAC. DN. **Qualidade em prestação de serviços**. 2. ed. Rio de Janeiro: Editoração Eletrônica, 2001.
- UGF, Unidade de Gestão e Formação. **O que é Prestação de Serviços?**. [S.l.], 2012. Disponível em: <<http://posugf.com.br/noticias/todas/1892-o-que-e-prestacao-de-servicos>>. Acesso em: 22 set. 2015.
- ZARATIN, Rodrigo. **Google mostra novidades do Android para empresas**. [S.l.], 2014. Disponível em: <<http://www.showmetech.com.br/google-mostra-novidades-android-para-empresas/>>. Acesso em: 22 set. 2015.

APÊNDICE A – Descrição dos Casos de Uso

Este Apêndice apresenta a descrição dos casos de uso presentes no modelo de casos de uso da subseção 3.2.3.

O Quadro 13 apresenta o detalhamento do caso de uso *Agendar horário*.

Quadro 13 – Caso de uso 01 – *Agendar horário*

Caso de uso	UC01 – <i>Agendar horário</i>
Descrição	Permite ao cliente realizar o agendamento de horário para um serviço prestado desejado.
Ator	Cliente.
Pré-condição	O ator deve estar autenticado no aplicativo <i>mobile</i> . O ator deve estar na tela de informações da prestadora
Fluxo principal	Ator seleciona a opção “ <i>Agendar Horário</i> ” Ator seleciona um serviço que deseja. Aplicação <i>mobile</i> apresenta os funcionários que prestam o serviço. Ator seleciona o funcionário que presta o serviço. Aplicação <i>mobile</i> apresenta os horários disponíveis para agendamento. Ator seleciona um horário. Ator agenda o horário desejado.
Pós-condição	Agendamento adicionado aos agendamentos do cliente para visualização. Agendamento adicionado à agenda do prestador de serviços (administrador).

O Quadro 14 apresenta o detalhamento do caso de uso *Manter Perfil*.

Quadro 14 – Caso de uso 02 – Manter perfil

Caso de uso	UC02 – Manter perfil
Descrição	Permite ao cliente e/ou administrador cadastrar-se no sistema, assim como alterar suas informações de perfil ou excluí-lo quando bem desejar.
Ator	Cliente. Administrador.
Pré-condição	Nenhuma.
Fluxo principal (Cliente)	Ator inicia aplicativo. Ator clica em "Não é cadastrado?". Aplicação direciona ator para tela de criação de perfil. Ator fornece dados necessários. Ator conclui operação de criação de perfil. É exibida mensagem de sucesso na criação do perfil. Aplicação direciona ator para tela inicial.
Fluxo principal (Administrador)	Ator acessa a página <i>web</i> do sistema. Ator clica em "Não é cadastrado? Inscreva-se". Sistema redireciona ator para tela de criação de perfil. Ator fornece dados necessários. Ator conclui operação de criação de perfil. É exibida mensagem de sucesso na criação do perfil. Sistema redireciona ator para tela de manter serviços.
Fluxo alternativo – editar perfil (Cliente)	Ator acessa a aplicação com seu nome de usuário e senha. Ator seleciona a seção "Meu Perfil" contida no menu. Ator altera os dados desejados do seu perfil. Ator conclui operação. Aplicativo exibe uma mensagem de sucesso na edição do perfil.
Fluxo alternativo – editar perfil (Administrador)	Ator acessa a página <i>web</i> do sistema com seu nome de usuário e senha. Ator seleciona a opção "Alterar Informações" contida no menu. Ator altera os dados desejados do seu perfil. Ator conclui operação. Exibida uma mensagem de sucesso na edição do perfil.
Pós-condição	Ator criou ou editou seu perfil.

O Quadro 15 apresenta o detalhamento do caso de uso Realizar login no aplicativo.

Quadro 15 – Caso de uso 03 – Realizar login no aplicativo

Caso de uso	UC03 – Realizar <i>login</i> no aplicativo
Descrição	Permite ao cliente e administrador efetuar <i>login</i> no aplicativo <i>mobile</i> .
Ator	Cliente. Administrador.
Pré-condição	O ator deve estar cadastrado no sistema.
Fluxo principal	Ator preenche os campos de nome de usuário e senha. Aplicativo valida nome de usuário e senha. Aplicativo direciona ator para a tela principal.
Pós-condição	O ator entra autenticado no aplicativo.

O Quadro 16 apresenta o detalhamento do caso de uso Manter cliente.

Quadro 16 – Caso de uso 04 – Manter cliente

Caso de uso	UC04 – Manter cliente
Descrição	Permite ao administrador manter o perfil de um cliente.
Ator	Administrador.
Pré-condição	O ator deve estar autenticado no sistema. O ator deve estar na tela de manter perfil de cliente.
Fluxo principal	Ator preenche os dados do cliente. Ator finaliza cadastro do cliente. É exibida mensagem de sucesso na criação do cadastro do cliente.
Fluxo alternativo – editar cadastro de cliente	Ator seleciona o cliente que deseja editar dados no <i>grid</i> . Sistema preenche os campos do formulário com os dados do cliente selecionado. Ator edita os dados que desejar. Ator finaliza operação. É exibida mensagem de sucesso na edição do cadastro do cliente.
Fluxo alternativo – excluir cadastro de cliente	Ator seleciona o cliente que deseja excluir no <i>grid</i> . Sistema preenche os campos do formulário com os dados do cliente selecionado. Ator clica no botão excluir. Sistema limpa dados do formulário e exclui registro da base. É exibida mensagem de sucesso na exclusão do cadastro do cliente.
Pós-condição	Ator criou, editou ou excluiu cadastro de cliente.

O Quadro 17 apresenta o detalhamento do caso de uso Procurar por prestadora de serviço específica”.

Quadro 17 – Caso de uso 05 – Procurar por prestadora de serviço específica

Caso de uso	UC05 – Procurar por prestadora de serviço específica
Descrição	Permite ao cliente procurar por uma prestadora de serviço pelo nome, onde o cliente digita o nome da prestadora desejada e o aplicativo lhe exhibe todos os resultados encontrados.
Ator	Cliente.
Pré-condição	O ator deve estar autenticado no sistema.
Fluxo principal	Ator seleciona a opção “Pesquisa por Prestadora” no menu. Ator digita o nome da prestadora de serviço desejada no campo de busca. Ator seleciona a opção “Pesquisar”. Aplicativo exhibe os resultados encontrados.
Pós-condição	O ator tem a visualização das prestadoras de serviço com o nome que o mesmo realizou a pesquisa.

O Quadro 18 apresenta o detalhamento do caso de uso Procurar por um tipo de serviço prestado.

Quadro 18 – Caso de uso 06 – Procurar por um tipo de serviço prestado

Caso de uso	UC06 – Procurar por um tipo de serviço prestado.
Descrição	Permite ao cliente realizar uma pesquisa por um tipo de serviço que ele deseja contratar. Sendo exibida assim, uma lista com as prestadoras que prestam esse tipo de serviço.
Ator	Cliente.
Pré-condição	O ator deve estar autenticado no sistema.
Fluxo principal	Ator seleciona a opção “Pesquisar por serviço” no menu. Ator seleciona o serviço prestado que o mesmo deseja no campo de busca. Ator seleciona a opção “Pesquisar”. Aplicativo exibe uma lista com as prestadoras do serviço solicitado.
Pós-condição	O ator tem a visualização das prestadoras de serviço que prestam o serviço pesquisado.

O Quadro 19 apresenta o detalhamento do caso de uso Realizar feedback sobre o serviço utilizado.

Quadro 19 – Caso de uso 07 – Realizar feedback sobre o serviço utilizado

Caso de uso	UC07 – Realizar feedback sobre o serviço utilizado.
Descrição	Permite ao cliente, depois de ter o serviço prestado para si, realizar um <i>feedback</i> , com uma nota de classificação e um comentário para o serviço.
Ator	Cliente.
Pré-condição	O ator deve estar autenticado no sistema. O <i>status</i> do agendamento deve estar em “Finalizado”.
Fluxo principal	Ator seleciona o agendamento. Ator seleciona opção “Realizar <i>feedback</i> ”. Ator seleciona uma nota e escreve um comentário. Ator seleciona opção “Concluir”. Aplicativo exibe mensagem de realização de <i>feedback</i> concluída.
Pós-condição	<i>Feedback</i> sobre o serviço prestado realizado com sucesso.

O Quadro 20 apresenta o detalhamento do caso de uso Visualizar Agendamentos.

Quadro 20 – Caso de uso 08 – Visualizar Agendamentos

Caso de uso	UC08 – Visualizar Agendamentos.
Descrição	Permite ao cliente visualizar seus agendamentos de horários, com os serviços que ainda não foram prestados e os que já foram prestados. Assim como excluir algum agendamento da lista.
Ator	Cliente.
Pré-condição	O ator deve estar autenticado no sistema. Para excluir um agendamento o <i>status</i> deve ser “Aberto”. O ator deve ter algum agendamento de horário.
Fluxo principal	O ator seleciona a opção “Agendamentos” no menu. O aplicativo exibe a tela de agendamentos com dois botões: “Pendentes” e “Realizados”. O ator seleciona uma das opções de agendamentos. O aplicativo exibe os agendamentos.
Fluxo alternativo – excluir agendamento	O ator seleciona um agendamento. O ator seleciona a opção excluir. O aplicativo exibe uma mensagem de confirmação para saber se o ator realmente deseja excluir o agendamento. O ator confirma a exclusão do agendamento.

	O aplicativo exibe uma mensagem de exclusão de agendamento bem sucedida.
Pós-condição	O ator visualiza seus agendamentos.

O Quadro 21 apresenta o detalhamento do caso de uso Manter horários de atendimento.

Quadro 21 – Caso de uso 09 – Manter horários de atendimento

Caso de uso	UC09 – Manter horários de atendimento.
Descrição	Permite ao administrador gerenciar os horários de atendimento da sua prestadora de serviços. Assim como alterar ou excluir algum horário de atendimento.
Ator	Administrador.
Pré-condição	O ator deve estar autenticado no sistema.
Fluxo principal	O ator seleciona a opção “Horário de Funcionamento” no menu. O aplicativo exibe a tela de horários de funcionamento da prestadora de serviços. Ator preenche os dados para o cadastro. Ator finaliza cadastro do horário de funcionamento. É exibida mensagem de sucesso na criação do cadastro do horário de funcionamento.
Fluxo alternativo – editar horário de funcionamento	Ator seleciona o horário de atendimento que deseja editar no grid. Sistema preenche os campos do formulário com os dados do horário de funcionamento selecionado. Ator edita os dados que desejar. Ator finaliza operação. É exibida mensagem de sucesso na edição do cadastro do horário de funcionamento.
Fluxo alternativo – excluir horário de funcionamento	Ator seleciona o horário de atendimento que deseja excluir no grid. Sistema preenche os campos do formulário com os dados do horário de atendimento selecionado. Ator clica no botão excluir. Sistema limpa dados do formulário e exclui registro da base. É exibida mensagem de sucesso na exclusão do cadastro do horário de atendimento.
Pós-condição	Ator criou, editou ou excluiu cadastro de horário de atendimento.

O Quadro 22 apresenta o detalhamento do caso de uso Manter funcionários.

Quadro 22 – Caso de uso 10 – Manter funcionários

Caso de uso	UC10 – Manter funcionários.
Descrição	Permite ao administrador gerenciar os funcionários da sua prestadora de serviços. Assim como alterar ou excluir algum funcionário.
Ator	Administrador.
Pré-condição	O ator deve estar autenticado no sistema.
Fluxo principal	O ator seleciona a opção “Cadastro de Funcionário” no menu. O aplicativo exibe a tela de cadastro de funcionário da prestadora de serviços. Ator preenche os dados para o cadastro do funcionário. Ator finaliza cadastro do funcionário. É exibida mensagem de sucesso na criação do cadastro do funcionário.
Fluxo alternativo – editar funcionário	Ator seleciona o funcionário que deseja editar no grid. Sistema preenche os campos do formulário com os dados do

	funcionário selecionado. Ator edita os dados que desejar. Ator finaliza operação. É exibida mensagem de sucesso na edição do cadastro do funcionário.
Fluxo alternativo – excluir funcionário	Ator seleciona o funcionário que deseja excluir no grid. Sistema preenche os campos do formulário com os dados do funcionário selecionado. Ator clica no botão excluir. Sistema limpa dados do formulário e exclui registro da base. É exibida mensagem de sucesso na exclusão do cadastro do funcionário.
Pós-condição	Ator criou, editou ou excluiu cadastro de funcionário.

O Quadro 23 apresenta o detalhamento do caso de uso *Realizar login*.

Quadro 23 – Caso de uso 11 – *Realizar login*

Caso de uso	UC11 – <i>Realizar login</i> .
Descrição	Permite ao administrador autenticar-se no módulo web do sistema.
Ator	Administrador.
Pré-condição	O ator deve estar cadastrado no sistema.
Fluxo principal	Ator preenche os campos de nome de usuário e senha. Sistema valida nome de usuário e senha. Sistema direciona ator para a página inicial.
Pós-condição	O ator entra autenticado no sistema.

O Quadro 24 apresenta o detalhamento do caso de uso *Visualizar perfil de uma prestadora de serviços*.

Quadro 24 – Caso de uso 12 – *Visualizar perfil de uma prestadora de serviços*

Caso de uso	UC12 – <i>Visualizar perfil de uma prestadora de serviços</i> .
Descrição	Permite ao cliente visualizar o perfil de uma prestadora de serviços, com suas informações básicas. O cliente pode visualizar o perfil da prestadora após uma pesquisa realizada.
Ator	Cliente.
Pré-condição	O ator deve estar autenticado no aplicativo.
Fluxo principal	Ator realiza uma busca (vide UC05 ou UC06). Aplicativo exibe prestadoras de serviço. Ator seleciona uma prestadora de serviço. Aplicativo exibe o perfil da prestadora com suas informações.
Pós-condição	O ator visualiza o perfil da prestadora de serviço.

O Quadro 25 apresenta o detalhamento do caso de uso *Manter agenda*.

Quadro 25 – Caso de uso 13 – Manter agenda

Caso de uso	UC13 – Manter agenda.
Descrição	Permite ao administrador da prestadora de serviços manter agendamentos para com seus clientes. Podendo cadastrar um novo agendamento ou editar um existente.
Ator	Administrador.
Pré-condição	O ator deve estar autenticado no sistema.
Fluxo principal	Ator seleciona a opção “Visualizar Agenda” no menu. Sistema exibe a agenda da prestadora de serviços. Ator seleciona uma data que deseja realizar um agendamento. Sistema exibe tela com os dados necessários para gravar um agendamento. Ator preenche dados necessários. Ator finaliza a operação de realizar agendamento. Sistema exibe mensagem confirmando sucesso ao realizar agendamento.
Fluxo alternativo – editar agendamento	Ator seleciona um agendamento visível na agenda. Sistema exibe tela com os dados do agendamento. Ator altera os dados desejados. Ator finaliza a operação de editar agendamento. Sistema exibe mensagem confirmando sucesso ao editar agendamento.
Pós-condição	O ator agendou um horário novo ou editou um existente.

O Quadro 26 apresenta o detalhamento do caso de uso Visualizar agenda.

Quadro 26 – Caso de uso 14 – Visualizar agenda

Caso de uso	UC12 – Visualizar agenda.
Descrição	Permite ao administrador visualizar os agendamentos de horário por dia, com o status dos horários, podendo ser “Aberto” ou “Finalizado”.
Ator	Administrador.
Pré-condição	O ator deve estar autenticado no aplicativo.
Fluxo principal	Ator seleciona a opção “Agenda”. Aplicativo exibe os horários de agendamento do dia atual. Ator visualiza os horários de agendamento do dia.
Pós-condição	O ator visualiza sua agenda.

APÊNDICE B – Dicionário de Dados

Este Apêndice apresenta a descrição das entidades do banco de dados do servidor apresentadas na seção de especificação deste trabalho. Os tipos de dados utilizados nos atributos são:

- a) *int*: armazena números inteiros;
- b) *varchar*: armazena conjuntos de caracteres;
- c) *datetime*: armazena data e hora;
- d) *decimal*: armazena números de precisão e escala fixos.

O Quadro 27 apresenta o dicionário de dados da tabela *jpr001_prestadora*.

Quadro 27 – Tabela *jpr001_prestadora*

jpr001_prestadora					
Armazena as prestadoras de serviços					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
<i>cd_prestadora</i>	Código da prestadora	<i>INT</i>		Sim	Não
<i>cd_endereco</i>	Código do endereço	<i>INT</i>		Não	Sim
<i>ds_razao_social</i>	Descrição da razão social	<i>VARCHAR</i>	100	Não	Não
<i>ds_cnpj</i>	CNPJ da prestadora	<i>VARCHAR</i>	20	Não	Não
<i>ds_telefone</i>	Telefone da prestadora	<i>VARCHAR</i>	15	Não	Não
<i>ds_email</i>	E-mail da prestadora	<i>VARCHAR</i>	50	Não	Não
<i>ds_senha</i>	Senha da prestadora	<i>VARCHAR</i>	30	Não	Não

O Quadro 28 apresenta o dicionário de dados da tabela *jpr002_servico_prestadora*.

Quadro 28 – Tabela *jpr002_servico_prestadora*

jpr002_servico_prestadora					
Armazena os serviços realizados pelas prestadoras					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
<i>cd_servico_prestadora</i>	Código do serviço	<i>INT</i>		Sim	Não
<i>cd_prestadora</i>	Código da prestadora	<i>INT</i>		Não	Sim
<i>cd_servico_disponivel</i>	Código de serviços disponíveis	<i>INT</i>		Não	Sim
<i>hr_duracao</i>	Tempo de duração do serviço	<i>DATETIME</i>		Não	Não

O Quadro 29 apresenta o dicionário de dados da tabela *jpr003_profissional*.

Quadro 29 – Tabela *jpr003_profissional*

jpr003_profissional					
Armazena os profissionais					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
cd_profissional	Código do profissional	<i>INT</i>		Sim	Não
cd_endereco	Código do endereço	<i>INT</i>		Não	Sim
nm_profissional	Nome do profissional	<i>VARCHAR</i>	50	Não	Não
ds_cpf	CPF do profissional	<i>VARCHAR</i>	20	Não	Não
ds_telefone	Telefone do profissional	<i>VARCHAR</i>	15	Não	Não

O Quadro 30 apresenta o dicionário de dados da tabela *jpr004_cliente*.

Quadro 30 – Tabela *jpr004_cliente*

jpr004_cliente					
Armazena os clientes					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
cd_cliente	Código do profissional	<i>INT</i>		Sim	Não
cd_endereco	Código do endereço	<i>INT</i>		Não	Sim
nm_cliente	Nome do cliente	<i>VARCHAR</i>	50	Não	Não
ds_cpf	CPF do cliente	<i>VARCHAR</i>	20	Não	Não
ds_telefone	Telefone do cliente	<i>VARCHAR</i>	15	Não	Não
ds_email	E-mail do cliente	<i>VARCHAR</i>	50	Não	Não
ds_senha	Senha do cliente	<i>VARCHAR</i>	30	Não	Não
eh_empresa	Indicador se o cliente é ou não uma prestadora também	<i>VARCHAR</i>	1	Não	Não

O Quadro 31 apresenta o dicionário de dados da tabela *jpr005_feedback*.

Quadro 31 – Tabela *jpr005_feedback*

jpr005_feedback					
Armazena os <i>feedbacks</i> realizados sobre os serviços prestados					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
cd_feedback	Código do <i>feedback</i>	<i>INT</i>		Sim	Não
vl_nota_servico	Valor da nota de serviço	<i>DECIMAL</i>		Não	Não
ds_feedback	Descrição do <i>feedback</i>	<i>VARCHAR</i>	500	Não	Não

O Quadro 32 apresenta o dicionário de dados da tabela *jpr006_uf*.

Quadro 32 – Tabela *jpr006_uf*

jpr006_uf					
Armazena as unidades federativas					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
cd_uf	Código da UF	<i>INT</i>		Sim	Não
sg_uf	Sigla da UF	<i>VARCHAR</i>	2	Não	Não
ds_uf	Descrição da UF	<i>VARCHAR</i>	45	Não	Não

O Quadro 33 apresenta o dicionário de dados da tabela *jpr007_cidade*.

Quadro 33 – Tabela jpr007_cidade

jpr007_cidade					
Armazena as cidades					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
cd_cidade	Código da cidade	<i>INT</i>		Sim	Não
cd_uf	Código da UF	<i>INT</i>		Não	Sim
ds_cidade	Descrição da cidade	<i>VARCHAR</i>	30	Não	Não

O Quadro 34 apresenta o dicionário de dados da tabela jpr008_endereco.

Quadro 34 - Tabela jpr008_endereco

jpr008_endereco					
Armazena os endereços					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
cd_endereco	Código do endereço	<i>INT</i>		Sim	Não
cd_cidade	Código da cidade	<i>INT</i>		Não	Sim
ds_endereco	Descrição do endereço	<i>VARCHAR</i>	200	Não	Não
ds_bairro	Bairro do endereço	<i>VARCHAR</i>	45	Não	Não
ds_cep	CEP do endereço	<i>VARCHAR</i>	45	Não	Não

O Quadro 35 apresenta o dicionário de dados da tabela jpr009_horario_prestadora.

Quadro 35 – Tabela jpr009_horario_prestadora

jpr009_horario_prestadora					
Armazena os horários de funcionamento das prestadoras					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
cd_horario_prestadora	Código do horário da prestadora	<i>INT</i>		Sim	Não
cd_prestadora	Código da prestadora	<i>INT</i>		Não	Sim
hr_funcionamento_ini	Horário de início de um período	<i>DATETIME</i>		Não	Não
hr_funcionamento_fin	Horário de fechamento de um período	<i>DATETIME</i>		Não	Não
ds_dias_funcionamento	Dias de funcionamento da prestadora	<i>VARCHAR</i>	30	Não	Não

O Quadro 36 apresenta o dicionário de dados da tabela jpr10_horario_profissional.

Quadro 36 – Tabela `jpr10_horario_profissional`

jpr10_horario_profissional					
Armazena os horários de trabalho dos profissionais					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
<code>cd_horario_profissional</code>	Código do horário do profissional	<i>INT</i>		Sim	Não
<code>cd_profissional</code>	Código do profissional	<i>INT</i>		Não	Sim
<code>hr_trabalho_ini</code>	Horário inicial de trabalho do profissional	<i>DATETIME</i>		Não	Não
<code>hr_trabalho_fin</code>	Horário final de trabalho do profissional	<i>DATETIME</i>		Não	Não
<code>ds_dias_trabalho</code>	Dias de trabalho do profissional	<i>VARCHAR</i>	30	Não	Não

O Quadro 37 apresenta o dicionário de dados da tabela `jpr011_agendamento`.

Quadro 37 – Tabela `jpr011_agendamento`

jpr011_agendamento					
Armazena os agendamentos realizados					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
<code>cd_agendamento</code>	Código do agendamento	<i>INT</i>		Sim	Não
<code>cd_profissional_servico</code>	Código do serviço prestado	<i>INT</i>		Não	Sim
<code>cd_cliente</code>	Código do cliente	<i>INT</i>		Não	Sim
<code>cd_feedback</code>	Código do <i>feedback</i>	<i>INT</i>		Não	Sim
<code>dt_agendamento</code>	Data do agendamento	<i>DATETIME</i>		Não	Não
<code>ds_status</code>	Situação do agendamento	<i>VARCHAR</i>	10	Não	Não

O Quadro 38 apresenta o dicionário de dados da tabela `jpr012_profissional_servico`.

Quadro 38 – Tabela `jpr012_profissional_servico`

jpr012_profissional_servico					
Armazena os profissionais ligados a determinado serviço					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
<code>cd_profissional_servico</code>	Código do serviço do profissional	<i>INT</i>		Sim	Não
<code>cd_servico_prestadora</code>	Código do serviço da prestadora	<i>INT</i>		Não	Sim
<code>cd_profissional</code>	Código do profissional	<i>INT</i>		Não	Sim

O Quadro 39 apresenta o dicionário de dados da tabela `jpr013_servico_disponivel`.

Quadro 39 – Tabela jpr013_servico_disponivel

jpr013_servico_disponivel					
Armazena os serviços disponibilizados para prestadoras					
Campo	Descrição	Tipo	Tamanho	Chave primária	Chave estrangeira
cd_servico_disponivel	Código do serviço	<i>INT</i>		Sim	Não
ds_servico	Descrição do serviço	<i>VARCHAR</i>	45	Não	Não