



LUCAS FLORIANO BERNARDES DE CASTRO

**SISTEMA PARA GERENCIAMENTO DE LAUDOS DE
AUDIOMETRIA**

LAVRAS – MG

2023

LUCAS FLORIANO BERNARDES DE CASTRO

SISTEMA PARA GERENCIAMENTO DE LAUDOS DE AUDIOMETRIA

Trabalho de Conclusão de Curso apresentado à
Universidade Federal de Lavras, como parte das
exigências do Curso de Bacharelado em Ciência da
Computação para a obtenção do título de Bacharel.

Prof. André Vital Saude
Orientador

LAVRAS – MG

2023

Dedico este trabalho primeiramente a Deus, pois é o pai que sempre me fortalece nos momentos de dificuldades. Dedico a minha família e amigos, que sempre estiveram ao meu lado nas provações e obstáculos da vida.

AGRADECIMENTOS

Agradeço impreterivelmente a minha mãe Raquel, que por mais tortuosa fosse a estrada a ser trilhada esteve ao nosso lado honrando-nos e dedicando sua vida à família. Agradeço a minha família por sempre estarem em comunhão nas adversidades da vida, sendo apoio e pilar dessa trajetória. Aos meus amigos por estarem presentes nessa jornada, fazerem parte dessa história e tantas vezes em auxílio nos momentos de necessidade. Por fim, a todos que contribuíram direta ou indiretamente para a realização dessa etapa em minha vida.

RESUMO

No dia a dia de um fonoaudiólogo, está entre suas responsabilidades atender diversos pacientes, fazer diversos exames audiométricos e registrar de forma sistemática estes exames. Pensando nessa problemática, este trabalho teve como objetivo o desenvolvimento de um sistema web para o gerenciamento de laudos de audiometria e cadastros de pacientes. Utilizando a linguagem JavaScript e bibliotecas como ReactJS e NodeJS, foi desenvolvido um sistema que gera gráficos com base nos dados coletados nos exames audiométricos e armazena os cadastros dos pacientes em um banco de dados. O sistema batizado de Phonoaudio, foi utilizado por um profissional fonoaudiólogo na dinâmica de sua rotina de atendimentos, com base na experiência de utilização e o feedback positivo do profissional que fez uso do sistema é possível afirmar que o trabalho obteve sucesso no seu objetivo.

Palavras-chave: Sistema. Audiometria. Laudo

ABSTRACT

In the daily life of a speech therapist, one of his responsibilities is to serve many patients, to do a lot of audiometric exams and to register these exams in a systematic way. Thinking about this problem, this work had as a goal the development of a web system to manage audiometric reports and patient records. Using a JavaScript language and libraries like ReactJs and NodeJS, it was developed a system to generate graphs based on collected data from audiometric exams and it stores the patient records in a database. The system named as Phonoaudio was used by a professional speech therapist while he was attending the patients in his routine, based on the experience using the system and the positive feedback from the professional who used it, it's possible to claim that the work had success in his goal.

Keywords: System. Audiometric. Patient Report

LISTA DE FIGURAS

Figura 2.1 – Estrutura básica de uma página HTML	12
Figura 2.2 – Conjunto de regras do CSS	13
Figura 2.3 – Exemplo de "Hello World"em JavaScript.	14
Figura 2.4 – Componente React.	15
Figura 2.5 – Implementação de gráfico utilizando o Google Charts.	17
Figura 2.6 – Resultado do componente implementado na figura 2.5.	18
Figura 2.7 – Exemplo de uma requisição POST com AXIOS.	19
Figura 2.8 – Exemplo de código em Node JS.	20
Figura 3.1 – Planilha Excel: Campos de Inserção	22
Figura 3.2 – Planilha Excel: Gráficos	23
Figura 4.1 – Diagrama Entidade Relacionamento	28
Figura 4.2 – Componente NavigationMenu	30
Figura 4.3 – Estilização de Componente com CSS	31
Figura 4.4 – Instalação Material UI	32
Figura 4.5 – Import Material UI	32
Figura 4.6 – Implementação do componente Dialog	33
Figura 4.7 – Instalação Google Charts	34
Figura 4.8 – Import Google Chart	34
Figura 4.9 – Implementação Google Charts	35
Figura 4.10 – Instalação Axios	35
Figura 4.11 – Import Axios	36
Figura 4.12 – Requisição Get com Axios	36
Figura 5.1 – Tela do Sistema	37
Figura 5.2 – Barra de Navegação	38
Figura 5.3 – Dados e Gráfico	39
Figura 5.4 – Área de cadastro	40
Figura 5.5 – Busca de pacientes	40

Figura 5.6 – Fluxograma de funcionamento 41

SUMÁRIO

1	Introdução	10
2	Conceitos e tecnologias	11
2.1	HTML - Linguagem de Marcação de Hipertexto	11
2.2	CSS - Folhas de Estilo em Cascata	12
2.3	JavaScript	14
2.4	React Js	15
2.4.1	Bibliotecas React	16
2.4.2	Node JS	19
3	Materiais e Métodos	21
3.1	Buscando Informações	21
3.2	Levantamento de Requisitos	24
3.3	Implementação	25
3.4	Revisão	26
4	Desenvolvimento	27
4.1	Rotina de Desenvolvimento	27
4.2	Backend	27
4.2.1	Banco de Dados	27
4.2.2	API	28
4.3	Frontend	29
4.3.1	Interação com o usuário	29
4.3.2	Estilização	30
4.3.3	Utilização de componentes	32
4.3.4	Visualização gráfica	34
4.3.5	Comunicação entre API e frontend	35
4.3.6	Segurança do Sistema	36
5	Resultados	37
6	Discussão	42

7	Conclusão	43
	REFERÊNCIAS	44

1 INTRODUÇÃO

O fonoaudiólogo é um profissional da área da saúde, que atua de maneira independente, podendo atuar no setor público ou privado tendo como algumas de suas responsabilidades o aperfeiçoamento dos aspectos fonoaudiológicos, juntamente com avaliação e diagnóstico, terapias de habilitação e reabilitação da função auditiva, linguagem oral e escrita, articulação da fala e dos sistemas orofacial, cervical, de deglutição, entre outros. Dentre todas as responsabilidades e etapas que envolvem o trabalho do fonoaudiólogo está a realização de exames de audiometria com o qual avalia a capacidade de audição de um indivíduo.

Com isso, destacamos que mesmo com os softwares já disponíveis no mercado, é uma dificuldade enfrentada pelo fonoaudiólogo ter, de forma prática, uma visualização gráfica dos testes de audiometria realizados associada a uma organização dos cadastros de seus pacientes. Mantendo junto aos registros de cada paciente um histórico de exames realizados.

Dessa forma o objetivo deste trabalho foi desenvolver um sistema web para que o profissional de fonoaudiologia pudesse cadastrar seus pacientes e ao realizar os exames de audiometria ter uma visualização gráfica dos resultados audiométricos. A fim de agilizar e sistematizar o atendimento ao paciente, deixando o processo de realização de exames audiométricos mais rápidos e práticos. Sendo para este profissional uma ferramenta gratuita que una todos esses recursos com praticidade, em relação à softwares pagos disponíveis no mercado.

O trabalho foi desenvolvido utilizando a linguagem de programação JavaScript com uso de ferramentas e bibliotecas dessa linguagem como ReactJS e NodeJS, para composição do sistema web que foi batizado de Phonoaudio. O sistema se baseia em duas principais funcionalidades, a exibição de um gráfico com os dados audiométricos inseridos e o cadastro de pacientes.

No capítulo 2 deste trabalho está apresentado os conceitos e tecnologias utilizados para o desenvolvimento do projeto. No capítulo 3 estão descritos os materiais e métodos que compuseram o desenvolvimento do sistema. O capítulo 4 detalha com especificações técnicas a construção do projeto. No capítulo 5 consta os resultados obtidos e a visualização da tela do sistema. O capítulo 6 denota uma análise dos resultados do que construído e desenvolvido. No capítulo 7 estão as conclusões.

2 CONCEITOS E TECNOLOGIAS

Neste capítulo serão apresentadas as tecnologias utilizadas durante toda a construção do sistema, destacam-se as tecnologias de desenvolvimento web, que foi a área que o projeto mais abrangeu. O desenvolvimento Web é um campo da tecnologia que se concentra na construção de sites, software, aplicativos e quaisquer outras ferramentas voltadas para a internet. A área de desenvolvimento Web é composta por duas sub-áreas, nomeadamente: o front-end e o back-end, sendo o back-end a área dedicada à infraestrutura do site, o front-end é responsável pela experiência do usuário, as telas onde o usuário interage com o sistema. (MILETTO; BERTAGNOLLI, 2014) Abaixo serão listadas as principais tecnologias utilizadas tanto no back-end quanto no front-end.

2.1 HTML - Linguagem de Marcação de Hipertexto

HTML (HyperText Markup Language) é a construção mais básica da web. HTML é frequentemente usado com CSS (Cascading Style Sheets), para descrever a aparência e, com JavaScript, para descrever sua funcionalidade e comportamento. O HTML usa "marcação" para anotar texto, imagens, tabelas e outros conteúdos para exibição em um navegador da Web.

Os elementos HTML são separados de outros textos no documento por "tags", que consistem em nomes de elementos entre " < > ". Os nomes dos elementos nas tags não diferenciam maiúsculas de minúsculas. Ou seja, pode ser escrito em maiúsculas, minúsculas ou uma mistura. Por exemplo, a tag <title> pode ser escrita como <Title>, <TITLE> ou de qualquer outra forma. Um documento HTML é dividido em dois blocos principais, separados pelas tags <head> e <body>. A tag <head> representa o cabeçalho da página onde são colocadas informações importantes, como por exemplo, o título da página (usando a tag <title>) ou metadados (usando a tag <meta>). (MOZILLA, 2022b). Na figura 2.1 podemos ver um exemplo de código HTML.

Figura 2.1 – Estrutura básica de uma página HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Fonte: <https://www.w3schools.com/html/default.asp>

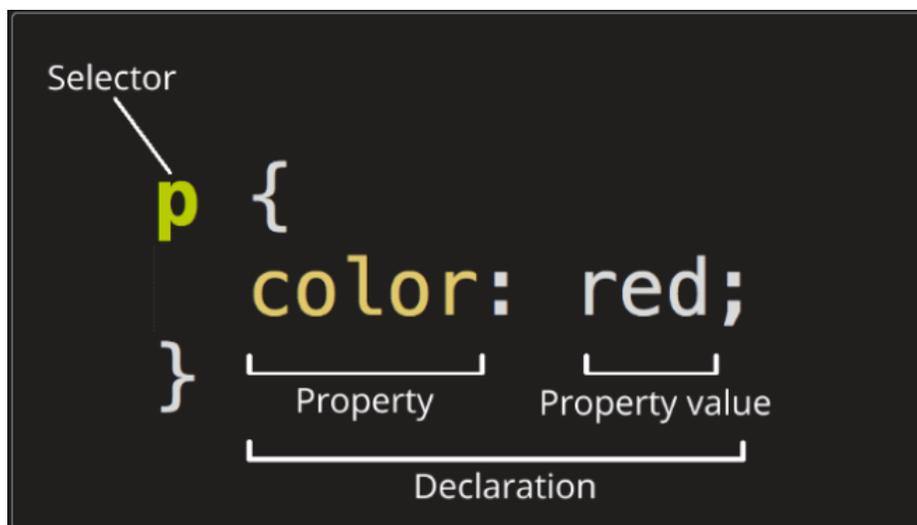
HTML refere-se a links que conectam páginas da Web entre si, seja no mesmo site ou entre sites. Os links são um aspecto essencial de um site. Ao enviar conteúdo para a Internet e criar links para páginas criadas por outras pessoas, você se torna um participante ativo na world wide web. (W3C, 2023)

2.2 CSS - Folhas de Estilo em Cascata

CSS (Cascading Style Sheets) é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML ou XML (incluindo várias linguagens XML como SVG, MathML, XHTML). CSS descreve como os elementos aparecem na tela, no papel, na fala ou em outras mídias.

CSS é uma das principais linguagens da web aberta e é padronizada em navegadores web de acordo com a especificação W3C. Desenvolvido em camadas, o CSS1 está atualmente obsoleto, o CSS2.1 é uma recomendação e o CSS3, agora dividido em pequenos módulos, caminha para a padronização. (MOZILLA, 2022a)

Figura 2.2 – Conjunto de regras do CSS



Fonte: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>

Como mostrado na figura 2.2 esse padrão que é usado em CSS é chamado de conjunto de regras. Observe os nomes das partes individuais:

Seleto (**seletor**) O nome do elemento HTML no início do conjunto de regras. Seleciona os elementos a serem modificados (neste caso `<p>` elementos).

Declaração Regra simples como `cor: vermelho;` especificando quais propriedades do elemento você deseja modificar.

Propriedades (Propriedade) Como estilizar um elemento HTML. (Nesse caso, a cor são as propriedades dos elementos `<p>`.) Em CSS, você pode escolher quais propriedades deseja que sua regra afete.

Valor da propriedade (valor da propriedade) À direita da propriedade após os dois pontos, temos um valor de propriedade que seleciona uma das muitas aparências possíveis para aquela propriedade (existem muitos valores de cores além do vermelho). (MOZILLA, 2022c)

O CSS foi bastante utilizado na construção do sistema, para deixá-lo da melhor forma para o usuário.

2.3 JavaScript

JavaScript® (às vezes abreviado para JS) é uma linguagem leve, interpretada e orientada a objetos com recursos de primeira classe, mais conhecida como uma linguagem de script para páginas da Web, mas também usada em muitos outros ambientes não relacionados a navegadores, como nodeJS, Apache CouchDB e Adobe Acrobat. JavaScript é uma linguagem prototípica, multi-paradigmática e dinâmica que suporta paradigmas como orientação a objetos, paradigmas imperativos e declarativos (como programação funcional). (ILYA KANTOR, 2023)

Colocar o código JavaScript diretamente dentro de um elemento `<script>` não é recomendado e deve ser usado apenas para fins de prova de conceito ou teste.

Na figura 2.3 um código JavaScript, onde é passado a função `alert()` com o texto “Hello, world!”:

Figura 2.3 – Exemplo de "Hello World" em JavaScript.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>JavaScript Hello World Example</title>
  <script>
    alert('Hello, World!');
  </script>
</head>
<body>
</body>
</html>
```

Fonte: <https://ricardo-reis.medium.com/hello-world-em-javascript-61fc5f8fc59c>

2.4 React Js

O ReactJS facilita a criação de interfaces de usuário interativas. Utilizado para criar exibições simples para cada estado em um aplicativo onde o React atualiza e renderiza com eficiência apenas os componentes necessários à medida que os dados mudam.

A visualização declarativa torna seu código mais previsível e fácil de depurar.

Como a lógica do componente é escrita em JavaScript e não em modelos, você pode passar facilmente diferentes tipos de dados em seu aplicativo e ainda manter o estado fora do DOM (Document Object Model) que está embutido no navegador.

Figura 2.4 – Componente React.

```
class HelloMessage extends React.Component {
  render() {
    return <div>Hello {this.props.name}</div>
  }
}

root.render(<HelloMessage name="Taylor" />);
```

Fonte: <https://pt-br.reactjs.org/>

Os componentes do React implementam um método "render()", uma função que recebe os dados de entrada e retorna o que deve ser exibido. Na figura 2.4 por exemplo é usado uma sintaxe semelhante a XML chamada JSX, uma extensão da linguagem JavaScript que é usada pelo React para criar interfaces de usuário. Os dados de entrada que são passados para o componente são acessíveis na função "render()" por meio do uso de "this.props". O React Js foi utilizado durante toda a construção do projeto e foi a principal ferramenta de criação do front-end do Phonoaudio. (META PLATFORMS, 2023)

2.4.1 Bibliotecas React

As bibliotecas do React fornecem maneiras mais eficientes de criar componentes personalizados, incluindo entradas, seleções, botões e novos componentes. Auxiliam bastante no desenvolvimento de componentes de maior complexidade, o que proporciona agilidade no desenvolvimento. Para o desenvolvimento do projeto foram utilizadas duas bibliotecas citadas a seguir:

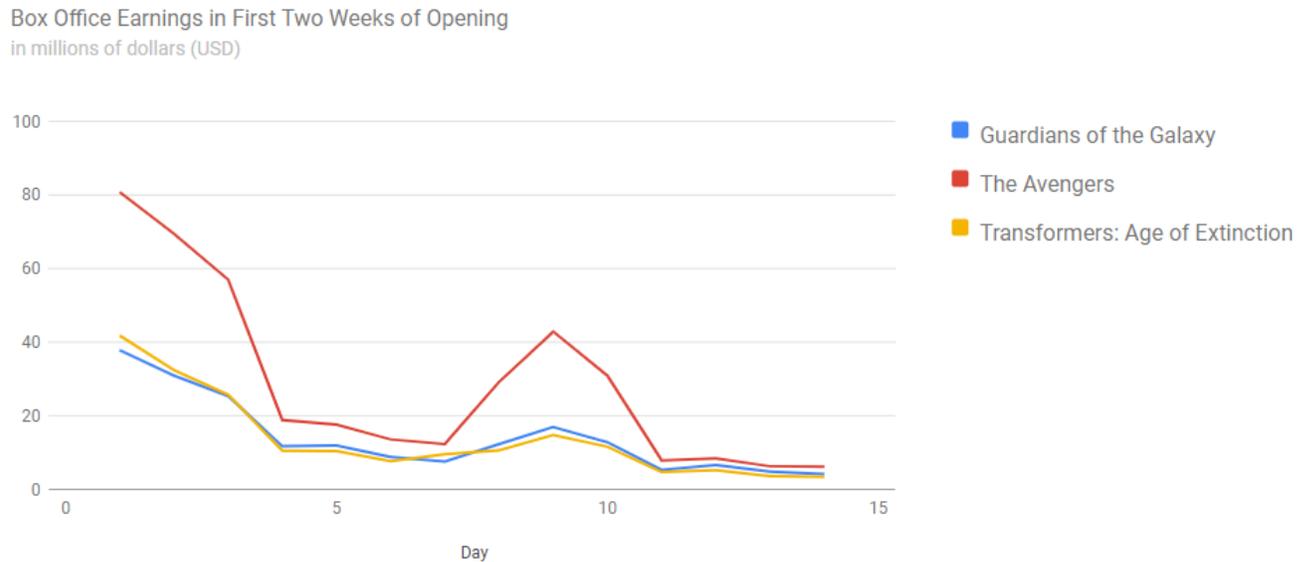
Google Charts: Construídos em JavaScript, ReactJS e Google Charts trabalham juntos para permitir que os desenvolvedores criem visualizações de dados atraentes e reutilizáveis, como gráficos de barras, pizza, bolhas entre outros. (GOOGLE, 2023)

Na figura 2.5 podemos ver como o google charts funciona, com poucas linhas de código, podemos construir gráficos bastante completos e intuitivos, como mostrado da figura 2.6.

Figura 2.5 – Implementação de gráfico utilizando o Google Charts.

```
1 import React from "react";
2 import { Chart } from "react-google-charts";
3
4 export const data = [
5   [
6     "Day",
7     "Guardians of the Galaxy",
8     "The Avengers",
9     "Transformers: Age of Extinction",
10  ],
11  [1, 37.8, 80.8, 41.8],
12  [2, 30.9, 69.5, 32.4],
13  [3, 25.4, 57, 25.7],
14  [4, 11.7, 18.8, 10.5],
15  [5, 11.9, 17.6, 10.4],
16  [6, 8.8, 13.6, 7.7],
17  [7, 7.6, 12.3, 9.6],
18  [8, 12.3, 29.2, 10.6],
19  [9, 16.9, 42.9, 14.8],
20  [10, 12.8, 30.9, 11.6],
21  [11, 5.3, 7.9, 4.7],
22  [12, 6.6, 8.4, 5.2],
23  [13, 4.8, 6.3, 3.6],
24  [14, 4.2, 6.2, 3.4],
25  ];
26
27 export const options = {
28   chart: {
29     title: "Box Office Earnings in First Two Weeks of Opening",
30     subtitle: "in millions of dollars (USD)",
31   },
32 };
33
34 export function App() {
35   return (
36     <Chart
37       chartType="Line"
38       width="100%"
39       height="400px"
40       data={data}
41       options={options}
42     />
43   );
44 }
```

Figura 2.6 – Resultado do componente implementado na figura 2.5.



Fonte: <https://www.react-google-charts.com/examples/line>

AXIOS: Axios é um cliente HTTP baseado em promessas para nodeJS e o navegador. É isomórfico (pode ser executado no navegador e nodeJS com a mesma base de código). No lado do servidor, ele usa o código nodeJS nativo - o módulo http, enquanto no lado do cliente (navegador) usa XMLHttpRequests. (ZABRISKIE, 2023)

Uma biblioteca com diversos recursos e utilização simples, que facilita a requisição entre nosso sistema e sua API, segue exemplo de uma requisição POST utilizando o AXIOS na figura 2.7.

Figura 2.7 – Exemplo de uma requisição POST com AXIOS.

```
axios.post('/user', {
  firstName: 'Santos',
  lastName: 'Dumont'
})
.then(function (response) {
  console.log(response);
})
.catch(function (error) {
  console.error(error);
});
```

Fonte: https://axios-http.com/ptbr/docs/post_example

2.4.2 Node JS

NodeJS pode ser definido como um tempo de execução Javascript do lado do servidor.

Isso significa que com o NodeJS é possível criar aplicações Javascript que rodam como uma aplicação autônoma no computador, não dependendo do navegador como usualmente é feito. (OPUS SOFTWARE, 2023)

Como um tempo de execução JavaScript assíncrono e orientado a eventos, o NodeJS foi projetado para o desenvolvimento de aplicativos da Web escaláveis. Na figura 2.8, várias conexões podem ser controladas simultaneamente. Uma função de retorno é chamada em cada conexão. Mas se nada precisa ser feito, o NodeJS ficará inativo.

Figura 2.8 – Exemplo de código em Node JS.

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Fonte: <https://nodejs.org/pt-br/about>

NodeJS foi a tecnologia utilizada para a estruturação do backend com a criação da API que faz a comunicação com o frontend e o acesso ao banco de dados.

3 MATERIAIS E MÉTODOS

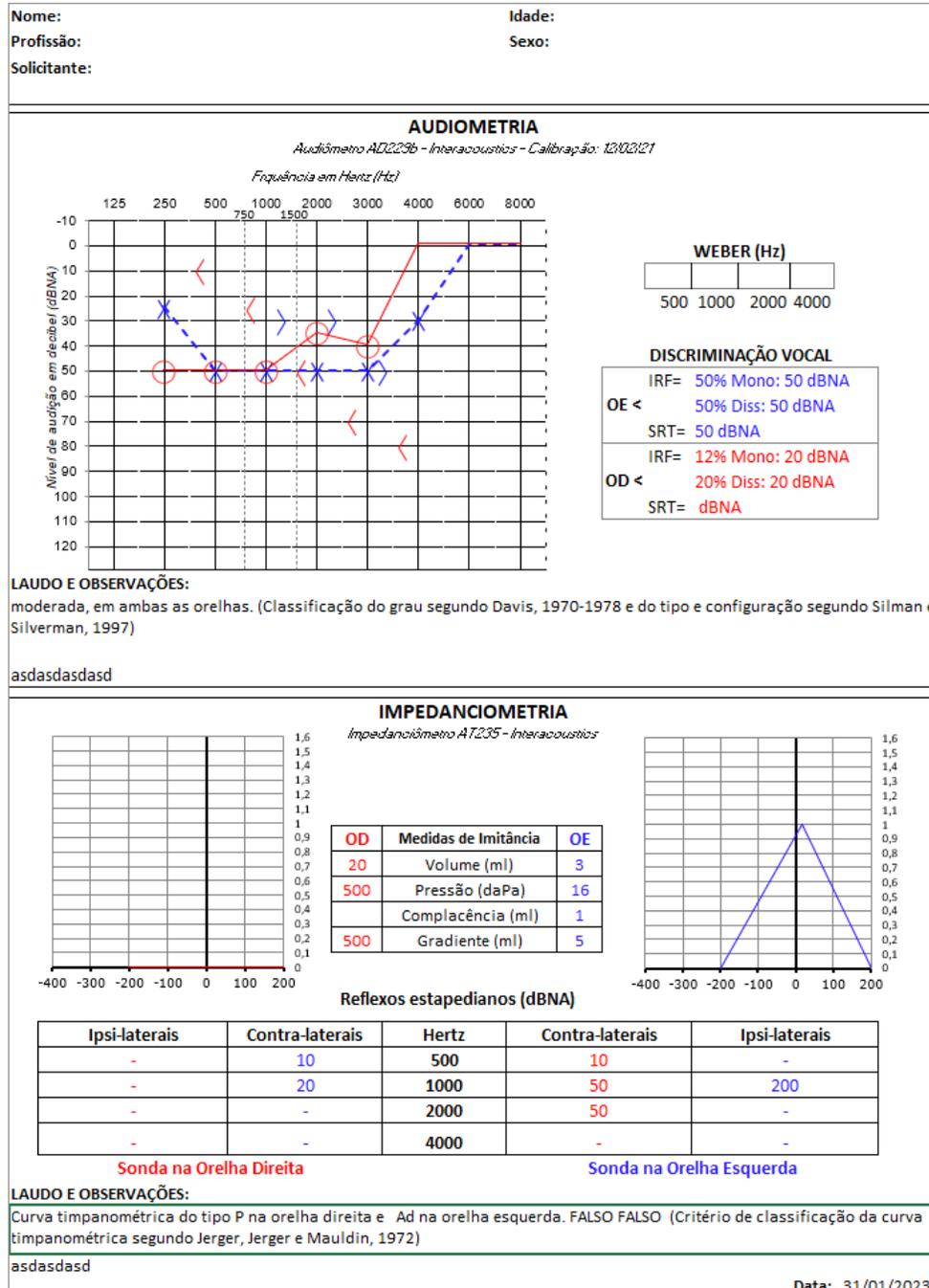
Nesta seção são descritos os métodos e processos envolvidos para desenvolver o projeto, o objetivo final foi desenvolver um sistema que atendesse aos requisitos e sanasse os problemas enfrentados pelos profissionais da área. A primeira etapa do trabalho foi a elucidação das necessidades do usuário e a concepção da solução.

3.1 Buscando Informações

Para fazer o intermédio entre o ambiente de desenvolvimento e cenário prático do consultório fonoaudiológico, o projeto contou com uma informante, fonoaudióloga, que apesar de não pagar pelo serviço, podemos chamar de cliente, pois é parte interessada e foi quem apresentou a lacuna no mercado, mostrando a relevância de se ter um produto livre que atendesse de forma mais adequada que os existentes. A cliente apresentou os requisitos desejados no sistema na forma de vídeos e conversas com o Product Owner que em questão está representado pelo orientador do trabalho André Vital Saúde, o qual teve contato com a cliente que forneceu exemplos de laudos e planilha Excel avançada como um protótipo funcional do produto desejado.

Nas figuras 3.1 e 3.2 abaixo temos a planilha Excel avançada fornecida pela cliente com o objetivo de apresentar muitos dos recursos desejados por ela no sistema a ser desenvolvido. Apesar de ser utilizada cotidianamente pela cliente, a planilha apresenta limitações por não ser um sistema web e estar limitada aos recursos do software Excel. No entanto, a planilha mostra o que a cliente considera como necessário no software, e o projeto do sistema foi realizado com base nessa planilha.

Figura 3.2 – Planilha Excel: Gráficos



Fonte: Protótipo Excel (2023)

3.2 Levantamento de Requisitos

Em posse da planilha Excel e de laudos usados como parametrização dos requisitos do software e o contato com a cliente que atua na área de fonoaudiologia e tem ciência de causa das funcionalidades almejadas, foi feito o levantamento dos requisitos de software que se esperavam do projeto e de seu funcionamento. E, a partir deste ponto, o software começou a ser projetado e desenvolvido. O objetivo foi sempre atender as necessidades da cliente com relação ao uso do sistema com funções que facilitariam o seu trabalho. A planilha Excel fornecida serviu como protótipo para a concepção das funcionalidades do sistema a ser desenvolvido. Esse protótipo foi usado para validar as funcionalidades e os gráficos esperados no sistema, denotando muito do comportamento e expectativas do sistema.

1. Cadastro: Esse requisito permite fazer o cadastro de um paciente no sistema, com as informações necessárias para registro e acompanhamento ao longo do tempo.
2. Informar Solicitante: Permite que ao, fazer um cadastro, seja possível informar o solicitante ligado ao paciente, seja um médico, clínica, solicitação particular, etc.
3. Laudo: Com as inserções do exame audiométrico, o campo Laudar, permite descrever o laudo do exame, com as descrições de sua condição auditiva, incluindo observações pertinentes feitas pelo fonoaudiólogo.
4. Registro de Exames: Com essa funcionalidade é possível registrar as informações obtidas pelo exame realizado. Quando é realizado um exame, os dados inseridos, incluindo o laudo obtido, ficam registrados no cadastro do paciente.
5. Plotagem de Gráfico: Essa funcionalidade do sistema permite que com os dados obtidos no exame audiométrico, o fonoaudiólogo os insira nos campos de inserção especificados e seja gerado um gráfico com os dados audiométricos.
6. Atualização instantânea do gráfico: A cada nova inserção de dados para gerar um ponto no gráfico, as coordenadas são atualizadas automaticamente, inserindo um novo ponto no gráfico, sem a necessidade de comando para atualização.

7. Registro no banco de dados: Todos os dados inseridos, desde dados cadastrais até exames audiométricos são registrados no banco de dados.
8. Consulta ao banco de dados: Permite que o sistema faça busca no banco de dados para resgatar exames anteriores.
9. Lista de exames: Ao consultar um paciente no banco de dados permite consultar o histórico dos exames deste paciente.

3.3 Implementação

O sistema foi implementado sob o conceito de frontend e backend com um banco de dados para armazenar as informações pertinentes. O frontend que é o responsável por toda interação com o usuário, entrada e saída de dados, foi desenvolvido com React. A escolha dessa tecnologia foi devida a diversos fatores: é uma tecnologia voltada ao desenvolvimento web; conta com muito suporte e material funcional como tutoriais e documentação, facilitando a continuidade do desenvolvimento; é muito compatível com os sistemas e ferramentas atuais, podendo ser o sistema aprimorado com facilidade; possui muitas bibliotecas complementares como frameworks e plugins, agilizando o desenvolvimento e fornecendo recursos complexos de forma prática.

O ReactJS possui diversas características marcantes, é uma excelente biblioteca JavaScript para o desenvolvimento de aplicações web. Muitas empresas o utilizam, o que o torna ainda mais desejada e o que o torna uma excelente escolha para o desenvolvimento de aplicações web. (TREINAWEB, 2023b)

O backend foi responsável pela gestão da API que recebe as requisições do frontend, interage com o banco de dados e retorna ao frontend as informações requisitadas.

O NodeJS foi escolhido para estruturar o banco de dados por ser uma tecnologia de alta compatibilidade com o ReactJS, possuindo também muitos recursos que vão tornar a comunicação com o frontend mais prática e melhor estruturada. A dupla ReactJS e NodeJS tem sido muito utilizada no mercado, com muita base de material, muito suporte ao desenvolvimento e aceitação pelos sistemas atuais.(TREINAWEB, 2023a)

O banco de dados foi modelado sob conceito de Entidade Relacionamento (ER) e desenvolvido com MySQL, um sistema gerenciador de banco de dados relacional de código aberto que utiliza a linguagem SQL (Structure Query Language – Linguagem de Consulta Estruturada).(ORACLE, 2023)

3.4 Revisão

Com uma versão funcional do sistema, muito próximo dos requisitos determinados, das funcionalidades requisitadas e apresentadas no protótipo desenvolvido em Excel, nos laudos apresentados, houve uma revisão desta versão do sistema, onde o desenvolvedor e o product owner averiguaram o funcionamento atual do sistema e foi determinado algumas modificações e otimizações. Duas determinações feitas foram:

1. Permitir que fosse exibido no sistema a lista de pacientes a medida que o nome fosse digitado na barra de pesquisa, assim pouparia tempo de busca, agilizaria o atendimento e a entrada dos dados no exame de audiometria.
2. Os campos de inserção iniciariam selecionados prontos para a digitação sem que o usuário clicasse com o mouse para poder dar entrada em cada campo, assim otimizando o processo do atendimento.

4 DESENVOLVIMENTO

Este capítulo tem como objetivo descrever como as tecnologias envolvidas no sistema foram utilizadas. O sistema que foi desenvolvido com a linguagem JavaScript foi nomeado de Phonoaudio e foi estruturado de forma que o backend é responsável pela gestão e armazenamento dos dados enquanto o frontend gerencia toda interação com o usuário, como entrada e saída de dados e representações gráficas.

4.1 Rotina de Desenvolvimento

A dinâmica de desenvolvimento do sistema se deu de forma remota com desenvolvedor único que não estava presente no campus da universidade mas que manteve contato constante com o orientador e product owner André Vital Saúde. As atualizações e troca de informações com o orientador ocorreram por meio de whatsapp, emails e reuniões no google meet. Assim foram feitas periodicamente revisões nas etapas desenvolvidas e sempre estabelecidas metas para a próxima etapa de desenvolvimento. Não havia estipulação exata de prazo para conclusão das etapas de desenvolvimento e reuniões, assim que a etapa era concluída pelo desenvolvedor havia o contato com o orientador para marcar a reunião que definiria os próximos passos do desenvolvimento, entretanto as reuniões ocorriam em média a cada 25 dias.

4.2 Backend

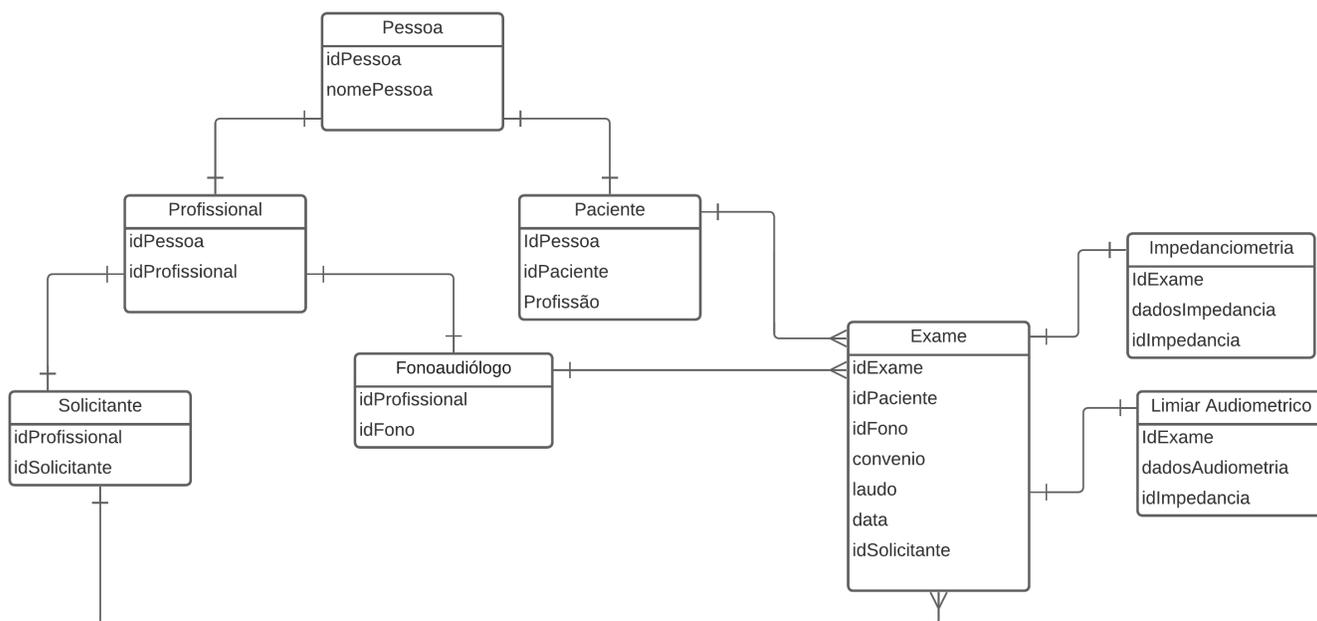
Responsável pela API que gerencia o banco de dados, o back-end faz toda a gestão que liga o front-end aos dados armazenados. Usando o NodeJS para implementação da API, faz-se as requisições ao banco de dados, para buscar, inserir, deletar, etc.

4.2.1 Banco de Dados

O banco de dados utilizado no sistema segue o modelo Entidade Relacionamento (ER), no diagrama da figura 4.1 é denotada a modelagem do banco de dados, onde o Exame possui um laudo, uma impedanciometria, um limiar audiométrico, é realizado por um profissional fonoaudiólogo, em

um paciente e tem um solicitante, o exame contém dados como convenio, identificações e outras informações pertinentes. O objetivo desta modelagem é manter um histórico de exames de um determinado paciente com todos os dados, desde seu convenio e profissional que realizou o exame até mesmo os dados coletados pelo profissional no exame.

Figura 4.1 – Diagrama Entidade Relacionamento



Fonte: Do Autor (2023)

4.2.2 API

A API que gerencia o acesso e requisições ao banco de dados, foi desenvolvida com a utilização do NodeJs e executa ações como:

- Cadastro completo: na primeira consulta é possível fazer o cadastro do paciente no banco de dados, inserindo as informações pertinentes ao paciente.
- Registro de exame: ao ser realizado um exame no paciente é possível inserir os dados e já o vincular ao paciente examinado, previamente cadastrado no sistema e selecionado.

- Busca de todos os pacientes: essa ação permite listar os pacientes cadastrados no sistema.
- Busca de paciente específico: essa ação permite visualizar o registro de um paciente pesquisado.
- Deletar paciente: com esta ação é possível excluir o registro paciente do banco de dados.
- Editar dados do paciente: possibilita fazer alterações no registro do paciente.

4.3 Frontend

Por se tratar de um sistema web, o frontend é a parte mais abrangente do projeto, as interações e interfaces gráficas pertinentes ao usuário são de delimitação do frontend. É moldado com a junção de diversas ferramentas como:

- ReactJS para estruturar o frontend
- CSS para estilizar os componentes visuais do frontend
- Material UI para fornecer componentes de otimização e customização no desenvolvimento
- Google Charts para construir gráficos
- Axios para estruturar a comunicação entre o frontend e a API do backend

4.3.1 Interação com o usuário

O ReactJS foi utilizado em todo frontend do sistema Phonoaudio para a criação de interfaces de usuário. Foi implementado com a ideia de oferecer reatividade e fornecer mais flexibilidade no desenvolvimento, permitindo que componentes do sistema fossem atualizados individualmente, sem a necessidade de renderizar toda a página em que o componente se encontra. Permitindo assim modularizar o código em componentes para que possam ser reutilizados sem que haja redundância no código. Na figura 4.2 mostrada a seguir, é demonstrada a implementação do componente NavigationMenu:

Figura 4.2 – Componente NavigationMenu

```
import React from "react";
import "./NavigationMenu.css";
import { AiOutlineFileText, AiOutlineReconciliation } from "react-icons/ai";
export const NavigationMenu = () => {
  return (
    <div className="navigation-wrapper">
      <div className="navigation-container">
        <h1>Phonoaudio</h1>
        <div className="navigation-buttons-container">
          <div className="navigation-buttons">
            <AiOutlineFileText size={22} />
            <a href="/Person">Exames</a>
          </div>
          <div className="navigation-buttons">
            <AiOutlineReconciliation size={22} />
            <a href="/">Registro</a>
          </div>
        </div>
      </div>
    </div>
  );
};
```

Fonte: Do Autor (2023)

Este componente é muito utilizado para fazer a navegação lateral do sistema, possibilitando organizar as funcionalidades do sistema.

4.3.2 Estilização

Uma vez que o ReactJS faz uso de tags HTML para implementar seus componentes e construir suas interfaces, podemos utilizar o CSS para estilizá-los, assim como em sites HTML. O sistema foi desenvolvido utilizando um arquivo .css onde cria-se classes que são usadas para estilizar os componentes, evitando assim uma sobrescrita de código de estilização. Uma estilização utilizada no sistema foi usada no componente NavigationMenu e é exibida na figura 4.3:

Figura 4.3 – Estilização de Componente com CSS

```
.navigation-wrapper {
  position: absolute;
  width: 256px;
  left: 0px;
  top: 0px;
  bottom: 0px;
  height: 150%;
  background: #ffffff;
  /* navigation menu / vertical shadow */

  box-shadow: 4px 0px 16px rgba(16, 30, 115, 0.08);
}

.navigation-container {
  padding-left: 5%;
}

.navigation-container h1 {
  margin-top: 20px;
  font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Oxygen,
  | Ubuntu, Cantarell, "Open Sans", "Helvetica Neue", sans-serif;
  color: #336cfb;

  margin-bottom: 40%;
}

.navigation-container a {
  /* Subtitle 2 - bold 14 (20, 0.1px) */

  font-family: "Lato";
  font-style: normal;
  font-weight: 700;
  font-size: 20px;
  line-height: 20px;
  /* identical to box height, or 143% */

  letter-spacing: 0.1px;

  /* Neutral / Gray dark */

  color: #52575c;
  margin-bottom: 10%;
}

.navigation-buttons {
  display: flex;
}
```

Fonte: Do Autor (2023)

4.3.3 Utilização de componentes

O ReactJS nos permite utilizar plugins para aumentar o desempenho e agilidade no desenvolvimento e como se baseia em componentização, há diversos plugins que trazem variados componentes comumente utilizados, que já estão implementados e ao serem utilizados trazem ferramentas e recursos complexos já prontos e testados, que deixam o sistema com menos probabilidade a erros e poupam o tempo do desenvolvedor. O Material UI é um exemplo deste tipo de plugin, bastante conhecido, conta com diversos componentes otimizados e customizáveis. Sua instalação é simples, bastando utilizar os comandos exibidos na figura 4.4. E para utilizá-lo faz-se a importação como qualquer outro componente do React, assim como mostra a figura 4.5::

Figura 4.4 – Instalação Material UI

npm

```
npm install @mui/material @emotion/react @emotion/styled
```

yarn

```
yarn add @mui/material @emotion/react @emotion/styled
```

Fonte: <https://mui.com/material-ui/getting-started/installation/>

Figura 4.5 – Import Material UI

```
import Button from "@material-ui/core/Button";  
import TextField from "@material-ui/core/TextField";  
import Dialog from "@material-ui/core/Dialog";  
import DialogActions from "@material-ui/core/DialogActions";  
import DialogContent from "@material-ui/core/DialogContent";  
import DialogTitle from "@material-ui/core/DialogTitle";
```

Fonte: Do Autor (2023)

O sistema Phonoaudio fez uso do plugin Material UI, na figura 4.6 é demonstrado a implementação de um de seus componentes:

Figura 4.6 – Implementação do componente Dialog

```
import React, { useState } from "react";
import Button from "@material-ui/core/Button";
import TextField from "@material-ui/core/TextField";
import Dialog from "@material-ui/core/Dialog";
import DialogActions from "@material-ui/core/DialogActions";
import DialogContent from "@material-ui/core/DialogContent";
import DialogTitle from "@material-ui/core/DialogTitle";
import Axios from "axios";

export default function FormDialog(props) {
  const [editValues, setEditValues] = useState({
    cpf: props.cpf,
    name: props.name,
    insurance: props.insurance,
  });

  const handleChangeValues = (values) => { ...
  };

  const handleClose = () => { ...
  };

  const handleEditPerson = () => { ...
  };

  const handleDeletePerson = () => { ...
  };

  return (
    <div>
      <Dialog
        open={props.open}
        onClose={handleClose}
        aria-labelledby="form-dialog-title"
      >
        <DialogTitle id="form-dialog-title">Editar</DialogTitle>
        <DialogContent> ...
        </DialogContent>
        <DialogActions> ...
        </DialogActions>
      </Dialog>
    </div>
  );
}
```

Fonte: Do Autor (2023)

4.3.4 Visualização gráfica

Uma das funções mais importantes do sistema é a plotagem de gráfico com os dados inseridos pelo fonoaudiólogo, seguindo a ideia de plugins para complementar o desenvolvimento do sistema, foi utilizado o plugin google charts, desenvolvido especificamente para realizar plotagem de gráficos em sistemas web, trazendo diversos modelos de gráficos customizáveis. Para instalação do Google Charts utilizamos os comandos demonstrados na figura 4.7 e o utilizamos fazendo a importação como mostrado na figura 4.8:

Figura 4.7 – Instalação Google Charts

Install this library with your favorite package manager:

```
yarn add react-google-charts
```

or

```
npm install --save react-google-charts
```

Fonte: <https://www.react-google-charts.com/>

Figura 4.8 – Import Google Chart

```
import { Chart } from "react-google-charts";
```

Fonte: Do Autor (2023)

Na implementação, utilizamos o Google Charts como uma tag do html passando os parâmetros desejados, seguindo a ordem de tipo do gráfico, altura, largura, os dados para serem plotados e as opções como nome e informações que aparecem nos eixos do gráfico (legendas) como mostra a figura 4.9.

Figura 4.9 – Implementação Google Charts

```
<Chart  
  chartType="ScatterChart"  
  width="80%"  
  height="300px"  
  data={Dchart}  
  options={DOptions}  
>
```

Fonte: Do Autor (2023)

4.3.5 Comunicação entre API e frontend

Para fazer a comunicação entre a API e o frontend foi utilizado o plugin AXIOS com o objetivo de gerir essa conexão. Podemos fazer sua instalação utilizando os comandos demonstrados na figura 4.10.

Figura 4.10 – Instalação Axios

Usando o npm:

```
$ npm install axios
```

Usando o bower:

```
$ bower install axios
```

Usando o yarn:

```
$ yarn add axios
```

Fonte: <https://axios-http.com/ptbr/docs/intro>

Importamos o plugin como um componente, utilizando o demonstrado na figura 4.11.

Figura 4.11 – Import Axios

```
import Axios from "axios";
```

Fonte: Do Autor (2023)

O Axios nos permite fazer as requisições à API. Na figura 4.12 a seguir, temos uma requisição do tipo GET, onde é feito a busca de pacientes e os dados retornados são salvos em um state com nome “listPerson”.

Figura 4.12 – Requisição Get com Axios

```
Axios.get("http://localhost:3001/persons").then((response) => {  
  setListPerson(response.data);  
});
```

Fonte: Do Autor (2023)

4.3.6 Segurança do Sistema

O projeto e as definições esperadas para o desenvolvimento deste trabalho não idealiza estruturas focadas na segurança do sistema, tendo em vista que o trabalho prevê um sistema pequeno, focado uma usabilidade do cotidiano de um profissional fonoaudiólogo. Por não se tratar do desenvolvimento de um sistema robusto de larga escala e comercial, o ponto desenvolvido por este trabalho de conclusão de curso não determina mecanismos para assegurar a segurança dos dados do sistema, apesar de reconhecermos a importância do trabalho em segurança de software, o sistema no estágio atual tem premissa de uso interno e local de um consultório de um profissional fonoaudiólogo. Estimula-se o desenvolvimento de determinadas estruturas de segurança para trabalhos futuros e possíveis continuidades no desenvolvimento do projeto.

5 RESULTADOS

Para o desenvolvimento do sistema que foi nomeado de Phonoaudio foram seguidas as etapas discriminadas anteriormente, com tudo foram feitas observações do que era esperado pela cliente além de obtenção de um protótipo e comunicação entre o product owner e a parte interessada para que fossem entendidas as dores de mercado que o projeto atenderia, de modo que sempre houvesse um acompanhamento do desenvolvimento, para que o projeto estivesse cada vez mais alinhado e atendesse cada vez melhor as necessidades em relação à sistemas e ferramentas já disponíveis no mercado.

É mostrado na figura 5.1 a tela inicial do sistema no ponto atual de desenvolvimento, com os campos e recursos relacionados as funcionalidades e requisitos que o sistema atende.

Figura 5.1 – Tela do Sistema

Dados

Nome:	Jean Carlos	Sexo:	Masculino
Profissão:	Eng. Civil	Nascimento:	1988-02-14
Solicitante:	Dr. Solicitante	Documento:	147852369
Convênio:	Unimed		

Gráficos

OE
IRF = 2 Mono: 1 dBNA 3 Diss: 1 dBNA
SRT = 3 dBNA

OD
IRF = 2 Mono: 1 dBNA 3 Diss: 1 dBNA
SRT = 3 dBNA

Laudo e Observações:

Gráficos

Cadastro do usuário

Documento: 147852369 | Nome: Jean Carlos | Profissão: Eng. Civil

Solicitante: Dr. Solicitante | Convênio: Unimed | Fono: Lucia

Sexo: Masculino | DN: 14/02/1988

Laudar: Laudar

Exame: Audiometria Impedanciometria

Limiares Audiométricos

0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4
0.6	0.5	0.4	0.3	0.5	0.4	0.3	0.2

IRF: 2 % Mono 3 % Diss 1 dBNA SRT: 3 dBNA

IRF: 2 % Mono 3 % Diss 1 dBNA SRT: 3 dBNA

Fonte: Do Autor (2023)

Na figura 5.2 é mostrado o menu lateral, uma barra de navegação utilizada para fazer a transição entre as telas do sistema.

Figura 5.2 – Barra de Navegação

Phonoaudio

 Pacientes

 Registro

Salvar (Ctrl+S)

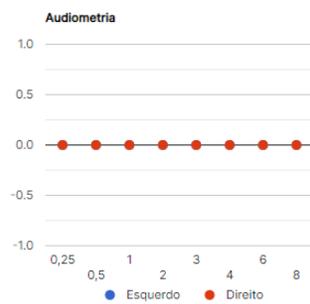
Fonte: Do Autor (2023)

Na figura 5.3 é mostrado a área do sistema que exibe dados do paciente examinado e o gráfico que exibe as inserções dos dados audiométricos.

Figura 5.3 – Dados e Gráfico

Dados

Nome:	Sexo:	Outro
Profissão:	Nascimento:	
Solicitante:	Documento:	
Convênio:		

Gráficos

OE
 IRF= 0 Mono: 0 dBNA 0 Diss: 0
 dBNA
 SRT = 0 dBNA

OD
 IRF= 0 Mono: 0 dBNA 0 Diss: 0
 dBNA
 SRT = 0 dBNA

Laud e Observações:

//

 Fonte: Do Autor (2023)

Na figura 5.4 é exibida a área onde são inseridos os dados para cadastro de um novo paciente ou alteração de dados de pacientes já cadastrados.

Figura 5.4 – Área de cadastro

Cadastro do usuário

Documento: Documento do pacien

Nome: Nome do pacien

Profissão: Aposentado (a)

Solicitante: Dr. Vinicius Malaquias Ramos

Convênio: Convênio do pacien

Fono: Fono

Sexo:

DN: dd/mm/aaaa

Laudar: Laudar

NORMAL

Fonte: Do Autor (2023)

Na figura 5.5 é mostrada a área do sistema que se faz a busca por pacientes e seu histórico de exames.

Figura 5.5 – Busca de pacientes

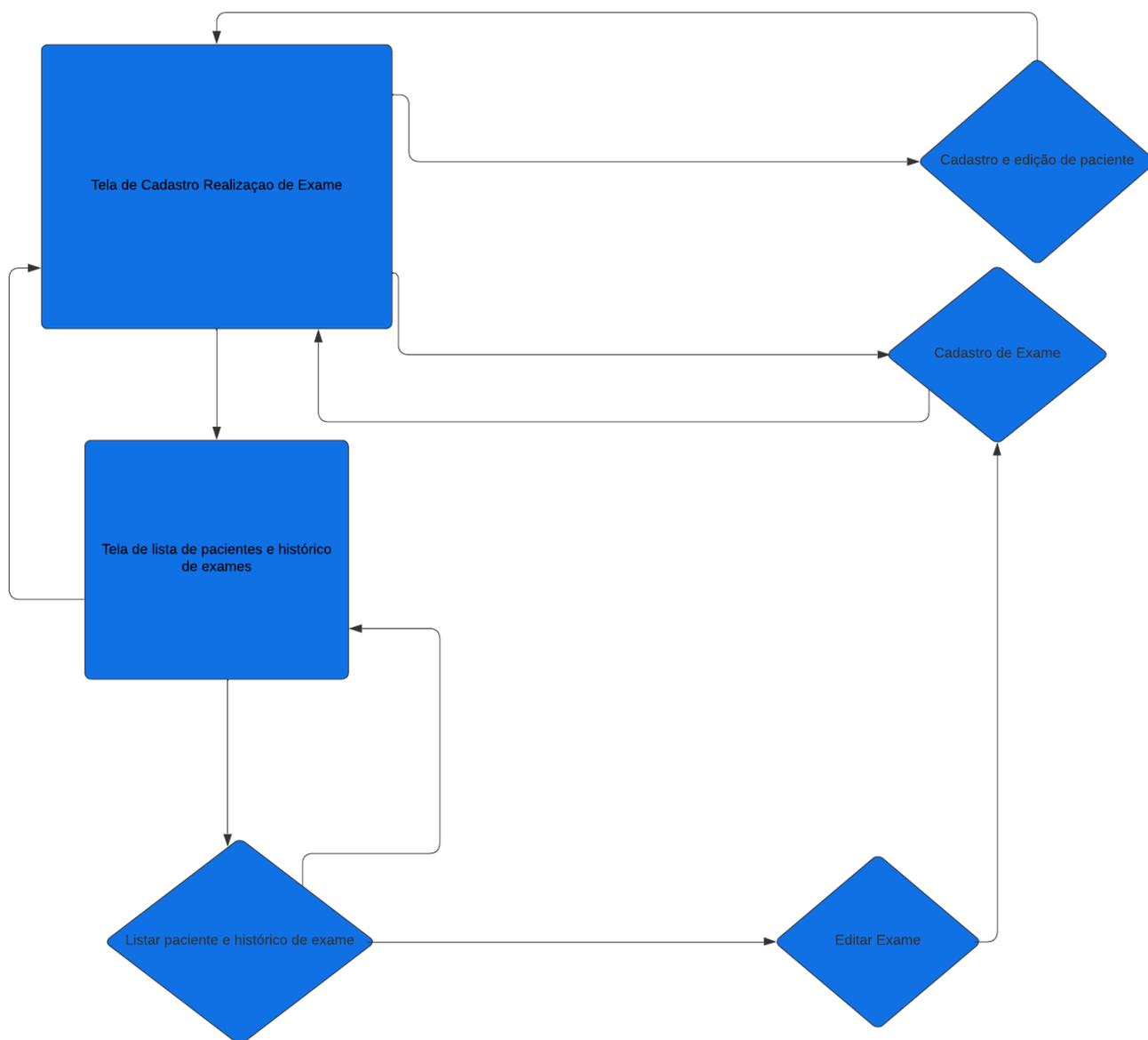
Busca:

CPF	Nome	Editar	Exames
12345678	Paulo Mercio de Souza	<input type="button" value="Editar"/>	<input type="button" value="Ver exames"/>
14785236	Helena das Dores Moura	<input type="button" value="Editar"/>	<input type="button" value="Ver exames"/>
74185296	Liane Batista Silva	<input type="button" value="Editar"/>	<input type="button" value="Ver exames"/>
96385214	Patrick Barros da Silva	<input type="button" value="Editar"/>	<input type="button" value="Ver exames"/>

Fonte: Do Autor (2023)

Na figura 5.6 é apresentado um fluxograma de funcionamento do sistema no estágio atual de desenvolvimento. Apresentando a ordem de navegação entre as telas e entre as funcionalidades do sistema.

Figura 5.6 – Fluxograma de funcionamento



Fonte: Do Autor (2023)

6 DISCUSSÃO

Após todo o processo para o desenvolvimento do sistema e devidas otimizações, correções e as demais etapas serem realizadas, incluindo todo acerto com relação a entrega de funcionalidades e aparência do sistema, para que haja um uso intuitivo somado a um layout amigável para o usuário, nos vemos num momento de realizarmos uma avaliação do que foi desenvolvido e elaborado no sistema.

Deparamos-nos com o resultado de um ponto do desenvolvimento do sistema e concordamos com um índice satisfatório proporcionado no estágio atual, observando que no presente momento o sistema fornece as funcionalidades requisitadas, baseando e aprimorando as funcionalidades contidas no protótipo, se apresenta como uma solução que se adéqua para o ambiente em que será utilizado, se mostra intuitivo e proporcional ao que se esperava com os recursos disponíveis como tempo, ferramentas de desenvolvimento e desenvolvedor único trabalhando no projeto. Foram realizados no sistema alguns testes para averiguar as funcionalidades, a usabilidade e constância do sistema. Foi realizado testes de unidade para testar métodos e funções individuais de classes, componentes e módulos usados pelo software, testes de integração para verificar se diferentes módulos ou serviços usados pelo sistema estão funcionando corretamente, como a comunicação com o banco de dados. Testes funcionais também foram realizados, para averiguar os dados retornados pelas funções. Foram realizados também testes de ponta a ponta para replicar o comportamento de um usuário com o software em um ambiente usual completo. Com isso, nos damos por satisfeitos com o projeto, deixando o sistema em um ponto satisfatório para o que foi proposto, entregando uma agilidade na visualização dos exames audiométricos, que já atende parcialmente as necessidades do profissional e que após avaliação da cliente com possíveis ajustes, poderá integrar o dia a dia do profissional fonoaudiólogo.

7 CONCLUSÃO

Neste trabalho foi desenvolvido um sistema web para gerenciamento de laudos de audiometria, que permite o cadastro de pacientes e seus respectivos exames, com o objetivo de agilizar e sistematizar o atendimento do fonoaudiólogo em seu ambiente de trabalho. O sistema utilizou o padrão frontend para fazer a interação com o usuário e plotagem de gráficos e backend para gerenciar o banco de dados, foi desenvolvido utilizando a linguagem JavaScript com uso de bibliotecas como ReactJS e NodeJS.

O sistema passou por diversos testes no decorrer do desenvolvimento, realizados pelo desenvolvedor e product owner, para validar a entrega das funcionalidades e comportamento do sistema, atestando as expectativas projetadas para o software. Será necessário agora uma avaliação com a cliente para atestar o produto final gerado com o desenvolvimento deste trabalho, para que de fato o sistema entre em produção e seja aplicado no ambiente de atendimento do profissional fonoaudiólogo.

Para o desenvolvimento deste projeto o curso de ciência da computação contou com diversas disciplinas que foram fundamentais para a compreensão e elaboração do sistema desenvolvido assim como para o manuseio das tecnologias utilizadas, disciplinas estas como Introdução aos Algoritmos, Estrutura de Dados, Sistemas Gerenciadores de Banco de Dados, Engenharia de Software, Interação Humano-Computador, entre outras.

Com os obstáculos vencidos no decorrer do desenvolvimento deste trabalho observa-se possibilidades para trabalhos futuros, entre elas pode-se destacar a implementação de funcionalidades como a possibilidade de gerar um arquivo PDF com o histórico de exames do paciente, autenticação de usuário para uso de múltiplos profissionais. A criação de mecanismos de segurança de software para a proteção dos dados contidos no sistema é um recurso a ser desenvolvido em trabalhos futuros e se faz necessário para a possibilidade deste sistema se tornar comercial. Outro trabalho futuro poderia ser a criação de uma API integrada ao sistema para gerenciamento de consultas dos pacientes cadastrados, podendo fornecer algumas informações ao paciente como seu relatório de exames. Assinatura digital, relatórios de atendimento, incorporação de exames são outras funcionalidades que podem ser desenvolvidas futuramente.

REFERÊNCIAS

- GOOGLE. **Gráficos interativos para navegadores e dispositivos móveis**. Lavras, 2023. Disponível em: <<https://developers.google.com/chart?hl=pt-br>>. Acesso em: 28 Jan. 2023.
- ILYA KANTOR. **The Modern JavaScript Tutorial**. Lavras, 2023. Disponível em: <<https://javascript.info/>>. Acesso em: 28 Jan. 2023.
- META PLATFORMS. **React - Uma biblioteca JavaScript para criar interfaces de usuário**. Lavras, 2023. Disponível em: <<https://pt-br.reactjs.org/>>. Acesso em: 03 Jan. 2023.
- MILETTO, E. M.; BERTAGNOLLI, S. de C. **Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP-Eixo: Informação e Comunicação-Série Tekne**. [S.l.]: Bookman Editora, 2014.
- MOZILLA. **CSS**. Lavras, 2022. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 03 Jan. 2023.
- MOZILLA. **HTML: Linguagem de Marcação de Hipertexto**. Lavras, 2022. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 03 Jan. 2023.
- MOZILLA. **Regras do CSS**. Lavras, 2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/CSS_basics>. Acesso em: 03 Jan. 2023.
- OPUS SOFTWARE. **Node.js – O que é, como funciona e quais as vantagens**. Lavras, 2023. Disponível em: <<https://www.opus-software.com.br/node-js/>>. Acesso em: 03 Jan. 2023.
- ORACLE. **MySQL Documentation**. Lavras, 2023. Disponível em: <<https://dev.mysql.com/doc/>>. Acesso em: 02 Fev. 2023.
- TREINAWEB. **5 motivos para estudar React**. Lavras, 2023. Disponível em: <<https://www.treinaweb.com.br/blog/5-motivos-para-estudar-react>>. Acesso em: 03 Fev. 2023.
- TREINAWEB. **Conheça o React, biblioteca para desenvolvimento Web**. Lavras, 2023. Disponível em: <<https://www.treinaweb.com.br/blog/conheca-o-react-biblioteca-para-desenvolvimento-web>>. Acesso em: 28 Jan. 2023.
- W3C. **HTML5: Curso W3C Escritório Brasil**. Lavras, 2023. Disponível em: <<https://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>>. Acesso em: 28 Jan. 2023.
- ZABRISKIE. **Cliente HTTP baseado em promessas para o navegador e node.js**. Lavras, 2023. Disponível em: <<https://axios-http.com/ptbr/>>. Acesso em: 28 Jan. 2023.