UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

JUAN HERMANN SERAMUCIN

**SISTEMA WEB PARA CONTROLE E GERENCIAMENTO DE AGENDA EM CENTROS ESTÉTICOS**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO

2018

JUAN HERMANN SERAMUCIN

**SISTEMA WEB PARA CONTROLE E GERENCIAMENTO DE AGENDA EM CENTROS ESTÉTICOS**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 1, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Profa. Andreia Scariot Beulke

Coorientador: Prof. João Guilherme Brasil Pichetti

PATO BRANCO

2018

**RESUMO**

Seramucin, Juan Hermann. SISTEMA *WEB* PARA CONTROLE E GERENCIAMENTO DE AGENDA EM CENTROS ESTÉTICOS. 2018. 26 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2018.

A Internet, desde seus primórdios, vem oferecendo recursos e padrões cada vez mais eficientes para auxiliar seus usuários a desenvolverem suas tarefas e rotinas, pois permite acesso rápido e independente de plataforma.

Este projeto tem como objetivo desenvolver um sistema *web* que controle e gerencie os agendamentos de dias, horários e serviços oferecidos pelos centros estéticos. Além disso o cliente poderá agendar um serviço de forma rápida e eficiente.

Dentre as tecnologias a serem utilizadas no trabalho proposto, está a linguagem Java e o framework Spring que tem por objetivo acelerar o processo de configuração da aplicação. O Spring Security permite deixar o sistema mais seguro, pois garante a autenticação e autorização dos usuários, permitindo que somente quem é cadastrado no sistema tenha acesso aos recursos restritos. Além disso, é importante que o sistema seja responsivo, pois o acesso poderá ocorrer por diversos dispositivos, como tablets, celulares e computadores e, assim, é importante manter o layout e acesso às finalidades adequados para cada dispositivo utilizado para o acesso.

**Palavras-chave**: Centros estéticos. Sistema *web.* PrimeFaces.

**ABSTRACT**

Seramucin, Juan Hermann. SISTEMA *WEB* PARA CONTROLE E GERENCIAMENTO DE AGENDA EM CENTROS ESTÉTICOS. 2018. 26 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2018.

The Internet, since at begginings, has been providing more eficiente features standards to help it users developt their tasks and routines as it allows fast access and a platform-independent.

This Project aims to develop a web system that controls and manages schedules of days, times and services offered by “beauty centers”. In addon cliente will be are to schedules a servisse quitcky and efficiently.

Among the tecnologies used iin the proposed work in Java language and the framework Spring that ais to accelerate the process of configuring the app. The Spring Security allows you to make the system safe, as it ensures user authentication and authorization, allowing only those who are registred in the system to access on the restricted resources. In addition, it is importante that the system be responsive because access may through a lot of devices, such as tablets, cell phones and computers, so it is importante to montain the layout and access to the propriate purposes for each device used for access.

**Keywords**: Web system. Beauty centers. PrimeFaces.

**LISTA DE FIGURAS**

[Figura 1 – Casos de uso 20](#_Toc516679263)

[Figura 2 - Banco de dados 25](#_Toc516679264)

**LISTA DE QUADROS**

[Quadro 1 - Lista de ferramentas e tecnologias utilizadas 16](#_Toc516679201)

[Quadro 2 - Requisitos funcionais do sistema 19](#_Toc516679202)

[Quadro 3 – Requisitos não funcionais do sistema 20](#_Toc516679203)

[Quadro 4 – Operação “incluir” dos casos de uso de cadastro 21](#_Toc516679204)

[Quadro 5 – Operação “alterar” dos casos de uso de cadastro 22](#_Toc516679205)

[Quadro 6 - Operação “excluir” dos casos de uso de cadastro 22](#_Toc516679206)

[Quadro 7 – Operação “consultar” dos casos de uso de cadastro 23](#_Toc516679207)

[Quadro 8 - Caso de uso para o chat do sistema 23](#_Toc516679208)

[Quadro 9 – Caso de uso dos relatórios do sistema. 24](#_Toc516679209)

[Quadro 10 – Caso de uso de bloqueio do sistema. 24](#_Toc516679210)

[Quadro 11 - Caso de uso validação de cliente do sistema. 25](#_Toc516679211)

**LISTA DE SIGLAS E ACRÔNIMOS**

ABIHPEC Associação Brasileira da Indústria de Higiene Pessoal, Perfumaria e Cosméticos

AJAX *Asynchronous Javascript and XML*

CGI Comitê Gestor da Internet

CSS *Cascading Style Sheet*

HTML *HyperText Markup Language*

HTTP *HyperText Protocol  Transfer*

IoC Inversão de Controle

RIA *Rich Internet Applications*

URL *Uniform Resource Locator*

WWW *World Wide Web*

**SUMÁRIO**

[1 INTRODUÇÃO 8](#_Toc517163403)

[1.1 CONSIDERAÇÕES INICIAIS 8](#_Toc517163404)

[1.2 OBJETIVOS 9](#_Toc517163405)

[1.2.1 Objetivo Geral 9](#_Toc517163406)

[1.2.2 Objetivos Específicos 9](#_Toc517163407)

[1.3 JUSTIFICATIVA 10](#_Toc517163408)

[1.4 ESTRUTURA DO TRABALHO 11](#_Toc517163409)

[2 APLICAÇÕES WEB 12](#_Toc517163410)

[2.1 DESENVOLVIMENTO WEB TRADICIONAL 12](#_Toc517163411)

[2.2 DESENVOLVIMENTO COM INTERFACE RICA 13](#_Toc517163412)

[2.3 COMUNICAÇÃO SÍNCRONA E ASSÍNCRONA 14](#_Toc517163413)

[3 MATERIAIS E MÉTODO 16](#_Toc517163414)

[3.1 MATERIAIS 16](#_Toc517163415)

[3.2 MÉTODO 16](#_Toc517163416)

[4 RESULTADOS 18](#_Toc517163417)

[4.1 ESCOPO DO SISTEMA 18](#_Toc517163418)

[4.2 MODELAGEM DO SISTEMA 19](#_Toc517163419)

[4.2.1 Requisitos Funcionais 19](#_Toc517163420)

[4.2.2 Requisitos Não Funcionais 19](#_Toc517163421)

[REFERÊNCIAS 26](#_Toc517163422)

# 1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais, os objetivos e a justificativa da realização deste trabalho.

## 1.1 CONSIDERAÇÕES INICIAIS

A Internet, desde seus primórdios, vem oferecendo recursos e padrões cada vez mais eficientes para auxiliar seus usuários a desenvolverem suas tarefas e rotinas, pois permite acesso rápido e independente de plataforma.

Com a tecnologia disponível em vários segmentos da indústria e do comércio, os sistemas de informação permitem que muitos serviços sejam realizados de forma automática, além de permitir o gerenciamento e controle das informações registradas por meio desses sistemas.

É notório o aumento de usuários de Internet no Brasil proporcionado pela facilidade de acesso disponível por diversos dispositivos que oferecem esse serviço, como, computadores, celulares e TVs e pela variedade de atividades que a Internet oferece seja para uso profissional ou pessoal. Dados do Comitê Gestor da Internet (CGI) apontam que cerca de 54% das residências possuem acesso à Internet. O número de empresas que possuem *web* sites varia de acordo com o porte, sendo 87% para grandes empresas, 75% para médias empresas e 52% para pequenas empresas. Para perfis em redes sociais a pesquisa aponta que 60% dos usuários são de empresas de grande porte, 56% de médio porte e 50% de pequeno porte. Dentre as atividades mais frequentes realizadas na Internet os dados revelam que 99% dos usuários utilizam a Internet para envio de e-mails, 94% para buscar informações sobre produtos ou serviços, 88% para fazer pagamentos, transferências e consultas bancárias via Internet banking. (CGI, 2016).

Além dos serviços citados existem softwares ou aplicações *web* que visam oferecer a realização de atividades específicas de acordo com o segmento da empresa. No que tange o comércio de centros estéticos, por exemplo, as atividades realizadas são relacionadas ao tratamento corporal e facial de seus clientes. As alternativas de serviços a serem ofertados

variam de acordo com o tipo de cliente que envolve os gêneros feminino e masculino de várias idades (SEBRAE, 2015).

De acordo com a Associação Brasileira da Indústria de Higiene Pessoal, Perfumaria e Cosméticos (ABIHPEC), no mercado de centros de estética fatura cerca de 38 bilhões no Brasil, com crescimento de 10% ao ano e o Brasil é o terceiro maior mercado do mundo nessa área (ABIHPEC, 2015).

No que tange os salões de beleza desempenham um papel fundamental para elevar a autoestima de seus clientes. Segundo a Euromonitor International (2017), a indústria de produtos voltados para beleza e cuidados pessoais, cresceu à uma taxa média ao ano de 13% entre 2004 e 2014. Em um mercado que movimenta anualmente US$30 bilhões no país, esse nível de crescimento anual adiciona US$2 bilhões dentro do mercado global de beleza.

Nesse contexto, este projeto tem como objetivo desenvolver um sistema *web* que controle e gerencie os agendamentos de dias, horários e serviços oferecidos pelos centros estéticos. O cliente poderá agendar um serviço de forma rápida e eficiente, além de se comunicar diretamente com o estabelecimento por via chat próprio.

## 1.2 OBJETIVOS

A seguir são apresentados o objetivo geral e específicos do sistema proposto neste trabalho. O objetivo geral está relacionado com o resultado principal da realização deste trabalho e os objetivos específicos complementam o geral em termos de funcionalidades do sistema.

### 1.2.1 Objetivo Geral

Desenvolver um sistema *web* para controle e gerenciamento de agenda de um centro estético.

### 1.2.2 Objetivos Específicos

Para alcançar o objetivo geral, serão estabelecidos os seguintes objetivos específicos.

- Permitir a centralização e automatização das informações cadastradas para o agendamento.

- Proporcionar flexibilidade, agilidade e eficiência no agendamento de clientes para realização de um serviço.

- Permitir a consulta de dados por meio de relatórios.

## 1.3 JUSTIFICATIVA

Para uma empresa manter-se no mercado é necessário, além de oferecer serviços e produtos de qualidade, que esteja em constante atualização seja para os produtos e serviços que oferecem ou pelo atendimento a seus clientes. Para isso, é fundamental que as empresas invistam em tecnologias, equipamentos e serviços de qualidade para que possam auxiliar no desenvolvimento de suas atividades e funções. Os sistemas de informação são ferramentas necessárias para o controle e gerenciamento dessas atividades, pois dentre suas diversas funcionalidades, está a parte de cadastros e relatórios, que são funcionalidades básicas mas de grande importância, pois permite a centralização dos mais variados dados em um banco de dados. Além disso, existem funcionalidades específicas visando atender as necessidades de cada tipo de empresa seja no nível operacional, tático ou estratégico.

A facilidade de acesso às mais variadas informações e o atendimento à serviços de vários segmentos proporcionados pela uso da tecnologia faz com que as pessoas necessitem cada vez mais de atendimentos rápidos e pontuais.

No que compete aos centros de estética, os clientes são pessoas de gêneros, raças e idade distintas que buscam a prestação de serviços relacionados ao tratamento estético corporal e facial. Esses centros trabalham com hora marcada para cada serviço prestado que tem um tempo limite entre o início e o fim do atendimento e são realizados por profissionais distintos de acordo com a sua especialidade. Assim, a agenda desse tipo de estabelecimento é importante para o controle e gerenciamento dos serviços oferecidos para que ocorram nos horários estabelecidos para cada cliente. Normalmente, o agendamento de um determinado serviço ocorre por uma chamada telefônica ou mensagem instantânea, sendo necessário um atendente para controlar a agenda entre os profissionais que executam os serviços e os clientes do estabelecimento.

Nesse sentido, este trabalho se justifica pela proposta de oferecer aos clientes de um centro de estética a possibilidade de realizar um agendamento on-line, escolhendo dia, horário, tipo de serviço e profissional que deseja para executar o serviço. Além disso, o sistema visa centralizar os dados dos agendamentos em um único banco de dados o que permite que a empresa possa realizar consultas a diversas informações, como, por exemplo, serviços realizados por profissional, serviços mais realizados, entre outros.

Dentre as principais tecnologias a serem utilizadas no desenvolvimento do trabalho proposto, está a linguagem Java e o framework Spring que tem por objetivo acelerar o processo de configuração da aplicação. Além disso, é baseado nos padrões de Inversão de Controle (IoC) e Injeção de Dependências (SOUZA, 2015). O Spring Security permite deixar o sistema mais seguro, pois garante a autenticação e autorização dos usuários, permitindo que somente quem é cadastrado no sistema tenha acesso aos recursos restritos. Além disso, é importante que o sistema seja responsivo, pois o acesso poderá ocorrer por diversos dispositivos, como tablets, celulares e computadores e, assim, é importante manter o layout e acesso às finalidades adequados para cada dispositivo utilizado para o acesso.

## 1.4 ESTRUTURA DO TRABALHO

Este texto está organizado em capítulos e este é o primeiro que apresenta a introdução, as considerações iniciais, os objetivos e a justificava de realização deste trabalho. O Capítulo 2 apresenta o referencial teórico. No Capítulo 3 são apresentados os materiais e o método utilizados para o desenvolvimento do trabalho. No Capítulo 4 está o resultado da realização do trabalho que apresenta a modelagem do sistema.

# 2 APLICAÇÕES WEB

Este capítulo apresenta a fundamentação teórica deste trabalho relacionado as tecnologias utilizadas para desenvolvimento do trabalho. Estas tecnologias que estão relacionadas ao conceito de aplicações ricas e *Asynchronous Javascript and XML* (AJAX).

## 2.1 DESENVOLVIMENTO WEB TRADICIONAL

As aplicações *web* possuem características específicas com relação a sistemas convencionais, pois em uma aplicação *web*, além dos aspectos de Engenharia definidos nas aplicações tradicionais, envolvem características específicas, como o ambiente baseado em navegação, por exemplo. Além disso, é importante não confundir uma aplicação *web* com um *web* site, pois uma aplicação *web* enfatiza aspectos relacionados à aplicabilidade e funcionalidade, enquanto os *web* sites têm ênfase na apresentação, estruturação e navegação.

Em uma aplicação *web* tudo é processado por um servidor que recebe uma requisição do cliente, ou seja, as aplicações *web* são feitas por requisição e resposta, no qual o cliente (usuário) faz uma requisição para um servidor e o servidor responder para o cliente. O browser permite ao usuário solicitar um recurso e o servidor pode responder com vários recursos, conforme a solicitação, como: uma página *HyperText Markup Language* (HTML), uma figura ou documentos que serão apresentas ao usuário. Os conteúdos dessas páginas são estilizados com recursos do *Cascading Style Sheet* (CSS) e do JavaScript para comandos interativos.

A comunicação entre o cliente e servidor é realizada por meio de um protocolo, chamado *HyperText Protocol  Transfer* (HTTP) que é baseado em requisições e respostas. Uma requisição possui um método HTTP, a página que será acessada e os dados do formulário. A resposta possui um status que informa se a solicitação foi realizada, o tipo e o conteúdo.

Cada requisição o cliente específica o nome de um método a ser aplicado em um recurso no servidor e o *Uniform Resource Locator* (URL) desse recurso. Os métodos HTTP mais conhecidos são o *get* e o *post*. No método *get* o URL é dado como argumento, no qual o servidor responderá retornando os dados identificados por este URL. O método *post* especifica o URL de um recurso que pode tratar os dados fornecidos no corpo do pedido.

Conforme destaca Fraternali, Rossi e Sánchez-Figueroa (2010) a *World Wide Web* (WWW) foi uma plataforma para acessar recursos estáticos e dinâmicos codificados em HTML e que a interação do usuário era limitada a navegar pelas páginas por meio dos links e inserir dados em formulários sendo páginas *web* eram simples e limitadas. O autor enfatiza, ainda, que as primeiras tentativas de estender a funcionalidade das interfaces (como *applets* em Java e scripts do lado cliente) enriqueceram a navegação baseada em HTML, com objetos interativos, efeitos de animações e validações de entrada. No entanto, a utilização desses novos recursos causaram problemas de padronização e problemas de arquitetura.

A partir disso, o desenvolvimento de aplicações *web* passaram por transformações, pois houve um progresso no aumento de funcionalidades de tarefas realizadas na *web* abrindo espaço para soluções *web* mais modernas que se assemelham a aplicações desktop e permitem interações sofisticas com o usuário, processamento do lado cliente, comunicação assíncrona e multimídia.

Assim, as tecnologias evoluíram e sugiram as *Rich Internet Applications* (RIAs) que desempenham um importante papel no desenvolvimento de interfaces *web*.

## 2.2 DESENVOLVIMENTO COM INTERFACE RICA

Conforme destaca Fraternali, Rossi e Sánchez-Figueroa (2010) o termo RIA refere-se a um conjunto heterogêneo de soluções que permitem adicionar novas capacidades à *web* tradicional baseada em hipertexto. Conforme destaca Meliá et al. (2010) oferecem melhor capacidade de resposta e uma experiência para o usuário mais extensa que as aplicações *web* tradicionais.

As *Rich Internet Application* (RIA), surgiram em 1995, mais especificamente com a criação dos *applets*. As RIAs proporcionam a criação de aplicações *web* que permitem aos navegadores uma maior riqueza de recursos ao invés de utilizar somente HTML tradicional da época. O uso de *applets* era a alternativa encontrada por muitos desenvolvedores quando necessitavam criar aplicações *web* mais complexas. Ao longo dos anos, o HTML e o JavaScript evoluíram, solucionando algumas limitações e problemas existentes nas versões iniciais. A grande evolução, entretanto, veio por meio do surgimento do AJAX que é conjunto de técnicas para desenvolvimento *web* focadas em JavasScript dotadas de recursos que proporcionam uma interação mais rica e rápida para o usuário (PINA; OLIVEIRA, p. 2, ANO).

Fraternali, Rossi e Sánchez-Figueroa (2010) afirmam que as RIAs são aplicações cliente-servidor e que são utilizadas no desenvolvimento de aplicações desktop e *web* e fornecem benefícios de implantação e manutenção dos aplicativos da *web*, ao mesmo tempo em que suportam uma Interface do Usuário (UI) do cliente muito mais rica. Além disso, os RIA introduzem novas características arquitetônicas no campo das aplicações tradicionais da *Web*, e. uma UI com estados conectados e desconectados, uma comunicação cliente e servidor inteligente com solicitações assíncronas.

Como dito por [Vinícius Baggio Fuentes](https://www.google.com.br/search?hl=pt-BR&tbo=p&tbm=bks&q=inauthor:"Vinícius+Baggio+Fuentes"&source=gbs_metadata_r&cad=7) (2014, p. 74) depois da repopularização do JavaScrip com o uso intenso do AJAX, os browsers tem se tornados muito mais poderosos que antigamente. E é nesse requisito que se refere uma interface rica, na qual não existe a necessidade de atualização de todas as páginas, recurso que é era possível antigamente somente com o uso do HTML, ele trabalha com o conceito do AJAX no qual não existe a necessidade de atualizar a pagina para atualizar alguma informação que contenha nela.

Graças ao RIA ter sua “Cliente Engine” existe vários benefícios ao se criar uma interface rica, como, uma melhor resposta entre servidor/cliente passando a impressão ao usuário utilizar uma programa desktop, uma comunicação assíncrona fazendo com que quando o usuário clique em uma opção sem a necessidade de esperar a resposta do servidor.

## 2.3 COMUNICAÇÃO SÍNCRONA E ASSÍNCRONA

No desenvolvimento de aplicações *web* há dois tipos de requisições a serem feitas para um servidor: a síncrona e a assíncrona. Ao utilizar requisição síncrona o browser fica parado aguardando o término da requisição, o que contradiz todo o propósito do AJAX.

Na comunicação síncrona os processos de origem e destino são sincronizados a cada mensagem, o browser ficará parado até a requisição for completada. Esse procedimento acontece da seguinte forma (COLOURIS et al., 2013 p. 147): quando um envio é realizado, o processo de origem é bloqueado até que a recepção correspondente seja realizada. Assim, quando uma recepção é realizada, o processo é bloqueado enquanto a mensagem não chegar e a transmissão da mensagem ocorre paralelamente com o processo de origem. Colouris et al., (2013, p. 148) explicam, também, que na comunicação assíncrona a operação não bloqueia o processo, ou seja, o processo origem pode prosseguir assim que a mensagem seja copiada para um buffer local e a transmissão da mensagem ocorre em paralelo com o processo de origem.

Um exemplo de requisição assíncrona é o e-mail do Google, o Gmail. Por meio da técnica de *pooling* ele cria constantemente requisições para verificar a chegada de novos e-mails no servidor. Dessa forma, enquanto o e-mail não chega, o usuário pode interagir com outras funcionalidades do sistema de e-mails (MILETTO[; BERTAGNOLLI, 2014](https://www.google.com.br/search?hl=pt-BR&tbo=p&tbm=bks&q=inauthor:"Evandro+Manara+Miletto"&source=gbs_metadata_r&cad=6)).

# 3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e o método utilizados para a realização deste trabalho. Os materiais estão relacionados às tecnologias e ferramentas utilizadas e o método apresenta a sequência das principais atividades realizadas.

## 3.1 MATERIAIS

O Quadro 1 apresenta a listagem das principais ferramentas e tecnologias que serão utilizadas no desenvolvimento do sistema proposto.

|  |  |  |  |
| --- | --- | --- | --- |
| **Ferramenta / Tecnologia** | **Versão** | **Disponível em** | **Aplicação** |
| **Astah Community** | 7.2.0 | http://astah.net/editions/community | Modelagem do sistema: desenvolvimento do diagrama de classes e de casos de uso |
| **Bootstrap** | 4 | https://getbootstrap.com/docs/3.3/ | *Framework* *front-end* |
| **PrimeFaces** | 6.2 | https://www.primefaces.org/downloads/ | *Framework* *front-end* |
| **Spring Tools** | 4.3 | <https://spring.io/tools> | Ambiente de desenvolvimento para o sistema |
| **Java** | 8 | http://www.java.com/pt\_BR/download/win10.jsp | Linguagem de programação |
| **PostgreSQL** | 10 | https://www.postgresql.org/ | Banco de dados |
| **pgAdmin 4** | 3.0 | https://www.pgadmin.org/download/ | Ferramenta para administração do banco de dados. |
| **Spring Data** | 2.0.0 | https://projects.spring.io/spring-data-jpa/ | *Framework* de implementação de repositórios |
| **Spring Boot** | 1.5.8 | https://projects.spring.io/spring-boot/ | *Framework* para criação de aplicações |
| **Spring Security** | 4.2.3 | https://projects.spring.io/spring-security/ | *Framework* de autenticação e segurança da aplicação. |
| **JSF** | 2.3 | https://javaee.github.io/javaserverfaces-spec/ | Especificação Java para a construção de interfaces |
| **Lombok** | 1.18.8 | https://projectlombok.org/ | Biblioteca Java focada em produtividade e redução de código redundante. |
| Jaspersoft  Studio | 6.4.3 | https://community.jaspersoft.com/download | Gerador de relatórios. |

Quadro 1 - Lista de ferramentas e tecnologias utilizadas

## 3.2 MÉTODO

O método utilizado neste trabalho foi o processo sequencial linear de Pressman (2008) que define as fases de levantamento de requisitos, análise e projeto e desenvolvimento do sistema. Neste trabalho são apresentadas as duas primeiras etapas.

Na fase inicial do levantamento de requisitos foi realizada entrevista com um profissional da área de estética para poder compreender as necessidades de uma empresa que atua nessa área. Dentre os requisitos levantados, observou-se a necessidade de ter um sistema que controlasse, mais especificamente, a agenda de um centro de estética, no qual o cliente tivesse mais autonomia e flexibilidade em agendar seus próprios horários, sem a necessidade de um atendente. Além disso, o cliente pode escolher por qual profissional deseja ser atendido.

Na fase de análise e projeto do sistema tendo como base os requisitos preestabelecidos, definiram-se os casos de uso do sistema que foram documentados gerando informações para a modelagem do banco de dados.

Após a modelagem dos casos de uso, foi elaborado o diagrama de entidades e relacionamentos do banco gerando as tabelas com seus campos, tipos, tamanhos e relacionamentos.

# 4 RESULTADOS

Este capítulo apresenta o resultado deste trabalho que é o desenvolvimento de um sistema *web* para controle e gerenciamento de agenda de centros estéticos.

## 4.1 ESCOPO DO SISTEMA

O sistema proposto é para a *web* e permite controlar e gerenciar a agenda de um centro de estética. Os clientes poderão realizar o agendamento de um serviço *on-line*, escolhendo dia, horário, tipo de serviço e profissional desejado para executar o serviço.

O sistema terá quatro atores, sendo, o administrador que terá acesso a todas as funcionalidades do sistema, o funcionário que terá acesso às funcionalidades relacionadas à sua agenda, o atendente que terá acesso às funcionalidades relacionadas aos agendamentos e bloqueios e o cliente que poderá realizar seu cadastro para autenticação para, posteriormente, realizar seus agendamentos.

As funcionalidades do sistema em relação aos cadastros envolve o de usuários, serviço e da agenda. Os clientes poderão realizar o cadastro de um agendamento. Para isso, ele deverá selecionar os serviços desejados, e se houver mais de um funcionário cadastrado no sistema, aparecerá uma caixa para a seleção do mesmo. Para a Seleção da data será disponibilizado um *input* com uma interface em formato de calendário. E por último terá um *ComboBox* com todos os horários disponíveis*.*

Para que o cliente possa realizar o agendamento o sistema deverá permitir os cadastros de serviços e dos profissionais que atendem no estabelecimento e vinculá-los a um determinado serviço.

Para fazer um agendamento o cliente deverá registra-se e posteriormente autenticar-se no sistema. Assim, poderá agendar seu horário por meio da seleção de um ou mais serviços, escolha do profissional, selecionar o dia e o horário do dia para realização do serviço. Ao realizar o cadastro, o cliente pode realizar somente um agendamento e o administrador será responsável por aceitar ou não o cadastro do cliente, e após validado o cadastro o cliente poderá realizar outros agendamentos.

Quando o cliente realizar um agendamento por outro meio que não seja o sistema, o funcionário deverá registrá-lo no sistema para manter o controle e gerenciamento da agenda.

O sistema também permitirá que o administrador possa emitir relatórios de clientes que mais utilizam os serviços do estabelecimento, dos serviços mais realizados, entre outros. Além disso, os clientes poderão entrar em contato com o estabelecimento por meio de um *chat* para que possam esclarecer dúvidas sobre os serviços oferecidos.

## 4.2 MODELAGEM DO SISTEMA

Este capítulo apresenta a modelagem deste trabalho.

### 4.2.1 Requisitos Funcionais

O Quadro lista os Requisitos Funcionais (RF) identificados para o sistema.

|  |  |  |
| --- | --- | --- |
| **Identificação** | **Nome** | **Descrição** |
| RF01 | Manter agenda | Manutenção dos dados relacionados aos agendamentos. |
| RF02 | Manter de Usuários | Manutenção dos dados relacionados aos usuários do sistema. |
| RF03 | Manter Tipos de Serviço | Manutenção dos tipos de serviços que podem ser, limpeza de pele, depilação, entre outros. |
| RF04 | Bloqueio de Agenda | Permitir ao usuário administrado fazer e editar bloqueio em horários na agenda. |
| RF05 | Validar cadastro de novos clientes | Permitir ao usuário administrador aceitar ou não, o cadastro de novos usuários. |
| RF06 | Emitir relatórios | Proporcionar relatórios para o administrador sobre as atividades do estabelecimento, como, por exemplo, atividades prestadas por funcionário, serviços mais procurados, horários mais requisitados, entrada mensal, serviços realizados por funcionário. |
| RF07 | Comunicação via *chat* | Permitir que os clientes se comuniquem com os funcionários ou administrador por meio de *chat* próprio. |

Quadro 2 - Requisitos funcionais do sistema

### 4.2.2 Requisitos Não Funcionais

O Quadro lista os Requisitos Não Funcionais (RNF) identificados para o sistema.

|  |  |  |
| --- | --- | --- |
| **Identificação** | **Nome** | **Descrição** |
| RNF01 | Efetuar *login* | O sistema validará *login* e senha para conceder acesso ao sistema. |
| RNF02 | Campos de preenchimento obrigatórios | Os campos que são de preenchimento obrigatório serão validados por meio de uma função do sistema. |
| RNF03 | Campos com máscaras de entrada | Os campos que possuem caracteres especiais serão validados por meio de máscaras de entrada. |
| RNF04 | Vínculo | Dados relacionados devem estar vinculados. Por exemplo, um serviço deve estar relacionado a um funcionário. O sistema também não deve permitir a exclusão de dados vinculados (não é possível excluir um funcionário se possuir serviços relacionados). |
| RNF05 | Relatórios | A data inicial deve ser maior que a data final. |
| RNF06 | Remover agendamento | O cliente não poderá remover um agendamento com período inferior a 24 horas para ser realizado. |
| RNF07 | Selecionar serviços | As opções de serviço serão mostradas conforme o funcionário que foi selecionado. |
| RNF08 | Controle de horário na agenda | Só poderá ser mantido na agenda se o funcionário selecionado estiver livre no horário selecionado. |

Quadro 3 – Requisitos não funcionais do sistema

A Figura 1 apresenta os casos de uso definidos para o sistema que é composto pelo administrador, funcionário e cliente. O administrador tem acesso à todas as funcionalidades do sistema. O cliente e o funcionário possuem acesso às funcionalidades de relatórios, agendamentos, chat e ao cadastro de usuários.

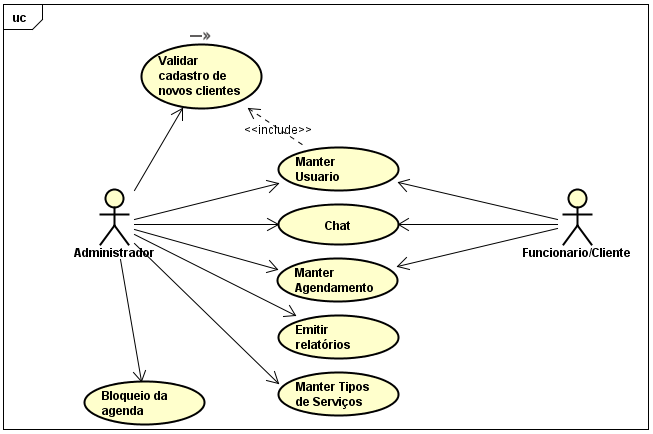


Figura 1 – Casos de uso

No Quadro 4 é descrito a operação do tipo “manter” que refere-se à inclusão dos registros de usuários, agendamentos e tipos de serviço. O funcionamento da inclusão de um novo registro é padrão para todos os casos de uso relacionados a esse tipo de operação.

|  |  |
| --- | --- |
| **Caso de uso:**  Inserir registros.  **Descrição:**  Ator inclui dados no sistema.  **Atores:**  Administrador, Funcionário e Cliente.  **Pré-condição:**  Não há.  **Sequência de Eventos:**  1. Ator acessa a tela para cadastrar as informações solicitadas.  2. O sistema insere as informações no banco de dados.  **Pós-Condição:**  Registro inserido no banco de dados. | |
| **Nome do fluxo alternativo (extensão)** | **Descrição** |
| 1. Campos obrigatórios não informados. | 1.1. Ator deixa de informar dados obrigatórios e clica em salvar.  1.2. O sistema valida que não foram informados todos os campos obrigatórios e exibe mensagem ao usuário sem salvar o registro.  1.3. O sistema permanece na tela de inclusão mantendo os dados informados anteriormente. |

Quadro 4 – Operação “incluir” dos casos de uso de cadastro

O Quadro 5 apresenta o funcionamento da operação de alterar registro de todos os casos de usos identificados como “manter”.

|  |  |
| --- | --- |
| **Caso de uso:**  Alterar registro.  **Descrição:**  Ator altera dados no sistema.  **Atores:**  Administrador, Funcionário e Cliente.  **Pré-condição:**  Dados cadastrados no sistema.  **Sequência de Eventos:**  1. O ator acessa a tela para visualização de dados já cadastrados.  2. O sistema apresenta o registro selecionado para a alteração.  3. Ator altera os dados do registro e clica em salvar.  4. O sistema valida as informações e salva as informações no mesmo registro.  **Pós-Condição:**  Registro alterado no banco de dados. | |
| **Nome do fluxo alternativo (extensão)** | **Descrição** |
| 1. Campos obrigatórios não informados. | 1.1. Ator apaga dados obrigatórios e clica em salvar.  1.2. O sistema valida que não foram informados todos os campos obrigatórios e exibe mensagem ao usuário sem salvar o registro.  1.3. O sistema permanece na tela de edição mantendo as alterações realizadas. |
| 2. Campos informados em formato incorreto. | 2.1. Ator altera os dados deixando em um formato incorreto e clica em salvar.  2.2. O sistema valida que os dados não estão no formato esperado e exibe mensagem ao usuário sem salvar o registro.  2.3. O sistema permanece na tela de edição mantendo as alterações realizadas. |

Quadro 5 – Operação “alterar” dos casos de uso de cadastro

O Quadro 6 apresenta o funcionamento da operação de exclusão dos casos de uso do tipo “manter”.

|  |  |
| --- | --- |
| **Caso de uso:**  Excluir registro.  **Descrição:**  Ator solicita a exclusão de dados no sistema.  **Atores:**  Administrador, Funcionário e Cliente.  **Pré-condição:**  Dados cadastrados no sistema.  **Sequência de Eventos:**  1. O ator acessa a tela para visualização de dados já cadastrados.  2. O sistema apresenta o registro selecionado para a exclusão.  3. Ator clica em excluir registro.  4. O sistema exclui as informações do banco de dados e exibe as informações do status do procedimento.  **Pós-Condição:**  Registro excluído no banco de dados. | |
| **Nome do fluxo alternativo (extensão)** | **Descrição** |
| Cliente deseja excluir consulta | 1.1 O cliente não poderá excluir um agendamento com período inferior a 24 horas para o serviço ser realizado. |
| 1. Exclusão de registro com vínculos no sistema | 1.1. Ator clica em excluir um registro que possui vínculos no sistema.  1.2. O sistema verifica que o registro tem vínculos, não o exclui e exibe mensagem de alerta ao usuário. |

Quadro 6 - Operação “excluir” dos casos de uso de cadastro

O Quadro 7 apresenta o funcionamento da operação de consultar um registro dos casos de uso do tipo “manter”.

|  |  |
| --- | --- |
| **Caso de uso:**  Consultar registro.  **Descrição:**  Ator solicita a consulta de dados cadastrados no sistema.  **Atores:**  Administrador, Funcionário e Cliente.  **Pré-condição:**  Dados cadastrados no sistema.  **Sequência de Eventos:**  1. Ator acessa a tela para visualização de dados já cadastrados.  2. O sistema exibe os dados da consulta ao usuário.  **Pós-Condição:**  Dados são exibidos aos usuários. | |
|  |  |
|  |  |

Quadro 7 – Operação “consultar” dos casos de uso de cadastro

O Quadro 8 apresenta o caso de uso relacionado ao chat do sistema.

|  |  |
| --- | --- |
| **Caso de uso:**  Chat (recurso utilizado para que os usuários possam se comunicar por meio da troca de mensagens).  **Descrição:**  Ator solicita a troca de mensagens.  **Atores:**  Administrador, Funcionário e Cliente.  **Pré-condição:**  Usuários cadastrados no sistema.  **Sequência de Eventos:**  1. Ator acessa a tela para visualização de usuários cadastrados.  2. Ator escolhe com quem deseja trocar mensagens.  3. Ator manda mensagens para o escolhido.  **Pós-Condição:**  Mensagem recebida por outro usuário. | |
|  |  |
|  |  |

Quadro 8 - Caso de uso para o chat do sistema

O Quadro 9 apresenta o caso de uso referente aos relatórios do sistema.

|  |  |
| --- | --- |
| **Caso de uso:**  Visualizar relatórios.  **Descrição:**  Visualizar os relatórios de atividades prestadas por funcionário, serviços mais prestados, horários mais requisitados, valor de entrada (lucro) e funcionários.    **Ator:**  Administrador e Funcionário.  **Pré-condição:**  Relatório gerado/visualizado.  **Sequência de Eventos:**  1. O ator acessa a tela de relatórios;  2. O ator escolhe os parâmetros de filtragem para o relatório;  3. O sistema gera o arquivo e/ou a página de impressão.  **Pós-Condição:**  Relatório gerado e visualizado. | |
| **Nome do fluxo alternativo (extensão)** | **Descrição** |
| 1. Não existem dados referentes ao relatório | 1.1. O sistema informa ao ator que não foi possível gerar o relatório.  1.2. Retorna ao fluxo principal do caso de uso. |
| 2. Data incorreta | 2.1 Ator seleciona data final menor que a data inicial.  2.2 Sistema exibe mensagem informando que o erro ocorrido. |

Quadro 9 – Caso de uso dos relatórios do sistema.

|  |  |
| --- | --- |
| **Caso de uso:**  Bloquear agenda  **Descrição:**  Ator deseja bloquear algum horário na agenda do estabelecimento.  **Atores:**  Administrador.  **Pré-condição:**  Não há.  **Sequência de Eventos:**  1. O ator acessa a tela de bloqueio da agenda;  2. O ator escolhe o(s) dia(s) e/ou a(s) hora(s) que deseja bloquear a agenda;    **Pós-Condição:**  Agenda bloqueada o tempo escolhido. | |
|  |  |
|  |  |

Quadro 10 – Caso de uso de bloqueio do sistema.

|  |  |
| --- | --- |
| **Caso de uso:**  Validar cadastro de novos clientes  **Descrição:**  Ator deseja aceitar ou não novos clientes cadastrados.  **Atores:**  Administrador.  **Pré-condição:**  Novos clientes adicionados  **Sequência de Eventos:**  1. O ator acessa a tela de novos clientes;  2. O ator escolhe um cliente;  3. O ator verifica suas informações do cadastro;  4. O ator aceita ou não o cliente cadastrado;  **Pós-Condição:**  Novos clientes aceitos ou não. | |
|  |  |

Quadro 11 - Caso de uso validação de cliente do sistema.

A Figura 3 apresenta o diagrama de entidade e relacionamentos do banco de dados.

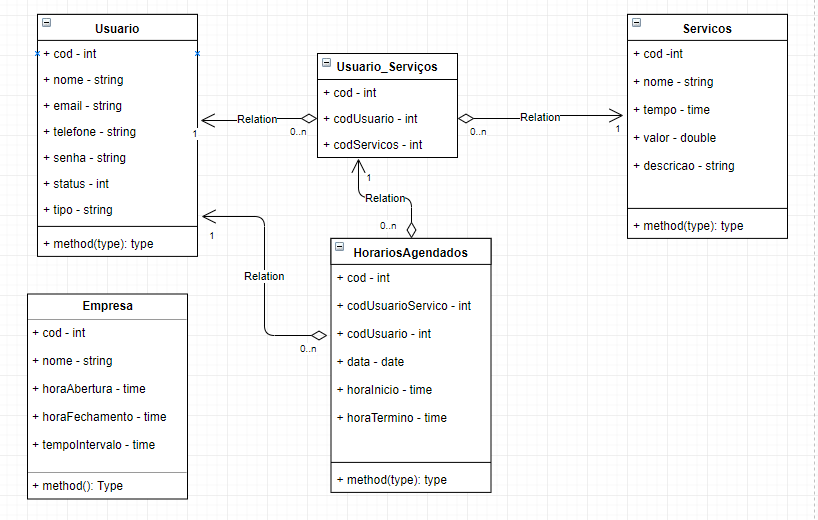


Figura 2 - Banco de dados

# REFERÊNCIAS

ABIHPEC. **MEIs do mercado de beleza crescem 567% em 5 ano**. 2015

Disponível em: <https://abihpec.org.br/2015/04/meis-do-mercado-de-beleza-crescem-567-em-5-anos/>. Acesso em: 26 mar 2018.

cgi.br. **Cresce a proporção de empresas brasileiras que utilizam conexões à Internet mais velozes, aponta Cetic.br**. 2016. Disponível em: <https://www.cgi.br/noticia/releases/cresce-a-proporcao-de-empresas-brasileiras-que-utilizam-conexoes-a-internet-mais-velozes-aponta-cetic-br/>. Acesso em: 26 mar. 2018.

SOUZA, Alberto. **Spring MVC**: Domine o principal framework *web* Java. Editora: Casa do Código, 2015.

Euromonitor International. [Indústria: Esta](http://agedobrasil.com.br/industria-quais-os-produtos-cosmeticos-mais-usados-no-brasil/) na hora de buscar crescimento fora do Brasil, 2018. Disponível em: < https://blog.euromonitor.com/2017/10/hora-buscar-crescimento-brasil.html>.Acesso em: 28 março 2018.

MELIÁ Santiago; GÓMEZ Jaime; PÉREZ Sandy; DÍAZ Oscar. Facing Architectural and Technological Variability of Rich Internet Applications. **IEEE Internet Computing**, 2010.

Fraternali Piero; Rossi Gustavo; Sánchez-Figueroa Fernando. Rich Internet Applications. **IEEE Computer Society**, 2010

COULOURIS George;DOLLIMORE Jean; KINDBERG Tim, BLAIR Gordon. **Sistemas Distribuídos** - 5ed: Conceitos e Projeto. Editora: Bookman, 2013.

FUENTES Vinícius Baggio. **Ruby on Rails**: Coloque sua aplicação web nos trilhos. Casa do Código: São Paulo,2014.

MILETTO Evandro Manara; BERTAGNOLLI Silvia de Castro. **Desenvolvimento de Software II**: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP. Editora: Bookman, 2014.

PINA, DSA; OLIVEIRA, LEMC. **RIA-RICH INTERNET APPLICATIONS**: UMA REVISÃO DOS PRINCIPAIS EXPOENTES DA ÁREA. Disponivel em : < <http://scholar.googleusercontent.com/scholar?q=cache:m1ezUDCoNZsJ:scholar.google.com/&hl=pt-BR&as_sdt=0,5>.Acesso> em: 17 maio 2018.

PRESSMAN, Roger. **Engenharia de software**. Rio de Janeiro: McGraw-Hill, 2008.