

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO RIO GRANDE DO SUL
CAMPUS RESTINGA**

**ODONTOFÁCIL: SISTEMA DE GERENCIAMENTO PARA
CONSULTÓRIOS DE ODONTOLOGIA**

PAULO HENRIQUE SILVEIRA FARIA

**Porto Alegre
2017**

PAULO HENRIQUE SILVEIRA FARIA

**ODONTOFÁCIL: SISTEMA DE GERENCIAMENTO PARA
CONSULTÓRIOS DE ODONTOLOGIA**

Trabalho de Conclusão de Curso apresentado, junto ao Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal Educação, Ciência e Tecnologia do Rio Grande do Sul, como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Rafael Pereira Esteves

**Porto Alegre
2017**

PAULO HENRIQUE SILVEIRA FARIA

ODONTOFÁCIL: SISTEMA DE GERENCIAMENTO DE CONSULTÓRIOS DE ODONTOLOGIA

Trabalho de Conclusão de Curso apresentado como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Rafael Pereira Esteves

Aprovado em

Prof. Rafael Pereira Esteves (Orientador)

Prof. Roben Castagna Lunardi –IFRS
Campus Restinga

Prof. Jezer Machado de Oliveira – IFRS
Campus Restinga

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO SUL

Reitor: Prof. Osvaldo Casares Pinto

Pró-Reitora de Ensino: Profa. Clarice Monteiro Escott

Diretor do Campus Restinga: Prof. Gleison Samuel do Nascimento

Coordenador do Curso de Análise e Desenvolvimento de Sistemas: Prof. Rafael Pereira Esteves

Bibliotecária-Chefe do Campus Restinga: Paula Porto Pedone

Dedico este trabalho aos meus pais, que me apoiaram do início ao fim desta conquista para nossa família, se hoje estou me formando é graças a eles e minha namorada.

RESUMO

O objetivo deste trabalho é criar um sistema computacional para auxiliar no gerenciamento de consultórios odontológicos, pois atualmente existem muitos consultórios que executam suas tarefas manualmente, sem o apoio de sistemas para automatizar algumas tarefas do dia a dia. Uma das necessidades é fazer contato quando há um novo agendamento para um cirurgião dentista e paciente, assim, a secretária pode vincular o profissional e o paciente a uma consulta através da agenda do sistema e o profissional e o paciente recebem uma notificação em seu email contendo os dados do agendamento da consulta. O sistema também terá um módulo financeiro simples, para cadastros de despesas, proporcionando registros de entrada e saída de caixa, também será implementada a funcionalidade de uma rotina de backup, sendo gerada automaticamente no final do dia, ou quando for necessário. O sistema também irá contar com três tipos de permissões para poder restringir os acessos dentro da aplicação. O sistema também disponibiliza funcionalidades como, geração de relatórios como, atestados, orçamentos, despesas, receitas e recibo para comprovante de renda. Durante o desenvolvimento, as funcionalidades do sistema foram analisadas por uma cirurgiã dentista, colaborando com sugestões para melhorias das funcionalidades do mesmo.

Palavra-chave: Sistema, Consultório, Odontologia.

ABSTRACT

The main goal for this work is to develop a system to support management activities for dental offices, since nowadays most of them execute such tasks manually, without any support to automate that process. An example for the need of such automation is when the office need to make contact with the patient and the dentist when there is a new appointment. In this case, a secretary can make an appointment for both the professional and the patient using the proposed system and both will receive an e-mail with detailed information of appointment. Besides, the system has a financial module to record expense, a backup module for automatically saving data at the end of day or manually when needed, and a security module with three different levels of access. Other features include quotes and reports, sick notes, and income tax returns. During the development, the proposed features were monitored and analyzed by a dental surgeon who has pointed out suggestions for improvement.

Keywords: *System, office, dentistry.*

LISTA DE FIGURAS

FIGURA 1. MÓDULOS DO SISTEMA ODONTOFACIL	16
FIGURA 2. FLUXO BÁSICO DO SISTEMA ODONTOFACIL	18
FIGURA 3. DIAGRAMA EER ODONTOFACIL	19
FIGURA 4. TELA DE LOGIN DO SISTEMA ODONTOFACIL	21
FIGURA 5. TELA INICIAL DO SISTEMA ODONTOFACIL	23
FIGURA 6. RESPONSABILIDADE DO SISTEMA ODONTOFACIL	23
FIGURA 7. TELA DE AGENDAMENTO DE CONSULTAS ODONTOFACIL	24
FIGURA 8. MÉTODO NO ANGULARJS QUE CARREGA O USUÁRIO LOGADO	25
FIGURA 9. TELA DE LOGIN DO SISTEMA	25
FIGURA 10. DASHBOARD DO SISTEMA PERFIL ADMINISTRADOR	26
FIGURA 11. DASHOBOARD DO SISTEMA PERFIL SECRETÁRIA	27
FIGURA 12. DASHOBOARD DO SISTEMA PERFIL DENTISTA	27
FIGURA 13. RESPONSABILIDADE DO SISTEMA	28
FIGURA 14. AGENDA DO SISTEMA	29
FIGURA 15. MODAL AGENDAMENTO ADIMINISTRADOR E SECRETÁRIA	30
FIGURA 16. MODAL AGENDAMENTO OPÇÃO GERAR ATESTADO	30
FIGURA 17. MÉTODO DE SALVAR AGENDAMENTOS PARTE 1	32
FIGURA 18. MÉTODO DE SALVAR AGENDAMENTOS PARTE 2	32
FIGURA 19. MÉTODO DE SALVAR AGENDAMENTOS PARTE 3	33
FIGURA 20. SMS COM AS INFORMAÇÕES DO AGENDAMENTO	33
FIGURA 21. TEMPLATE NOTIFICAÇÃO POR EMAIL	34
FIGURA 22. TELA AGENDAMENTO COM PERFIL DENTISTA	34
FIGURA 23. MODAL DE AGENDAMENTO PERFIL DENTISTA	35
FIGURA 24. MÉTODO DE ENVIO DE LOG COM ERRO POR EMAIL	36
FIGURA 25. TELA DE CONSULTA DE RECEITAS	37
FIGURA 26. TELA DE CONSULTA DE DESPESAS	38
FIGURA 27. FUNÇÃO PARA LISTAGEM DE RECEITAS	38
FIGURA 28. RELATÓRIO LISTAGEM DE DESPESAS	39
FIGURA 29. RELATÓRIO LISTAGEM DE RECEITAS	40
FIGURA 30. ATESTADO	41

FIGURA 31. TELA DE ORÇAMENTOS	42
FIGURA 32. RELATÓRIO ORÇAMENTO	43
FIGURA 33. RECIBO PARA COMPROVANTE DE RENDA	44

LISTA DE TABELAS

TABELA 1. COMPARAÇÃO ENTRE SISTEMAS

21

SUMÁRIO

1. INTRODUÇÃO	11
1.1. PROBLEMA	Error! Bookmark not defined.1
1.2. PROPOSTA DE SOLUÇÃO	12
2. TRABALHOS RELACIONADOS	13
2.1. DIAGRAMA DE CLASSES	13
2.2. DIAGRAMA DE CLASSES	14
2.3. DIAGRAMA DE CLASSES	14
2.4. DIAGRAMA DE CLASSES	15
3. MODELAGEM DO SISTEMA	16
3.1. DIAGRAMA DE CLASSES	18
3.2. DIAGRAMA DE CLASSES	19
4. IMPLEMENTAÇÃO	21
4.2 AUTENTICAÇÃO E PERMISSÕES	22
4.2.1 DASHBOARD	26
4.2.2 AGENDAMENTO E CONSULTA	28
4.2.3 BACKUP	35
4.2.4 ENVIO DE LOG	36
4.2.5 BANCO DE DADOS	36
4.2.6 FINANCEIRO	37
4.2.7 ATESTADO	40
4.2.8 ORÇAMENTO	41
4.2.9 RECIBO	44
4.2.10 CRIPTOGRAFIA	45
5. CONCLUSÃO	45

1. INTRODUÇÃO

A Odontologia é uma área da saúde que tem crescido muito nos últimos tempos, ganhando espaço não só no âmbito particular, como também nos serviços públicos de saúde, onde o cuidado com a saúde bucal é essencial para a saúde do corpo como um todo.

No Brasil são 261 mil cirurgiões-dentistas formados [1]. No Rio Grande do Sul, só na cidade de Porto Alegre, são 4941 profissionais e outros milhares serão colocados no mercado de trabalho, todos os anos [2].

Com esse grande volume de profissionais, a área tem avançado, ganhando espaço nos serviços estéticos que podem, ou não, envolver a boca, como por exemplo a aplicação de toxina botulínica e preenchedores faciais, algo que era restrito a profissionais da medicina.

As novas opções de tratamento e, o acesso a informação por parte da população, que tem se conscientizado com a importância dos cuidados com a saúde pessoal, tem levado cada vez mais pacientes a procurar atendimento odontológico.

Esse crescimento aponta a necessidade de modernização e atualização dos profissionais da área, visando potencializar seu atendimento e fidelizar seus pacientes.

1.1 Problema

Hoje em dia, existem algumas soluções de sistemas para consultórios odontológicos, muitos deles com um número grande de funcionalidades que atendem satisfatoriamente as necessidades do consultório.

Sendo assim, foi feita uma pesquisa sobre os sistemas disponíveis no mercado o que possibilitou a identificação de algumas necessidades que ainda não são completamente atendidas por estas soluções, como por exemplo, na questão de restrições de permissões, onde normalmente os sistemas atuais disponíveis liberam apenas uma licença de uso na contratação do mesmo, impossibilitando, assim o acesso de todos os profissionais do consultório

(dentistas, secretária). Esse sistema disponibiliza a possibilidade de se contratar mais algumas licenças adicionais, porém o valor mensal da contratação fica muito elevado, tornando o custo da solução proibitivo para consultórios de pequeno e médio porte. Outras funcionalidades importantes são de comunicação, como notificações por e-mail e SMS, o que possibilitaria que pacientes e profissionais recebam as notificações de agendamentos na sua caixa de e-mail e no seu telefone, sendo ele *smartphone* ou não.

1.2 Proposta de Solução

O sistema proposto neste trabalho, denominado Odontofácil, identifica cada profissional do consultório através de seu nome de usuário login. Assim cada usuário terá permissão de acessar funcionalidade específicas do sistema. O sistema Odontofácil suporta atualmente 3 tipos de usuários: secretária(o), dentista e administrador. O secretária(o) é o profissional que trabalha na recepção que faz o agendamento/remarcação de consultas. O Dentista tem acesso a sua agenda específica. O administrador possui acesso irrestrito a todas funcionalidades do sistema, incluindo a funcionalidade de finanças, onde serão gravadas as informações de despesas mensais, do consultório como, gastos com materiais, conta de água, luz telefone, receitas com os valores recebidos pelas consultas.

Para cada agendamento realizado o Sistema OdontoFácil envia um e-mail e um SMS para o cliente e para o cirurgião dentista contendo as informações do agendamento, como data / hora da consulta. A aplicação também permite *backup* dos dados a critério do administrador achar interessante, possibilitando a armazenagem do *backup* em *pen drives*, nuvem ou onde achar interessante.

O sistema Odontofácil possibilita a geração de atestados para os pacientes durante as consultas caso seja necessário, o sistema também permite a geração de orçamentos para determinados tipos de procedimentos cirúrgicos.

Além deste capítulo introdutório, esta monografia está organizada da seguinte forma. O Capítulo 2 apresenta os principais trabalhos relacionados ao sistema proposto. O Capítulo 3 mostra a modelagem do sistema. O Capítulo 4 detalha como foi feita a implementação do Odontofácil e o Capítulo 5 conclui o trabalho.

2. TRABALHOS RELACIONADOS

Como primeira etapa do trabalho, analisou-se uma série de sistemas na Internet que atualmente estão disponíveis no mercado, que contemplam, pelo menos parcialmente, a proposta do trabalho. A seguir, os três principais sistemas encontrados são discutidos e comparados entre si e com o Odontofácil.

2.1 Simples Dental

Simples Dental [3] é um sistema de Internet para gerenciamento de consultórios odontológicos que disponibiliza as seguintes funcionalidades: agenda para dentistas com a possibilidade de cadastro de consultas, ou compromisso, com a possibilidade de envio de SMS para os envolvidos na consulta agendada, cadastro de dentistas, cadastro de pacientes, módulo financeiro para gerenciamento de fluxo de caixa, com a funcionalidade de gerar boletos, marketing de relacionamento, possibilitando envio de campanhas via SMS, e também disponibiliza uma versão mobile de sua agenda, com versões para Android e IOS.

Existem 3 tipos de planos, plano com parceria com a Dental Cremer, plano com a parceria com a Dental Speed e o plano diretamente com Simples Dental.

O plano de parceria com a Dental Cremer disponibiliza todas as funcionalidades descritas, sendo necessária a compra de materiais na Dental Cremer a cada 3 meses para renovar o plano.

O plano de parceria com a Dental Speed funciona da mesma forma que o de parceria com a Dental Cremer, disponibilizando todas as funcionalidades descritas, sendo necessária a compra de materiais na Dental Speed a cada 3 meses para renovar o plano.

Já no plano diretamente com a Simples Dental, são disponibilizadas todas as funcionalidades descritas porém com o pagamento mensal de R\$ 98,90 (valor na data de 16/09/2017).

2.2 Dental Office

Dental Office [4] é um sistema de Internet para gerenciamento de consultórios odontológicos que disponibiliza as seguintes funcionalidades: painel de controle, cadastros de clientes, anamneses, atestados, receituários, galeria de imagens e documentos, agenda de consultas, odontograma com visualização gráfica da arcada dentária e procedimentos de cada plano de tratamento com orçamento integrado, financeiro com emissão de boletos, marketing com SMS e emails automáticos, controle de estoque, relatórios e uma versão mobile com todas as funcionalidades do Sistema para Android e IOS.

O Sistema Dental Office existem duas opções de planos, o plano Cloud e o plano Desktop.

O plano Cloud tem valor mensal de R\$79,90 e oferece duas licenças com todas suas funcionalidades exceto, o modulo de Clinica, Boleto e Marketing que tem um adicional de R\$25,00 mensais cada.

O plano Desktop tem valor mensal de R\$79,90 e oferece apenas uma licença para uso, também conta com todas as funcionalidades descritas acima, exceto os módulos de, Clinica, Boleto e Marketing, que tem valor adicional de R\$25,00 mensais cada, possibilitando a contratação de licenças adicionais para uso em rede por R\$15,00 mensais cada.

2.3 SYSPSI

Syspsi é um sistema de Internet para gerenciamento de consultórios para Psicologia, com foco no agendamento de consultas e segurança de prontuários, disponibilizando as seguintes funcionalidades: módulo de agendamento de consultas com integração ao *Google Calendar*, financeiro, gerenciamento de consultas, gerenciamento de convenios, backup, emissão de relatórios financeiros e armazenamento de prontuários. O SYSPSI é um sistema de único usuário (psicólogo) e não permite mais de um profissional com diferentes perfis, além de não gerar atestados, orçamentos, recibos e não haver nenhum tipo de notificações de agendamentos para o cliente

2.4 Comparação entre Sistemas

Tabela 1. Comparação entre sistemas

Funcionalidades	Odonto Facil	Simples Dental	Dental Office	SYSPSI
Notificações e-mail	Sim	Sim	Sim	Não
Permissões por papéis de usuários	Sim	Não	Não	Não
Envio de SMS	Sim	Sim	Sim	Não
Agenda	Sim	Sim	Sim	Sim
Integração ao Google Calendar	Não	Não	Não	Sim
Módulo Financeiro	Sim	Sim	Sim	Sim
Número de Licenças	Ilimitado	1	2	Ilimitado
Localização do servidor	Local	Externo	Externo	Local
Backup	Sim	Sim	Sim	Sim
Orçamento	Sim	Sim	Sim	Não
Atestado	Sim	Não	Não	Não
Recibo para comprovante de renda	Sim	Não	Não	Não
Envio de Email para Administrador com log do sistema caso exista erro.	Sim	Não	Não	Não

No comparativo conseguimos ver que os sistemas disponíveis hoje no mercado para o segmento odontologia atendem a maioria das necessidades diárias de um consultório odontológico. Porém para um consultório com um número mais elevado de dentistas, o custo das licenças individuais eleva bastante o investimento anual, para o sistema Dental Office, o investimento anual seria de R\$ 958,80 para no máximo duas licença de uso, já no sistema oferecido pela Simples Dental o custo anual pode chegar a R\$1.186,80 para apenas uma licença de uso.

Uma funcionalidade que o OdontoFácil contempla, que os demais sistemas não disponibilizam é a questão de permissões por tipo de usuário, possibilitando assim a definição de papéis de usuários. Cada papel define o que cada tipo de usuário pode acessar dentro do sistema, O Odontofácil suporta até 3 tipos de permissões de acesso sendo estas, Secretária(o), Dentista e Administrador. Outra funcionalidade importante disponibilizada no Odontofácil é referente as notificações, por e-mail e SMS, facilitando a comunicação entre o pacientes e funcionários do consultório, o sistema também conta com um módulo de orçamentos, onde é possível gerenciar os orçamentos gerados para pacientes, guardando históricos e possibilitando a impressão do orçamento, para as consultas realizadas no consultório é possível gerar atestados pelo Odontofácil para facilitar o atendimento.

3. MODELAGEM DO SISTEMA

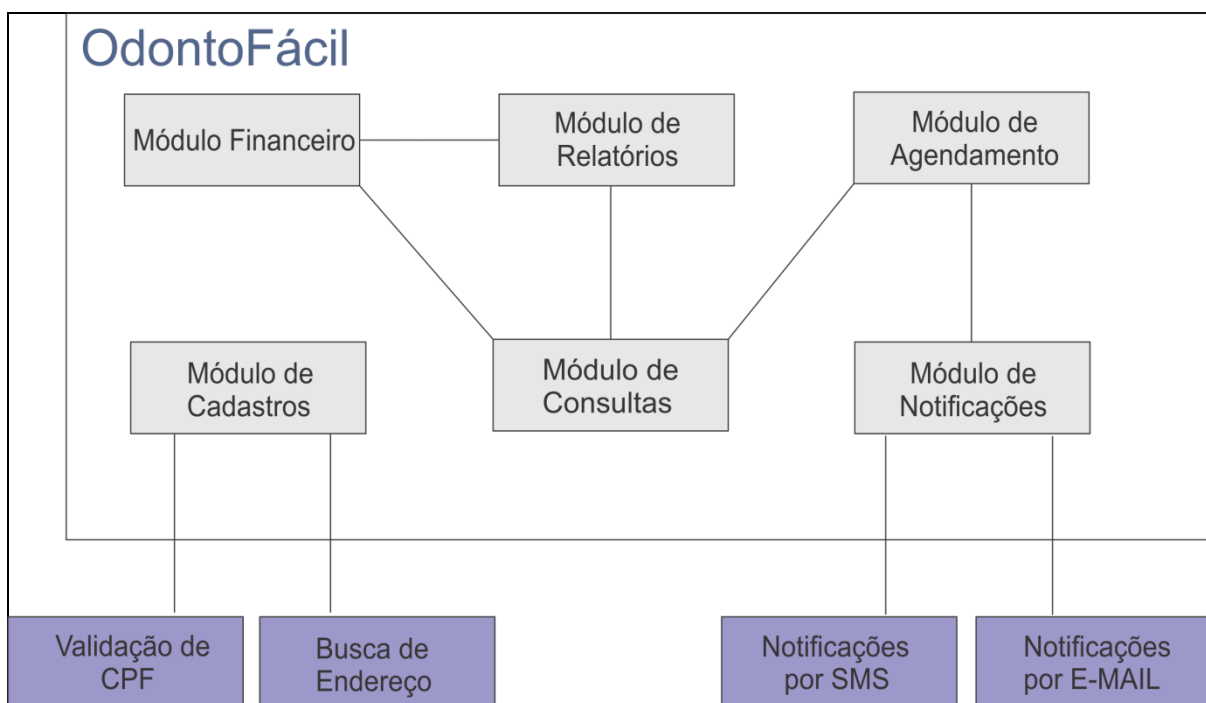


Figura 1. Módulos do Sistema OdontoFácil

Fonte: Próprio autor.

Como é possível observar na Figura 1, o sistema é composto por seis principais módulos, que são descritos a seguir:

Módulo de cadastro, responsável por gerenciar cadastros de funcionários e pacientes, possibilitando a listagem, inclusão, exclusão e edição, dos registros, este módulo possui integrações com duas funcionalidades externas, uma de validação de CPF disponibilizado pela receita federal e a outra de consulta de endereço pelo CEP disponibilizado pelo site dos correios.

Módulo financeiro, este é um módulo básico responsável pelo gerenciamento financeiro onde, será possível cadastrar despesas como, contas de água, luz e materiais, facilitando a visualização dos gastos mensais. O módulo financeiro também conta com uma página onde é possível listar as receitas cadastradas nas consultas realizadas, é possível selecionar um determinado período para verificar o valor de receitas cadastradas durante o período selecionado, também é possível gerar um relatório com os dados de receitas e despesas por um período pré determinado na tela de listagem de despesas e receitas.

Módulo de relatórios, este módulo é responsável pelo gerenciamento dos relatórios, onde será possível gerar, atestados ao final dos atendimentos, orçamentos para procedimentos futuros, relatórios financeiros e declaração para imposto de renda, no formato PDF, através do framework *Jasper Reports* [3].

Módulo de agendamento, responsável por gerenciar a marcação das consultas no sistema diretamente na agenda do Odontofácil, o funcionário seleciona a data e hora desejada para a consulta, quando a opção desejada é selecionada, o sistema apresenta um modal solicitando os dados necessários para o agendamento. Existem algumas regras de negócio que não permitem a criação de um agendamento para uma data anterior a data atual, e restrições de permissões por tipo de usuário, caso seja um usuário com permissão de Dentista, a agenda que será mostrada será apenas com agendamentos para aquele profissional, caso seja permissão de administrador ou secretária, o sistema mostrará todos os agendamentos do sistema.

Módulo de consultas, responsável por armazenar as informações de cada consulta realizada no consultório, possibilitando um armazenamento das informações sobre cada paciente.

Módulo de notificações, responsável por enviar notificações por e-mail e SMS para dentistas e pacientes, a cada agendamento realizado pelo sistema Odontofácil.

3.1 Fluxo básico do sistema OdontoFácil

A seguir, a Figura 2 ilustra um diagrama de atividades que descreve o fluxo de funcionamento básico do OdontoFácil.

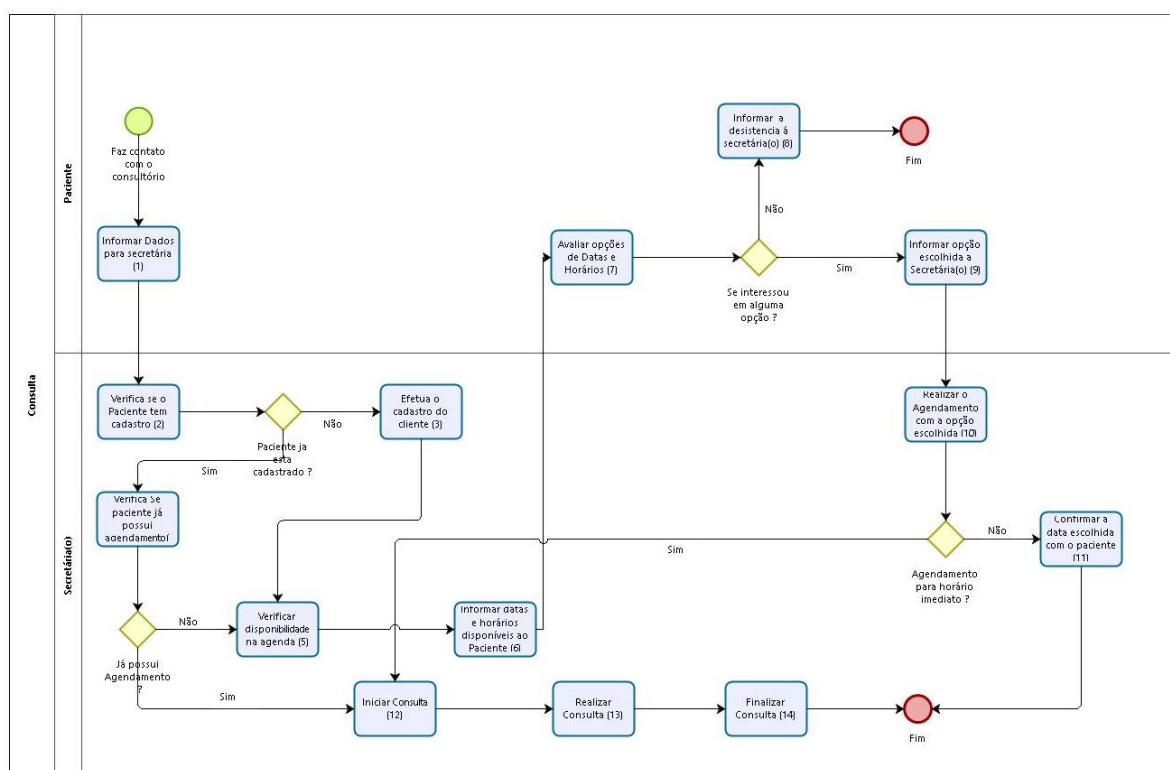


Figura 2. Fluxo da Consulta OdontoFácil

Fonte: Próprio autor.

Como pode-se observar na Figura 2 o fluxo da consulta inicia quando o paciente faz contato com o consultório pessoalmente ou por telefone. Neste momento o paciente informa os dados para a secretária (1). Em seguida, a secretária verifica se o paciente já tem cadastro no sistema (2), se o cliente não estiver cadastrado, a secretária efetua o cadastro do cliente (3). Caso o cliente já esteja cadastrado, a secretária verifica se o paciente já possui um agendamento (4). Caso o paciente já possua um agendamento para o horário atual, é iniciada a consulta com o Cirurgião Dentista (12), a consulta é realizada (13) e finalizada pelo profissional (14). Porém, caso o paciente não possua um agendamento, a secretária verifica a

disponibilidade de horários (5), a secretária informa ao paciente as datas e horários disponíveis (6), o paciente avalia as opções de data e horário informadas pela secretária (7). Caso o paciente não se satisfaça com nenhuma opção fornecida, o mesmo informa a desistencia do agendamento para a secretária (8). Caso o paciente opte por alguma das opções oferecidas, o mesmo informa a opção selecionada para a secretária (9), a secretária realiza o agendamento no sistema com a opção escolhida pelo paciente (10). Caso o agendamento seja para uma consulta imediata, o fluxo vai direto para a tarefa (12) e a consulta é iniciada, Caso o agendamento seja para uma hora ou data futura, a secretária confirma a data e hora do agendamento com o paciente (11).

3.2 Diagrama de Classes

A Figura 3 apresenta o diagrama de classes do OdontoFácil

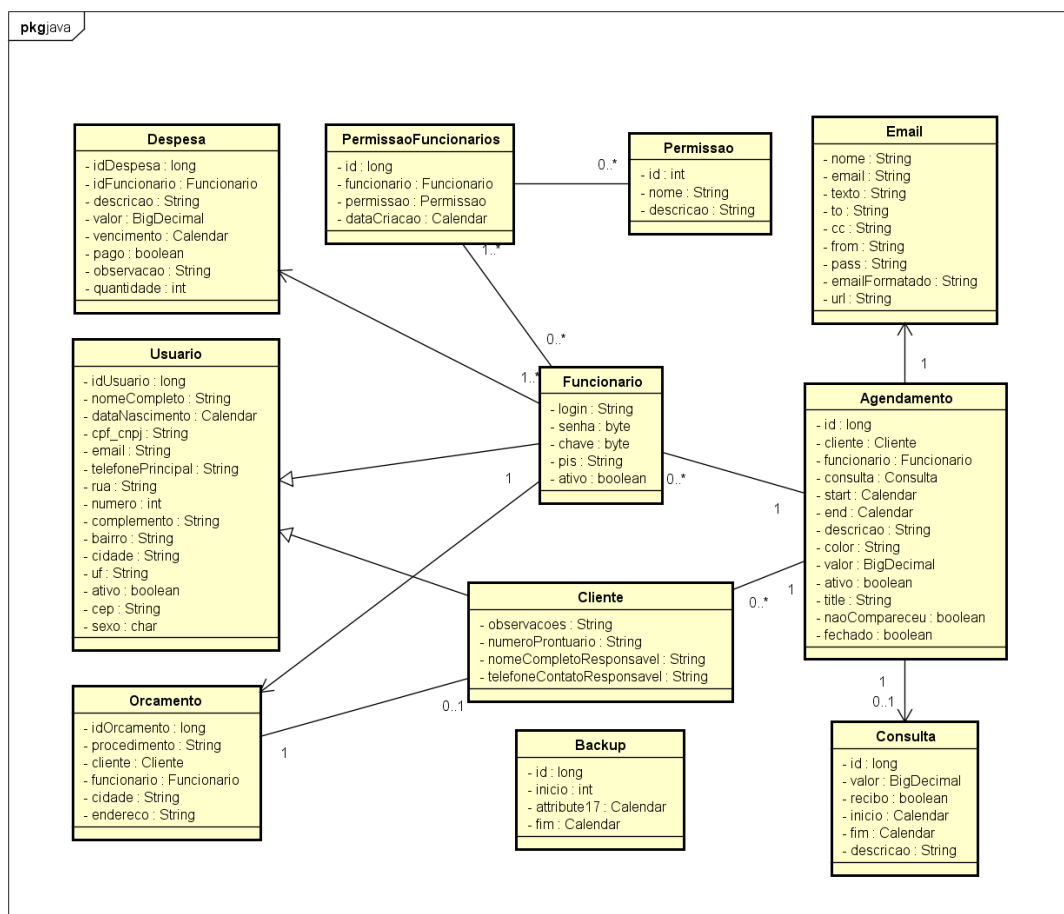


Figura 3. Diagrama de Classes OdontoFacil

Fonte: Próprio autor.

A partir do diagrama de classes é possível observar que as classes Funcionario e Cliente estendem a classe Usuario que contém atributos básicos para o cadastro de pessoa, onde tanto Funcionario quanto Cliente contém estes atributos. A classe Funcionario possui atributos específicos para o acesso ao sistema. A classe Cliente possui atributos relacionados às suas informações de contato do cliente bem como a identificação do prontuário do cliente. Vale ressaltar que o OdontoFácil não armazena os prontuários dos pacientes mas apenas guarda uma referência para que o funcionário (secretária, dentista) possa localizar o prontuário em arquivo físico. A classe Despesas guarda informações sobre as despesas do consultório como, por exemplo, conta de luz, conta de água, materias e aluguel, e possui um relacionamento com a classe Funcionario pois uma despesa sempre será cadastrada por algum funcionário. A classe Permissao descreve um papel do sistema (dentista, administrador e secretária) que por sua vez indicam o que cada funcionário pode fazer no sistema. A classe PermissoesFuncionarios foi criada pelo relacionamento da classe Funcionario com a classe Permissao, pois se trata de um relacionamento de muitos para muitos. A classe Orcamento disponibiliza as opções de salvar, listar, editar, e imprimir orçamentos, possibilitando o reuso de um orçamentos já criados no sistema e podendo ser impresso nas consultas. A classe Orcamento possui um relacionamento com a classe Funcionario e com a classe de Cliente pois sempre que um orçamento for criado ele seja criado por um funcionário para um paciente. A classe Agendamento, responsável por manter os agendamentos no sistema, enviar as notificações SMS e montar o email que será enviado para o paciente e o dentista relacionados no agendamento, possui uma ligação com a classe Consulta, pois uma consulta se inicia a partir de um agendamento. A classe Email responsável por realizar o envio de emails para o paciente e dentista associados ao agendamento, tem apenas um relacionamento com a classe Agendamento pois a cada agendamento realizado no sistema um email e um SMS são enviados pelo sistema para os envolvidos na consulta.

4. IMPLEMENTAÇÃO

O sistema Odontofácil foi desenvolvido em uma arquitetura bastante utilizada na atualidade, a arquitetura REST “Transferência de Estado Representacional” (*Representational State Transfer*) [7]. A arquitetura REST nos garante maior escalabilidade nos sistemas desenvolvidos, segurança e performance.

A Figura 4 demonstra a arquitetura e as camadas do sistema Odontofácil.

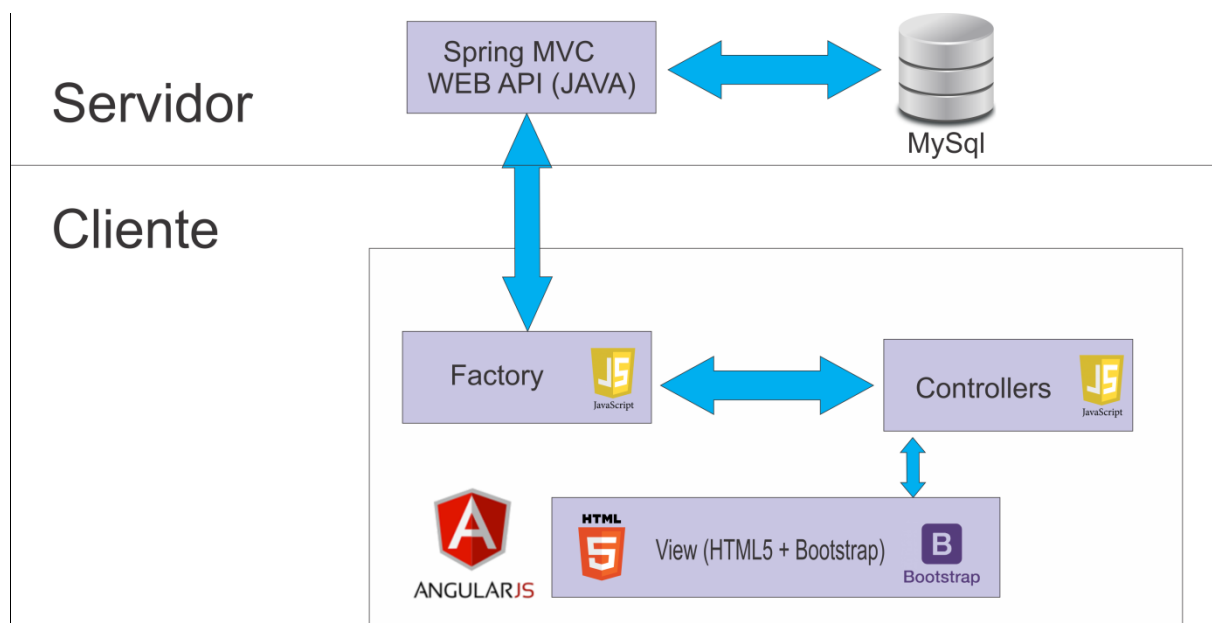


Figura 4. Arquitetura do Odontofácil

Fonte: Próprio autor.

O lado cliente da aplicação inclui diversas tecnologias como pode ser observado na Figura 4. *AngularJS* [8], *HTML5* [6], *Bootstrap* [9] são as principais tecnologias utilizadas no desenvolvimento do lado do cliente. *HTML5* é tecnologia utilizada para a criação de páginas Web em conjunto com *Bootstrap* para atribuir um layout agradável e sugestivo para as páginas Web, o *Bootstrap* também é o responsável por possibilitar a responsividade das páginas Web do sistema, assim sendo possível acessá-lo de diversas plataformas, como tablets e smartphones. O *AngularJS* é o responsável pelas validações de cadastros nas páginas Web e também responsável por fazer a comunicação entre o lado cliente e o lado servidor. A escolha da utilização do *AngularJS* foi feita pela facilidade e praticidade do framework, que contém

muitos serviços com funcionalidades já implementadas e disponíveis para utilização, com a necessidade apenas de incluir estes serviços nas classes que se deseja utiliza-los. Alguns dos serviços disponíveis do *AngularJS* são: *\$http*, *\$scope*, *\$rootScope*, *\$location* e etc.

No lado do servidor é utilizado a linguagem *Java* [10] com o pacote de frameworks contidos no *SpringBoot* [11], que são: *Spring Security* [12], *Spring MVC* [13] e *Spring Data* [14]. A escolha destas tecnologias para o lado do servidor foi de acordo com a necessidade atual do mercado de desenvolvimento *Java* onde o framework *Spring Boot* está em alta. O *Spring Security* é o responsável pela segurança do sistema, com ele é feita a autenticação e controle de sessão do Odontofácil, o *Spring Data* é o responsável pela camada de persistência do Odontofácil e o *Spring MVC* é o responsável por garantir a arquitetura MVC “*Model View Controller*” do Odontofácil e também pelas injeções de dependências dentro das classes do sistema.

4.2 Autenticação e Permissões

A Figura 5 ilustra o método de autenticação no sistema do lado cliente. Na linha 1 é feita a declaração do módulo utilizado que no caso é **OdontoFacil**. Após isso, são feitas as injeções dos serviços utilizados na classe. Da linha 4 á linha 6 tem-se declarações de objetos onde a primeira recebe o nome de *ctrl* que é atribuída pela própria classe. Na linha 8 é utilizado o serviço *\$watch* que é responsável por atualizar o objeto *ctrl.funcionario*, criado na linha 5, sempre que este objeto receba alguma alteração. Da linha 11 até a linha 31, é definido o método responsável por encaminhar os dados de autenticação, para a classe *loginFactory* que é a classe que faz o envio dos dados cadastrados no formulário de login (Figura 9), para o lado do servidor.

```

1 angular.module('odontoFacil').controller('loginController', ['$scope', '$rootScope', '$http', '$location',
2   '$mdDialog', 'loginFactory', 'funcionarioFactory', 'Session', 'homeFactory', 'utilService', function($scope, $rootScope,
3     $http, $location, $mdDialog, loginFactory, funcionarioFactory, Session, homeFactory, utilService) {
4   var ctrl = this;
5   ctrl.funcionario = {};
6   ctrl.x = {};
7
8   $scope.$watch(function () { return ctrl.funcionario; }, function (newValue, oldValue) {
9     });
10
11   var authenticate = function(credentials, callback) {
12     var headers = credentials ? {authorization : "Basic "
13       + btoa(credentials.username + ":" + credentials.password)
14     } : {};
15
16     loginFactory.login(headers).then(function(response) {
17       if (response.data.name) {
18         $rootScope.authenticated = true;
19         ctrl.funcionarioLogado(response.data.name);
20         $location.path('/home');
21       } else {
22         $rootScope.authenticated = false;
23         credentials = {};
24         $scope.frmLogin.$setPristine();
25       }
26       callback && callback();
27     }, function() {
28       $rootScope.authenticated = false;
29       callback && callback();
30     });
31   }

```

Figura 5. Método de Autenticação AngularJS

Fonte: Próprio autor.

Na Figura 6 são criadas duas *Strings* contendo duas consultas, estas *Strings*, *USUARIO_POR_LOGIN* e *PERMISSAO_POR_USUARIO* são utilizadas pelo *Spring Security* para autenticação e controle de permissões no sistema (Figura 7).

```

68 private static final String USUARIO_POR_LOGIN = "SELECT login, senha, ativo FROM usuario "
69   + "WHERE login = ?";
70
71
72 private static final String PERMISSAO_POR_USUARIO = "SELECT u.login,"
73   + "u.nome_completo FROM permissoes_funcionarios up "
74   + "JOIN usuario u ON u.id_usuario = up.id_usuario "
75   + "JOIN permissao p ON p.id = up.id_permissao "
76   + "WHERE u.login = ?";

```

Figura 6. Criação das consultas para autenticação Java

Fonte: Próprio autor.

Na Figura 7 é possível observar o método que gerencia a autenticação do lado do servidor, o método *configure* é o responsável por receber os dados de autenticação, primeiramente pode-se ver na linha 119 que o método instancia um objeto da classe *AuthenticationManagerBuilder* [15] do *Spring Security* que precisa de alguns parâmetros para realizar a autenticação no sistema. Da linha 120 até a linha 126 são feitas as atribuições necessárias para a autenticação no Odontofácil. Na linha 122 é passando uma *String* com o nome de *dataSource* que contem os dados para conexão com o banco de dados. Na linha 123

a senha digitada no formulário de login para acesso ao sistema é criptografada com o método *passwordEncoder* que já recebeu a senha digitada no formulário e recebe o tipo de implementação para gerar a criptografia da senha, que é *BCryptPasswordEncoder*. Nas linhas 124 e 125 são passadas as duas *Strings* que contém as consultas mostradas na Figura 6, para serem executadas, o retorno da consulta verifica se o usuário existe no banco. O *framework Spring Security*[12]

```
118 @Override
119 public void configure(AuthenticationManagerBuilder builder) throws Exception {
120     builder
121         .jdbcAuthentication()
122         .dataSource(dataSource)
123         .passwordEncoder(new BCryptPasswordEncoder())
124         .usersByUsernameQuery(USUARIO_POR_LOGIN)
125         .authoritiesByUsernameQuery(PERMISSAO_POR_USUARIO)
126         .rolePrefix("ROLE_");
127     System.out.println(builder);
128 }
129 }
```

Figura 7. Método de Autenticação Java

Fonte: Próprio autor.

A Figura 8 ilustra o carregamento do funcionário que está logado no sistema. A função faz uma chamada via *GET* para a api que retorna o funcionário da sessão que é mantida pelo *Spring Boot*, com o *id* de permissão do funcionário carregado, sabemos quais restrições de acesso o funcionário logado deverá ter.

```
100 ctrl.funcionarioLogado = function(x) {
101     $http({
102         method: 'GET',
103         url: 'https://localhost:8443/userLogado/'
104     }).then(function successCallback(response) {
105         Session.create(response.data);
106         ctrl.x = response.data;
107     }, function errorCallback(response) {
108         console.log(response.data);
109         console.log(response.status);
110     });
111 }
112 }
```

Figura 8. Método no AngularJS que carrega o usuário logado.

Fonte: Próprio autor.

A Figura 9 apresenta a tela de login do OdontoFacil com o logo do sistema e um botão para logar no sistema.

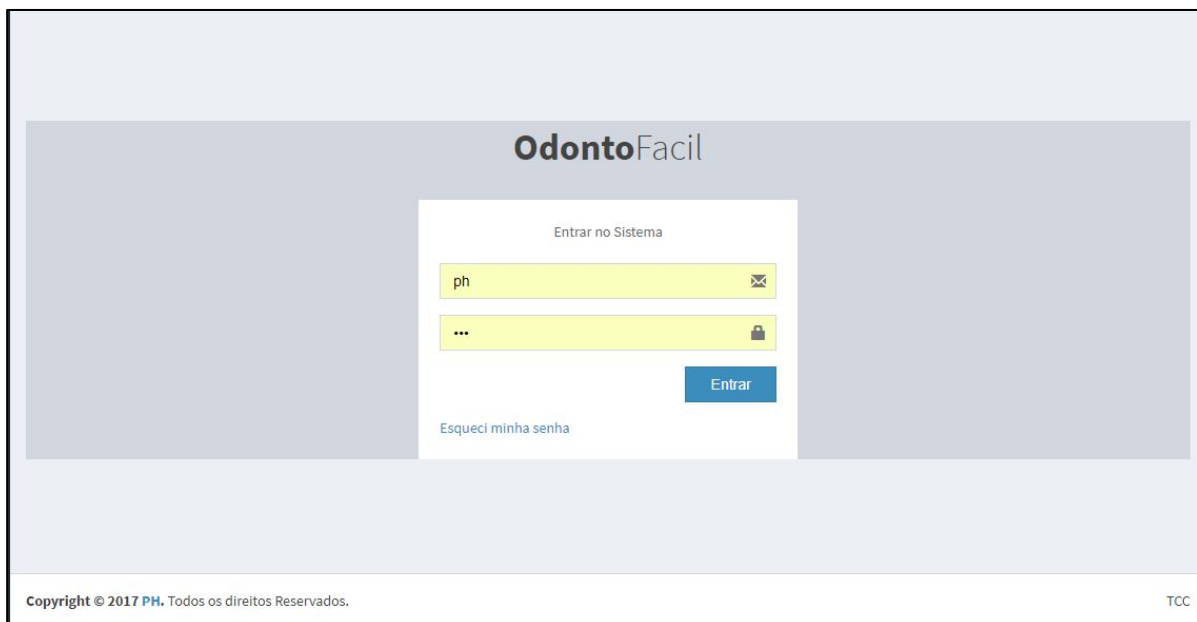


Figura 9. Tela de Login do Sistema

Fonte: Próprio autor.

4.2.1 Dashboard

A Figura 10 apresenta o Dashboard do sistema, acessado por um usuário com permissões de Administrador, com acesso a todas as funcionalidades no menu a esquerda: Home, Funcionários com um menu *dropdown* que tem as opções de cadastrar/editar e consultar funcionários, Agenda, Pacientes também com um menu *dropdown* com as opções de cadastrar/editar e consultar clientes, Financeiro que também conta com um menu *dropdown* com as opções de cadastrar/consultar despesas ou cadastrar/consultar receitas e a opção de Orçamentos que contém as funcionalidades de consultar, cadastrar, editar e imprimir

orçamentos. No centro da tela há *widgets*, com os últimos 8 pacientes cadastrados no sistema, aniversariantes do mês, receitas cadastradas no mês, número de consultas do dia, despesas do mês e novos usuários no mês.

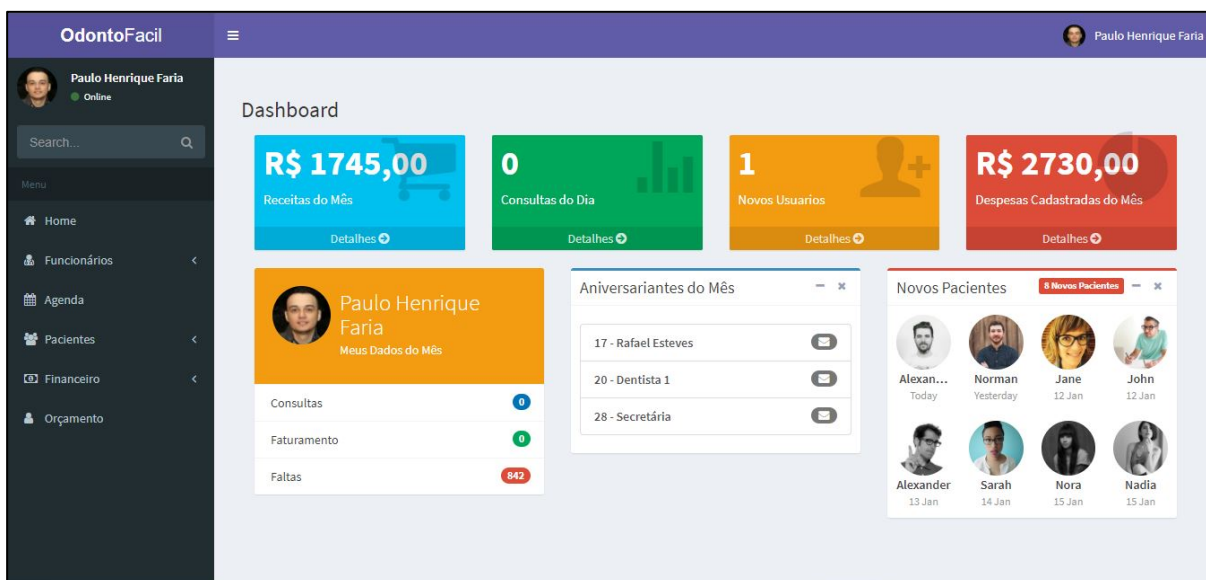


Figura 10. Dashboard do Sistema perfil de Administrador

Fonte: Próprio autor.

A Figura 11 apresenta o *Dashboard* do sistema acessado por um usuário com permissões de secretária. É possível observar que no perfil de secretária não existem as mesmas permissões que no perfil administrador. Existem restrições para que o perfil não tenha acesso as receitas cadastradas no mês dentro do menu *dropdown* financeiro, também não há possibilidade de acessar a tela de orçamentos. Apenas usuários com permissões de dentista e administrador possuem permissão para acessar a funcionalidade de geração de orçamentos.

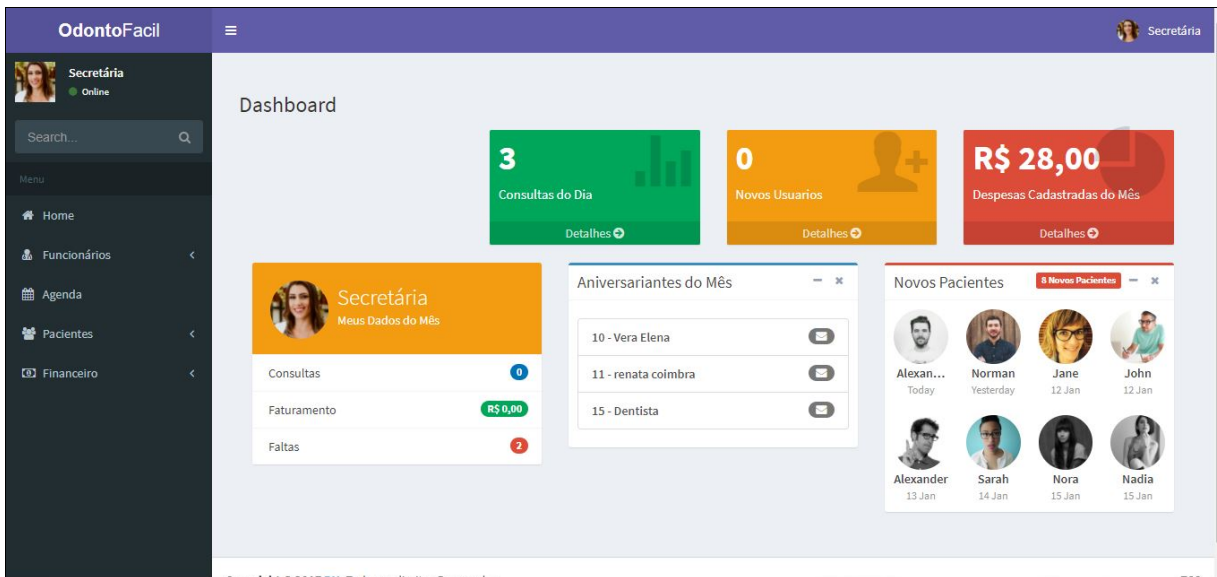


Figura 11. Dashboard do Sistema perfil de Secretária

Fonte: Próprio autor.

A Figura 12 apresenta o *Dashboard* do sistema acessado por um usuário com permissões de dentista. Nota-se que a permissão de dentista dentro do sistema é bem restrita, no menu principal a esquerda há apenas 3 opções sendo elas, home, agenda e orçamentos, também há restrições em relação aos *widjets* no centro da tela, se comparado com as outras duas permissões de acesso.

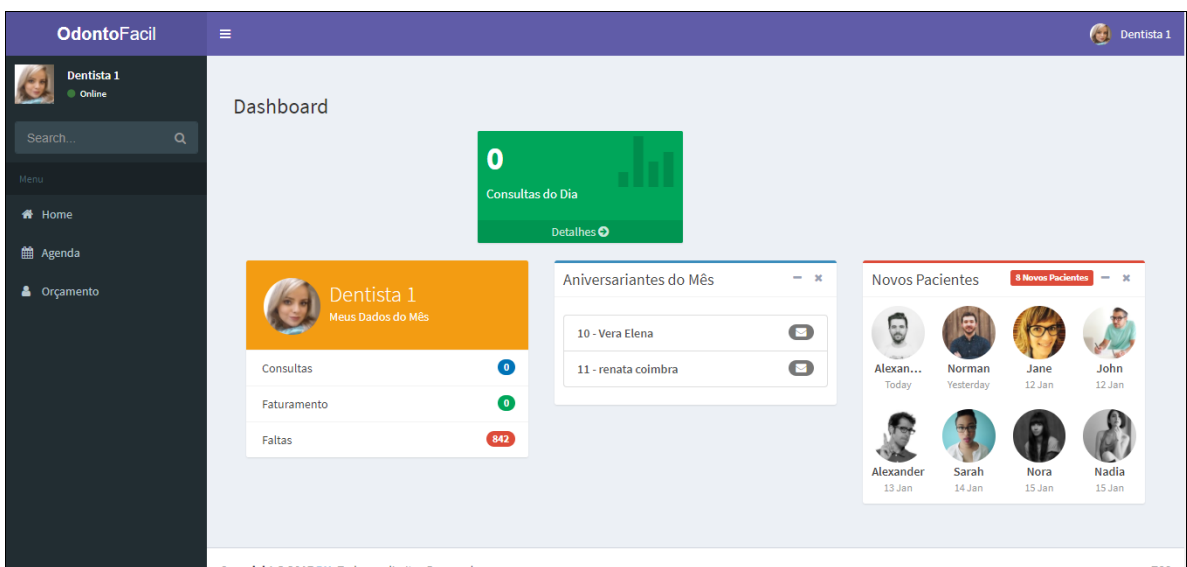


Figura 12. Dashboard do Sistema perfil de Dentista

Fonte: Próprio autor.

Na Figura 13 a permissão do usuário autenticado é de administrador. Também pode-se observar que é possível recolher o menu principal a esquerda e que o sistema é 100% responsivo podendo ser acessado através de tablets e smartphones.

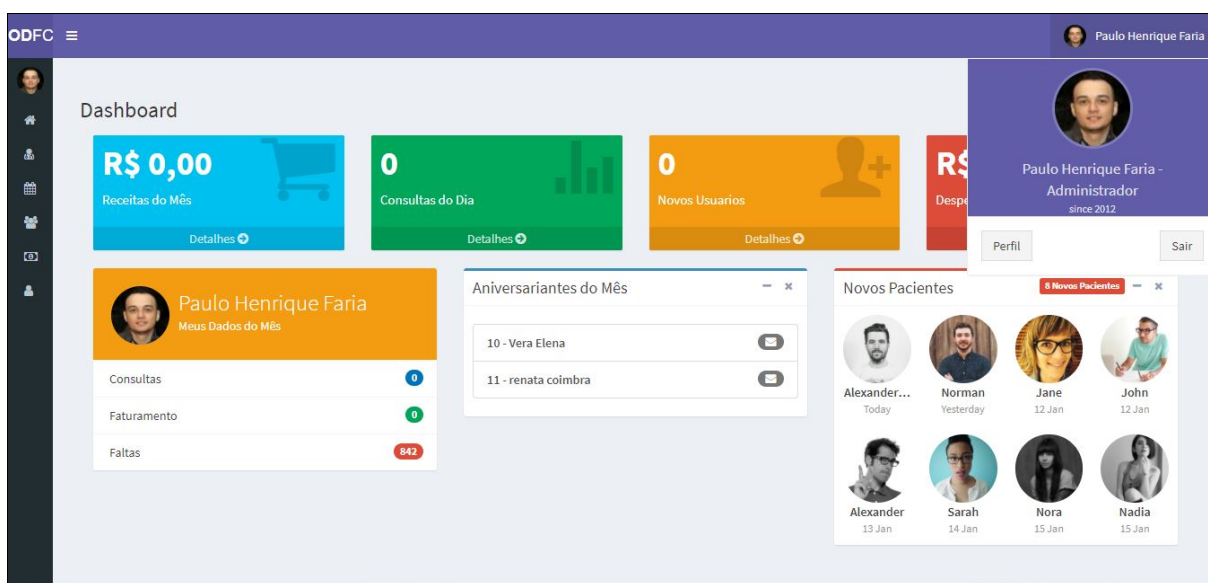


Figura 13. Responsividade do Sistema
Fonte: Próprio autor.

4.2.2 Agendamento e Consulta

A Figura 14 apresenta a tela de agendamento de consultas com permissões de Administrador. A Figura ilustra a agenda do framework FullCalendar [23], com alguns agendamentos marcados no sistema contendo o nome do paciente no título. Existem 3 status de agendamentos nesta tela, agendamentos na cor azul, são agendamentos que ainda não tiveram suas consultas finalizadas no sistema, sendo que para encerrar uma consulta no sistema é necessário preencher o valor recebido da consulta na tela de consulta. Agendamentos na cor verde são agendamentos que já tiveram sua consulta finalizada.

Agendamentos na cor vermelha são agendamentos que os pacientes não compareceram, ou cancelaram por algum motivo.

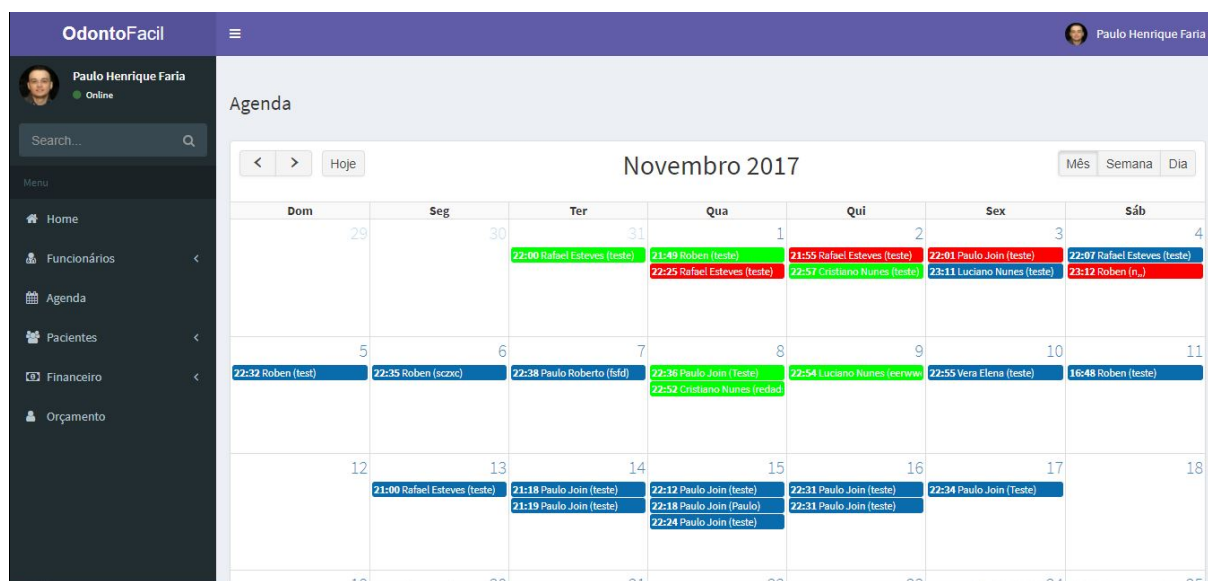


Figura 14. Agenda do Sistema
Fonte: Próprio autor.

Na Figura 15 é possível observar o modal que é aberto ao selecionar a data no calendario para realizar o agendamento no sistema. Este modal é para um novo agendamento em uma data futura. Os campos do formulário são: 2 *combobox* um para selecionar o paciente e o outro para selecionar o dentista, há também um campo para o horário da consulta e o campo de procedimentos da consulta, que será cadastrado pelo dentista durante ou após a consulta. Por ultimo há um *checkbox* para informar caso o paciente não tenha comparecido a consulta. Caso o valor da consulta ja tenha sido preenchida o modal habilita um botão com a opção de gerar atestado e um botão com a opção de gerar um recibo para imposto de renda como pode-se ver na Figura 16.

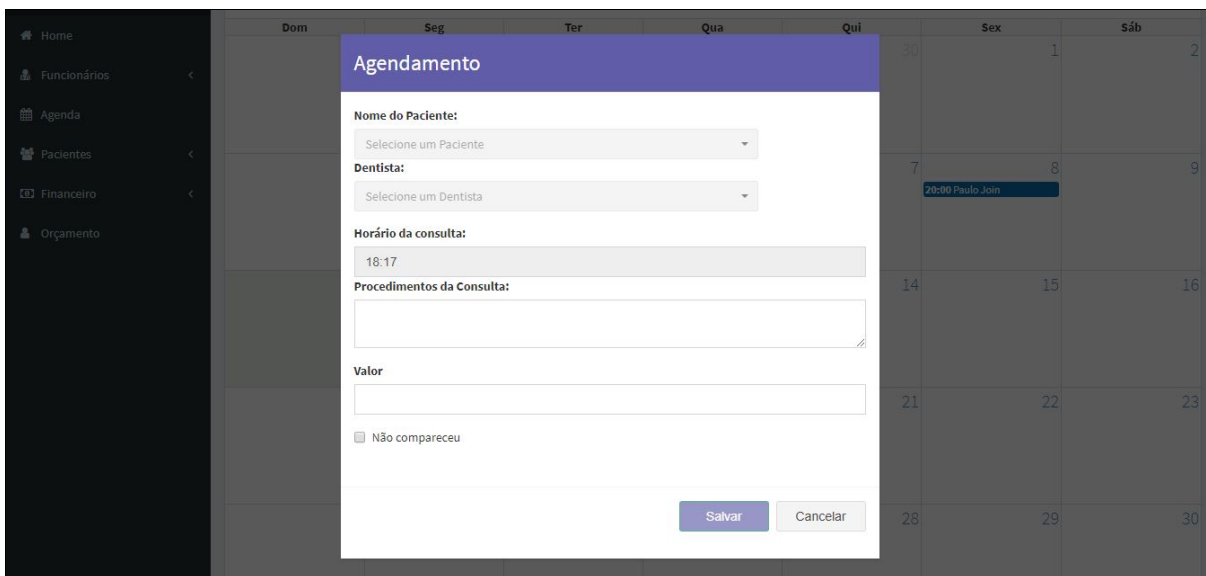


Figura 15. Modal Agendamento Administrador e Secretária.

Fonte: Próprio autor.

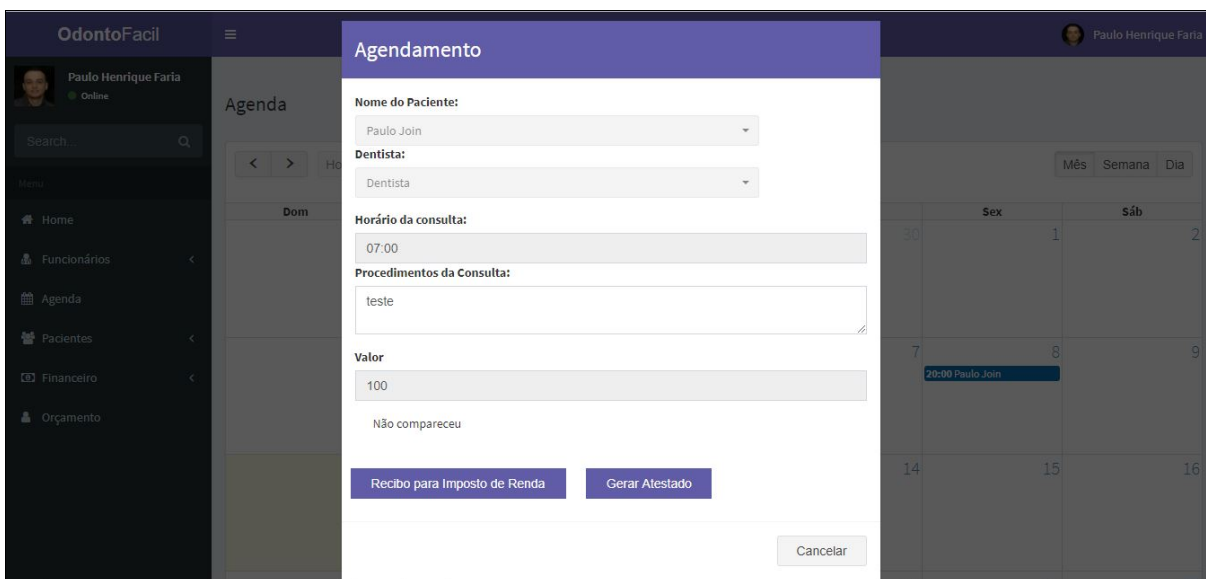


Figura 16. Modal Agendamento opção gerar atestado.

Fonte: Próprio autor.

Nas Figuras 17, 18 e 19 é descrito o método salvarAgendamento. Entre as linhas 240 e 242 [Figura 17] tem-se a declaração do método com a anotação *@RequestMapping* que o *Spring* utiliza para mapear a chamada do *web service*. Na linha 241 [Figura 17] as anotações de *produces* e *consumes* informam o formato de dado que será consumido e produzido no retorno do método. Neste caso é passado o parâmetro *APPLICATION_JSON_VALUE* que será o formato *json*[16]. Na linha 242 Figura 17 define-se o nome do método e a anotação *@RequestBody* e logo em seguida são declarados os objetos que serão recebidos no corpo do *json*, um objeto do tipo agendamento e o usuário logado. Na linha 273 Figura 18 é verificado se na tela de agendamento a *flag* *naoCompareceu* está marcada. Caso a *flag* estiver marcada, o agendamento recebe na linha 274 Figura 18 a cor vermelha na agenda. Na linha 275 Figura 18 é verificado se o valor foi preenchido para a consulta do agendamento. Caso o valor tenha sido preenchido na linha 277 Figura 18 a consulta é encerrada e o agendamento recebe a cor verde na agenda, caso contrário o agendamento recebe a cor azul na linha 279 Figura 18. Na linha 282 Figura 18 é enviado o agendamento para a classe *AgendamentoService* que é responsável por enviar este agendamento para a interface *AgendamentoRepository* que é a responsável por salvar o agendamento no banco de dados. Da linha 289 Figura 18 até a linha 308 Figura 19 são preenchidos os dados do agendamento para o envio de um email notificando o profissional cirurgião dentista e o paciente do agendamento feito no sistema Figura 21. Na linha 311 Figura 19 é efetuado o envio do email.

Na linha 310 Figura 19 é enviado o objeto agendamento para a classe *smsController*, esta classe é responsável por realizar o envio de um SMS para o paciente com os dados do agendamento feito pelo sistema Figura 20.

```

240= @RequestMapping(value = "/salvarAgendamento", method = {
241     RequestMethod.POST }, produces = MediaType.APPLICATION_JSON_VALUE, consumes = MediaType.APPLICATION_JSON_VALUE)
242 public Agendamento salvarAgendamento(@RequestBody Agendamento agendamento, Principal user) throws Exception {
243     System.out.println("salvarAgendamento: início");
244
245     Funcionario funcionario;
246
247     if (user != null) {
248         System.out.println("user.getName(): " + user.getName());
249
250         funcionario = funcionarioService.buscaPorLogin(user.getName());
251
252         if (agendamento.getFuncionario() == null) {
253             agendamento.setFuncionario(funcionario);
254         }
255
256         if (funcionario == null) {
257             System.out.println("Funcionario nulo em getFuncionarioLogado");
258
259             throw new Exception("Erro ao carregar funcionario. Faça login novamente.");
260         }
261     } else {
262         System.out.println("user nulo em getFuncionarioLogado");
263         throw new Exception("Erro ao carregar funcionario. Faça login novamente.");
264     }
265
266

```

Figura 17. Método de salvar agendamentos parte 1

Fonte: Próprio autor.

```

267     } else {
268         System.out.println("user nulo em getFuncionarioLogado");
269         throw new Exception("Erro ao carregar funcionario. Faça login novamente.");
270     }
271
272
273     if (agendamento.isNaoCompareceu()) {
274         agendamento.setColor(COR_AGENDAMENTO_NAO_COMPARECEU);
275     } else if (agendamento.getValor() != null) {
276         agendamento.setFechado(true);
277         agendamento.setColor(COR_AGENDAMENTO_FECHADO);
278     } else {
279         agendamento.setColor(COR_AGENDAMENTO_DEFAULT);
280     }
281
282     agendamento = agendamentoService.salvar(agendamento);
283
284     if (agendamento == null) {
285         System.out.println("Erro ao salvar no BD.");
286         throw new Exception("Não foi possível salvar o agendamento!");
287     } else {
288         if (!ehEdicao && agendamento.getStart().getTime().after(new Date())) {
289             Email email = new Email();
290             // Pra quem vai mandar
291             email.setTo(agendamento.getCliente().getEmail());
292             email.setCc(agendamento.getFuncionario().getEmail());
293             // Quem ta mandando
294             email.setFrom("tccodontofacil@gmail.com");

```

Figura 18. Método de salvar agendamentos parte 2

Fonte: Próprio autor.


```

295 // senha
296 email.setPass(password);
297 email.setTexto("");
298 email.setEmailFormatado("<html>"
299 + "<body>"
300 + "<div style=\"text-align: center;\">"
301 + "<span style=\"font-size:16px;\">Olá, "+ agendamento.getCliente().getNomeCompleto() +",</span></h2><br>"
302 + "<span style=\"font-size:16px;\">Este &eacute; um e-mail autom&aacute;tico "
303 + "para informar que seu agendamento esta marcado para as "+ agendamento.getStart().HOUR_OF_DAY
304 + ":"+ agendamento.getStart().MINUTE + " do dia "+ agendamento.getStart().getTime() +"/"
305 + ", com o Profissional "+ agendamento.getFuncionario().getNomeCompleto() + ".</span></h2>"
306 + "<p>"
307 + "<strong><span style=\"font-size:16px;\">Na Odonto F&aacute;cil, "
308 + " avenida João Antônio da Silveira Número 1580.</span></strong></p>");
309
310 SendEmailController sendMail = new SendEmailController(email);
311 smsController.EnviaSMS(agendamento);
312 }
313 }
314
315 if (agendamento.getColor() != null && agendamento.getColor().equals(COR_AGENDAMENTO_NAO_COMPARECEU)) {
316     agendamento.setNaoCompareceu(true);
317 }

```

Figura 19. Método de salvar agedamentos parte 3

Fonte: Próprio autor.

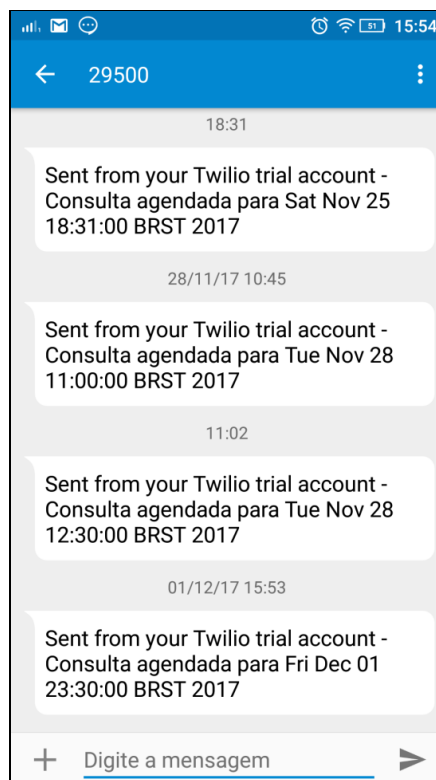


Figura 20. SMS com as informações do Agendamento.

Fonte: Próprio autor.

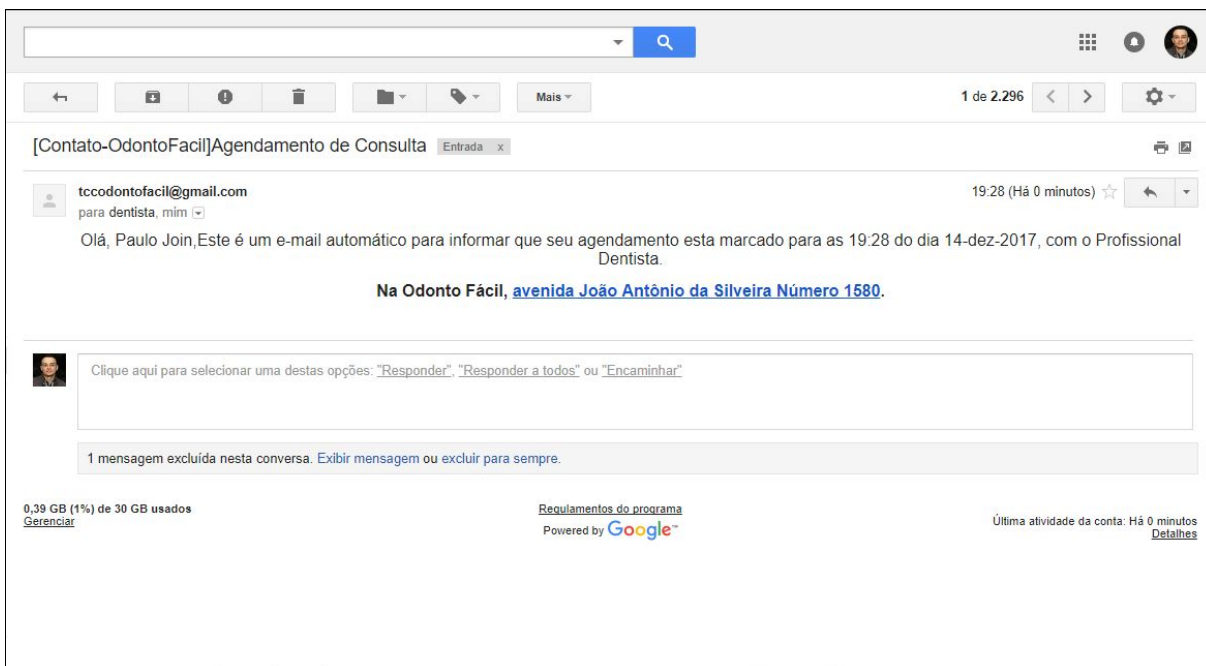


Figura 21. Template notificação por email

Fonte: Próprio autor.

Na agenda do perfil dentista (Figura 22), são listadas apenas as consultas agendadas para o dentista autenticado.

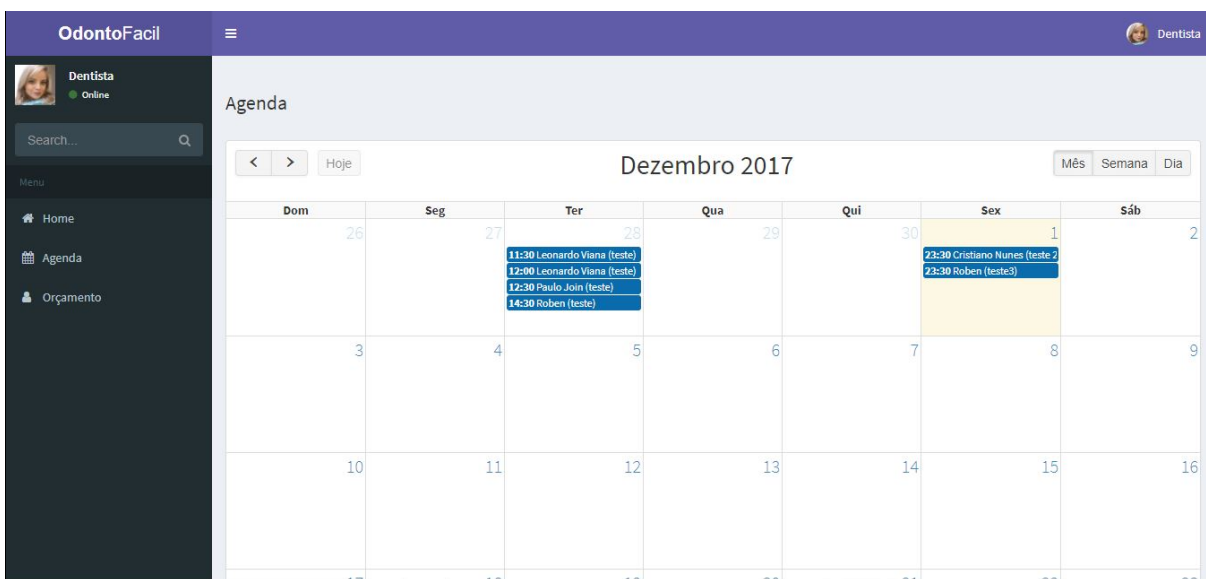


Figura 22. Tela de Agendamento com perfil de Dentista.

Fonte: Próprio autor.

No modal de agendamento para o perfil dentista (Figura 23), não está disponível a opção para selecionar o dentista para o agendamento, pois a regra do sistema define que o perfil dentista só pode marcar agendamentos para seu próprio usuário no sistema.

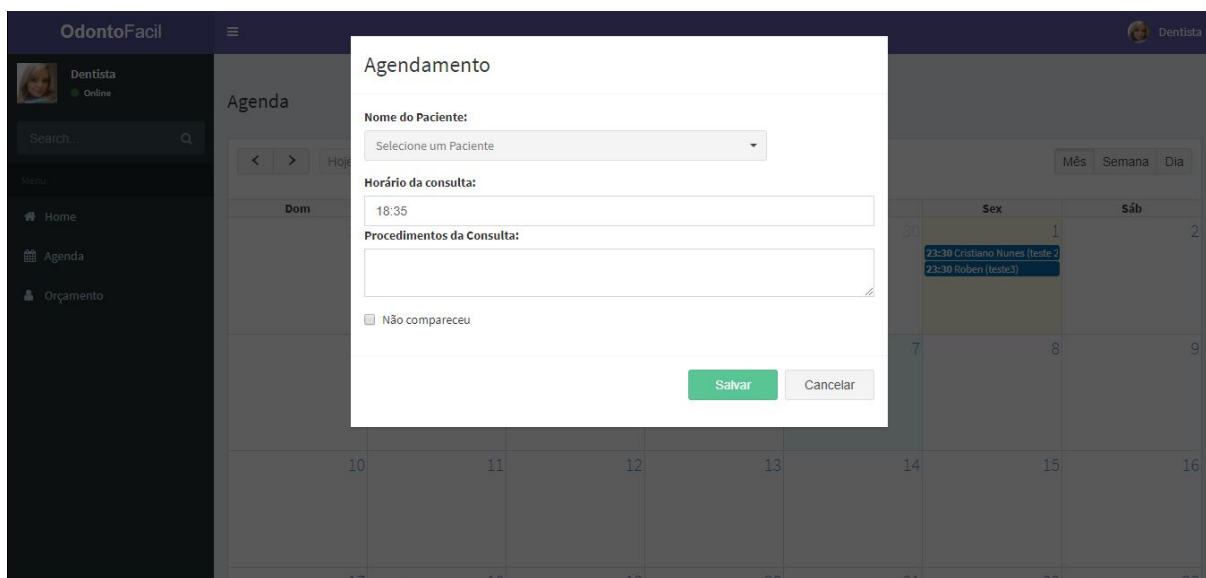


Figura 23. Modal de Agendamento com perfil de Dentista.

Fonte: Próprio autor.

4.2.3 Backup

O sistema Odontofácil contém uma funcionalidade de realizar backups diariamente. O backup do sistema é feito de forma incremental, mantendo um histórico de até 10 arquivos versionados. O sistema realiza automaticamente os backups quando um funcionário logar no sistema. Os dados são criptografados com a biblioteca *Cipher* [17]. A realização do backup é feita pela ferramenta *mysqldump* [18].

4.2.4 Envio de log

O sistema Odontofácil conta com uma funcionalidade de gerar logs caso ocorra algum erro no sistema, além de gerar um arquivo em um diretório específico que pode ser alterado para cada necessidade. O Odontofácil também faz o envio automaticamente com o arquivo gerado com o log do sistema por email, para o email do Administrador do sistema.

A Figura 24 ilustra método que realiza o envio do arquivo de log, como anexo no email que é enviado automaticamente pelo sistema caso ocorra alguma *exception*. Na linha 65 é informado onde está o arquivo contendo o log do sistema. Na linha 70 anexa-se o arquivo ao email.

```
56     try {
57         MimeMessage message = new MimeMessage(session);
58         if(email.getFrom() != null){
59             message.setFrom(new InternetAddress(email.getFrom()));
60             message.addRecipient(Message.RecipientType.TO, new InternetAddress(email.getTo()));
61             //ISSO é o titulo do EMAIL
62             message.setSubject("[Contato-OdontoFacil]Agendamento de Consulta");
63             message.setContent(this.corpo, "text/html");
64             MimeBodyPart mbp2 = new MimeBodyPart();
65             FileDataSource fds = new FileDataSource("C:\\Users\\phsil\\Desktop\\logsOdontoFacil\\erros.log");
66             mbp2.setDataHandler(new DataHandler(fds));
67             mbp2.setFileName(fds.getName());
68             Multipart mp = new MimeMultipart();
69             mp.addBodyPart(mbp2);
70             message.setContent(mp);
71             Transport.send(message);
72         }
73     } catch (MessagingException e) {
74         e.printStackTrace();
75         SalvarEnviarLogs.gravarArquivo(e);
76     }
77 }
```

Figura 24. Método de Envio de log com erro por email.

Fonte: Próprio autor.

4.2.5 Banco de Dados

No sistema Odontofácil é utilizado o *SGBD* [20] (Sistema de Gerenciamento de Banco de Dados) *Mysql* [19] que utiliza como linguagem *SQL* [21] (*Structured Query Language*), o *Mysql* é utilizado por ser o segundo *SGBD* mais utilizado na atualidade [22].

4.2.6 Financeiro

O Odontofácil disponibiliza um módulo financeiro que possibilita o cadastro de despesas e receitas do consultório, guardando históricos e possibilitando gerar relatórios com dados de um período desejado.

Na tela de listagem de receitas, (Figura 25) é possível verificar o total de valores pagos nas consultas no período selecionado. Na tela de listagem de despesas, (Figura 26) são apresentadas informações do total de despesas cadastradas no período selecionado no sistema, bem como, o total de despesas pagas e o total de despesas em aberto.

OdontoFacil Paulo Henrique Faria

Receitas

Período: 08/11/2017 a 11/12/2017 Pesquisar

Total de valores pagos em consultas no período: R\$ 1655,00

Receitas no período

Funcionario	Paciente	Descrição	Data Consulta	Valor	Pago
Paulo Henrique	Paulo Join	Teste	08/11/2017	R\$ 550,00	Sim
Dentista	Cristiano Nunes	Colocação de Aparelho	08/11/2017	R\$ 550,00	Sim
Dentista	Luciano Nunes	Limpeza gengival	09/11/2017	R\$ 275,00	Sim
Paulo Henrique	Paulo Join	teste	28/11/2017	R\$ 280,00	Sim

Imprimir receitas do período

Figura 25. Tela de consulta de receitas.

Fonte: Próprio autor.

Figura 26. Tela de consulta de despesas.

Fonte: Próprio autor.

Na Figura 27, a função mostrada recebe dois parâmetros da tela de listagem de receitas (Figura 25), *dataInicial* e *dataFinal*, que são os dois parâmetros passados na linha 141 para a api fazer a consulta no banco de dados e retornar os dados recebidos nas linhas 143 e 144, o objeto *lstReceitas* contendo a lista de consultas com o valor de cada receita e o objeto *totalConsultasPeriodo* contendo o valor somado de receitas durante o período selecionado. Da linha 146 a linha 149 possíveis erros que possam retornar da api são tratados.

```

139     ctrl.pesquisarReceitasPeriodo = function(dataInicial, dataFinal) {
140         ctrl.searchDisabled = true;
141         financeiroFactory.listarConsultasPorPeriodo(dataInicial, dataFinal).then(
142             successCallback = function(response) {
143                 financeiroFactory.setLstReceitas(response.data.lstAgendamentos);
144                 financeiroFactory.setTotalConsultasPeriodo(response.data.totalConsultas);
145             },
146             errorCallback = function(error) {
147                 financeiroFactory.setTotalConsultasPeriodo(0);
148                 ctrl.searchDisabled = false;
149                 utilService.tratarExcecao(error);
150             }
151         );
152     };

```

Figura 27. Função para listagem de receitas.

Fonte: Próprio autor.

A Figura 28 mostra o relatório de despesas, onde é possível listar as despesas de um determinado período definido como parâmetro na tela de despesas (Figura 26). No relatório constam informações como, id da despesa, descrição, se a despesa já foi paga, o vencimento e o valor da despesa.

Despesas				
OdontoFacil				
vencimento				
id	Descrição	Conta paga	Vencimento	Valor
4	Luz	Sim	18/12/17 00:00	328.00
				Total:328.00
Período de		Até		
01/12/17 00:00		31/12/17 23:59		
Domingo 10		Page 1 of 1		

Figura 28. Relatório listagem de despesas.

Fonte: Próprio autor.

A Figura 29 apresenta o relatório de receitas onde é possível listar as receitas de um determinado período definido como parâmetro na tela de receitas (Figura 25). No relatório constam informações como, id da consulta, descrição da consulta, o dentista que realizou a consulta, a data da consulta e o valor recebido na consulta.

Receitas				
OdontoFacil				
id	Descrição	Funcionario	dataConsulta	Valor
96	teste	Dentista	06/12/17 07:00	100.00
				Total:100.00
Período de		Até		
01/12/17 00:00		31/12/17 23:59		
Domingo 10		Page 1 of 1		

Figura 29. Relatório listagem de receitas.

Fonte: Próprio autor.

4.2.7 Atestado

A funcionalidade de gerar atestados foi implementada com o framework *Jasper Reports* [24]. O modelo utilizado é um template de exemplo da ferramenta *Ireport* [23]. O atestado contém o nome, cpf, logradouro e número dinâmicos passados através do código para preencher o documento.

Atestado
OdontoFacil

Atesto para devidos fins que Rafael Estevez
CPF: 02654612007 residente em Avenida Borges de Medeiros 4262
esteve sob tratamento Odontológico neste consultório, no
período das _____ às _____ horas do
dia ____/____/____.

Necessita de convalescença ? () SIM () NÃO

Período _____

Porto Alegre ____/____/____

Assinatura do Paciente Assinatura

Segunda-feira, 04 Page 1 of 1

Figura 30. Atestado nesta figura os dados mencionados são hipotéticos.
Fonte: Próprio autor.

4.2.8 Orçamento

O sistema Odontofácil contém um módulo de orçamentos onde é possível cadastrar, listar, imprimir e deletar orçamentos gerados para clientes em consultas. A funcionalidade de guardar histórico possibilita o reaproveitamento de orçamentos para dar mais agilidade aos atendimentos. Todos os sistemas pesquisados previamente não disponibilizam esta funcionalidade para guardar históricos de orçamentos.

ID	Título	Paciente	Dentista	Ações
3	Limpeza	Rafael Esteves	Paulo Henrique	
4	Colocação de Aparelho	Luciano Nunes	Dentista	
5	Tratamento Endodontico	Roben	Paulo Henrique	
6	Restauração	Cristiano Nunes	Paulo Henrique	
7	Prótese Parcial	Vera Elena	Dentista	
8	Raio X	Paulo Roberto	Paulo Henrique	
9	Raspagem Supragengival	Paulo Roberto	Dentista	
10	Prótese Total	Vera Elena	Dentista	
11	Raspagem Subgengival	Paulo Join	Dentista	
12	Implante	Paulo Join	Paulo Henrique	

Figura 31. Tela de Orçamentos.

Fonte: Próprio autor.

Na Figura 32 pode-se observar o modelo do orçamento gerado pelo sistema, que inclui o nome do paciente, procedimento, valor, nome do profissional cirurgião dentista e endereço, assim como nos demais templates de relatórios do sistema. O modelo de orçamento foi elaborado com o auxílio de uma profissional da área.

Orçamento

OdontoFacil

Paciente: Paulo Henrique

Procedimento: gerador de blablabla gerador de blablabla gerador de blablabla
gerador de blablabla gerador de blablabla gerador de blablabla
gerador de blablabla gerador de blablabla gerador de blablabla
gerador de blablabla gerador de blablabla gerador de blablabla
gerador de blablabla gerador de blablabla gerador de blablabla
gerador de blablabla gerador de blablabla gerador de blablabla
gerador de blablabla gerador de blablabla gerador de blablabla
gerador de blablabla gerador de blablabla gerador de blablabla
gerador de blablabla gerador de blablabla gerador de blablabla
gerador de blablabla gerador de blablabla gerador de blablabla
gerador de blablabla gerador de blablabla gerador de blablabla
gerador de blablabla gerador de blablabla gerador de blablabla

Valor: R\$ 380

Renata Capitão Coimbra

Avenida Borges de Medeiros 250
Porto Alegre

Figura 32. Relatório orçamento.

Fonte: Próprio autor.

4.2.9 Recibo

Uma funcionalidade que o sistema Odontofácil disponibiliza é a de gerar recibos das consultas realizadas pelos profissionais, para facilitar a posterior declaração em seu imposto de renda. Os demais sistemas pesquisados não disponibilizam esta funcionalidade, o que é mais um diferencial do sistema OdontoFacil. Este recibo inclui dados como o nome do profissional, nome do cliente, valor pago, CRO, CPF do profissional, endereço do profissional e cep.

Recibo

OdontoFacil

Renata Capitão Coimbra

CRO: 25523589 CPF: 02654612007
endereço: Avenida Borges de Medeiros Nº: 2500
CEP: 90110150

Recebi do Sr(a) Paulo Henrique Silveira Faria
o montante de R\$ 250 ,00 (valor por extenso) _____
_____ referente ao tratamento odontológico
realizado no mês de _____.

Por ser verdade assino.

Carimbo

Assinatura

Porto Alegre ____/____/____

Quinta-feira 07Page 1 of 1

Figura 33. Recibo para comprovante de renda, nesta figura os dados mencionados são hipotéticos.
Fonte: Próprio autor.

4.2.10 Criptografia

No Odontofácil é utilizado o framework *SpringSecurity* [12] para segurança da senha do funcionário, o tipo de chave utilizado pelo *framework* é uma chave simétrica, com o método de criptografia AES (*Advanced Encryption Standard*) de 256 bits, que também é utilizado para criptografia dos *Backups*.

5. CONCLUSÃO

A escolha deste TCC teve como foco o desenvolvimento de um sistema para consultórios odontológicos. Na pesquisa realizada previamente ao seu desenvolvimento foi constatada a necessidade de algumas funcionalidades que as soluções atuais não atendem, uma delas é a possibilidade de conter mais de um perfil de acesso, podendo ter restrições a determinados perfis. Esta necessidade foi relatada por uma profissional da área.

O sistema Odontofácil tem como diferencial além de 3 tipos de perfis a possibilidade de gerar atestados, orçamentos, recibos, além de conter notificações enviadas por email e SMS para pacientes e profissionais. O Odontofácil também disponibiliza uma funcionalidade que facilita na administração do sistema, pois caso o sistema retorne algum erro, um email com o log do sistema é enviado automaticamente para a conta do administrador do sistema. Assim, essas funcionalidades facilitam os processos de comunicação do consultório odontológico.

A escolha das tecnologias foi feita pela necessidade atual do mercado, pois os atuais *frameworks* utilizados para a linguagem de programação *JAVA* [10] são *Spring Boot* [11] e *AngularJS* [8] com a arquitetura *REST* [7].

Como sugestão de trabalhos futuros pode-se propor:

1. Integrar a agenda do sistema com o *framework* do *Google Calendar*;
2. Possibilitar para o paciente as opções de confirmar ou cancelar consultas por SMS;

3. Melhorar Layout da tela de login;
4. Disponibilizar exames online para pacientes;
5. Desenvolver um módulo de estoque para manter o gerenciamento dos consultórios.
6. Criar perfil para funcionários como auxiliar de dentista (técnico).

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Sindicato dos cirurgiões dentistas do Rio de Janeiro. Disponível em: <http://scdrj.org.br/?page_id=2239> Acessado em: 02 de outubro de 2017.
- [2] Conselho Federal de Odontologia do estado do Rio Grande do Sul. Disponível em: <<http://www.cfo.org.br>> Acessado em: 02 de outubro de 2017.
- [3] Simples Dental. Disponível em: <www.simplesdental.com> Acessado em: 02 de outubro de 2017.
- [4] Dental Office. Disponível em: <www.dentaloffice.com.br> Acessado em: 02 de outubro de 2017.
- [5] Jasper Reports. Disponível em: <<https://community.jaspersoft.com/project/jasperreports-library/>> Acessado em: 10 de outubro de 2017.
- [6] HTML 5. Disponível em: <https://www.w3schools.com/html/html5_intro.asp/> Acessado em: 12 de outubro de 2017.
- [7] REST. Disponível em: <<https://pt.wikipedia.org/wiki/REST/>> Acessado em: 18 de outubro de 2017.
- [8] AngularJS. Disponível em: <<https://angularjs.org/>> Acessado em: Acessado em: 18 de outubro de 2017.

- [9] Bootstrap. Disponível em: <<https://getbootstrap.com/>> Acessado em: 26 de outubro de 2017.
- [10] Java. Disponível em: <https://www.java.com/pt_BR/> Acessado em: 28 de outubro de 2017.
- [11] SpringBoot. Disponível em: <<https://projects.spring.io/spring-boot/>> Acessado em: 01 de novembro de 2017.
- [12] SpringSecurity. Disponível em: <<https://projects.spring.io/spring-security/>> Acessado em: 01 de novembro de 2017.
- [13] Spring MVC. Disponível em: <<https://projects.spring.io/spring-restdocs/>> Acessado em: 01 de novembro de 2017.
- [14] Spring Data. Disponível em: <<http://projects.spring.io/spring-data/>> Acessado em: 01 de novembro de 2017.
- [15] AuthenticationManagerBuilder. Disponível em: <<https://docs.spring.io/spring-security/site/docs/current/apidocs/org/springframework/security/config/annotation/authentication/builders/AuthenticationManagerBuilder.html>> Acessado em: 04 de novembro de 2017.
- [16] JSON. Disponível em: <<https://www.json.org/>> Acessado em: 17 de novembro de 2017.
- [17] Cipher. Disponível em: <<https://docs.oracle.com/javase/7/docs/api/javax/crypto/Cipher.html>> Acessado em: 17 de novembro de 2017.
- [18] Mysqldump. Disponível em: <<https://dev.mysql.com/doc/refman/5.7/en/mysqldump.html>> Acessado em: 22 de novembro de 2017.
- [19] Mysql. Disponível em: <<https://dev.mysql.com/doc/refman/5.7/en/mysqldump.html>> Acessado em: 22 de novembro de 2017.

[20] Sistema Gerenciador de Banco de Dados. Disponível em:

<https://pt.wikipedia.org/wiki/Sistema_de_gerenciamnto_de_banco_de_dados/>

Acessado em: 22 de novembro de 2017.

[21] SQL. Disponível em: <<https://www.w3schools.com/sql/>> Acessado em: 27 de

novembro de 2017.

[22] DB-Engines Ranking. Disponível em: <<https://db-engines.com/en/ranking/>>

Acessado em: 27 de novembro de 2017.

[23] Ireport. Disponível em: <[https://community.jaspersoft.com/project/ireport-](https://community.jaspersoft.com/project/ireport-designer/)

[designer/](https://community.jaspersoft.com/project/ireport-designer/)> Acessado em: 01 de dezembro de 2017

[24] Full Calendar. Disponível em: <<https://fullcalendar.io/>> Acessado em: 01 dezembro

2017