Disponível online em http://periodicos.estacio.br/index.php/e-revistafacitec/issue/current

Página inicial: http://periodicos.estacio.br/index.php/e-revistafacitec

Artigo Original

SISTEMA ODONTOLÓGICO CONNECTEETH

Igor Brayan Ferreira de Souza[a], Larissa Cunha dos Santos[b], Luíza Meireles Duarte[c], Maicon Luis Sampaio de Morais[d], Josyane Lannes Florenzano de Souza[e], Lucas Campos de Magalhães Nunes[f], Lucas Maurício Castro e Martin[g]

[a] *Discente do Centro Universitário Estácio de Brasília – DF*
[b]*Discente do Centro Universitário Estácio de Brasília – DF*
[c]*Discente do Centro Universitário Estácio de Brasília – DF*
[d]*Discente do Centro Universitário Estácio de Brasília – DF*
[e]*Docente do Centro Unviersitário Estácio de Brasilia – DF*
[f]*Docente do Centro Universitário Estácio de Brasília - DF*
[g]*Docente do Centro Universitário Estácio de Brasília - DF*

* Nome do *Autor Correspondente*.
Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail: author@institute.xxx

INFORMAÇÃO DO ARTIGO

Histórico do artigo:
Recebido em 00 Maio 00
Revisado em 00 Julho 00
Aceito em 00 Agosto 00

Palavras-chave: (3-5 palavras)
Sistema Odontológico, sistema Web

Keywords: (3-5 words)
Odontological System
Web System

RESUMO

O ambiente competitivo contemporâneo, assim como todo o sistema econômico, está sendo marcado por um modelo de maior liberalização da economia e por um aumento ainda mais expressivo do progresso tecnológico nos meios de comunicação e de informação. Com essa nova configuração, o mercado tem exigido das empresas constantes mudanças em seus planos organizacionais e gerenciais, além de uma postura centrada cada vez mais na ética e no profissionalismo. Desenvolver um sistema de gerenciamento odontológico, onde nele possa controlar os processos realizados dentro do consultório, com isso ter um impacto perante os concorrentes, ter um apoio a tomada de decisão com base de dados reais e um clique. Nosso sistema propõe assim uma proposta diferenciada do mercado hoje em dia, observando outros sistemas onde neles o principal foco vem sendo o mais do mesmo, onde eles priorizam um sistema com controle de agenda que sim em um consultório é extremamente importante. Mas percebemos que as coisas vêm mudando, de acordo com a reforma trabalhista sobre os novos métodos de trabalhos onde as empresas buscam mais empregados por meio de contratos, visamos a importância de focar em um sistema para o dentista. Com o carro chefe do nosso trabalho percebemos a importância do dentista que trabalha durante a semana em vários consultórios, um sistema integrado onde ele possa ter controle da sua agenda, poder cadastrar mais de um consultório, diferenciando clientes de cada consultório, ter uma forma mais otimizada de poder gerar um orçamento ao cliente sendo mais interativo, e também acabar com aqueles velhos prontuário escrito gerando muitas folhas arquivadas ou uma planilha no Excel.

ABSTRCT

The contemporary competitive environment, as well as the entire economic system, is being marked by a model of greater liberalization of the economy and by an even more expressive increase in technological progress in the media and information. With this new configuration, the market has required companies to constantly change their organizational and management plans, in addition to a posture increasingly focused on ethics and professionalism. Develop a dental management system, where it can control the processes carried out within the office, thereby having an impact on competitors, supporting decision making with real data and a click. Our system thus proposes a differentiated proposal from the market today, observing other systems where in them the main focus has been the most of the same, where they prioritize a system with agenda control that yes in an office is extremely important. But we realized that things have been changing, according to the labor reform on new work methods where companies seek more employees through contracts, we aim at the importance of focusing on a system for the dentist. With the flagship of our work we realized the importance of the dentist who works during the week in several offices, an integrated system where he can have control of his schedule, be able to register more than one office, differentiating customers from each office, having a more optimized to be able to generate a budget for the client being more interactive, and also to end those old written medical records generating many archived sheets or an Excel spreadsheet.

Introdução

O ambiente competitivo contemporâneo, assim como todo o sistema econômico, está sendo marcado por um modelo de maior liberalização da economia e por um aumento ainda mais expressivo do progresso tecnológico nos meios de comunicação e de informação. Com essa nova configuração, o mercado tem exigido das empresas constantes mudanças em seus planos organizacionais e gerenciais, além de uma postura centrada cada vez mais na ética e no profissionalismo. (Pires,2006).

Contudo, a falta de conhecimento sobre administração por parte dos cirurgiões-dentistas acarreta numa gestão realizada por decisões intuitivas e aleatórias, tornando a estratégia um ato emergente e não deliberado (WHITTINGTON apud ZIMBRES, 2006; MODAFFORE e FIGUEIREDO FILHO, 2010; KÜHNEN e RIBEIRO, 2011).

De acordo com Chiavenato e Sapiro (2009), em um mundo caracterizado por incertezas e concorrência feroz o planejamento estratégico tem sido imprescindível para o sucesso de uma organização. Com isso e importante o primeiro passo de um consultório odontológico e reconhecer como organização que transformará seus recursos em serviços. Ainda segundo os autores, o planejamento estratégico é uma ferramenta para inserir a organização e a sua missão no seu ambiente de atuação.

De acordo com o conselho federal de odontologia no Distrito Federal existem 1980 a EPAO (Entidades Prestadoras de Assistência

Odontológica), cadastradas onde ela se encontra de pequeno a grande porte, e grande parte delas principalmente de pequeno porte não utilizam nenhum sistema informatizado para organização dos processos da clínica, necessitando assim uma melhor maneira de agilizar seus processos e ter a informação ali quando precisa com um simples toque, e assim podendo gerar vantagens competitiva.

Justificativa

O desenvolvimento da ciência da informação e a introdução dos computadores no âmbito profissional trouxeram profundas transformações na odontologia beneficiando a sociedade, promovendo soluções para simplificar e otimizar a vida, no intuito de agilizar, tornar mais dinâmico o trabalho e automatizar processos repetitivos (SILVESTRE, 1998).

> Para Silvestre (1998), afirma que diversas especializações e inovações tecnológicas dos consultórios odontológicos hoje é uma realidade impulsionada pela competitividade, pois há pouco investimento na construção de sistemas que tenham um controle diversificado onde o dentista pode trabalhar em uma ou mais clinicas e com isso em um sistema poder gerir seus dados, sua agenda e tendo uma facilidade em apresentar um orçamento limpo para seu cliente. A obtenção da informação por meio eletrônico reflete positivamente no resultado do atendimento com qualidade, pois a perda de informação com prontuários obsoletos e pouco precisos podem causar danos irreversíveis para o profissional e principalmente ao paciente.

Por meio deste dados podemos afirmar que a informatização é importante para obter vantagens competitivas e com isso agilizar os processos, visando otimizar junto ao um sistema os processos da clínica odontológica, tendo em vista que toda empresa desde o pequeno a grande porte, que tenha computador e serviços precisam dessa união para melhor obter seus resultados.

Objetivo Geral

Desenvolver um sistema de gerenciamento odontológico, onde nele possa controlar os processos realizados dentro do consultório, com isso ter um impacto perante os concorrentes, ter um apoio a tomada de decisão com base de dados reais e um clique. Nosso sistema propõe assim uma proposta diferenciada do mercado hoje em dia, observando outros sistemas onde neles o principal foco vem sendo o mais do mesmo, onde eles priorizam um sistema com controle de agenda que sim em um consultório é extremamente importante. Mas percebemos que as coisas vêm mudando, de acordo com a reforma trabalhista sobre os novos métodos de trabalhos onde as empresas buscam mais empregados por meio de contratos, visamos a importância de focar em um sistema para o dentista. Com o carro chefe do nosso trabalho percebemos a importância do dentista que trabalha durante a semana em vários consultórios, um sistema integrado onde ele possa ter controle da sua agenda, poder cadastrar mais de um consultório, diferenciando clientes de cada consultório, ter uma forma mais otimizada de poder gerar um orçamento ao cliente sendo mais interativo, e também acabar com aqueles velhos prontuário escrito gerando muitas folhas arquivadas ou uma planilha no Excel.

Objetivos Específicos

Como objetivos específicos, temos:

- Desenvolver um sistema disponível via web;
- Ter uma interface intuitiva e responsiva;
- Orçar procedimentos;
- Gerenciar a agenda de atendimentos;

Fundamentação Teórica

Casos de Uso

Os casos de uso têm, conforme Sommerville (2007, p. 103), o objetivo de identificar as interações que podem ocorrer no sistema. Essas interações podem ser descritas como cenários. Por meio dessas informações, o analista é capaz de elaborar os requisitos, pois é mais fácil o usuário descrever como são desempenhadas suas atividades no dia-a-dia do que abstrair uma informação para mencioná-la.

De acordo com Sommerville (2007, p.102), um cenário deve conter as seguintes

informações:

- Uma descrição do que os usuários esperam do sistema no início do cenário
- Uma descrição do fluxo normal de eventos no cenário
- Uma descrição do que pode dar errado e como isso é tratado
- Informações sobre outras atividades que podem ocorrer simultaneamente
- Uma descrição do estado de sistema no fim do cenário

Primeiramente, o usuário deve definir qual será o objetivo do cenário a ser descrito. Em seguida, o fluxo principal do cenário, por exemplo, um simples cadastro de um produto. Caso ocorra alguma inconsistência no fluxo principal, o sistema deve tratar o erro e então informar o erro ao usuário, por exemplo, uma mensagem na tela. Por fim, é descrito o resultado que o sistema deve obter após a execução das tarefas. Assim, essas informações podem ser descritas em forma de texto ou utilizando diagramas de caso de uso na notação UML (Unified Modeling Language).

Posteriormente, segundo Pressman (2006, p. 130), o analista deve identificar quem são os atores que irão interagir com o sistema. Ator é aquele que realiza uma determinada funcionalidade do sistema, podendo ser uma pessoa, um serviço ou um outro sistema. Em seguida, devem ser levantadas as interações que ocorrem com cada ator. Sendo assim, um diagrama de caso uso é criado, conforme representado na FIG. 1. Um documento final é criado contendo o conjunto de casos de uso, na qual representa todas as possíveis interações que ocorrem no sistema.

De acordo com Sommerville (2007, p. 101) a técnica de caso de uso é mais bem aplicada quando o usuário descreve como ele interage com o sistema. Entretanto, essa técnica não é recomendada em situações que o analista deseja obter requisitos de domínio, pois na descrição do cenário, o usuário pode não mencionar qual é o domínio que determinada funcionalidade necessita para o seu funcionamento.

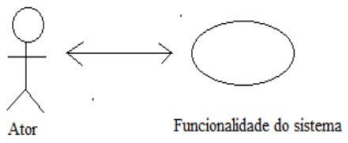

Figura 1: Modelo de Diagrama de Caso de Uso (SOMMERVILLE, 2007)

Engenharia de Requisitos de Software

As descrições das funções que um sistema deve incorporar e das restrições que devem ser satisfeitas são os requisitos para o sistema. Isto é, os requisitos de um sistema definem o que o sistema deve fazer e as circunstâncias sob as quais deve operar. Em outras palavras, os requisitos definem os serviços que o sistema deve fornecer e dispõem sobre as restrições à operação do mesmo (SOMMERVILLE, 2007).

Requisitos são, normalmente, classificados em requisitos funcionais e requisitos não funcionais. Requisitos funcionais, como o próprio nome indica, apontam as funções que o sistema deve fornecer e como o sistema deve se comportar em determinadas situações. Já os requisitos não funcionais descrevem restrições sobre as funções oferecidas, tais como restrições de tempo, de uso de recursos etc. Alguns requisitos não funcionais dizem respeito ao sistema como um todo e não a funcionalidade específica. Dependendo da natureza, os requisitos não funcionais podem ser classificados de diferentes maneiras, tais como requisitos de desempenho, requisitos de portabilidade, requisitos legais, requisitos de conformidade etc. (SOMMERVILLE, 2007).

O levantamento de requisitos é um processo iterativo, com uma continua validação de uma atividade para outra (ver fig 2).

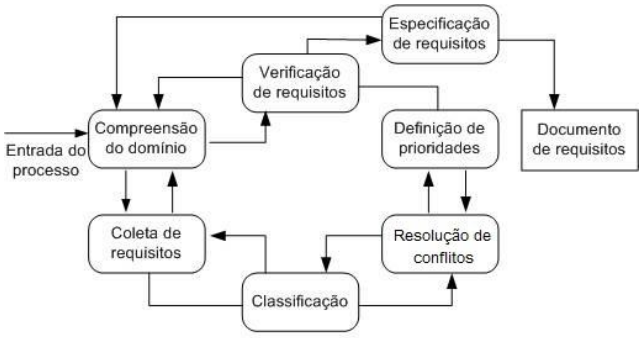

Figura 2: Processo de Levantamento e Análise de Requisitos (SOMMERVILLE, 2003)

Existem diversas técnicas para realizar o levantamento dos requisitos, tais como entrevista questionário, observação, casos de uso, *brainstorm* entre outros.

2.3 Entrevista

A técnica de entrevista é, segundo Brum e Pena (2011), simples e bastante eficiente na fase de levantamento dos requisitos, na qual o analista será o entrevistador que obtém os dados informados pelo cliente. De acordo com Sommerville (2007, p. 101), essa entrevista pode ser formal ou informal.

A **entrevista formal** é composta de perguntas pré-definidas que devem ser respondidas em sequência, sempre evitando desviar o foco. Já a **entrevista informal** contém algumas perguntas pré-definidas, sendo elas de extrema importância para dar início as entrevistas. Contudo, para que tais informações sejam mais bem compreendidas, é possível o usuário dar maiores esclarecimentos sobre o assunto abordado. Geralmente ambos são utilizados, pois determinados requisitos requerem explicações por parte do usuário a fim de que o analista obtenha a compreensão correta.

Brum e Pena (2011) ressaltam vários benefícios na utilização da técnica de entrevista. Por exemplo, na entrevista informal, o usuário pode desviar em certos momentos dos objetivos a fim de que o analista entenda melhor o processo do negócio. Além disso, caso seja necessário, as perguntas não precisam seguir a ordem definida previamente

Segundo Sommerville (2007, p.101), o analista formula questões para os usuários sobre o sistema que eles usam e o sistema a ser desenvolvido. As entrevistas são uteis para se obter um entendimento geral sobre o que os usuários fazem, como eles irão interagir com o sistema e quais as dificuldades que enfrentam com os sistemas atuais.

2.4 Framework - Laravel

Laravel é um framework PHP utilizado para o desenvolvimento web, utilizando da arquitetura MVC (Model-View-Controlle) e tem como principal característica ajudar a desenvolver a aplicações seguras e performáticas de forma rápida, com código limpo e simples, já que ele incentiva o uso de boas práticas de programação.

Para a criação de interface gráfica, o Laravel utiliza uma Engine de template (software que pega um texto e faz a substituição de partes deste texto por alguma informação a ser definida, seja por um código, seja por um padrão que será determinado por alguma regra fixa) chamada Blade, que traz uma gama de ferramentas que ajudam a criar interfaces bonitas e funcionais de forma rápida e evitar a duplicação de código.

Para se comunicar com um Banco de Dados o Laravel utiliza uma implementação simples do ActiveRecord chamada de Eloquent ORM, que é uma ferramenta que traz várias funcionalidades para facilitar a inserção, atualização, busca e exclusão de registros. Com configuração simples e pequena e com pouco código podemos configurar a conexão com banco de dados e trabalhar com ele. Atualmente o framework está em sua versão 7.0.

2.5 Gerenciador de Repositório de Código – Git Lab

De acordo o site https://software.com.br/, o Git Lab é um sistema de controle de versão distribuído de código aberto e gratuito, projetado a lidar com tudo, de projetos pequenos a grandes.

Sua principal funcionalidade é fazer o controle de versões de modo colaborativo, ou seja, é possível que o mesmo arquivo seja modificado ao mesmo tempo por dois ou mais desenvolvedores diferentes, e que ambas as alterações sejam salvas sem que nenhum código seja sobrescrito.

Para manter o modo colaborativo, o Git utiliza o conceito de ramificação ou branch, onde cada branch é uma linha do tempo que possui marcos ou commits, e nesse branch os arquivos podem ser alterados livremente sem impactar outras ramificações.

O Git Lab é uma plataforma de hospedagem de código-fonte. Permitem que os desenvolvedores contribuam em projetos privados ou abertos. Nesta plataforma, cada projeto em que é desenvolvido contendo um

código-fonte e um repositório, onde por exemplo o frontend é desenvolvido separadamente do código backend, sendo uns hospedados em repositórios separados.

2.6 Linguagem de Programação – JavaScript

Segundo Gonçalves (2006), o JavaScript foi criado por Brendan Eich da Netscape na versão 2.0 no ano de 1995, para controlar o navegador, interagindo com as páginas Web. É uma linguagem de programação de scripts, com o propósito de trazer interatividade às páginas web.

Assim o JavaScript, CSS (Cascade Style Sheet), HTML, DHTML em inglês Dynamic HTML, aliada ao DOM (Document Object Model) permite que as aplicações Web sejam modificadas na máquina cliente, sem acesso ao servidor web. Esta linguagem também é utilizada nas aplicações de desktops. Aplicações como os mais usados, Mozilla Firefox, Internet Explorer, Thunderbird, são alguns exemplos. (GONÇALVES, 2006).

2.7 Linguagem de Estrutura de Projetos

Em Pender (2004), a UML (Unified Modeling Language) foi criada por desenvolvedores com o intuito de solucionar problemas antes da implementação do código evitando o trabalho extra de reescrita e a cada mudança no sistema que levaria projetos mais lentos com custos de manutenção mais altos, permite a comunicação, organização da documentação do sistema quando a ideia é trabalhar orientado a objeto. Tornou-se um padrão para a modelagem de software orientado a objeto e tem sido adotada por empresas do mundo inteiro, com o intuito de evitar problemas futuros.

> UML (Unified Modeling Language) é a sucessora da onda de métodos de análise e projeto orientado a objetos (OOA& D) que surgiu no final dos anos oitenta e no início dos anos noventa. Mais especificamente, ela unifica os métodos de Booch, Rumbaugh OMT (Object Modeling Technique) e Jacobson, mas o seu alcance é bem maior. UML passou por um processo de padronização pela OMG (Object Managment Group) e é agora um padrão OMG..(FOWLER; SCOTT, 2000, p.19)

Quando se usa a UML, podem-se produzir projetos com elegância, segundo Wazlawick (2004, p.19) descreve: “O software elegante é aquele cuja estrutura é intrinsecamente mais fácil de compreender”.

Existiam muitos problemas na fase de análise de requisitos, análise de sistemas e design. A interação da equipe com o uso da UML auxiliou no desenvolvimento do trabalho criando assim uma forma abrangente de entender o projeto. Pode-se criar qualquer tipo de aplicação que se deseje, o projeto deve ser modelado e representado por diagramas usando alguns mecanismos que permitem a comunicação para alavancar o conhecimento e a comunicação entre a equipe. (UML, 2001)

O UML está sendo a base para muitas ferramentas de desenvolvimento, incluindo modelagem visual, simulações e ambientes de desenvolvimento, integrando muitas ideias adversas, e esta integração acelera o uso do desenvolvimento de softwares orientandos a objetos.

- Diagramas Estruturais:
    - Diagramas de Classe: Este é o fundamental e serve de apoio a outros diagramas, pois mostra suas classes, métodos, atributos e relacionamentos. Segundo Guedes em seu livro “UML – Uma Abordagem Prática”, o objetivo do diagrama de classes é mostrar os relacionamentos existentes entre as classes que são abstraídas no projeto, e como esses relacionamentos colaboram para a execução de um processo específico.
    - Diagrama de Objeto**:** Está relacionado com o diagrama de classe, mas este mostra a visão dos valores armazenados nos objetos do diagrama de classe em

determinada execução do processo.

- Diagrama de Componentes: está associado a linguagem de programação e mostra os componentes do software e relacionamentos.
- Diagrama de Hardware: Determina as características físicas do sistema.
- Diagrama de Pacotes: Mostra os subsistemas englobados no sistema de forma a destacá-los.
- Diagrama de Estrutura: Descreve a estrutura interna de um classificador. Diagramas Comportamentais Diagrama de Caso de Uso: Para levantamento e análise dos requisitos do sistema. Diagrama Máquina de Estado: Procura mostrar as transformações de um objeto dentro de um processo. Diagrama de Atividade: Descreve os passos para uma conclusão de atividade. Diagrama de Iteração: Dividido em:

  1. Diagrama De Sequência: Descreve a ordem temporal em que as mensagens são trocadas entre os objetos.
  2. Diagrama Geral interação: Variação dos diagramas de atividades que fornece visão geral dentro do sistema ou processo do negócio.
  3. Diagrama De comunicação: concentra-se em como os objetos estão vinculados.
  4. Diagrama De tempo: Descreve a mudança de estado ou condição de uma instância de uma classe ou seu papel durante o tempo.

Astah é uma ferramenta que permite criar vários diagramas que são necessários para documentação do software e que alguns serão usados nesse projeto. De acordo com Daves (2013):

> O Astah Professional, uma ferramenta CASE de criação de diagramas UML, além de outros diagramas, tais como diagrama de entidade-relacionamento, diagrama de fluxo de dados e outras funcionalidades úteis à fase de especificação e projeto de um sistema. Anteriormente a ferramenta era conhecida por Jude, tendo o nome alterado para Astah*. A ferramenta pode ser encontrada no site http://astah.changevision.com/en/product/astah-professional.html onde é oferecido o download do Astah* Professional e uma licença provisória de 20 dias no próprio site. Após esse período a ferramenta para de funcionar devendo-se comprar a licença ou usar sua versão free, que não possui o mesmo conjunto de diagramas disponíveis

2.8 Modelo de Sistema – Model-View-Controller (MCV)

Considerado como o primeiro padrão de projeto criado, por um engenheiro civil chamado Christopher Alexander em meados da década de 70. É considerado um padrão de projeto uma solução já testada e documentada que possa resolver um problema específico em projetos distintos.

Conceito principal do modelo MVC é utilizar uma solução já definida para separar partes distintas do projeto reduzindo suas dependências ao máximo.

Desenvolver uma aplicação utilizando algum padrão de projeto pode trazer alguns dos seguintes benefícios:

- Aumento de produtividade;
- Uniformidade na estrutura do software;
- Redução de complexidade no código;

- As aplicações ficam mais fácies de manter;
- Facilita a documentação;
- Estabelece um vocabulário comum de projeto entre desenvolvedores;
- É considerada uma boa prática utilizar um conjunto de padrões para resolver problemas maiores que, sozinhos, não conseguiriam;
- Ajuda a construir softwares confiáveis com arquiteturas testadas;
- Reduz o tempo de desenvolvimento de um projeto.

O MVC é utilizado em muitos projetos devido à arquitetura que possui, o que possibilita a divisão do projeto em camadas muito bem definidas. Cada uma delas, o Model, o Controller e a View, executa o que lhe é definido e nada mais do que isso.

A utilização do padrão MVC traz como benefício isolar as regras de negócios da lógica de apresentação, a interface com o usuário. Isto possibilita a existência de várias interfaces com o usuário que podem ser modificadas sem que haja a necessidade da alteração das regras de negócios, proporcionando assim muito mais flexibilidade e oportunidades de reuso das classes.

Uma das características de um padrão de projeto é poder aplicá-lo em sistemas distintos. O padrão MVC pode ser utilizado em vários tipos de projetos como, por exemplo, desktop, web e mobile.

2.9 Servidor HTTP – Apache

O Apache é um software de código aberto de propriedade da Apache Software Foundation (ASF) que garante desempenho, estabilidade e segurança para um servidor web.

Por meio do processo baseado em mérito da ASF conhecido como "The Apache Way", mais de 730 membros voluntários individuais e mais de 7.000 colaboradores de códigos de todos os continentes colaboram com sucesso em inovações em Inteligência Artificial e Deep Learning, Big Data, Gerenciamento de Compilação, Computação em Nuvem, Conteúdo Gerenciamento, DevOps, IoT e Edge Computing, Mobile, Servidores e Web Frameworks, entre outras categorias".

Foi criado em 1995 e é a tecnologia central responsável pelo crescimento inicial da internet. De código aberto onde recebe milhares de contribuições de desenvolvedores de todo mundo. Onde programadores configuram seus próprios módulos, aplicam funcionalidades especificas e aprimoram seus recursos para atuar em projetos variados na área da tecnologia.

Hoje em dia várias empresas de grande porte utilizam as funcionalidades do apache como por exemplo os gigantes da Netflix, Airbnb, Ebay, Cisco, BBC, Nike entre outras.

Hoje em dia existem diversas vantagens em utilizar o Apache dentre elas as principais são:

- Preço – Sendo de código aberto, está disponível para download ou para modificação por qualquer pessoa gratuitamente.
- Recursos – é uma poderosa ferramenta, com recursos muito grande:
    - Painel de controle administrativo;
    - Envio de mensagens de erro personalizadas;
    - Esquemas de autenticação;
    - Modulo de host virtual que permite executar vários sites simultaneamente;
    - Serviço de nome de domínio;
    - SMTP (Simple Mail Transfer Protocol);
    - TP (File Transfer Protocol);
    - Compatibilidade – É compatível com inúmeras configurações de hardwares e sistema operacional;
    - Suporte – Tendo uma documentação para suporte técnico sempre disponível;
    - Modularidade – Um dos recursos mais poderosos, tendo a capacidade de lidar com um alto volume de tráfego com uma mínima configuração adicional.

2.10 Sistema de Gerenciamento de Banco de Dados

A ferramenta MySql é um Sistema de Gerenciamento de Banco de Dados (SGBD), possui benefícios como a alta capacidade de

acúmulo da base de dados e também ser mundialmente conhecida pela sua distribuição ser gratuita. Capaz de gerenciar uma grande quantidade de dados e informações, podendo usufruir de recursos com stored procedures, triggers e functions além de garantir uma boa segurança e integração dos dados armazenados. O SGBD MySql é de licença dupla, os usuários podem usar como um produto Open Source/Free software sob os termos da General Public License (GNU) gratuito. (INFORMAÇÕES, 2008).

3. DESENVOLVIMENTO

3.1 Proposta de Trabalho

A proposta de trabalho é unificar em apenas um sistema o(s) consultório(s), a agenda, pacientes e orçamentos que o profissional da área da saúde bucal realiza os procedimentos odontológicos, tendo em vista que os dentistas atualmente trabalham de forma autônoma, podendo trabalhar em vários consultórios durante a semana, assim, precisando de uma ferramenta para auxiliar no gerenciamento de suas atividades.

3.1.1 Motivação para Novo Sistema

3.2 O sistema proposto

A proposta de solução para o problema é um sistema de gerenciamento, dessa forma, atuando como uma das ferramentas principais de trabalho do dentista, onde nele poderá cadastrar consultório(s), pacientes e seus respectivos prontuários, e orçamentos.

3.2.1 Requisitos do Sistema

Código: RF004	**Nome: Manter Orçamento**
Prioridade:	(x)Essencial ( )Importante ( )Desejável
Descrição:	O sistema deverá permitir o cadastro do orçamento, contendo (campos)
Regras de negócio:	1. Não é obrigatório ter um paciente/prontuário cadastrado para cadastrar o orçamento; 2. O orçamento deverá ter validade de até 30 dias; 3. O orçamento poderá ser vinculado a um prontuário e/ou paciente.

Tabela 4: Requisitos Funcionais - RF004 - Manter Orçamento

REQUISITOS NÃO FUNCIONAIS		
Código	**Requisitos**	**Categoria**
RNF001	Executar via web	Usabilidade
RNF003	Escalabilidade	Desempenho
RNF004	Sistema Operacional – Windows ou Linux (sistema)	Compatibilidade
RNF005	CHROME – Browser (técnico)	Compatibilidade

Código: RF005	**Nome: Manter Prontuário**
Prioridade:	(x)Essencial ( )Importante ( )Desejável
Descrição:	O sistema deverá permitir cadastro de prontuário, contendo
Regras de negócio:	1. É obrigatório possuir paciente cadastrado; 2. É obrigatório vincular o prontuário a um paciente.

Tabela 5: Requisitos Funcionais - RF005 - Manter Prontuário

Código: RF006	**Nome: Gerenciar Agenda**
Prioridade:	(x)Essencial ( )Importante ( )Desejável
Descrição:	O sistema deverá permitir o cadastro de atendimentos na agenda.
Regras de negócio:	1. Obrigatório possuir consultório cadastrado para relaizar o agendamento; 2. Obrigatório possuir paciente cadastrado para realizar o agendamento.

Tabela 6: Requisitos Funcionais – RF006 – Gerenciar Agenda

REQUISITOS NÃO FUNCIONAIS		
Código	**Requisitos**	**Categorias**
RNF001	Executar via web	Usabilidade
RNF003	Escalabilidade	Desempenho
RNF004	Sistema Operacional – Windows ou Linux (sistema)	Compatibilidade
RNF005	CHROME – Browser (técnico)	Compatibilidade

Tabela 7: Requisitos Não Funcionais

A Figura 3 mostra o diagrama de caso de uso do sistema proposto

3.1.1 Projetos de Interface

3.2 Projeto Físico

3.3.1 Modelo Físico de Dados

Dicionário de Dados

Tabela	usuario			
Descrição	Armazena as informações do usuário.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
Id	Código identificador do usuário.	INT	-	PRIMARY KEY
nome	Nome do usuário.	VARCHAR	45	Not null
email	E-mail do usuário.	VARCHAR	45	Not null
senha	Senha do usuário.	VARCHAR	45	Not null
telefone	Telefone do usuário.	VARCHAR	45	Not null
cro	CRO (Conselho Regional de Odontologia) do usuário.	INTEGER		Not null

Tabela 8: Dicionário de Dados – Usuário

Tabela	Usuario_has_Clinica			
Descrição	Armazena as informações do usuário e da clínica, realizando a relação dos dois.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
Usuario_id	Codigo identificador do usuário.	INT	-	
Clinica_id	Código identificador da clínica.	INT	-	

Tabela 9: Dicionário de Dados - Tabela associativa Usuário e Clínica

Tabela	Clinica			
Descrição	Armazena as informações da clínica.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
Id	Código identificador da clínica.	INT	-	PRIMARY KEY
nome	Nome da clínica.	VARCHAR	45	Not null
cnpj	CNPJ da clínica.	VARCHAR	45	Not null
endereço	Endereço da clínica.	VARCHAR	45	Not null
telefone	Telefone da clínica.	VARCHAR	45	Not null

Tabela 10: Dicionário de Dados – Clínica

Tabela	Clinica_has_Paciente			
Descrição	Armazena as informações da clínica e do paciente, realizando a relação dos dois.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
Clinica_id	Código identificador da clínica.	INT	-	
Paciente_id	Código identificador do paciente.	INT	-	

Tabela 11: Dicionário de Dados - Tabela associativa Clínica e Paciente

Tabela	Paciente			
Descrição	Armazena as informações do paciente.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
id	Código identificador do paciente.	INT	-	PRIMARY KEY
nome	Nome da paciente.	VARCHAR	45	Not null
cpf	CPF do paciente.	VARCHAR	45	Not null
datanascimento	Data de nascimento do paciente.	VARCHAR	45	Not null
telefone	Telefone do paciente.	VARCHAR	45	Not null
sexo	Sexo do paciente.	VARCHAR	45	Not null
cep	CEP do paciente.	VARCHAR	45	-
endereco	Endereço do paciente.	VARCHAR	45	-
email	E-mail do paciente.	VARCHAR	45	-

Tabela 12: Dicionário de Dados – Paciente

Tabela	Agendamento			
Descrição	Armazena as informações do agendamento.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
id	Código identificador do agendamento.	INT	-	PRIMARY KEY
Paciente_Id	Representa o código do paciente que pertence ao agendamento.	INT	-	FOREIGN KEY
Clinica_Id	Representa o código da Clínica que pertence ao agendamento.	INT	-	FOREIGN KEY
data	Data do agendamento.	VARCHAR	45	Not null
horário	Horário do agendamento.	VARCHAR	45	Not null
titulo	Título do agendamento	VARCHAR	45	Not null

Tabela13: Dicionário de Dados – Agendamento

Tabela 15: Dicionário de Dados - Tabela associativa Prontuário e Paciente

Tabela	Prontuario			
Descrição	Armazena as informações do prontuário.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
id	Código identificador do paciente.	INT	-	PRIMARY KEY

Tabela 141:Dicionário de Dados - Prontuário

Tabela	Prontuario_has_Paciente			
Descrição	Armazena as informações do prontuário e do paciente.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
Prontuario_id	Código identificador do prontuário.	INT	-	
Paciente_id	Código identificador do paciente.	INT	-	

Tabela 182: Dicionário de Dados – Procedimento

Tabela	anamnese			
Descrição	Armazena as informações da anamnese.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
Id	Código identificador da anamnese.	INT	-	Primary Key
resposta	Resposta em que o usuário selecionar.	Boolean	-	-
Perguntas_id	Código identificador das perguntas da anamnese.	INT	-	Foreign Key
Prontuario_id	Código identificador do prontuário	INT	-	-

Tabela 16: Dicionário de Dados - Anamnese

Tabela	Perguntas			
Descrição	Armazena as informações das perguntas da anamnese.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
id	Código identificador das perguntas.	INT	-	PRIMARY KEY
perguntas	armazena perguntas da anamnese	VARCHAR	255	

Tabela 17:Dicionário de Dados - Perguntas

Tabela	Procedimento			
Descrição	Armazena as informações do paciente.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
id	Código identificador do procedimento.	INT	-	Primary Key
nome	Nome do procedimento.	VARCHAR	45	Not null
Especialidade_int	Representa o código da Especialidade que pertence ao procedimento.	INT	45	Foreign Key
Nome_id	Nome e identificação do procedimento.	VARCHAR	45	Not null

Tabela 18 – Dicionário de Dados - Procedimento

Tabela	Prontuario_has_Orcamento			
Descrição	Armazena as informações do prontuário e orçamento.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
Prontuario_id	Código identificador do prontuário.	INT	-	
Orcamento_id	Código identificador do orçamento.	INT	-	

Tabela19: Dicionário de Dados - Tabela associativa Prontuário e Orçamento

Tabela	Orcamento			
Descrição	Armazena as informações do orçamento.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
id	Código identificador do orçamento.	INT	-	Primary Key
nome	Nome do orçamento.	VARCHAR	45	Not null
cpf	CPF do paciente que deseja realizar o orçamento.	INT	45	Not null

Tabela 20: Dicionário de Dados – Orçamento

no diagrama de implantação da figura 10.

Tabela	Dente			
Descrição	Armazena as informações do dente.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
id	Código identificador do dente.	INT	-	Primary Key
nome	Nome do dente.	VARCHAR	45	Not null
valor	Valor do dente.	VARCHAR	45	Not null
valor2	Valor do dente.	VARCHAR	45	-

Tabela 21: Dicionário de Dados - Dente

Tabela	Especialidade			
Descrição	Armazena as informações da especialidade.			
Nome	**Descrição**	**Tipo de Dado**	**Tamanho**	**Restrições de Domínio (Primary Key, Foreign Key, Not Null, Check, Default, Identify)**
id	Código identificador da especialidade.	INT	-	Primary Key
nome	Nome da especialidade.	VARCHAR	45	Not null

Tabela 223: Dicionário de Dados - Especialidade

O sistema foi é desenvolvido em uma arquitetura de três camadas seguindo o padrão de projeto MVC, tendo uma separação do código, da regra de negócio e da interface usuário.

Na primeira camada temos a interface, onde o sistema é apresentado visualmente, foi desenvolvido utilizando o framework Bootstrap 4 para desenvolver o front-end com CSS3 e JavaScript juntamente com o HTML5, citado nas tecnologias utilizadas.

Na segunda camada de aplicação utilizamos o framework Laravel, para desenvolver o código-fonte com o PHP (HyperText Preprocessor). Na terceira camada, usamos o MySql como o SGBD, para armazenamento de dados, onde é implementado através da função Migration, que nela é realizado o código baseado na modelagem do sistema proposto.

O sistema funciona de forma distribuída tendo 3 ambientes para sua execução que são: um servidor Linux onde rodará toda a aplicação integrando o front-end e o back-end pelo Laravel, um servidor onde rodará com o MySQL, e um computador pessoal ou smartphone para rodar a interface do sistema, via browser, como descrito

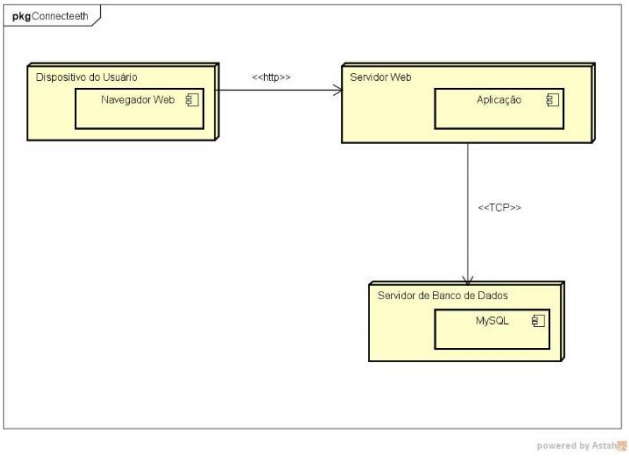

Figura 10: Ambiente do Sistema - Diagrama de Implantação

Considerações Finais

O sistema busca conectar e otimizar os serviços de saúde bucal tendo:

- O sistema ajudará com uma tela de orçamento, mas interativa trazendo valores e posição do dente selecionado, auxiliando assim o dentista com poucos toques fazer um orçamento preciso e com mais agilidade;
- Uma interface amigável e de fácil uso;
- O Sistema e de fácil acesso podendo acessar de vários dispositivos que tenha conexão com internet e um browser instalado;
- O sistema traz ferramentas para o auxílio da tomada de decisão como: calendário de agendamento de pacientes, o dentista que trabalha em várias clínicas pode em um único sistema se organizar, ter o prontuário do paciente digital trazendo mais agilidade e precisão nas informações;

Como trabalhos futuros para o sistema fica a necessidade de completar a implementação do sistema devido sua complexidade, foram desenvolvidas apenas as funções de maior importância para o funcionamento do sistema, no entanto dependendo a especialidade de cada profissional surge a necessidade de melhorar a performance do sistema completando a implementação do modulo financeiro trazendo o auxílio ainda maior de controle e gerando relatórios como nota fiscal entre outras, odontograma (prontuário odontológico com imagem) e o desenvolvimento de uma função que consiga fazer todo sua panorâmica e que esteja toda disponibilidade no seu prontuário

Referências Bibliográficas

1 – ALEXANDER, Christopher Alexander: A Pattern Language, Oxford Press, Oxford, R. Unido, 1978.

2 - Azure, **O que é o Azure**, 2020 Disponível em: https://azure.microsoft.com. Acessado em março de 2020.

3 - BRUM, Bruno Conde Perez; PENA, Leonardo. **Principais técnicasde levantamento de requisitos de sistemas. Engenharia de requisitos** – Técnicas. Disponível em: http://brunobrum.wordpress.com. Acessado em: 05 mai. 2020.

4 - DevMedia, **Técnicas para o Levantamento de Requisitos**. Disponível em https://www.devmedia.com.br. Acessado em 20 de março de 2020.
.

5 - GitLab, **Docs**. Disponível em https://docs.gitlab.com. Acessado em 20 de abril de 2020.

6 - Laravel, **Documentacion**. Disponível em https://laravel.com. Acessado em 15 de março de 2020
.

7 - LIMA, Davi de. **Modele software com Astah Community**.[s.1], 20 jun de 2016. Disponível em https://www.techtudo.com.br. Acessado em 05 mai. 2020.

8 - MACORATTI, José Carlos. **Padrões de Projeto: O modelo MVC - Model View Controller**. [*S. l.*], [201-]. Disponível em http://www.macoratti.net. Acessado em: 02 jun. 2020.

9 - PRESSMAN, Roger S. Engenharia de Software. 6ª ed. Rio de Janeiro: McGraw-Hill, 2006.

10 - SOMMERVILLE, Ian. Engenharia de

Software. 8ª ed. São Paulo: Pearson Addison- Wesley, 2007.