

Sistema de Gestão para Clínicas Odontológicas

Frederico G. Hartmann¹, Gustavo Stangherlin Cantarelli¹

¹Curso de Sistemas de Informação – Universidade Franciscana
CEP 97010-032 – Santa Maria – RS – Brasil

f.hartmann@ufn.edu.br, gus.cant@gmail.com

Abstract. *This article presents a proposal for a web system for managing a dental clinic, where dentists are able to schedule their appointments and store data on their clients and treatments. The system also generates reports with information about clients and appointments. The system will be developed with Javascript React library and it will use the NodeJS platform and the PostgreSQL DBMS. The methodology chosen for implementation was Feature-Driven Development (FDD).*

Resumo. *Este artigo apresenta a proposta de um sistema web para a gestão de uma clínica odontológica, por meio do qual dentistas podem agendar suas consultas e guardar dados de seus clientes e tratamentos. O sistema também gera relatórios com informações sobre clientes e agendamentos. O sistema foi desenvolvido utilizando a biblioteca Javascript React e utiliza plataforma NodeJS, juntamente com o SGBD PostgreSQL. A metodologia escolhida para a implementação foi a Feature-Driven Development (FDD).*

1. Introdução

Os avanços tecnológicos vêm transformando o trabalho nas mais diversas empresas, fazendo com que aquelas, que não se adaptam às mudanças, acabem ficando para trás daquelas que utilizam as tecnologias a seu favor. O número de empresas informatizadas no Brasil vem aumentando, tendo em vista a busca por vantagens competitivas, diminuição dos custos de tecnologia, exigência de parceiros comerciais e, até mesmo, por imposições legais [Estevam *et al.* 2020].

Dessa forma, as empresas que abraçam o uso de tecnologias de informação acabam por ter vantagens que se apresentam na forma de melhorias nas tomadas de decisões, no controle de informações, no atendimento aos clientes e avanços no processo produtivo, podendo, assim, aumentar a produtividade [Estevam *et al.* 2020].

Entre as áreas que estão cada vez mais avançadas tecnologicamente, está a área da Odontologia, que tem recebido suporte para trazer melhores tratamentos aos seus clientes e tornando o trabalho do cirurgião-dentista mais efetivo.

1.1. Justificativa

O papel não apresenta boa eficiência para guardar grande quantidade de informações, devido ao custo de armazenamento, aos riscos de segurança, com documentos que podem ser perdidos ou danificados, e à falta de acesso rápido às informações [Chao 2016]. Tendo isso em mente, surgiu a ideia de melhorar os processos de uma clínica de Odontologia a partir de tecnologias que possam retirar estas informações do papel, facilitando processos e diminuindo a ocorrência de erros.

1.2. Objetivo geral

Este trabalho teve como objetivo o planejamento e desenvolvimento de um sistema de gerenciamento em plataforma web, que auxilie no gerenciamento de clínicas odontológicas de forma a guardar informações importantes de maneira reunida e organizada.

1.3. Objetivos específicos

Os objetivos específicos deste trabalho são:

- Fazer uso da biblioteca Javascript React juntamente a plataforma NodeJS para o desenvolvimento do software;
- Fazer uso do SGBD PostgreSQL, devido sua fácil manutenção e quantidade de documentação;
- Utilizar as boas práticas da metodologia FDD (*Feature Driven Development*) no projeto, de forma a tentar obter melhores resultados;
- Criar interfaces intuitivas e de fácil utilização;

2. Referencial Teórico

Nesta seção, serão apresentadas revisões bibliográficas e pesquisas relacionadas, com o objetivo de fundamentar o trabalho. Aqui, também serão abordadas as tecnologias que serão utilizadas para o planejamento e desenvolvimento da aplicação.

2.1. Odontologia

A odontologia é um dos pilares que garante a saúde da população. Os dentistas são os profissionais responsáveis pela manutenção da saúde e recuperação funcional do aparelho mastigatório. Ou seja, cuidam não somente dos dentes, mas também das estruturas adjacentes que os suportam e de certas porções da cavidade bucal, todas partes que precisam estar saudáveis para manter a harmonia do corpo humano. Outra função importante do dentista é a capacidade de detectar e evitar doenças através de exames realizados na boca, tendo em vista que várias possuem manifestação oral [Chaves 1986; Rodrigues 2015].

2.2. Avanço da tecnologia na Odontologia

A tecnologia vem evoluindo junto com a humanidade e, com isso, profissionais de todas as profissões devem estudá-las e usá-las de forma competente. As inovações tecnológicas vêm surgindo juntamente com o desenvolvimento das ciências [Brigagão 2015].

A Odontologia não escapa dessas evoluções, sendo ela mesma uma junção de conhecimentos científicos e técnicos que visam trazer serviços essenciais para a saúde humana [Brigagão 2015].

As novas tecnologias podem ajudar os cirurgiões-dentistas das mais variadas formas, dando suporte antes, durante e depois dos atendimentos ocorrerem. Exemplos dessas tecnologias que têm chamado a atenção dos dentistas são o uso de aparelhos ortodônticos invisíveis, estes feitos com um material que se assemelha a um plástico transparente, com o uso de impressão 3D em resina, dessa forma, permitindo ao cirurgião-dentista produzir dentaduras, por exemplo, sem depender de laboratórios.

Embora não tão recente, os avanços tecnológicos na forma de softwares também vêm se tornando cada vez mais essenciais para o sucesso da gestão odontológica. Entre os problemas encontrados nas clínicas dentárias, estão a grande quantidade de papel que é gerada, que acarreta dificuldades na hora de se fazer armazenamentos e de buscar as informações, ou nas poucas funções que uma simples planilha eletrônica apresenta, sem conseguir apresentar dados significativos sobre as finanças, por exemplo. Ao se utilizar tecnologia para se melhorar o armazenamento desses dados, é colocada à disposição do profissional maior controle sobre as informações, economizando o tempo do profissional e garantindo que eles se encontrem atualizados.

2.3. NodeJS

É uma plataforma de desenvolvimento de aplicações de lado servidor, altamente escalável e de baixo nível, na qual programas escritos em JavaScript são interpretados pela *engine* V8, a mesma utilizada pelo navegador *Google Chrome* [Pereira 2014].

A característica principal do NodeJS é sua arquitetura não bloqueante, ou seja, não há paralização, o que resulta em uma boa performance em relação ao uso de memória, devido à diminuição do tempo ocioso do processador. Além disso, sistemas que utilizam NodeJS não sofrem de *deadlocks*, visto que eles utilizam somente um *thread* por processo [Pereira 2014].

JavaScript é uma linguagem de programação web, de alto nível, dinâmica, interpretada e não tipada. Esta é utilizada pela grande maioria dos sites e possui interpretadores em todos os navegadores modernos [Flanagan 2013].

2.4. React

O React é uma biblioteca JavaScript para criação de interfaces para usuários, desenvolvido por engenheiros de software do Facebook para resolver desafios ligados a interfaces complexas com dados que mudam com o passar do tempo [Banks e Porcello 2017].

Tendo em vista que o React não é um *framework*, ele não traz consigo ferramentas da forma que um *framework* faria, deixando a decisão de quais ferramentas usar para ser feita pelo desenvolvedor [Banks e Porcello 2017].

2.5. PostgreSQL

O PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional, o qual, como o nome sugere, usa e estende a linguagem SQL, combinado com funcionalidades que guardam e escalam de forma segura dados complexos. Outras características que tornam esse SGBD uma boa escolha para ser utilizado é o fato dele ser gratuito, *open source*, extensível e seguir as propriedades ACID, que são: atomicidade, consistência, isolamento e durabilidade [PostgreSQL 2021].

3. Trabalhos Correlatos

Nesta seção, serão apresentados os trabalhos relacionados ao sistema proposto, os quais possuem algumas características semelhantes ao presente trabalho, com o intuito de contribuir como base de conhecimento para a elaboração e organização desse projeto.

3.1. *Web application for professional union using React and Node.js*

O trabalho de Koirala (2020) fala sobre o desenvolvimento de uma aplicação para o auxílio de união de profissionais de arquitetura a colocarem suas operações diárias em uma plataforma digital, sendo este seu principal objetivo. Com a aplicação, é esperada a diminuição nos custos, diminuindo o trabalho manual e a espera nas comunicações com os clientes, e a manutenção dos dados guardados de maneira mais segura para diminuir erros.

Outro objetivo do trabalho é o estudo de tecnologias de desenvolvimento modernas, implementando-as durante o desenvolvimento da aplicação. A aplicação foi desenvolvida em ambiente Windows, usando ReactJS, Redux e JavaScript para o *frontend*, NodeJS para o *backend* e MongoDB para o armazenamento de dados, além de outras bibliotecas para melhorar a performance em termos de segurança e interfaces de usuário.

3.2. Sistema odontológico Connecteeth

O trabalho de Souza *et al.* (2020) também trata do desenvolvimento de um software de agendamento, entretanto, possui como diferencial o fato de permitir que o cirurgião-dentista trabalhe em várias clínicas diferentes, além das funcionalidades de agendamento e prontuário.

A justificativa do trabalho é utilizar a tecnologia para conseguir vantagens competitivas aos usuários e otimizar processos de empresas de pequeno a grande porte.

Os objetivos do software são dar vantagem competitiva, com um apoio à decisão e permitir também que o usuário gerencie seus agendamentos e clientes categorizados por clínica, sendo este seu principal diferencial.

O sistema foi desenvolvido usando a linguagem de programação JavaScript, fazendo uso do *framework* Laravel, enquanto seus dados são gerenciados pelo SGBD MySQL.

3.3. Sisodonto: sistema para gerenciamento e agendamento odontológico

O trabalho de Dutra e Souza (2021) fala sobre um artigo que trata do desenvolvimento para um software de gerenciamento de clínicas odontológicas. O software possui função de login com três tipos de cadastros, agendamento, orçamento, fichas clínicas e dois relatórios, de agendamentos por data e orçamentos por paciente.

A justificativa do trabalho é levar soluções tecnológicas a problemas sofridos em clínicas pequenas e médias, tornando o trabalho mais dinâmico com a automatização de trabalhos repetitivos.

Os objetivos do software são diminuir o tempo de atendimento ao paciente, armazenar informações de forma segura, diminuir a utilização e acúmulo de papel, fornecer informações necessárias para a realização de tratamentos e fornecer apoio à decisão na forma de relatórios.

Para a modelagem do sistema foi utilizada a *Unified Modeling Language* (UML). O sistema foi desenvolvido usando a linguagem de programação C#, enquanto seus dados são armazenados em um banco SQL, gerenciado pelo SGBD MySQL

Com o artigo, é possível compreender que a Odontologia vem se desenvolvendo junto da tecnologia, com sistema auxiliando tanto na tomada de decisões quanto na gestão de clínicas, beneficiando pacientes e profissionais.

3.4. Codental

O Codental é um software brasileiro de agendamento odontológico, o qual se baseia em três menus, onde suas funcionalidades estão divididas, sendo: Agenda, Pacientes e Financeiro [Codental 2021].

No menu Agenda, os dentistas de uma clínica conseguem ver e gerenciar seus agendamentos, o sistema possui função de envio de mensagens via SMS e WhatsApp, para assuntos referentes a seus agendamentos.

No menu Pacientes, o software permite o cadastro dos pacientes, guardando diversas informações pessoais e de históricos, além de uma anamnese e a possibilidade de salvar arquivos que forem considerados importantes.

Já no menu Financeiro, é possível verificar as contas a serem pagas e recebidas, gerenciar pagamentos e emitir boletos a seus pacientes.

Outras funcionalidades do software são a geração de boletos e atestados, além de um chat por onde os dentistas cadastrados no sistema podem conversar.

Na Figura 1 é possível observar a função e agenda do software. Nesta tela, é possível adicionar, alterar e excluir agendamentos dos diferentes dentistas cadastrados. Ademais, nesta mesma tela, também é possível se redirecionar as páginas com as funções de gerenciamento de pacientes e financeiro, além de um chat para mandar mensagens diretamente para os atendentes da Codental [Codental 2021].

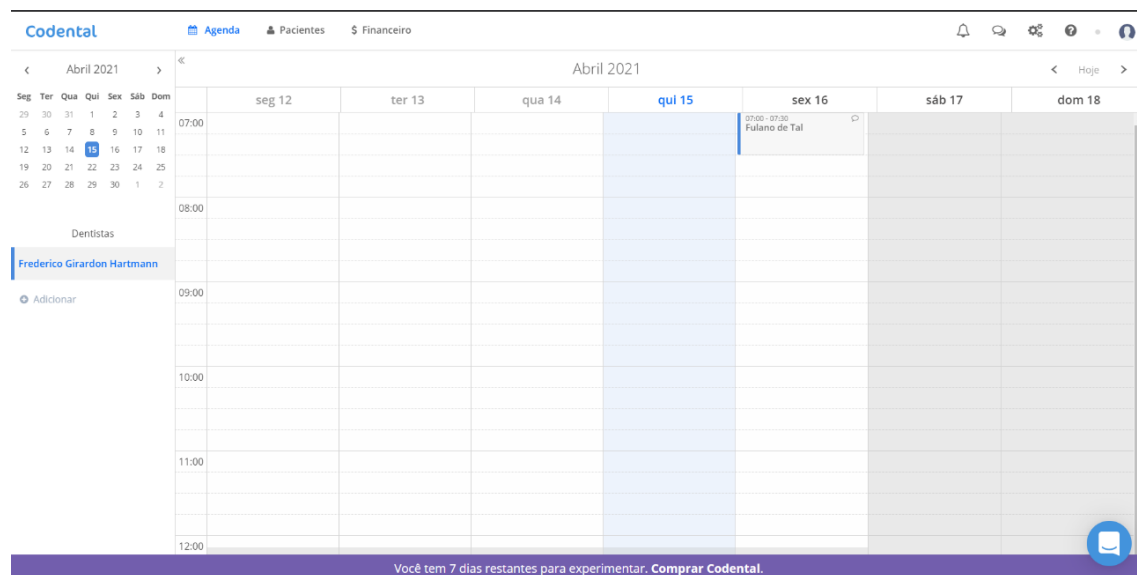


Figura 1. Agenda do Software Codental

3.5. Conclusão dos trabalhos correlatos

Os trabalhos pesquisados foram de alta relevância para o andamento deste trabalho. Sendo que dois dos trabalhos relacionados possuem propósito semelhante ao deste projeto, por serem softwares para agendamentos odontológicos.

No trabalho de Dutra e Souza (2020), o software se encontra restrito ao desktop, o que faz seu papel quanto ao requisito “sair do papel”, mas torna o usuário dependente de um computador, tornando o software menos conveniente.

O trabalho de Souza *et al.* (2020) possui uma proposta ainda mais semelhante, um software on-line, no entanto, possui poucas funções além do agendamento, controle de informações dos pacientes e pagamentos.

O software Codental (2021) possui também várias das características que se deseja implantar no sistema que será desenvolvido com este trabalho. Todavia, certas partes do software Codental, principalmente as configurações de mensagens, podem não ser facilmente compreendidas por aqueles que nunca utilizaram algum software de gestão.

Os trabalhos conseguiram atingir o objetivo de retirar dos papéis as diversas informações que poderiam ser perdidas por falta de cuidado daqueles que as manuseassem. Entretanto, em nenhum deles foram apresentadas formas de automatizar o contato dos pacientes com informações a respeito de seus próprios dados e agendamentos.

4. Desenvolvimento

As metodologias ágeis têm como objetivo o desenvolvimento de projetos de maneira rápida, mantendo as aplicações abertas à inserção de novas funcionalidades, quando for necessário [Goyal 2008]. Para este trabalho, optou-se por utilizar algumas das boas práticas da metodologia ágil, o FDD.

O *Feature Driven Development* (FDD) é um processo de desenvolvimento que contém características de processos ágeis, como o foco na programação, interações constantes com o cliente e a entrega frequente de versões do produto. Além disso, também possui alguns benefícios de processos rigorosos, como modelagem, planejamento prévio e controle do projeto. Por prover práticas somente para o desenvolvimento do software, não traz limitações a outros fatores do projeto, tais como seleção de tecnologias ou ferramentas a serem utilizadas, dentre outros [Silva, Hoentsch e Silva 2009].

O processo abrange e dá suporte a cinco fases, cada uma possuindo sua importância para o desenvolvimento da aplicação, sendo elas: desenvolver um modelo abrangente, construir uma lista de funcionalidades, planejar por funcionalidades, arquitetar por funcionalidades e construir por funcionalidades [Silva, Hoentsch e Silva 2009].

4.1. Desenvolver modelo abrangente

Este é o processo inicial do projeto, nele é realizado o estudo do escopo e contexto do sistema, para, então, a partir destes, serem realizados estudos mais detalhados do negócio para cada área a ser trabalhada [Goyal 2008].

Na Figura 2, é apresentado o Diagrama de Domínio inicial do projeto, podendo-se verificar suas entidades e dependências. Como foco deste diagrama, encontra-se a relação entre os dentistas, clientes e agendamentos, na qual os dentistas cadastrarão seus clientes para, mais tarde, cadastrar os agendamentos que possuem esses clientes.

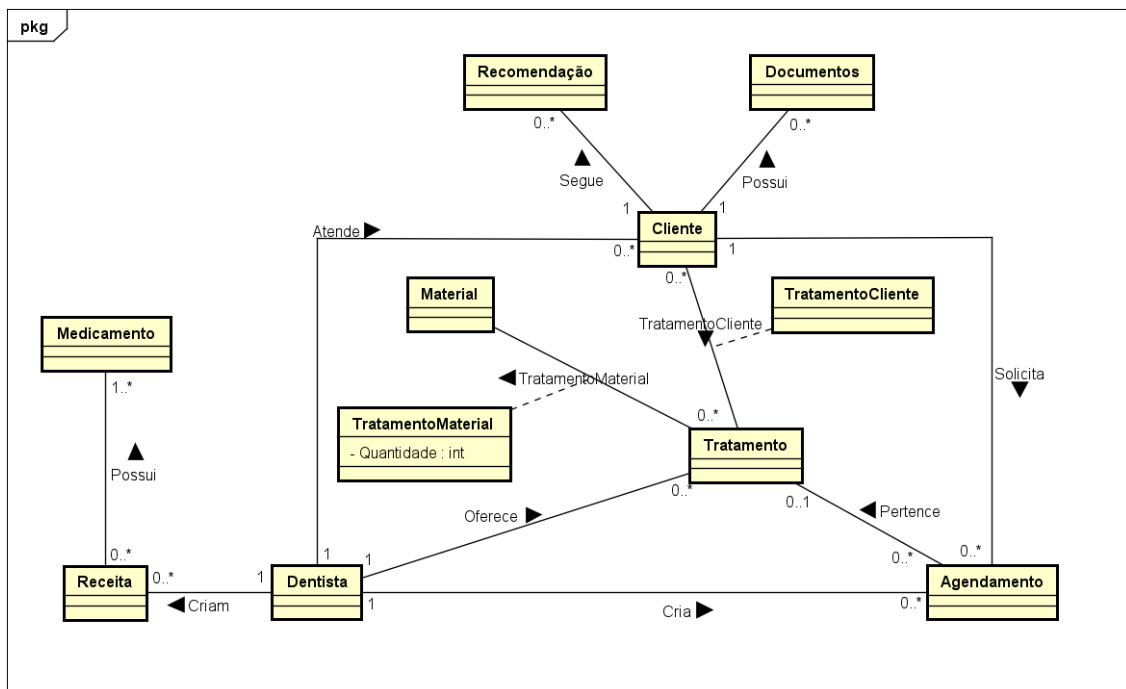


Figura 2. Diagrama de Domínio

4.2. Construção da lista de funcionalidades e planejamento

A construção da lista de funcionalidades consiste em identificar funcionalidades de forma a satisfazer os requisitos do sistema. Já no planejamento, é produzido o plano de como as funcionalidades serão desenvolvidas [Goyal 2008]. Dessa forma, foram construídas listas de funcionalidades contendo as regras de negócio, requisitos funcionais e requisitos não funcionais (Apêndice A).

A Figura 3 apresenta a representação das funcionalidades do sistema, na forma de um Diagrama de Caso de Uso. O Diagrama de Caso de Uso, embora não possua informações detalhadas sobre o funcionamento do software, serve para dar uma visão sobre o relacionamento entre atores e as funcionalidades do sistema.

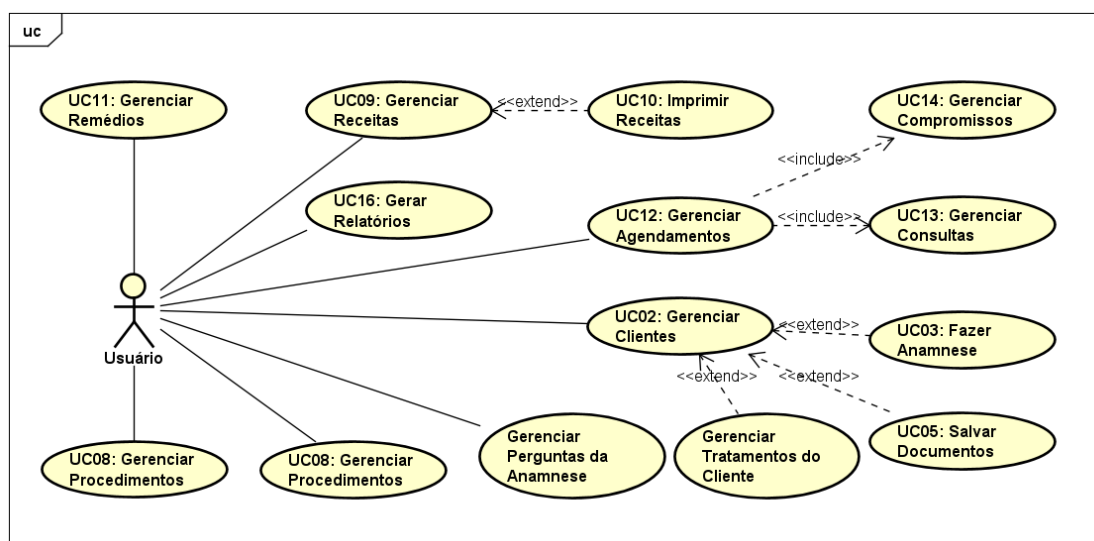


Figura 3. Diagrama de Caso de Uso

4.3. Detalhamento por funcionalidades

Nesta etapa, é feito o detalhamento de cada uma das funcionalidades do projeto [Goyal 2008]. Para esse detalhamento, foram utilizados o Diagrama de Atividades e o Diagrama de Entidade e Relacionamento.

Na Figura 4, está representado o Diagrama de Atividades referente ao caso de uso de gerenciar agendamento. O sistema permite o registro de dois tipos de agendamentos, consultas e compromissos. Na Figura 4, estão representados os processos básicos de acesso ao sistema feito por um cirurgião-dentista e a inserção de um novo agendamento, podendo o usuário escolher entre inserir uma consulta ou um compromisso.

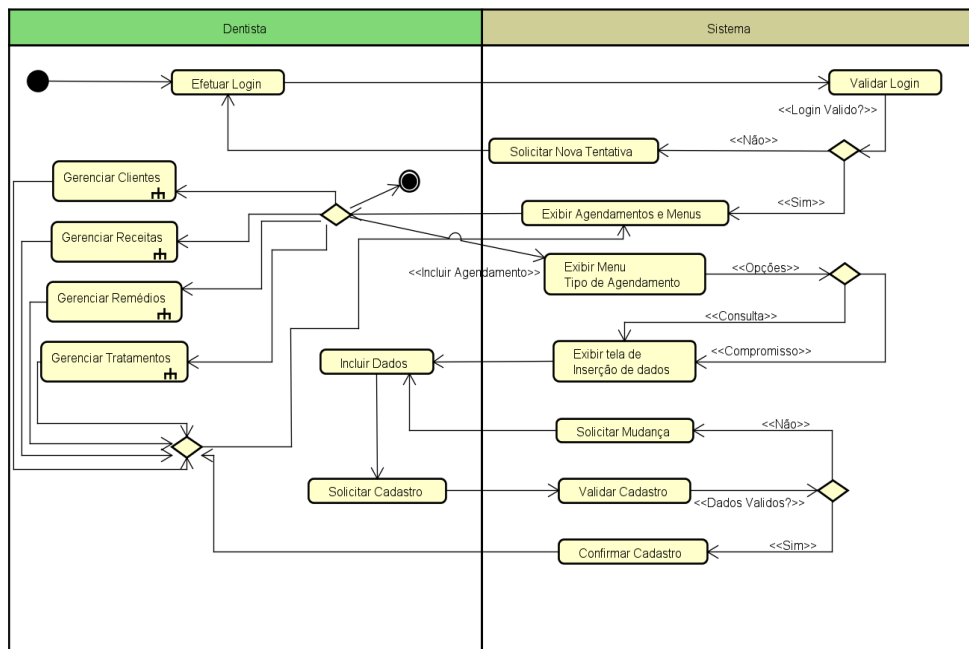


Figura 4. Diagrama de Atividade de Gerenciar Agendamentos

A Figura 5 apresenta o Diagrama de Entidade e Relacionamento (DER), por meio do qual é possível ver a estrutura do banco de dados e a relação entre as diferentes tabelas dele.

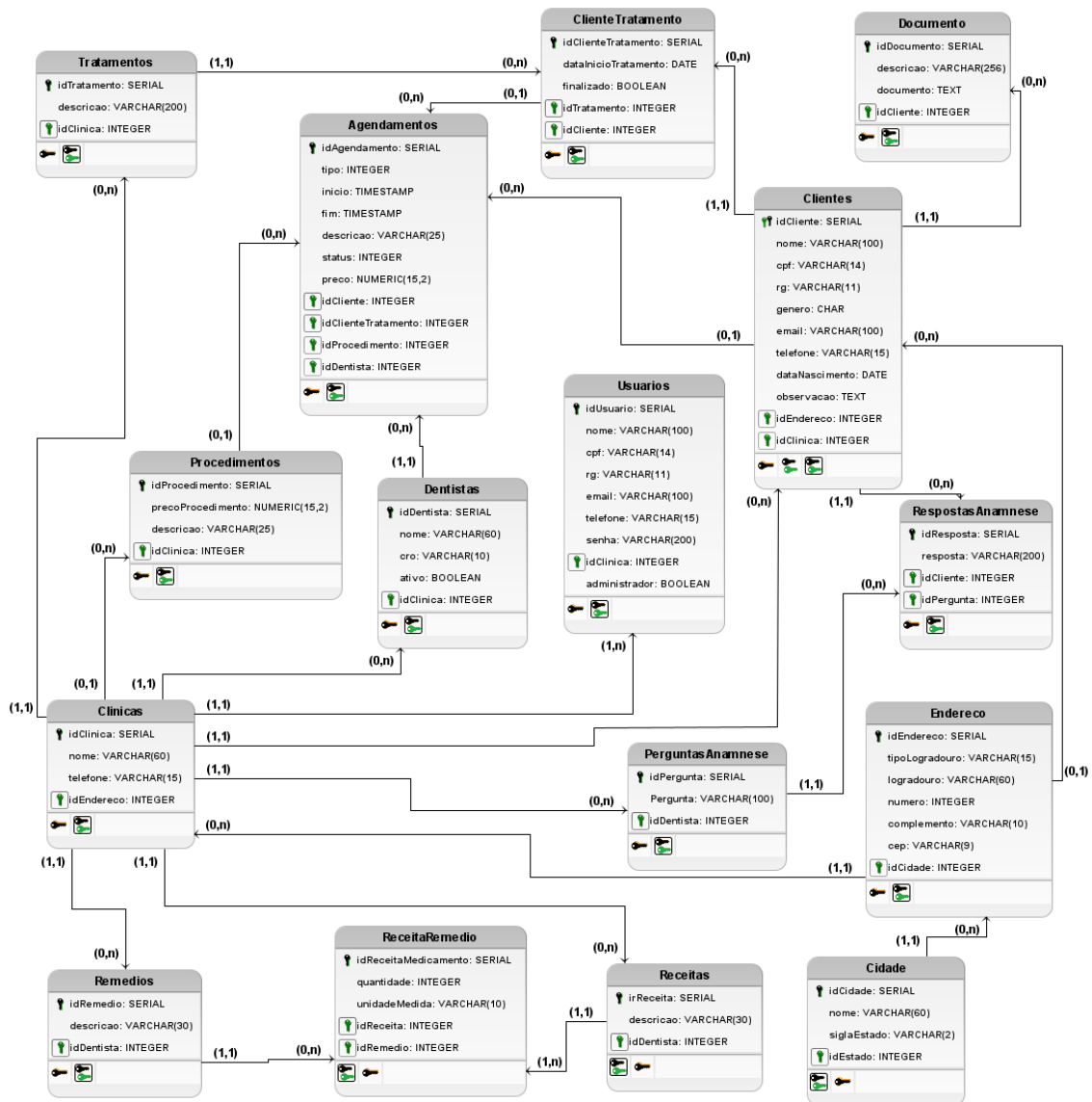


Figura 5. Diagrama de Entidade Relacionamento

4.4. Construir por funcionalidade

Nesta etapa do FDD, a aplicação é desenvolvida através da utilização das tecnologias propostas. Dentre os códigos implementados, alguns, dentre os considerados mais relevantes para o sistema, serão apresentados a seguir.

Uma das funcionalidades mais importantes do sistema é o gerenciamento dos clientes da clínica odontológica, podendo qualquer usuário visualizar, inserir, editar e excluir clientes da clínica a qual pertence. Embora possam ser inseridas diversas informações acerca desses clientes, a única obrigatória é o nome, para tentar tornar o cadastro o mais simples e rápido ao usuário. A Figura 6 apresenta um trecho da função responsável pela listagem dos clientes presentes na base de dados.

```

const idClinica = req.idClinica;
const pgSelect = {
  text: `
SELECT
  idCliente AS id, nome,
  CASE WHEN email is null THEN '-' ELSE email END AS email,
  CASE WHEN telefone is null THEN '-' ELSE telefone END AS telefone
FROM Clientes
WHERE idClinica = $1
ORDER BY nome
`,
  values:[idClinica]
}

pool.connect((err, client, done) => {
  if (err) throw err;
  client.query(pgSelect, (err, result) => {
    if (err) throw err;
    else {
      done();
      res.json(result.rows);
    }
  });
});
});

```

Figura 6. Trecho da função “listar”, implementada em “ClienteController” na *api*

A Figura 6 mostra a montagem do comando a ser executado no banco de dados, o qual irá buscar as informações dos clientes a serem listadas na tela de busca de clientes. No caso de uma busca bem-sucedida, os resultados serão enviados de volta na forma de um *json*.

Outra importante função do sistema é o gerenciamento de agendamentos, que deve mostrar os agendamentos feitos para os dentistas da clínica. Um agendamento deve conter sempre um horário para seu início e para seu fim, assim como um tipo, que determina se o agendamento é uma consulta ou um compromisso do dentista, que pode ou não ter relação com a clínica. A Figura 7 mostra a função responsável pela inserção de agendamentos na base de dados.

```

const dados = req.body;
const pgInsert = {
  text: `
    INSERT INTO Agendamentos (
      tipo, inicio, fim, descricao, status, preco, idDentista,
      idCliente, idClienteTratamento, idProcedimento
    ) VALUES (
      $1, $2, $3, $4, $5, $6, $7, $8, $9, $10
    )`,
  values: [dados.tipo, dados.inicio, dados.fim, dados.descricao, dados.status, dados.preco,
    dados.idDentista, dados.idCliente, dados.idClienteTratamento, dados.idProcedimento]
}

pool.connect((err, client, done) => {
  if (err) throw err;
  client.query(pgInsert, (err, result) => {
    if (err) throw err;
    else {
      done();
      res.json({
        statusCode: 200,
        title: "Cadastrar Agendamento",
        cadastrado: true,
        message: "Agendamento cadastrado com sucesso!",
      });
    }
  });
});
});

```

Figura 7. Trecho da função “cadastrar”, implementada em “AgendamentoController” na *api*

Na Figura 7, é possível ver como a função recebe os dados vindos da *request*, os inserindo em um comando. Após se conectar ao banco, a função executa o comando e, em caso de sucesso, retorna um *json* indicando que o cadastro do agendamento foi efetuado.

A última função a ser mencionada nesta seção é a de gerenciamento de dentistas. Um dentista deve sempre conter um nome e deverá saber se ele está ativo ou não. Além disso, opcionalmente, é possível inserir o número da inscrição no Conselho Regional de Odontologia (CRO), que será mostrado quando esse dentista tentar imprimir uma receita. A Figura 8 mostra a função responsável pela alteração dos dados dos dentistas de agendamentos na base de dados.

```

const dados = req.body;
const idDentista = req.params.id;

const pgUpdate = {
  text: `UPDATE Dentistas SET nome = $1, cro = $2, ativo = $3 WHERE idDentista = $4`,
  values: [dados.nome, dados.CRO, dados.ativo, idDentista]
}

pool.connect((err, client, done) => {
  if (err) throw err;
  client.query(pgUpdate, (err, result) => {
    if (err) throw err;
    else {
      done();
      res.json({
        statusCode: 200,
        title: "Editar Dentista",
        cadastrado: true,
        message: "Dentista atualizado com sucesso!",
      });
    }
  });
});
});

```

Figura 8. Trecho da função “atualizar”, implementada em “DentistaController” na api

A Figura 8 mostra como a função recebe os dados pelo corpo da *request* e o ID do dentista a ser alterado via parâmetro. Com essas informações, o comando é montado e a informação de seu sucesso é retornada ao usuário.

5. Resultados

O resultado deste trabalho foi a criação de um sistema web simples e eficaz. Ele utiliza uma API, a qual é responsável pelo gerenciamento das informações e criação dos objetos a serem usados pelo do sistema.

As interfaces de listagem seguem sempre um mesmo padrão, com uma tabela para apresentação dos registros buscados pela *api*, assim como botões para adição, exclusão e, na maioria dos casos, edição. Outra característica das interfaces de listagem é a presença de um filtro, onde os usuários conseguem filtrar registros baseados nas informações presentes nas colunas da lista.

Devido ao grande número de interfaces desenvolvidas para a aplicação, serão apresentadas as com maior importância para o projeto, sendo estas as de Agendamentos e Clientes.

Na Figura 9, a interface apresentada é a do quadro de agendamentos, que possui informações referentes aos tipos de agendamentos, horários e nomes de clientes ou descrição dos agendamentos de um dentista selecionado. Ao selecionar um espaço vazio, o usuário será levado à criação de um novo agendamento, onde, primeiramente, necessitará selecionar um tipo de agendamento para, depois, inserir as demais informações sobre o mesmo e assim o salvar no banco de dados. Já ao clicar em um agendamento, mais informações a respeito dele serão apresentadas, assim como opções para editá-lo ou excluí-lo.

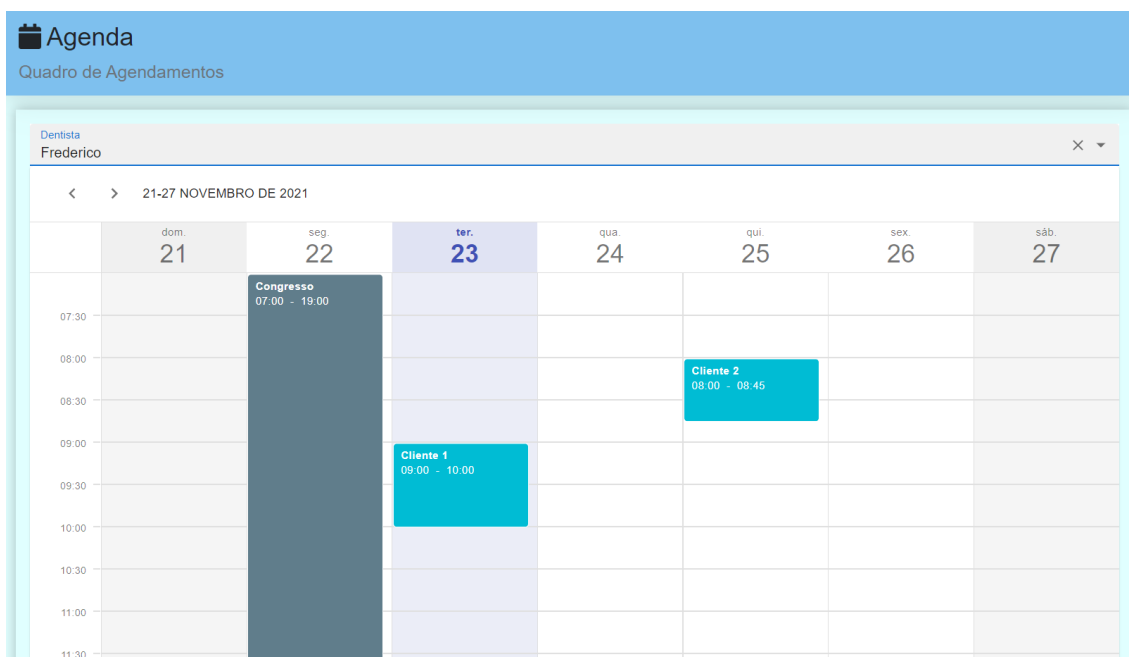


Figura 9. Interface do quadro de agendamentos

Na Figura 10, a interface apresentada é a lista de clientes cadastrados, a qual possui informações a respeito do cliente, sendo estas informações seu nome, e-mail e telefone. No cabeçalho da listagem, existe um botão com o símbolo de soma, que permite ao usuário adicionar novos clientes. Dentro da listagem, existem mais três botões, dois que servem para a edição e exclusão e mais um que serve para levar ao usuário a uma tela onde ele pode ver maiores detalhes a respeito do cliente, assim como gerenciar documentos, respostas de anamnese e tratamentos do usuário.

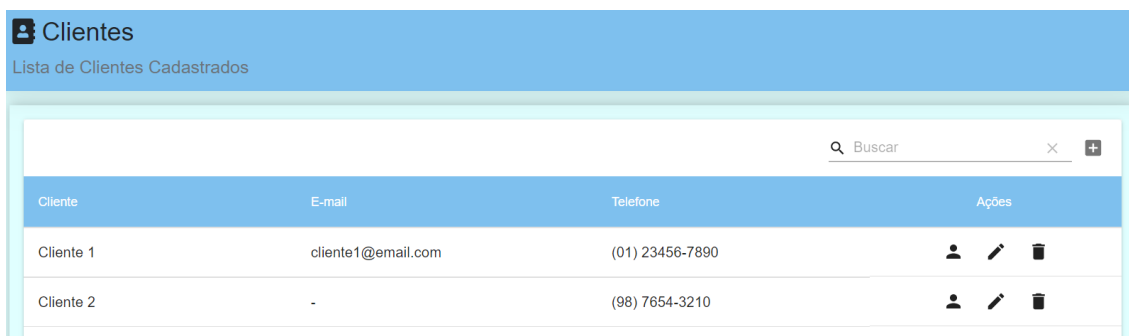


Figura 10. Interface da lista de Clientes

Na Figura 11, mostra-se a interface de edição de dentistas. Nessa interface, o usuário consegue alterar o nome, CRO e se o dentista está ativo ou não no sistema. Dentistas que forem desativados não aparecerão como opção na maior parte dos formulários que necessitem da seleção de um dentista (como no quadro de agendamentos), entretanto, dados referentes a seus agendamentos continuarão intactos.

Figura 11. Interface da edição de Dentista

Na Figura 12, mostra-se a interface de criação de relatórios de Agendamentos. Nela o usuário consegue filtrar agendamentos selecionando clientes, data e estado do pagamento. Ao clicar no botão visualizar, os agendamentos que passarem na filtragem serão mostrados na tela, juntamente com algumas de suas informações. O usuário pode então salvar o resultado da busca clicando no botão no canto superior direito.

Cliente	Data	Horário	Pago	Preço
Cliente 1	26/11/2021	10:00 - 11:00	Não	10.00
Cliente 1	07/12/2021	08:00 - 09:00	Não	0.00
Cliente 1	08/12/2021	08:00 - 09:00	Não	200.00

Figura 12. Interface da criação de relatório de Agendamentos

6. Conclusão

Este trabalho teve como objetivo a realização de um projeto de um sistema web para a organização e gerenciamento de clínicas odontológicas, a fim de diminuir as dificuldades de se armazenar as informações de forma segura.

Com base nas pesquisas relacionadas, foi possível verificar que os softwares de gerenciamento para clínicas odontológicas já existentes, embora possuam funcionalidades em comum, podem apresentar também várias funcionalidades que os diferenciam. Devido a isso, torna-se difícil encontrar um software que possua tudo que o cirurgião-dentista deseja, mas que também não sobrecarregue o mesmo com funções que ele não usufruirá.

Com a implementação desta aplicação, espera-se contribuir para a organização de consultórios odontológicos, dando aos dentistas fácil acesso aos dados sobre seus horários e clientes. Além disso, a aplicação também deve diminuir o trabalho de secretários ao tentar manter o cliente atualizado com o horário de suas consultas, podendo, inclusive, ajudar os dentistas a não perder horários devido ao esquecimento dos clientes.

Para o desenvolvimento deste projeto, optou-se pelo uso da metodologia ágil FDD, tendo em vista que ela comporta mudanças em qualquer etapa do projeto. Foram utilizados os principais diagramas dessa metodologia para exemplificar a construção deste trabalho, representando os processos, arquitetura e requisitos do aplicativo. Com o prosseguimento do cronograma proposto, a próxima fase será a implementação da aplicação, tendo por base o presente projeto.

Dentre as dificuldades enfrentadas durante o desenvolvimento dos trabalhos está a burocracia por trás do envio automatizado de mensagens, sendo necessária uma conta WhatsApp Business e uma forma de se obter a aceitação por parte das pessoas detentoras dos números que receberiam as mensagens.

Para os trabalhos futuros, ficam algumas sugestões, as quais são funcionalidades que não tiveram tempo para serem desenvolvidas. A primeira sugestão é a criação de odontogramas (diagramas gráficos que representam a boca do paciente, contendo informações sobre o estado de seus dentes), que não foram feitos devido à falta de componentes já existentes que tivessem as funcionalidades desejadas. A segunda sugestão é a criação de canais de comunicação com os clientes, na forma de mensagens de WhatsApp e de um portal para clientes, onde eles possam adquirir informações sobre seus agendamentos e recomendações feitas por seus dentistas.

Como última sugestão de trabalho futuro fica a implementação de uma forma de distribuição e comercialização do software, sendo sugerido o modelo SaaS (*Software as a Service*, ou software como serviço, em português), além da própria distribuição dele, via servidor próprio ou alugado.

Referências

- Banks, A. & Porcello, E. (2017) “Learning React: Functional Web Development with React and Redux”, O’Reilly Media, Estados Unidos.
- Brigagão, P. G. (2015). Tecnologia e Odontologia. In *Revista da ACBO*, v. 47, n. 2, p. 1-2, <http://www.rvacbo.com.br/ojs/index.php/ojs/article/view/262>, Junho.
- Chao, C. (2016). “Implementing a Paperless System for Small and Medium-Sized Businesses (SMBs)”, <https://scholarsbank.uoregon.edu/xmlui/handle/1794/19630>, Dezembro.
- Chaves, M. M. (1986). “Odontologia Social”, Artes Medicas, São Paulo.
- Codental. (2021) “Codental”, <https://www.codental.com.br>, Junho.
- Dutra, R. A. F. & Souza E. F. (2020) “Sisodonto: Sistema para gerenciamento e agendamento odontológico”, <http://pensaracademico.facig.edu.br/index.php/repositoriootcc/article/view/651>, Junho.
- Estevam, K. F. O. *et al.* (2020). O avanço da tecnologia da informação nos últimos 20 anos em pequenas empresas. In *Revista Acadêmica de Tecnologias em Educação*, n. esp., p. 1-8, <https://periodicosunimes.unimesvirtual.com.br/index.php/tecnologias-em-edu/article/view/1152>, Junho.
- Flanagan, D. (2013) “JavaScript: o guia definitivo”, Bookman, Porto Alegre.
- Goyal S. (2008). “Feature Driven Development”, <http://csis.pace.edu/~marchese/CS616/Agile/FDD/fdd.pdf>, Julho.

- Koirala D. (2020) “Web application for professional union using React and Node.js”, <https://www.theseus.fi/handle/10024/347879>, Junho.
- Pereira, C. R. (2014) “Aplicações web real-time com Node.js”, Casa do Código, São Paulo.
- PostgreSQL. (2021) “About”, <https://www.postgresql.org/about/>, Junho.
- Rodrigues, M. E. J. (2015). A importância da odontologia para a saúde da população. In Revista Sustinere, v. 3, n. 2, p. 191-192, <https://www.e-publicacoes.uerj.br/index.php/sustinere/article/view/20145/14498>, Julho.
- Silva, F. G., Hoentsch S. C. P. & Silva, L. (2009). Uma análise das metodologias ágeis FDD e scrum sob a perspectiva do modelo de qualidade MPS.BR. In *Scientia Plena*, v. 5, n. 2, p. 1-13, <https://www.scientiaplena.org.br/sp/article/view/678>, Junho.
- Souza, I. B. F. *et al.* (2020). Sistema Odontológico Connecteeth. In *e-Revista Facitec*, v. 11, n. 1, p. 1-19, <http://revistaadmmade.estacio.br/index.php/e-revistafacitec/article/view/8369>, Junho.

Apêndice A. Regras de Negócio, Requisitos Funcionais e Não Funcionais

As funcionalidades do sistema podem ser visualizadas a partir das regras de negócio, requisitos funcionais e não funcionais.

Na Tabela 1, são apresentadas as regras de negócio do projeto.

Tabela 1. Regras de Negócio

Id	Descrição
RN01	Clínicas não terão acesso a dados de outras Clínicas.
RN02	Não poderão ser excluídos clientes com agendamentos ou tratamentos.
RN03	Não poderão ser excluídos tratamentos com clientes em tratamento.
RN04	Não poderão ser excluídos medicamentos que fazem parte de receitas.
RN05	Somente o nome do cliente é necessário para seu cadastro.

Na Tabela 2, são apresentados os requisitos funcionais do projeto, juntamente com o tempo de desenvolvimento estimado para cada uma das funcionalidades do sistema na última coluna.

Tabela 2. Requisitos Funcionais

Id	Descrição	Complex.	Est. (h)
RF01	Permitir cadastro de clientes, contendo Nome, CPF, RG, Data de Nascimento, Gênero, E-mail, Telefone e Observações. Somente nome é obrigatório.	Média	15
RF02	Permitir o cadastro de agendamentos, podendo este ser do tipo consulta ou compromisso.	Alta	5
RF03	Ao cadastrar uma consulta, a ela deve conter o cliente, o horário de início e fim e o estado e o preço. Opcionalmente, o agendamento pode fazer parte de um tratamento sendo realizado pelo cliente ou procedimento, além de possuir uma observação.	Alta	10
RF04	Ao cadastrar um compromisso, devem ser informados a descrição e horário de início e fim.	Baixa	10
RF05	Permitir o cadastro de perguntas para a anamnese dos pacientes.	Baixa	3
RF06	Permitir o registro de odontograma, anamnese e documentos para o paciente.	Baixa	10
RF07	Permitir o cadastro de tratamentos, informando uma descrição.	Baixa	10
RF08	Permitir o cadastro de procedimentos, informando uma descrição e valor sugerido.	Baixa	10
RF09	Permitir o cadastro de remédios, contendo o nome.	Baixa	2
RF10	Permitir o cadastro de receita, contendo uma descrição, remédios que a compõem e suas quantidades.	Baixa	5
RF11	Permitir a impressão de receitas, podendo serem feitas alterações em seu conteúdo antes da impressão.	Baixa	5
RF12	Permitir a geração de relatórios, que contém informações sobre pacientes e finanças.	Baixa	10

RF13	O sistema deve enviar mensagem para os clientes para lembrá-los de seus agendamentos.	Alta	10
-------------	---	------	----

Na Tabela 3, são apresentados os requisitos não funcionais do projeto.

Tabela 3. Requisitos Não Funcionais

Id	Descrição	Complexidade
RNF01	Sistema será desenvolvido utilizando NodeJS e ReactJS.	Alta
RNF02	Os dados do sistema serão armazenados um banco de dados SQL, gerenciado pelo SGBD PostgreSQL.	Alta
RNF03	Fácil utilização.	Baixa

Apêndice B. Descritivos de Caso de Uso

Na sequência, são apresentados dois Descritivos de Caso de Uso que são considerados os mais importantes para o desenvolvimento do software.

Na Tabela 4, é representado o Descritivos de Caso de Uso da funcionalidade de Gerenciar Agendamento, com o agendamento de consultas, compromissos ou alterações neles.

Tabela 4. Descritivo de Caso da Funcionalidade - Gerenciar Agendamento

Identificação	UC12
Caso de Uso	Gerenciar Agendamento
Descrição	Caso de uso responsável pela Inclusão, Alteração, Exclusão e Consulta de Agendamentos no Sistema.
Ator Principal	Dentista
Pré Condições	
PC1	Dentista logado no sistema [UC001].
Pós Condições	
PO1	Agendamento incluído, alterado, excluído ou consultado com sucesso.
Fluxo Principal	
FP1	O sistema exibe os agendamentos da semana do dentista.
FP2	O Dentista pode clicar em um agendamento [FA1] ou em horário vazio, para agendar uma consulta [FA2] ou compromisso [FA3].
FP3	Fim do caso de uso.
Fluxo Alternativo	
FA1	Clicar em um Agendamento. FA1.1 – O Dentista visualiza as informações detalhadas sobre o agendamento. FA1.2 – O Funcionário altera os dados necessários. FA1.5 - O sistema grava os dados. FA1.6 – O sistema retorna ao [FP01].
FA2	Agendar Consulta. FA2.1 – O Dentista informa o horário de finalização e que o agendamento se trata de uma consulta, além de confirmar a data e horário de inicial do agendamento. FA2.2 – O Dentista informa o paciente, status da consulta, valor da consulta e, opcionalmente, um tratamento em andamento o paciente. FA2.4 - O sistema grava os dados FA2.5 – O sistema retorna ao [FP01].
FA3	Agendar Compromisso. FA3.1 – O Dentista informa o horário de finalização e que o agendamento se trata de um agendamento, além de confirmar a data e horário de inicial do agendamento. FA3.2 – O Dentista informa uma descrição. FA2.4 - O sistema grava os dados FA2.5 – O sistema retorna ao [FP01].

Fluxo de Exceções	
FE1	O agendamento não será permitido caso ocorra conflito de horários.
Regras de Negócio	RN05
Casos de Uso Incluídos	UC13, UC14, UC15

Na Tabela 5, é representado o Descritivos de Caso de Uso da funcionalidade de Gerenciar Cliente, com a inclusão, alteração, exclusão e consulta deles.

Tabela 5. Descritivo de Caso da Funcionalidade - Gerenciar Cliente

Identificação	UC002
Caso de Uso	Gerenciar Clientes
Descrição	Caso de uso responsável pela Inclusão, Alteração, Exclusão e Clientes no sistema.
Ator Principal	Dentista
Pré Condições	
PC1	Dentista logado no sistema [UC01].
Pós Condições	
PO1	Cliente incluído, alterado, excluído ou consultado com sucesso.
Fluxo Principal	
FP1	O sistema exibe as opções de Incluir, Alterar, Excluir ou Consultar Cliente ao Dentista.
FP2	O Funcionário escolhe uma das opções: Incluir [FA01], Alterar [FA02], Excluir [FA03] ou Consultar [FA04].
FP3	Fim do caso de uso.
Fluxo Alternativo	
FA1	Incluir Cliente FA1.1 – O Dentista informa o nome e CPF do cliente. FA1.2 – O dentista opcionalmente informa o RG, Data de Nascimento, Gênero, E-mail, Telefone, Endereço e Observações FA1.5 - O sistema grava os dados. FA1.6 – O sistema retorna ao [FP01].
FA2	Alterar Livro FA2.1 – O Dentista localiza o Cliente no sistema [FA4]. FA2.2 – O Funcionário altera os dados necessários. FA2.4 - O sistema grava os dados. FA2.5 – O sistema retorna ao [FP01].
FA3	Excluir Livro FA3.1 - O Dentista localiza o Cliente no sistema [FA4]. FA3.2 – O sistema verifica se o Cliente pode ser excluído [FE1] FA3.3 – O sistema registra a exclusão. FA3.4 - O sistema retorna ao [FP01].
FA4	Consultar Livro FA4.1 – O Dentista informa os dados a serem localizados. FA4.2 – O sistema exibe listagem com possíveis resultados. FA4.3 – O Dentista seleciona um resultado.

	FA4.4 – O sistema exibe informações de acordo com o resultado selecionado. FA4.5 – O sistema exibe opções: Incluir [FA1], Alterar [FA2], Excluir [FA3] ou retornar [FP01]
	Fluxo de Exceções
FE1	Se o Cliente possuir agendamentos não pagos o mesmo não poderá ser excluído.
Regras de Negócio	RN02, RN01
Casos de Uso Estendidos	UC03, UC04, UC05, UC06

Apêndice C. Diagrama de Classe

A Figura 13 apresenta o Diagrama de Classe que foi construído durante a fase de projeto, porém as classes representadas nele não foram implementadas durante a fase de desenvolvimento.

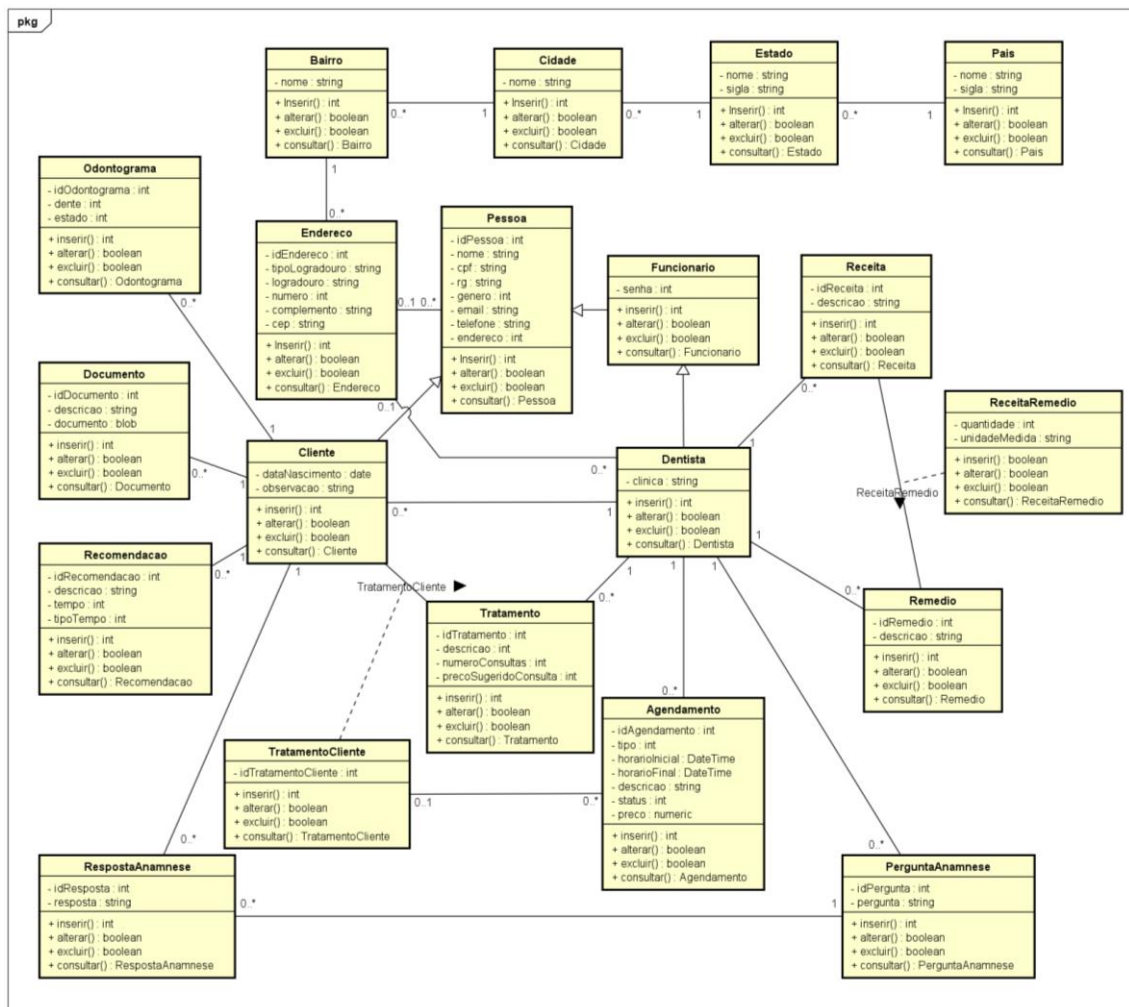


Figura 13. Diagrama de Classe