

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE GOIÁS - CÂMPUS JATAÍ**

DEPARTAMENTO DE ÁREAS ACADÊMICAS

**CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

Elizeu Oliveira da Silva

20172020230012

**REVERBERA – *SITE* DISPONIBILIZADOR DE
AUDIOLIVROS COM ACESSIBILIDADE VOLTADA PARA
DEFICIENTES VISUAIS**

Jataí

2022



INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
SISTEMA INTEGRADO DE BIBLIOTECAS

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

- | | |
|--|---|
| ☐ Tese | ☐ Artigo Científico |
| ☐ Dissertação | ☐ Capítulo de Livro |
| ☐ Monografia – Especialização | ☐ Livro |
| ☒ TCC - Graduação | ☐ Trabalho Apresentado em Evento |
| ☐ Produto Técnico e Educacional - Tipo: _____ | |

Nome Completo do Autor: Elizete Oliveira da Silva

Matrícula: 20172020230012

Título do Trabalho: Resenha: Site disponibilizador de audiolivros com acessibilidade rellada para deficientes visuais

Autorização - Marque uma das opções

- ☒ Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
- ☐ Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data ____/____/____ (Embargo);
- ☐ Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

- ☐ O documento está sujeito a registro de patente.
☐ O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
☐ Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

- o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Palma - Goiás, 26 / 04 / 23
Local Data

Elizete Oliveira da Silva

Assinatura do Autor e/ou Detentor dos Direitos Autorais

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE GOIÁS - CÂMPUS JATAÍ**

DEPARTAMENTO DE ÁREAS ACADÊMICAS

**CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

Elizeu Oliveira da Silva

RELATÓRIO TÉCNICO

**REVERBERA – *SITE* DISPONIBILIZADOR DE
AUDIOLIVROS COM ACESSIBILIDADE VOLTADA PARA
DEFICIENTES VISUAIS**

Relatório apresentado como requisito para obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistemas pelo
Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Orientador: Prof. Me. Danillo Vaz Borges de Assis

Jataí

2022

Dados Internacionais de Catalogação na Publicação na (CIP)

Silva, Elizeu Oliveira da.

Reverbera: site disponibilizador de audiolivros com acessibilidade voltada para deficientes visuais [manuscrito] / Elizeu Oliveira da Silva.

-- Jataí: IFG, Coordenação dos cursos de Informática – Tecnologia em Análise e Desenvolvimento de Sistemas, 2022.

73 f.; il.

Orientador: Prof. Me. Danillo Vaz Borges de Assis.

Bibliografias.

Apêndices.

1. Acessibilidade. 2. Web. 3. Deficiência visual. 4. Site. Tecnologias assistivas. I. Assis, Danillo Vaz Borges. II. IFG, Câmpus Jataí. III. Título.

Ficha catalográfica elaborada pela Seção Téc.: Aquisição e Tratamento da Informação.
Bibliotecária – Rosy Cristina Oliveira Barbosa – CRB 1/2380 – Campus Jataí. Cód. F025/2023-1.

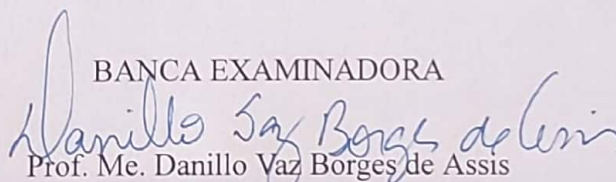
ELIZEU OLIVEIRA DA SILVA

**REVERBERA – SITE DISPONIBILIZADOR DE AUDIOLIVROS COM
ACESSIBILIDADE VOLTADA PARA DEFICIENTES VISUAIS**

Trabalho de Conclusão de Curso apresentado a
Coordenação do Curso de Tecnologia em Análise e
Desenvolvimento de Sistemas do Instituto Federal de
Educação, Ciência e Tecnologia de Goiás – Câmpus Jataí,
como parte dos requisitos para a obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistemas

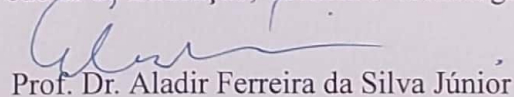
Relatório Técnico defendida(o) e aprovada(o), em 15 de março de 2023, pela banca
examinadora constituída pelos seguintes membros:

BANCA EXAMINADORA


Prof. Me. Danillo Vaz Borges de Assis

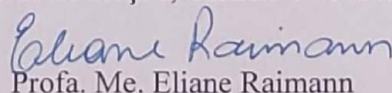
Presidente da banca / Orientador(a)

Instituto Federal de Educação, Ciência e Tecnologia de Goiás


Prof. Dr. Aladir Ferreira da Silva Júnior

Membro

Instituto Federal de Educação, Ciência e Tecnologia de Goiás


Profa. Me. Eliane Raimann

Membro

Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Jataí – GO

2023

RESUMO

O presente relatório aborda como foi desenvolvida uma plataforma *online* com acessibilidade para atender o público de pessoas com deficiências visuais no que tange o acesso ao mundo literário por meio de audiolivros, versão em áudio narrada de conteúdos de livros. Através de pesquisa bibliográfica e de exemplos de aplicações práticas, foi percebida a carência de *sites* que disponibilizam audiolivros de maneira gratuita em um ambiente digital que dê suporte à ferramentas assistivas, tecnologias que garantem o acesso destes usuários no meio *web*. Diante desta problemática, foi iniciado o desenvolvimento do *site* Reverbera, utilizando da metodologia ágil *Personal Scrum* e seus artefatos correspondentes juntamente com um estudo acerca da aplicação de recursos assistivos no desenvolvimento de uma navegação acessível voltada ao uso de leitores de tela, tecnologia alvo do trabalho. A plataforma finalizada contou com um reprodutor de áudio personalizado e acessível, recursos que favorecem a navegação como: controle de tamanho textual e modo de alto contraste para usuários com baixa visão, além de um painel administrativo para cadastro de audiolivros.

Palavras-chave: Acessibilidade, Web, Deficiência Visual, Site, Tecnologias Assistivas.

LISTA DE ABREVIATURAS E SIGLAS

BEM	Elemento Modificador de Bloco (<i>Block Element Modifier</i>)
CSS	Folhas de Estilo em Cascata 3 (<i>Cascading Style Sheets 3</i>)
HTML	Linguagem de Marcação de Hipertexto 5 (<i>Hypertext Markup Language 5</i>)
IDE	Ambiente de Desenvolvimento Integrado (<i>Integrated Development Environment</i>)
PcD	Pessoa com deficiência
TA	Tecnologias assistivas
WAI-ARIA	Iniciativa de Acessibilidade na <i>Web</i> para Aplicações Ricas e Acessíveis (<i>Web Accessibility Initiative for Accessible Rich Internet Applications</i>)
WCAG 2.0	Diretrizes para Conteúdos de Acessibilidade <i>Web</i> 2.0 (<i>Web Content Accessibility Guidelines 2.0</i>)
W3C	World Wide Web Consortium

LISTA DE FIGURAS

Figura 1 – Demonstração da aplicação de um atributo WAI-ARIA	11
Figura 2 – Demonstração da modificação um atributo WAI-ARIA	12
Figura 3 – Diagrama de Casos de uso desenvolvido no software Astah	13
Figura 4 – Mind Map desenvolvido no software Xmind Zen	14
Figura 5 – Divisão do Product Backlog no Trello	15
Figura 6 – Divisão do Sprint Backlog no Trello	16
Figura 7 – Detalhes da tarefa no Trello	17
Figura 8 – Logo do Site	17
Figura 9 – Diagramas demonstrando a variação da percepção de cores de acordo com alguns tipos de daltonismo	18
Figura 10 – Exemplo de tag do HTML com seus atributos e respectivos valores	19
Figura 11 – Seletor CSS aplicando estilos a um título	20
Figura 12 – Exemplo da aplicação das nomenclaturas de classes conforme a <i>BEM</i>	21
Figura 13 – Conteúdo da <i>tag head</i> do cabeçalho	23
Figura 14 – Trecho de código do cabeçalho	24
Figura 15 – Trecho de código do menu de categorias	25
Figura 16 – Aparência no <i>front end</i> do menu de categorias	26
Figura 17 – Funções em javascript do gerenciamento de fonte	27
Figura 18 – Função em javascript de verificação do modo de alto contraste	28
Figura 19 – Exemplo de página com o modo de alto contraste ativado	29
Figura 20 – Trecho de inicialização dos <i>players</i>	31
Figura 21 – Trecho de seleção dos elementos com o recurso <code>querySelector</code>	32
Figura 22 – Trecho da inicialização da barra de tempo	33
Figura 23 – Trecho da inicialização da barra de tempo	33
Figura 24 – Trecho da função <code>playOrPause</code>	34
Figura 25 – Conteúdo da função <code>update</code>	36
Figura 26 – Conteúdo da função <code>clickedTimeBar</code>	37
Figura 27 – Conteúdo da função <code>enhanceAudioTips</code>	38
Figura 28 – Visão de parte do painel administrativo do Wordpress	39
Figura 29 – Visão superficial de alguns dos diretórios do Wordpress	40

Figura 30 – Conteúdo do arquivo “style.css”	40
Figura 31 – Conteúdo da função responsável pelo registro da taxonomia	41
Figura 32 – Figura 31 – Grupo de campos do <i>post type</i> Audiolivro	42
Figura 33 – Trechos da integração do arquivo “header.php”	44
Figura 34 – Trechos demonstrando a inserção de <i>links</i> através do laço de repetição	44
Figura 35 – Exemplo de um <i>breadcrumb</i> so site Reverbera	45
Figura 36 – Variáveis da <i>single</i> de audiolivro	46
Figura 37 – Parte da <i>single</i> de audiolivro	47
Figura 38 – Parte da <i>single</i> de autor	49
Figura 39 – Parte da <i>archive</i> de audiolivro	50
Figura 40 – Conteúdo da página 404	51
Figura 41 – Função para ocultar a barra administrativar	51
Figura 42 – Função registerScripts	52

SUMÁRIO

1. INTRODUÇÃO	7
2. DESENVOLVIMENTO	10
2.1. APLICABILIDADE DA ACESSIBILIDADE VISUAL NO SITE REVERBERA	10
2.2. METODOLOGIA	12
2.3. DESENVOLVIMENTO DO PRODUTO	17
2.3.1. CRIAÇÃO DA IDENTIDADE DO SITE	17
2.3.2. DESENVOLVIMENTO DO FRONT END	19
2.3.2.1. Ferramentas utilizadas e linguagens	19
2.3.2.2. Criação das páginas	22
2.3.2.3. Desenvolvimento dos recursos assistivos	26
2.3.3. DESENVOLVIMENTO DO BACK END	38
2.3.3.1. Criação das estruturas do CMS	40
2.3.3.2. Integração	43
3. CONSIDERAÇÕES FINAIS	53
REFERÊNCIAS	56
APÊNDICE A – MEMORIAL DESCRITIVO	57
APÊNDICE B – DIAGRAMA DE CASOS DE USO	58
APÊNDICE C – MIND MAP	59
APÊNDICE D – MANUAL DE INSTALAÇÃO	60
APÊNDICE E – MANUAL DE UTILIZAÇÃO	68
APÊNDICE F – AVALIAÇÃO DE ACESSIBILIDADE	70

1. INTRODUÇÃO

Do total da população brasileira, quase 24% (45,6 milhões de pessoas) declararam ter algum tipo de deficiência, sendo a deficiência visual a mais comum, atingindo 3,5% da população (CENSO DEMOGRÁFICO, 2010). Além disso, estudos revelam que a cada 5 segundos uma pessoa se torna cega no mundo e que o número de deficientes visuais tende a dobrar até o ano de 2020 (OLIVEIRA *et al.*, 2016). Segundo o Instituto Benjamin Constant (2007):

Considera-se Pessoa com Deficiência (PcD) aquela que apresente, em caráter permanente, perdas ou reduções de sua estrutura, ou função anatômica, fisiológica, psicológica ou mental, que gerem incapacidade para certas atividades, dentro do padrão considerado normal para o ser humano. A deficiência visual é a perda ou redução de capacidade visual em ambos os olhos em caráter definitivo, que não possa ser melhorada ou corrigida com o uso de lentes, tratamento clínico ou cirúrgico. Existem também pessoas com visão subnormal, cujos limites variam com outros fatores, tais como: fusão, visão cromática, adaptação ao claro e escuro, sensibilidades a contrastes, etc. (INSTITUTO BENJAMIN CONSTANT, 2007).

Autores como Menezes e Ribeiro (2008) destacam que:

Ler, escrever, conhecer, interpretar, são atividades comuns para grande parcela da população mundial. Porém, amplo é o número de pessoas impossibilitadas de realizar este simples gesto físico de forma prática, como os deficientes visuais. (MENEZES; RIBEIRO (2008, p. 61)

A formação do indivíduo se estabelece em grande parte por meio da leitura, que o influencia a pensar analisando o seu dia-a-dia, a sociedade e, de modo particular, ampliando e diversificando suas visões sobre o mundo, com relação à vida sobre si mesma. É por meio do texto que se adquire e formata-se posicionamentos, sendo assim possível formar as opiniões individuais (KRUG, 2015).

O acesso principal à leitura de obras literárias por parte destas pessoas se dá por meio de materiais em Braille¹, que possuem baixa produção e variedade devido ao complexo tratamento para transcrição dos livros à tinta (MENEZES; RIBEIRO, 2008).

¹ O Braille é um sistema que foi oficializado em 1852 para possibilitar que pessoas com deficiência visual, parcial ou total, tivessem acesso à leitura. Todo o sistema é formado por caracteres em relevo que permitem o entendimento por meio do tato.

Como alternativa aos livros em Braille, surgiram os *audiobooks* ou audiolivros, que, segundo Menezes e Ribeiro (2008, p. 61), são suas versões faladas, de maneira interpretada utilizando sonorização a fim de transmitir “[...] sentimentalismo [...]”. Eles possuem benefícios importantes como a possibilidade de inserção do indivíduo deficiente visual no mundo literário mesmo que ainda não tenha aprendido o Braille, além de ocuparem menos espaço físico ou nenhum em relação aos exemplares escritos (MENEZES; RIBEIRO, 2008; OLIVEIRA et al., 2016).

O surgimento dos *audiobooks* no Brasil ocorreu na década de 70 e sua distribuição física é feita gratuitamente através de instituições filantrópicas. Porém, no meio *online*, grande parte dos exemplares são fornecidos por *sites* de vendas cujo público não são deficientes visuais (MENEZES; RIBEIRO, 2008).

O uso da tecnologia permitiu a saída de parte das PcD - Pessoas com deficiência, termo que será referido como PcD de agora em diante, da exclusão social e digital, que perdurou por um longo período, dos deficientes visuais (MENEZES; RIBEIRO, 2008). Portanto, o surgimento de novos aplicativos para *smartphones* e computadores em geral contribui para que tal fenômeno continue ocorrendo (SANTOS et al., 2017). Devido a esta revolução no campo da acessibilidade, as ferramentas que dispõem dessa característica são conhecidas como tecnologias assistivas - expressão que doravante será referenciada como Tecnologias Assistivas, doravante denominada neste trabalho com a sigla TA.

TA é o termo utilizado para recursos ou serviços cujo intuito é facilitar as tarefas cotidianas de deficientes, proporcionando o aumento de suas capacidades e a conquista de sua autonomia. Nesse contexto, alguns exemplos desse tipo de equipamento são os ampliadores e leitores de tela, softwares de reconhecimento de voz e teclados alternativos, que são tecnologias voltadas para o acesso à *web* (MELO et al., 2009).

A utilização do computador associado a um dispositivo ou software leitor de tela é essencial para a inclusão de um usuário deficiente visual no meio digital (PANSANATO; RODRIGUES; SILVA, 2016). Conforme Melo et al. (2009), os leitores de tela possuem a capacidade de ler informações textuais via sintetizadores de voz.

Uma outra preocupação a ser empreendida no sentido da acessibilidade está no uso correto dos recursos de criação de páginas *web*, conhecidos por *Web Standards*, os quais constituem a base para a acessibilidade na *web*. O ato de ignorá-los pode implicar em dificuldades na navegação via tecnologias assistivas, tais como fazer o leitor de tela ler de

maneira desenfreada ou permanecer mudo pelo excesso de elementos gráficos na página (MELO *et al.*, 2009; PANSANATO; RODRIGUES; SILVA, 2016).

A acessibilidade na Web diz respeito a habilitar qualquer pessoa, usando qualquer tecnologia adequada à navegação Web, a visitar qualquer website, obter a informação oferecida e interagir com o website (DIAS, 2003; HULL, 2004 apud MELO *et al.*, 2009, p. 102).

Foram identificados alguns *sites* nacionais relacionados à proposta como: Ubook², LivrosNarrados³, Universidade Falada⁴ e Tocalivros⁵. Observou-se que alguns não são totalmente gratuitos, pois exigem assinatura, não possuem navegação totalmente acessível, e não oferecem opções de visualização como ampliação de caracteres e alto contraste para pessoas com baixa visão. Além disso, os aplicativos chamados de *players* de áudio existentes em alguns desses *sites* são desprovidos de acessibilidade nos controles, tais como, tocar ou pausar, aumentar ou diminuir volume e avançar ou retroceder na faixa de áudio.

A partir das informações até aqui apresentadas, percebeu-se a carência de uma plataforma *online* que forneça audiolivros de maneira gratuita e cuja navegação seja plenamente acessível através dos leitores de tela.

Para tentar amenizar tal carência, o objetivo deste trabalho foi desenvolver um sistema para a plataforma *web* com finalidade de disponibilização de audiolivros, voltado ao público deficiente visual, utilizando boas práticas de acessibilidade segundo os *Web Standards*.

A fim de alcançar esses objetivos foi necessário cumprir os seguintes passos: I) Compreender o contexto do desenvolvimento de *softwares* utilizando tecnologias assistivas; II) Entender e aplicar os recursos de acessibilidade web; III) Contribuir de maneira a facilitar o acesso a audiolivros; IV) Desenvolver um software capaz de dar às PcD condições de acesso à conteúdos adaptados à suas necessidades;

Utilizou-se como metodologia para o desenvolvimento deste produto, a *Personal Scrum*, uma variação da metodologia ágil Scrum, que segundo Costa (2016), é sugerida no desenvolvimento de trabalhos individuais cuja duração é previamente conhecida.

² Endereço eletrônico: <https://www.ubook.com/>.

³ Endereço eletrônico: <http://livrosnarrados.blogspot.com/>.

⁴ Endereço eletrônico: <https://www.universidadefalada.com.br/>.

⁵ Endereço eletrônico: <https://www.tocalivros.com/>

Nas seções seguintes serão abordados os aspectos gerais do desenvolvimento como a aplicação da metodologia e o uso dos seus artefatos, a criação da logo do software, o uso da acessibilidade no desenvolvimento da plataforma. Também será apresentado o detalhamento do desenvolvimento das etapas de codificação da plataforma, tanto no *front end* quanto no *back end*.

2. DESENVOLVIMENTO

Neste capítulo serão abordadas as etapas necessárias para que houvesse a criação do produto em toda a sua totalidade.

2.1. APLICABILIDADE DA ACESSIBILIDADE VISUAL NO SITE REVERBERA

A W3C possui duas grandes diretrizes a respeito do desenvolvimento de páginas *web* com acessibilidade que são: a WCAG 2.0 - Diretrizes para Conteúdos de Acessibilidade *Web* 2.0 (*Web Content Accessibility Guidelines 2.0*), e a WAI-ARIA - Iniciativa de Acessibilidade na *Web* para Aplicações Ricas e Acessíveis (*Web Accessibility Initiative for Accessible Rich Internet Applications*). Elas consistem em orientações para que os envolvidos na produção de um *site* com acessibilidade possam se guiar. A primeira documentação tem como objetivo enumerar padrões a serem seguidos com relação ao conteúdo, já a segunda visa informar sobre técnicas e padrões a serem seguidos no desenvolvimento do código-fonte do site a fim de se alcançar uma usabilidade e acessibilidade satisfatórias para usuários com necessidades especiais.

O World Wide Web Consortium (W3C) é um consórcio internacional em que organizações filiadas, uma equipe em tempo integral e o público trabalham juntos para desenvolver padrões para a web. O W3C já publicou mais de cem padrões, como HTML, CSS, RDF, SVG e muitos outros. Todos os padrões desenvolvidos pelo W3C são gratuitos e abertos, visando garantir a evolução da web e o crescimento de interfaces interoperáveis. (W3C BRASIL).

A WCAG 2.0 divide as suas orientações em quatro grandes princípios: I) Perceptível; II) Operável; III) Compreensível; IV) Robusto. Cada um deles aborda o tipo de conteúdo e estabelece como este deve ser apresentado para que qualquer usuário possa compreendê-lo e o princípio seja cumprido.

O princípio Perceptível trata de requisitos para que as informações e os componentes de interface possam ser compreendidos, tais como fornecimento de alternativas de conteúdo não-textual para textuais, variações na apresentação do conteúdo sem que este perca suas informações e a clara distinção das percepções sensoriais em conteúdos não textuais.

O princípio Operável estabelece que todos componentes da interface do utilizador devem ser operados através do teclado. Além disso, estabelece padrões para lidar com conteúdos que envolvam espaço temporal, orienta como criar conteúdos que não ocasionam convulsões no usuário e informa sobre requisitos de navegação.

O princípio Compreensível trata da interação do usuário com as interfaces da página fazendo recomendações acerca do comportamento dos elementos. Já o princípio Robusto visa atentar o leitor quanto a necessidade de compatibilidade do *site* com os diversos tipos de tecnologias assistivas. Neste trabalho o WCAG 2.0 contribuiu com as recomendações de conteúdo voltadas às deficiências visuais. O WAI-ARIA teve um impacto mais significativo no desenvolvimento de funcionalidades.

A diretriz WAI-ARIA surgiu com a necessidade de melhorar a percepção das páginas *WEB* pelas tecnologias assistivas, em especial os leitores de tela, pelo fato de alguns elementos HTML - Linguagem de Marcação de Hipertexto 5 (*Hypertext Markup Language 5*) que será referida como HTML de agora em diante, quando utilizados para criação de componentes mais complexos não ofertarem uso acessível com estas tecnologias. Desta forma ela dá instruções de como utilizar seus componentes para melhorar a acessibilidade de uma aplicação.

Os componentes atuam como uma extensão do HTML nativo (figura 1) fazendo com que o leitor de tela seja capaz dar ao usuário informações mais fiéis sobre os elementos da página. Isto ocorre pelo fato de que os componentes desta extensão são atributos inseridos nas *tags*, componentes primários da escrita de elementos na linguagem HTML, para agregar mais semântica ao elemento. Estes atributos possuem o termo “*aria*” como componente de sua nomenclatura e possuem três tipos de classificação: Papel, Estado e Propriedade.

Figura 1 – Demonstração da aplicação de um atributo WAI-ARIA

```
<button tabindex="0" title="Ligar alto contraste" aria-pressed="false"
class="c__acessible-button c__contrast-button c__transition-300"
```

Fonte: Autoria Própria

Os atributos do tipo Papel geralmente são inseridos na *tag* com o nome *role* e tem como função sobrescrever a semântica natural da *tag*, modificando a forma como um dispositivo leitor de tela irá identificá-la. Um botão com este atributo com o valor *link*, por exemplo, será identificado pela tecnologia assistiva como link. Enquanto este grupo modifica a semântica para um valor fixo, os atributos do tipo Estado e Propriedade sugerem uma semântica dinâmica que pode ser modificada através de javascript conforme figura 2.

Figura 2 – Demonstração da modificação um atributo WAI-ARIA

```
volumeControlBar.setAttribute("aria-valuenow", volumeValue);  
volumeControlBar.setAttribute("aria-valuetext", volumeText);
```

Fonte: Autoria Própria

Dentro da diretriz WAI-ARIA existem diversas documentações e para o desenvolvimento do produto foi utilizada a *WAI-ARIA Authoring Practices* por se tratar de um guia mais direto e com o foco exclusivo no desenvolvimento de aplicações. A abordagem de seu uso será detalhada em capítulos posteriores.

2.2. METODOLOGIA

Por se tratar de um produto tecnológico voltado para a *web*, foi necessária para a sua execução uma abordagem fluida, que concilie a pesquisa bibliográfica com o desenvolvimento do software de maneira ágil.

Soares (2004, p. 7) afirma que a *web* é “[...] um ambiente de desenvolvimento dinâmico e com mudanças constantes, as metodologias tradicionais orientadas a documentação são menos adequadas que as metodologias ágeis [...]”. Dadas essas características, utilizou-se a metodologia Scrum Individual (do inglês, *Personal Scrum*), uma variação da metodologia ágil SCRUM para delinear o processo de desenvolvimento do software deste trabalho.

No *Personal Scrum* o indivíduo adota todas responsabilidades envolvendo os papéis relacionados ao SCRUM - *Product owner*, *Scrum master* e *Delepomment Team* sendo mantidos os artefatos estabelecidos originalmente - *Product backlog* e *Sprint backlog*. A principal diferença entre o SCRUM e o PERSONAL SCRUM reside na duração dos eventos, tais como *Sprints* de uma semana, por exemplo (COSTA, 2016).

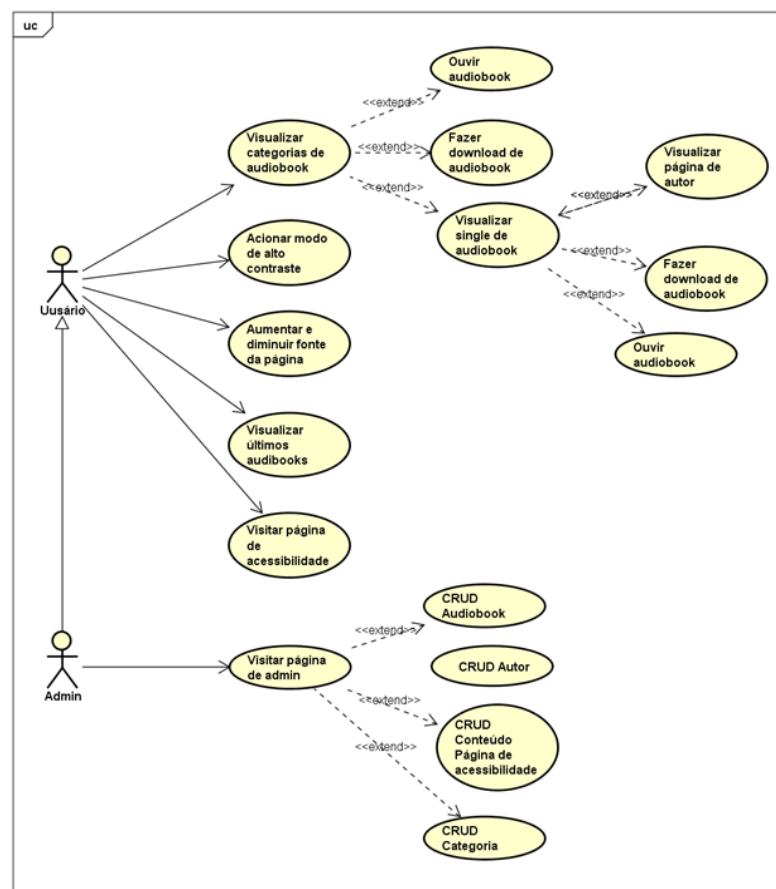
É caracterizada como *Product backlog* a lista de funcionalidades que devem ser implementadas no software e *Sprint backlog* o conjunto dessas funcionalidades a serem realizadas em uma *Sprint*, período de uma semana (PAGOTTO *et al.*, 2016).

A aplicação da metodologia iniciou-se com a criação da *Requeriment*, que segundo Pagotto (2016), é a definição do escopo do produto, tendo como artefatos a Planta de Desenvolvimento e o *Product Backlog*.

A Planta de Desenvolvimento abrangeu a criação de um Memorial Descritivo (Apêndice A), para definir-se os requisitos do trabalho, Diagrama de Casos de Uso (Apêndice B) e um *Mind Map* (Apêndice C), documento que define visualmente a estrutura de um *site*.

No Memorial Descritivo foram estabelecidas as funcionalidades assistivas, estruturas essenciais e informações a serem exibidas no produto. Tais descrições possibilitaram a elaboração do Diagrama de Casos de Uso, criado a partir da ferramenta de criação de diagramas UML denominada *Astah*. Neste documento foram definidos dois atores: usuário comum e administrador e atribuídos os seus respectivos casos de usos, conforme a Figura 3.

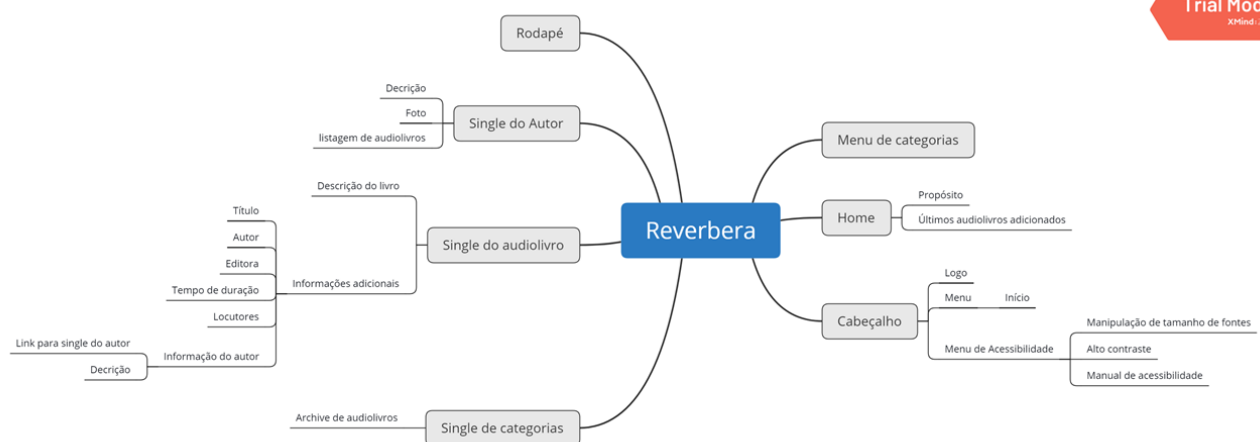
Figura 3 – Diagrama de Casos de uso desenvolvido no software Astah



Fonte: Autoria Própria

Após esta etapa de criação dos casos de uso, desenvolveu-se o *Mind Map*, utilizando como ferramenta o *software Xmind Zen*. A Figura 4 demonstra como foram descritas as informações concernentes a elementos internos de estruturas do *site* como: cabeçalho, menus, conteúdos de páginas e rodapé.

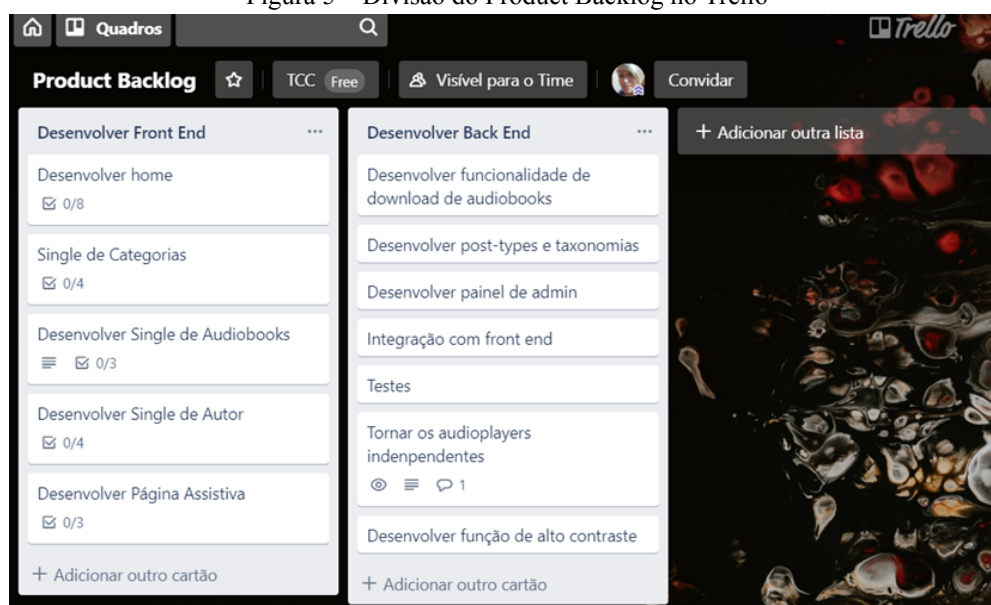
Figura 4 – Mind Map desenvolvido no software Xmind Zen



Fonte: Autoria Própria

Com o estabelecimento do *Mind Map*, concluiu-se a etapa de criação da planta de desenvolvimento e deu-se início ao controle de atividades a partir do desenvolvimento do *Product Backlog*, tendo como auxiliar a ferramenta *online* Trello. Ela foi escolhida por se tratar de um ambiente de divisão de tarefas que utiliza o modelo *Kanban* que, conforme Costa (2016) é sugerida para definir as datas iniciais e finais, além das tarefas dos *sprints* e itens do *product backlog*. Dividiu-se as atividades em dois grandes grupos: “Desenvolver *Front End*” e “Desenvolver *Back End*” conforme a Figura 5.

Figura 5 – Divisão do Product Backlog no Trello

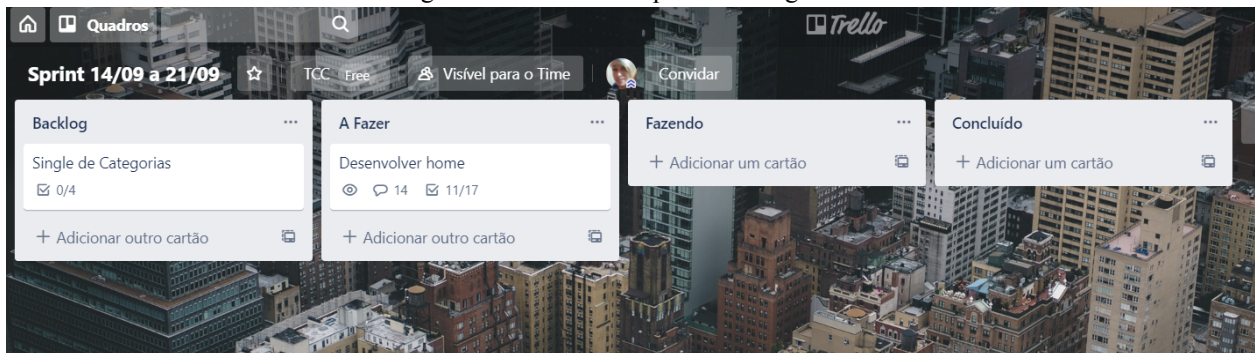


Fonte: Autoria Própria

A divisão das tarefas em grupos de atividades correlacionados surgiu com a necessidade de uma melhor gestão das tarefas conforme o seu tipo. O grupo *Desenvolver Front End* englobou todas as etapas necessárias para o desenvolvimento da interface do site. Já o grupo *Desenvolver Back End* se referiu à organização das etapas referentes ao funcionamento interno da plataforma. Após este processo, foram movidas as tarefas do Trello para a sua execução em uma *Sprint* (período de 7 dias) para um *Sprint Backlog*.

Cada *Sprint Backlog* compreendeu em um quadro diferente de tarefas que possuía como nome *Sprint*, acrescido do período inicial e final deste espaço de tempo. Além disso, o quadro possui divisões que correspondem a *status* da tarefa (vide figura 6) que são: I) *Backlog* - local onde são inseridas todas as tarefas movidas do *Product Backlog*; II) *A fazer* - local onde a tarefa em execução fica quando é pausada; III) *Fazendo* - local que comporta as tarefas sendo executadas naquele momento; IV) *Concluído* - localidade onde todas as tarefas finalizadas naquele *Sprint* foram alocadas.

Figura 6 – Divisão do Sprint Backlog no Trello

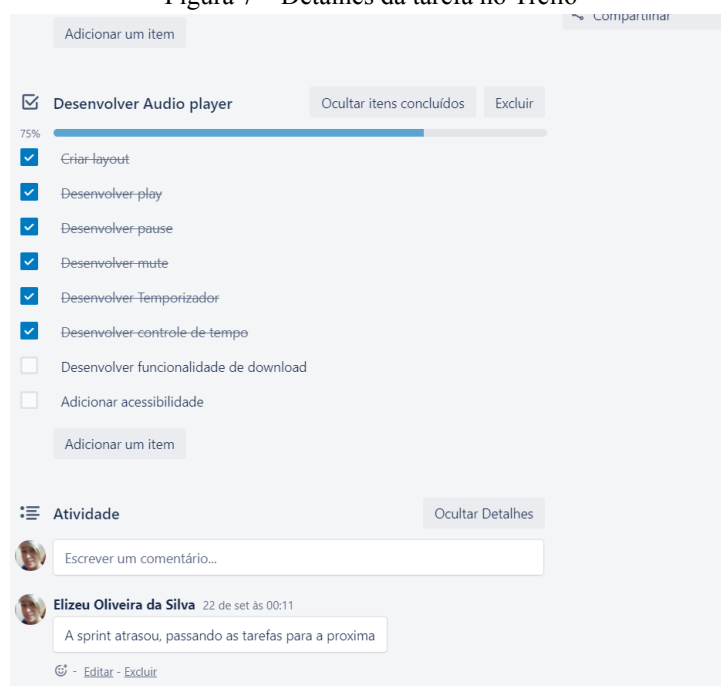


Fonte: Autoria Própria

O *Trello* possibilitou a inserção de detalhes importantes dentro de cada tarefa como as *User Stories* correspondentes e comentários acerca da execução do processo (figura 7), além de registrar automaticamente o tempo em que ocorreu uma mudança de *status* da tarefa, desta forma foi possível a gestão eficiente das tarefas através do uso desta ferramenta.

As tarefas que não puderam ser concluídas dentro da *Sprint* na qual foram desenvolvidas eram clonadas e realocadas para um novo *Sprint* Backlog, desta forma evitando-se a perda de registros importantes. Esta etapa das *Sprints* consistiu no processo final da metodologia, que compreendeu na execução de tarefas dentro de um período específico. Na seção seguinte será descrito o processo de compreensão da implementação focada em acessibilidade e como impactou no desenvolvimento do produto.

Figura 7 – Detalhes da tarefa no Trello



Fonte: Autoria Própria

2.3. DESENVOLVIMENTO DO PRODUTO

Todo o código fonte foi escrito na língua inglesa por se tratar de uma linguagem universal e ser uma boa prática de programação. O desenvolvimento do produto compreendeu-se em três etapas:

I) Criação da identidade visual do site; II) Desenvolvimento do *front end*; III) Desenvolvimento do *back end* e integração com o *front end*;

Nas seções seguintes será abordado cada aspecto destas etapas e seu detalhamento.

2.3.1. CRIAÇÃO DA IDENTIDADE DO SITE

Uma página *web* bem estruturada possui uma identidade visual com cores predefinidas que desta forma proporcionam uma melhor usabilidade. No caso do produto deste trabalho, as cores foram baseadas na logo (figura 8).

Figura 8 – Logo do Site

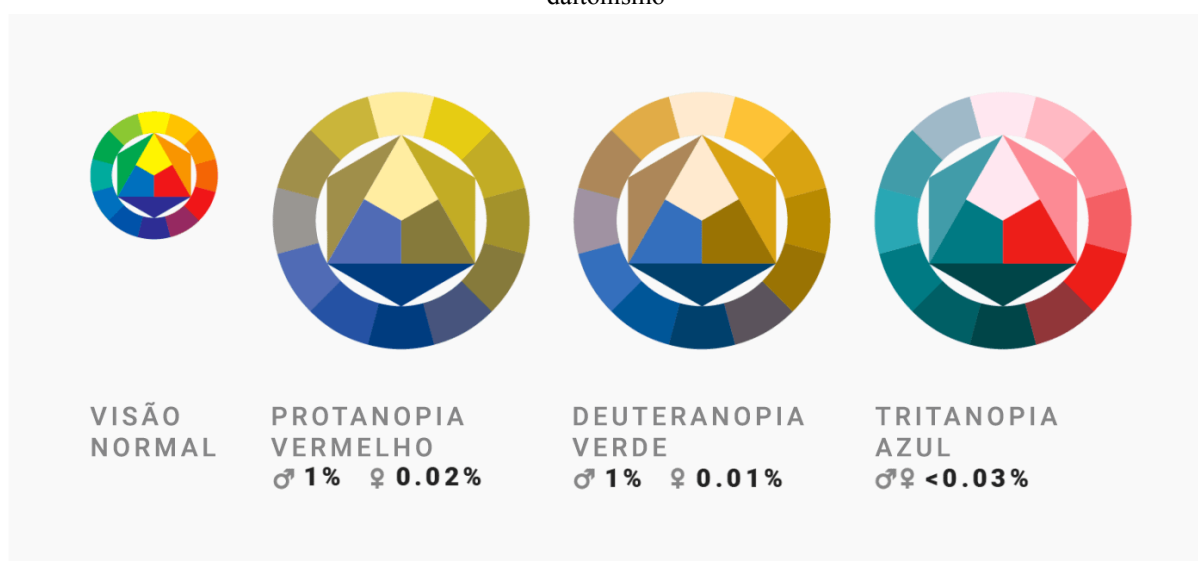


Fonte: Autoria Própria

A logo foi desenvolvida no software de edição gráfica Adobe Photoshop CC2019, desenvolvido pela Adobe, que foi escolhido pela familiaridade de uso do autor deste trabalho e pelos recursos de testes de cores para usuários daltônicos.

O daltonismo se caracteriza pela deficiência na percepção das cores, de tal forma com que o indivíduo enxergue um número de cores e tonalidades extremamente reduzido em comparação ao um indivíduo que não possua este problema. Embora existam diversas variações desta condição, de forma geral, os daltônicos possuem uma melhor percepção das cores azuis e tons de cinza conforme mostra a figura 9.

Figura 9 – Diagramas demonstrando a variação da percepção de cores de acordo com alguns tipos de daltonismo



DALTONISMOS TOTAIS (-ANOPIA)

Fonte: COMO PESSOAS DALTÔNICAS ENXERGAM AS CORES DE SEUS GRÁFICOS. Disponível em: <https://escoladedados.org/tutoriais/como-pecoas-daltonicas-enxergam-as-cores-de-seus-graficos/>. Acesso em: 02/11/2022

Em função da melhor percepção deste grupo de cores, foram escolhidos os azuis e cinza com os respectivos códigos de cor hexadecimais: #007dc9, #00001D e #E5E5E5 e o símbolo do logotipo representa um leitor recebendo o conhecimento de um livro através do som. Outro aspecto da criação da identidade do site foi a elaboração de seu nome, desenvolvido de forma com que combinasse com o logotipo.

O nome Reverbera foi imaginado pensando no verbo reverberar que significa propagar ondas sonoras, espalhar algo, termo que condiz muito com o propósito da plataforma que é contribuir com a divulgação do conhecimento literário por meio de

audiolivros e arquivos de áudio. Com a finalização desta etapa, se tornou possível o início do desenvolvimento do *front end* do produto.

2.3.2. DESENVOLVIMENTO DO FRONT END

O termo *front end* representa o conjunto de elementos que compõem a interface visual do usuário e corresponde, no âmbito da *web*, ao desenvolvimento do *layout* das páginas e das funcionalidades responsáveis pela interação do usuário com os seus componentes, que é feito basicamente com linguagens de marcação próprias para esta finalidade, como o HTML e o CSS3 - Folhas de Estilo em Cascata 3 (*Cascading Style Sheets 3*), que a partir de agora será referido como CSS . Essas linguagens são processadas pelo navegador, também conhecido como *browser*, que é uma aplicação na qual o usuário utiliza para navegar na *WEB*.

Nas próximas seções serão abordadas essas linguagens que foram utilizadas no *front end*, bem como ferramentas empregadas no desenvolvimento, criação das páginas e elaboração das funcionalidades de acessibilidade.

2.3.2.1. Ferramentas utilizadas e linguagens

O HTML tem como propósito a estruturação do conteúdo das páginas, determinando os títulos, parágrafos, links e imagens, por exemplo, que são representados no código por meio de *tags* e que podem possuir propriedades com valores ou não, conhecidos como atributos, vide figura 10.

Figura 10 – Exemplo de tag do HTML com seus atributos e respectivos valores

```
<h1 id="mainTitle" class="c__title c__white-text c-title--main-title c__center">
```

Fonte: Autoria Própria

Já o CSS age na estilização das estruturas do HTML através da aplicação de regras que apontam quais elementos deverão ser alvo de um certo conjunto de estilos específicos, a exemplo na figura 11. Estas regras são conhecidas como seletores do CSS.

Figura 11 – Seletor CSS aplicando estilos a um título

```
h2 {  
    margin-top: 20px;  
    margin-bottom: 10px;  
    font-size: 2.4em;  
    line-height: 36px;  
    font-weight: 700;  
}
```

Fonte: Autoria Própria

Para organização do código de CSS foi utilizada a metodologia *BEM* - Elemento Modificador de Bloco (*Block Element Modifier*) que será referida como *BEM* de agora em diante, criada para otimizar a criação de classes, que são um tipo de atributo de *tags* utilizando-se o termo *class* e que serve para referenciar o elemento em *scripts*⁶ e folhas de estilo, termo dado a arquivos CSS. A *BEM* desempenha seu papel por meio de uma associação hierárquica que evidencia se o elemento é um bloco ou um componente de bloco, tornando a leitura de código mais simples e intuitiva. Este método facilita a criação de seletores nas folhas de estilo, pois a relação de nomes dos elementos é mais clara e objetiva.

A relação de nomes é feita da seguinte maneira:

I) Atribui-se o nome do elemento de acordo com a sua função no *layout* ou aquilo que ele representa, *navigation* para um elemento de navegação, por exemplo;

II) Se o elemento é pertencente a um componente, ou seja um filho, utiliza-se o conjunto de caracteres “__” após o nome do componente e adiciona-se uma nomenclatura que represente o filho em si, se estamos nos referindo a um link de uma navegação, o nome da classe seria: “navigation__link”;

III) A metodologia *BEM* também indica se a classe é uma modificação do estilo original do elemento, para isso é feita a junção do nome da classe do elemento mais os caracteres “--” e o nome que elucida a modificação de estilo, se queremos aplicar um link de navegação com a fonte maior, podemos nomear a classe modificadora como: “navigation__link--large”;

⁶ *Script* é um conjunto de instruções para que uma função seja executada em determinado aplicativo.

A aplicação da metodologia *BEM* foi empregada com a adição de uma modificação que especifica se o elemento é pertencente a um componente ou a uma lista de elementos, utilizando-se o prefixo “c-” ou “l-” respectivamente, e outra que determina se a classe é pertencente a um elemento que não tem componente pai através do prefixo “c__”, vide figura 12.

Figura 12 – Exemplo da aplicação das nomenclaturas de classes conforme a *BEM*



```
.l-flex--stretch {
  align-items: stretch;
}

.l__col-8 {
  width: 66%;
  padding: 12px;
}

.c__acessible-button {
  padding-right: 8px;
  padding-left: 8px;
  background-color: transparent;
  color: #00001d;
}
```

Fonte: Autoria Própria

Além das linguagens de marcação HTML e CSS, para a criação do *front end* foi utilizada a linguagem de programação javascript, que no âmbito da *web* tem como principal objetivo a associação dos elementos do *layout* à interações do usuário ou modificações de estado da página.

Para auxiliar na criação do código, utilizou-se a *IDE*⁷ PhpStorm, desenvolvida pela JetBrains, escolhida pela familiaridade do autor deste trabalho, por oferecer suporte a diversas linguagens e fornecer recursos que aumentam a produtividade no momento de desenvolvimento.

⁷ Do inglês *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo.

No acompanhamento do comportamento do *site* em leitores de tela foi utilizado principalmente a extensão de navegador *Chromevox*, um leitor de tela para o navegador *Chrome*, o *browser* principal usado para a execução do site na etapa de desenvolvimento.

O gerenciamento do versionamento de código foi feito através da ferramenta *GitKraken* que utiliza como base o *GIT*.

Com as ferramentas e linguagens de marcação e programação utilizadas para o desenvolvimento desta etapa já apresentadas, será abordado a seguir o detalhamento da criação das páginas antes do *back end* e integração.

2.3.2.2. Criação das páginas

Inicialmente foi feita a divisão de diretórios na qual a pasta raiz tem o nome *reverbera*, contendo a página inicial, nomeada por convenção como *index* e uma subpasta chamada *assets* que serviu para armazenar as pastas *css*, *js*, *fonts* e *images*. A pasta *css* tem como propósito guardar todos os arquivos de folha de estilos, a *js* armazena os arquivos de *javascript*, *fonts* as fontes de tipografia que neste projeto serviram para a inserção de ícones e, a pasta *images* para a separação das imagens.

O primeiro componente construído na *index* foi o cabeçalho onde foi inserida a *tag head*, responsável pela especificação de informações diversas para o navegador acerca da página carregada, conhecidas como *metatags* e representadas pela tag *meta*. Em seguida, foi referenciado o *favicon*, ícone que exibe a logo da plataforma na aba do navegador e em outros meios, por meio de diferentes formas para cada tipo de dispositivo de acesso e por fim, especificado o título com a *tag title*, vide figura 13.

Figura 13 – Conteúdo da *tag head* do cabeçalho

```
<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1" name="viewport">
  <meta name="msapplication-TileColor" content="#da532c">
  <meta name="theme-color" content="#ffffff">

  <!--   Favicon-->
  <link rel="apple-touch-icon" sizes="180x180"
    href="<? = get_template_directory_uri(); ?>/assets/images/apple-touch-icon.png">
  <link rel="icon" type="image/png" sizes="32x32"
    href="<? = get_template_directory_uri(); ?>/assets/images/favicon-32x32.png">
  <link rel="icon" type="image/png" sizes="16x16"
    href="<? = get_template_directory_uri(); ?>/assets/images/favicon-16x16.png">
  <link rel="manifest" href="<? = get_template_directory_uri(); ?>/assets/images/site.webmanifest">
  <link rel="mask-icon" href="<? = get_template_directory_uri(); ?>/assets/images/safari-pinned-tab.svg"
    color="#5bbad5">

  <title>Reverbera - Áudiolivros Gratuitos</title>
  <?php
  if (is_home()) {
    echo "Reverbera - Áudiolivros Gratuitos";
  } else {
    //shows site name with the current page name
    wp_title('|', true, 'right');
    bloginfo('name');
  }
  ?>
</title>
</head>
```

Fonte: Autoria Própria

Após o estabelecimento das informações da *head*, foram criados os elementos visíveis do cabeçalho, começando pela *tag body*, que agrupa os elementos do corpo, aqueles que são renderizados da página. O cabeçalho foi envolvido pela *tag header*, natural do HTML e que fornece a semântica que indica para o leitor de tela que o elemento em questão é um cabeçalho. A *header* recebeu como atributos *tabindex*, *accesskey* e uma classe. O primeiro atributo serve para tornar um elemento passível de foco, ou seja, acessível através da tecla *TAB* aspecto da navegação por tabulação recomendada de maneira obrigatória segundo os *Web Standards* da *WCAG*. O segundo atributo serve para que seja possível acessar o elemento através de atalho do navegador conforme o valor estabelecido.

Dentro da *header* foi criado um contêiner, representado pela classe “*c-container*”, para delimitar o tamanho máximo dos elementos pertencentes ao cabeçalho para uma melhor visualização em telas grandes por conta de não permitir que estes se espalhem em todo o comprimento. Foi adicionado um elemento para a navegação de menu em dispositivos móveis e a ele atribuído o atributo do *WAI-ARIA role* com o valor “*navigation*” para indicar ao leitor de tela que se trata de um elemento de navegação. Em seguida, criou-se uma lista não ordenada de elementos, *tag ul*, para os *links* de navegação do cabeçalho. Essa lista

agrupou cada *link*, *tag a*, dentro de uma *tag li*, que indica um item de lista, desta forma o usuário de leitor de tela tem como informação que se trata de uma lista com x elementos em forma de *links*, parte do conteúdo do cabeçalho descrito pode ser acompanhado na figura 14.

Figura 14 – Trecho de código do cabeçalho

```
<body class="body">
  <header id="menu" tabindex="0" accesskey="2" class="c-header">
    <div class="c-container">
      <div data-collapse="medium" data-animation="default" data-duration="400" role="navigation"
        class="c-header__nav w-nav">
        <ul class="l-flex l-flex--center l-flex__wrap">
          <li class="c-header__brand">
            <a title="Início" href="<?= home_url(); ?>" class="w-inline-block w--current">
              
            </a>
          </li>
          <li>
            <a href="#conteudo" class="c__link c__transition300">
              Ir para o conteúdo [1]
            </a>
          </li>
          <li>
            <a href="#" class="c__link c__transition300">
              Ir para o menu [2]
            </a>
          </li>
        </ul>
      </div>
    </div>
  </header>
</body>
```

Fonte: Autoria Própria

A navegação do cabeçalho abrangeu a logo, inserida através da tag *img*, com o *link* para a página inicial, atalhos de navegação e botões com recursos de acessibilidade que serão abordados mais adiante. Com a criação do cabeçalho partiu-se para outra estrutura comum a todas as páginas: o menu de categorias de audiolivros.

O menu de categorias foi desenvolvido para que em dispositivos móveis ele possa ser minimizado ou maximizado para ocultar ou exibir o seu conteúdo, a técnica será explicada quando o assunto sobre responsividade for abordado. Ele teve o seu título, *tag h2*, com um *id* atribuído para que quando o menu de categorias receba foco pelo leitor de tela seja anunciado o nome do título, isso ocorre porque foi utilizado um atributo do *WAI-ARIA* no menu de categorias chamado *aria-labeledby*, nele é especificado o *id* de um elemento e isso faz com que ele receba o valor HTML deste elemento, conforme observável na figura 15. Em seguida, criou-se uma navegação, *tag nav*, com o atributo *role* com o valor *navigation* para assegurar que os vários tipos de leitores de tela informem o tipo do elemento corretamente. Essa navegação acoplou uma lista não-ordenada de *links* fictícios, pois estes seriam substituídos no

momento da integração com o *back-end*, isso foi feito para que a estrutura pudesse ser estilizada e preparada para receber os dados dos conteúdos futuros.

Figura 15 – Trecho de código do menu de categorias

```
<div data-collapse="medium" data-animation="default" data-duration="400"
    aria-labelledby="categoryMenuTitle"
    tabindex="0" class="c-category-menu c__center w-nav">

    <h2 id="categoryMenuTitle" class="c-category-menu__title c__white-text">
        Categorias de audiolivros
    </h2>
    <nav role="navigation" class="nav-menu w-nav-menu">
        <ul tabindex="0" role="menu" class="c-gategory-menu__list">

            <!-- Renders the category list -->
            <?php $categories = get_categories('taxonomy=categorias&post_type=audiolivro');
            foreach ($categories as $category): ?>
                <li class="c-category-menu__link c__transition300">
                    <!-- insert the category name in the url -->
                    <a href="<?= get_post_type_archive_link('audiolivro') ?>?categoria=<?= $cat
                        <?= $category->name; ?>
                    </a>
                </li>
            <?php endforeach; ?>
        </ul>
    </nav>
```

Fonte: Autoria Própria

Outra estrutura que se repete em todas as páginas que foi desenvolvida em seguida foi a seção principal, que engloba o título da página, o menu de categorias de audiolivros e a área de conteúdo da página. Um aspecto importante nas criações de títulos é a hierarquia, no HTML as *tags* de títulos compreendem em *h1*, *h2*, *h3*, *h4*, *h5* e *h6*, sendo que a primeira indica o título principal da página e só deve haver um *h1* em cada uma e as outras são usadas como subtítulos. É de grande importância o uso correto dessa hierarquia na criação de títulos pois os usuários de tecnologias assistivas costumam fazer navegação por títulos para encontrar mais rapidamente o tópico que é de seu interesse e a estruturação dessas *tags* feita de maneira incorreta gera problemas de percepção do conteúdo da página.

Acima do menu de categorias foi criada uma seção que também é um componente que se repete em todas as páginas e que engloba o título principal, utilizando a *tag h1*. Esta seção teve um plano de fundo mais conhecido como *background* de um fone de ouvido, equipamento bastante usado para ouvir arquivos de áudio e que também remete a temática do site.

O menu de categorias foi inserido em uma seção ao lado de uma estrutura para o conteúdo principal, *tag main* nativa do HTML. Nessa estrutura foi inserido um texto falando

sobre a missão do site e abaixo criada uma área intitulada Últimos Lançamentos reservada para agrupar os últimos 6 audiolivros inseridos no site. Com as estruturas mencionadas e a estilização feita através do CSS parte página ficou com aparência conforme a figura 16.

Figura 16 – Aparência no *front end* do menu de categorias



Fonte: Autoria Própria

A última estrutura criada foi o rodapé, que é delimitado pela *tag* “footer” e comporta um *link* para o questionário de avaliação do *site* e as chamadas de arquivos escritos em javascript, que serão discutidos na próxima sessão.

2.3.2.3. Desenvolvimento dos recursos assistivos

Agora serão abordadas as etapas de criação de recursos assistivos do site Reverbera que foram: gerenciamento de tamanho da fonte do site, recurso de alto contraste e o *player* de audiolivros com acessibilidade voltado ao uso através de leitores de tela.

O recurso de gerenciamento de fonte é recomendação obrigatória da *WCAG 2.0* e tem a sua importância para usuários com baixa visão. No site deste trabalho, o usuário consegue manipulá-lo através de botões localizados no cabeçalho presente em todas as páginas. A funcionalidade foi criada por meio da combinação de HTML complementado com *WAI-ARIA*, CSS e Javascript. Esses botões compreendem em dois: o primeiro, representado na interface com o símbolo de “A+”, tem a função de aumentar o tamanho da fonte do site e o segundo representado com o símbolo de “A-” tem o propósito de reduzir o tamanho da fonte.

O botões de gerenciamento de fonte têm em sua composição do HTML, os seguintes atributos: *id*, para referenciamento no javascript, *title*, um atributo que indica uma dica textual

ao se passar o *mouse*, *aria-label* para mostrar para o usuário de leitor de tela a função do botão e *role*, pois foi utilizada uma *tag de link* e este atributo atribui ao elemento o reconhecimento como botão em tecnologias assistivas e por fim, o atributo *tabindex* para que os botões sejam acessíveis através da tecla *tab*.

O funcionamento do gerenciamento de fonte ocorre através de duas funções escritas em javascript, vide figura 17, em um arquivo chamado “fontSize.js” localizado na pasta “js” dentro de “assets”: uma para aumentar a fonte e outra para diminuí-la em *pixels*, um tipo de unidade de medida na computação. Elas são acionadas através de eventos que ocorrem por meio de clique ou quando é utilizada a tecla *enter* durante o foco desses botões. As funções detectam o tamanho de fonte do corpo da página, *tag body*, o manipulam e os outros tamanhos de fontes dos diferentes elementos do *layout* se adequam a proporção pelo fato de usarem a unidade de medida relativa “*em*” no CSS.

Figura 17 – Funções em javascript do gerenciamento de fonte

```
function increaseFontSize() {
    let body = document.querySelector( selectors: 'body');
    let size = window.getComputedStyle(body).getPropertyValue( property: 'font-size');
    size = size.replace( searchValue: 'px', replaceValue: '');
    size++;
    document.body.style.fontSize = size + "px";
    document.body.style.lineHeight = size + "px";
}

function decreaseFontSize() {
    let body = document.querySelector( selectors: 'body');
    let size = window.getComputedStyle(body).getPropertyValue( property: 'font-size');
    size = size.replace( searchValue: 'px', replaceValue: '');
    size--;
    document.body.style.fontSize = size + "px";
    document.body.style.lineHeight = size + "px";
}
```

Fonte: Autoria Própria

Outro recurso assistivo implementado no Reverbera, de recomendação obrigatória conforme a *WCAG 2.0*, foi o modo de alto contraste que, assim como o gerenciamento de tamanho de fonte, foi desenvolvido através da mesma combinação de linguagens. Visualmente é apresentado como um botão de alternância localizado no cabeçalho no qual possui dois modos: ligado ou desligado. Esse botão no HTML, recebeu a *tag* nativa *button*

que já possui semântica própria e teve os seguintes atributos inseridos: *tabindex*, *aria-pressed*, um recurso do *WAI-ARIA* que indica estado para demonstrar no leitor de tela se ele foi ou não pressionado e *onclick* que se trata de um atributo de chamada de função javascript através do clique do mouse.

O conteúdo javascript responsável por essa funcionalidade foi inserido em um arquivo chamado “*contrast.js*” localizado na pasta “*js*” dentro de “*assets*”. Nesse arquivo foram criadas duas funções com diferentes papéis. A primeira, vide figura 18, verifica se o navegador possui um objeto de *local storage*⁸ com o nome “*contrast*” e valor “*on*” e que, em caso positivo, insere a chamada da folha de estilo responsável pelas modificações visuais de alto contraste, além de modificar valor do atributo “*title*” do botão para “desligar alto contraste” e mudar o valor de seu “*aria-pressed*” para “*true*”, indicando o estado pressionado no leitor de tela ao passar pelo botão. A segunda função habilita ou desabilita a folha de estilo de alto contraste em detrimento do clique do botão responsável e faz o gerenciamento da semântica, modificando os valores dos atributos do botão já mencionados e indicando o seu estado atual.

Figura 18 – Função em javascript de verificação do modo de alto contraste

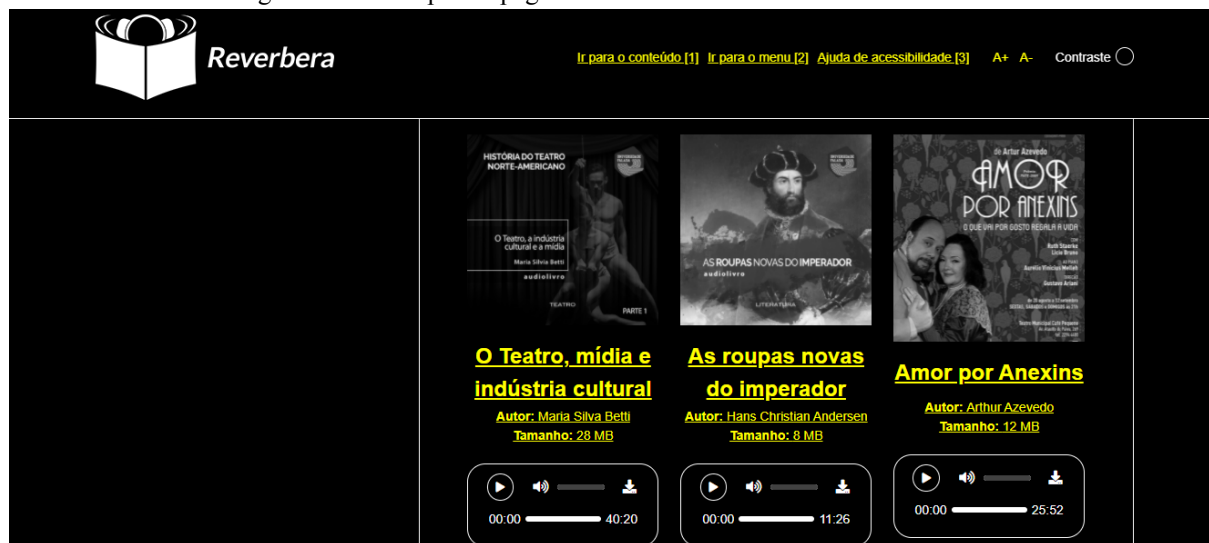
```
function checkContrast(href) {  
  let currentValue = window.localStorage.getItem( key: 'contrast');  
  let button = document.querySelector( selectors: '.c__contrast-button');  
  
  if (currentValue == 'on') {  
    //adds a new stylesheet  
    let link = document.createElement( tagName: 'link');  
    link.rel = 'stylesheet';  
    link.type = 'text/css';  
    link.href = href;  
    link.id = 'contrast';  
    document.getElementsByTagName( qualifiedName: 'head')[0].appendChild(link);  
  
    //accessibility state  
    button.setAttribute( qualifiedName: 'title', value: 'Desligar alto contraste');  
    button.setAttribute( qualifiedName: 'aria-pressed', value: 'true');  
  }  
}
```

Fonte: Autoria Própria

⁸ Objeto de dados com uma chave e valor salvo no navegador e que não expira automaticamente.

A folha de estilo que aplica as modificações para aplicar o alto contraste foi denominada “*contrast.css*” e alocada na pasta “*css*” dentro de “*assets*”. Fundamentalmente essa folha de estilo faz com que os *links* apresentem-se com a cor amarela, o fundo do site de cor preta, imagens em escala de cinza e os demais textos com a cor branca, conforme figura 19. Como a funcionalidade cria uma *local storage* no navegador, mesmo se o usuário sair da página do *site*, quando ele abri-lo novamente, ele estará com o modo de alto contraste ativado, caso ele tenha o feito previamente.

Figura 19 – Exemplo de página com o modo de alto contraste ativado



Fonte: Autoria Própria

Foi criada uma página de ajuda de acessibilidade conforme recomendado pela *WCAG 2.0*. Nela é apresentado quais são os recursos assistivos disponibilizados pelo *site* e como utilizá-los, além de algumas recomendações de *softwares* leitores de tela e os controles do *player* de audiolivros do *site* que será abordado a seguir.

O HTML possui uma *tag* nativa para *player* de áudio, denominada *audio*, porém, ela não possui suporte semântico algum, o que faz com que seja necessário o desenvolvimento de uma aplicação própria para se utilizar em um *site* com acessibilidade.

O desenvolvimento do *player* de áudio começou com a criação de uma *tag div*, que serve para agrupar elementos e que pode ser usada para referenciamento em estilizações no CSS ou em manipulações através do javascript, ela recebeu atributos para aplicação de foco, o *tabindex*, identificação do título do audiolivro utilizando o *aria-label* com o valor “Player de audiolivro:” concatenado com o nome do audiolivro e *role* com o valor “*application*” para

informar que se trata de uma aplicação durante o uso do leitor de tela. Dentro da *div* foi inserida a *tag audio* nativa pois, ela foi utilizada como base para o funcionamento do *player* pelo fato de já oferecer funções de reprodução como: reproduzir, pausar, volume e manipular a linha do tempo da reprodução. A *tag audio* recebeu uma *tag source* que serve para informar a origem do arquivo de áudio e o seu tipo, além do que esse elemento foi ocultado da interface por meio de estilização. Em seguida, foram desenvolvidos os controles conforme orientado pelo *WCAG 2.0* e que serão detalhados a seguir.

O primeiro controle criado foi o de reproduzir/pausar que recebeu a *tag button* e os atributos: *title* e *aria-label* com o valor “Reproduzir audiolivro”, *aria-pressed* com o valor “false” para indicar que a reprodução ainda não começou, *class* com as classes responsáveis pela estilização e uma específica chamada “*js-playbutton*” para identificação nas funções de javascript responsáveis pelo funcionamento do *audioplayer* que serão abordadas mais adiante. Dentro desse botão e dos demais presentes no *audioplayer* foi inserido uma *tag span*, utilizada em pequenos trechos com um estilo diferente do grupo na qual pertence, ela recebeu um atributo do *WAI-ARIA* chamado *aria-hidden* que serve para ocultar o elemento da percepção do leitor de tela, caso o valor seja “true”, ele foi utilizado porque essa *span* serviu para englobar ícones para identificar cada botão, tendo propósito único para usuários que não são cegos.

O segundo foi o botão de mutar/desmutar a reprodução e teve os mesmos atributos que o primeiro botão apresentado acima. A classe para identificá-lo no javascript foi a “*js-mutebutton*” e o valor dos atributos *title* e *aria-label* foi estabelecido como “Mutar audiolivro”.

O terceiro controle criado foi a barra de volume que recebeu como classe de referência a “*js-volumecontrolbar*” e os seguintes atributos: *aria-valuemin* com valor 0 para indicar no leitor de tela o valor mínimo do volume como 0, *aria-valuenow* que demonstra o valor atual do controle e *aria-value-text* para demonstrar o valor de uma maneira melhor quando o leitor de tela tem compatibilidade, que no caso deste controle foi em porcentagem, *aria-label* para indicar o nome do controle, *role* com o valor “*slider*” para indicar que é um controle deslizante, *aria-valuemax* para indicar o valor máximo do volume, que no caso é 100 e *tabindex* para tornar o foco neste elemento possível. Dentro da barra de volume foi inserida uma *div* com a classe “*js-volumecontroller*” para ser utilizada na função de indicação de estado da barra.

O quarto controle criado foi um botão de *download*, ou seja para salvar o arquivo no dispositivo do usuário, do audiolivro do *player* em formato de *link*, *tag a*, com os seguintes atributos: *title* e *arial-label* com o valor “Fazer *dowload* do audiolivro”, *download* para que o arquivo seja baixado diretamente assim que o botão for clicado caso o navegador seja compatível com isso, caso não foi adicionado um atributo *target* com o valor “*_blank*” para abrir o arquivo em outra guia do navegador.

O último controle feito na parte de desenvolvimento do HTML do *audioplayer* foi a barra de tempo de reprodução, que consistiu basicamente de dois elementos que indicam o tempo atual e o final da reprodução do audiolivro e que receberam como classes identificadoras no javascript a “*js-currenttime*” e a “*js-duration*”. Ambos elementos foram ocultos dos leitores de tela usando o atributo *aria-hidden* já mencionado. O último elemento deste controle foi a barra de controle que recebeu os atributos: *role* com o valor “*slider*” e *aria-label* com o valor “controle de tempo”. Teve como *classe* para o javascript a “*js-timecontrolbar*” e recebeu um elemento interno com a classe nomeada de “*js-timecontroller*” para auxiliar no desenvolvimento desse controle. Agora será abordado como foram feitas cada funcionalidades referentes aos controles do *audioplayer*.

Todo o código de javascript referente ao *audioplayer* foi alocado no arquivo chamado “*audioplayer.js*” dentro da pasta “*js*”. Inicialmente foi criado um trecho que seleciona todos os *audioplayers* da página e os inicializa conforme a figura 20.

Figura 20 – Trecho de inicialização dos *players*

```
let audioPlayer = document.querySelectorAll( selectors: '.c-audio-player');  
  
audioPlayer.forEach( callbackfn: function (element : Element ) {  
    initializeAudioPlayer(element)  
});
```

Fonte: Autoria Própria

A inicialização foi feita através de uma função intitulada “*initializeAudioPlayer*” que faz a seleção dos elementos HTML do *audioplayer* como o *player* nativo de áudio, *tag audio*, botões dos controles, barras de tempo e de volume e os prepara para o funcionamento da aplicação. Cada elemento foi selecionado através do recurso do javascript chamado “*querySelector*”⁹, utilizando-se de uma classe criada com o propósito de referenciá-lo

⁹ Funcionalidade que faz a seleção de elementos e utiliza como parâmetro *tags* ou atributos do HTML.

conforme foi mencionado nos parágrafos acima, e armazenado em uma variável específica de acordo com o seu propósito, de acordo com a figura 21.

Figura 21 – Trecho de seleção dos elementos com o recurso *querySelector*

```
//Adds functionality to the custom audio player
function initializeAudioPlayer(audioPlayer) {
    let audioBook = audioPlayer.querySelector('audio');
    let playButton = audioPlayer.querySelector('.js-playbutton');
    let muteButton = audioPlayer.querySelector('.js-mutebutton');

    let timeControlBar = audioPlayer.querySelector('.js-timecontrolbar');
```

Fonte: Autoria Própria

Além da seleção de elementos, foi trabalhada nessa mesma função a inicialização da barra de tempo, que ocorreu inicialmente com um trecho que captura os dados que são fornecidos pelo *player* nativo do HTML por meio de um recurso detector de eventos chamado “*eventListener*” e neste caso foi utilizado o parâmetro “*loadedmetadata*”. Esse detector de eventos acopla uma função anônima, tipo de função não declarada, nela foram criadas duas variáveis para representar a duração total do arquivo de áudio, a primeira para os minutos e a segunda para os segundos. Ambas as variáveis receberam o valor convertido para o tempo correspondente em minutos e segundos respectivamente que foram inseridos no HTML do *player* na parte que exibe a duração final. Além disso, nessa função os valores do *WAI-ARIA* já mencionados anteriormente da barra de tempo foram adicionados, conforme visível na figura 22.

Figura 22 – Trecho da inicialização da barra de tempo

```
//initialize aria values of time control bar
audioBook.addEventListener("loadedmetadata", function () {
    //fixes the duration display
    let minutes = parseInt(audioBook.duration / 60);
    let seconds = parseInt(audioBook.duration % 60);
    duration.innerHTML = minutes + ":" + seconds;

    timeControlBar.setAttribute("aria-valuemax", parseInt(audioBook.duration));
    timeControlBar.setAttribute("aria-valuenow", "0");
    timeControlBar.setAttribute(
        "aria-valuetext",
        parseInt(audioBook.currentTime / 60).toString() +
        "minutos e" +
        parseInt(audioBook.currentTime % 60).toString() +
        "segundos" +
        " " +
        "de" +
        " " +
        minutes.toString() +
        "minutos e" +
        seconds.toString() +
        "segundos"
    );
});
```

Fonte: Autoria Própria

Além dos elementos iniciados declarados nos parágrafos anteriores, foram salvos em variáveis o tamanho da barra de tempo referente a tela da navegação do usuário, os elementos responsáveis por exibir a duração e o tempo atual da faixa de áudio, a barra de progresso da faixa de áudio, o botão de download do audiolivro e os elementos da barra de volume conforme a figura 23.

Figura 23 – Trecho da inicialização da barra de tempo

```
timeControlBar.setAttribute('aria-valuemin', '0');

let timeControlBarSize = timeControlBar.clientWidth;
let progressBar = audioPlayer.querySelector('.js-timecontroller');
let duration = audioPlayer.querySelector('.js-duration');
let currentTime = audioPlayer.querySelector('.js-currenttime');

let volumeControlBar = audioPlayer.querySelector('.js-volumecontrolbar');
let volumeControlBarSize = volumeControlBar.clientWidth;
let volumeController = audioPlayer.querySelector('.js-volumecontroller');

let downloadButton = audioPlayer.getElementsByTagName('a')[0];
```

Fonte: Autoria Própria

Foi apresentada a parte de inicialização de elementos do *audioplayer* e agora será abordada a parte das funções propriamente ditas e que são responsáveis pelo funcionamento do *audioplayer*.

As funções são ativadas através do disparo de gatilhos de eventos que são implementados através de “eventListeners” e são acionados através de ações do usuário.

A primeira função criada foi a “playOrPause”, que tem o propósito de pausar ou reproduzir a faixa do audiolivro de acordo com o estado atual da reprodução. Ela tem como parâmetros o arquivo de áudio, o botão de reprodução, a barra de tempo, o tempo atual, o tamanho da barra de tempo e a barra de progresso. Caso o audiolivro não esteja pausado e não terminou a reprodução, a faixa de áudio é pausada, o ícone do botão de reprodução muda para indicar que está pausado e os valores WAI-ARIA são alterados para que o usuário tenha a informação ao passar pelo botão de que ele está pausado e que o próximo clique ou botão ENTER irá reproduzi-lo e por fim o tempo de reprodução é pausado. Caso contrário, é ativada a reprodução do audiolivro e feito os processos do caso anterior, porém, voltados para indicar o estado de áudio sendo reproduzido, vide figura 24. Além disso, quando a reprodução do arquivo é realizada, é chamada uma função intitulada “*update*”, que será abordada no próximo parágrafo, e todos os *players* da página são pausados caso estejam reproduzindo.

Figura 24 – Trecho da função playOrPause

```
//Play or Pause current audiobook
function playOrPause(
  audioBook,
  playButton,
  timeControlBar,
  currentTime,
  timeControlBarSize,
  progressBar
) {
  if (!audioBook.paused && !audioBook.ended) {
    audioBook.pause();
    playButton.innerHTML = "⏸";

    //accessibility state
    playButton.setAttribute("title", "Reproduzir audiolivro");
    playButton.setAttribute("aria-pressed", "false");
```

A função *“update”* é chamada quando se é feita a reprodução do audiolivro e seu papel é atualizar na interface os indicadores do tempo atual da faixa de reprodução. Caso o arquivo ainda esteja sendo reproduzido, essa função é chamada constantemente por um mecanismo chamado *“setInterval”*¹⁰ a cada meio segundo. O elemento responsável por exibir o tempo atual da reprodução é atualizado com o valor convertido, pois é recebido nativamente em milissegundos, acrescido com um zero à esquerda caso necessário. Além disso, a barra de progresso cresce de acordo com um cálculo que é feito pelo tempo atual da faixa multiplicado pelo tamanho da barra de tempo divididos pela duração da faixa de áudio. Os valores WAI-ARIA da barra de controle de tempo também são atualizados, sendo eles o *“aria-valuenow”* e o *“aria-valuetext”* indicando o tempo atual para o usuário de leitor de tela. Caso a reprodução do audiolivro já tenha sido encerrada, os valores do tempo atual e largura da barra de tempo são zerados, o ícone do botão é alterado e o *“setInterval”* é desativado, o conteúdo da função pode ser visto pela figura 25.

¹⁰ A função *setInterval()* chama uma função em intervalos especificados em milissegundos.

Figura 25 – Conteúdo da função *update*

```
//Updates current audiobook's time
function update(
  audioBook,
  playButton,
  timeControlBar,
  currentTime,
  timeControlBarSize,
  progressBar
) {
  if (!audioBook.ended) {
    let minutes = parseInt(audioBook.duration / 60);
    let seconds = parseInt(audioBook.duration % 60);
    let playedMinutes = addZero(parseInt(audioBook.currentTime / 60));
    let playedSeconds = addZero(parseInt(audioBook.currentTime % 60));
    currentTime.innerHTML = playedMinutes + ":" + playedSeconds;

    //calculates the progress bar's width percentage based on the audiobook's current time
    let size = parseInt(
      (audioBook.currentTime * timeControlBarSize) / audioBook.duration
    );
    progressBar.style.width = size + "px";

    //updates time control bar aria
    timeControlBar.setAttribute(
      "aria-valuenow",
      parseInt(audioBook.currentTime)
    );
    timeControlBar.setAttribute(
      "aria-valuetext",
      parseInt(audioBook.currentTime / 60).toString() +
        "minutos e" +
        parseInt(audioBook.currentTime % 60).toString() +
        "segundos" +
        " " +
        "de" +
        " " +
        minutes.toString() +
        "minutos e" +
        seconds.toString() +
        "segundos"
    );
  } else {
    currentTime.innerHTML = "00:00";
    playButton.innerHTML = "⏮";
    progressBar.style.width = "0px";
    window.clearInterval(updateTime);
  }
}
```

Fonte: Autoria Própria

A função “muteOrUnmute” realiza o papel de habilitar ou desabilitar o som da reprodução do audiolivro e atua de duas formas: caso esteja mudo, habilita o som, troca o ícone do botão e os valores WAI-ARIA que informam o nome do botão e se ele está pressionado ou não. Caso não esteja mudo, o som é desabilitado e são feitos os mesmos tratamentos do primeiro caso.

Foi desenvolvida também uma função responsável pela interação do mouse com a barra de tempo e que foi denominada “clickedTimeBar”. Sua atuação é feita através da detecção da posição do mouse no momento em que se ocorre o clique na barra de tempo. Através dessa captura é calculado o tempo relativo na faixa de áudio e comprimento da barra de progresso, conforme a figura 26.

Figura 26 – Conteúdo da função clickedTimeBar

```
function clickedTimeBar(event, audioBook, timeControlBar, progressBar, timeControlBarSize) {  
  if (!audioBook.ended) {  
    //gets the position of mouse relative to the element  
  
    let mouseX = event.pageX - timeControlBar.offsetLeft;  
    //calculates the time percentage based on mouse position on element  
    let newTime = mouseX * audioBook.duration / timeControlBarSize;  
  
    audioBook.currentTime = newTime;  
    progressBar.style.width = mouseX + 'px';  
  }  
}
```

Fonte: Autoria Própria

A WCAG 2.0 recomenda que sejam implementados atalhos de teclas em controles de aplicações para usuários de tecnologias assistivas, e que no caso de *players* de áudio sejam aplicadas as teclas: cima, baixo, direita, esquerda, *home* e *end*. Esses atalhos foram aplicados na barra de tempo e na barra de volume dos *players*.

Na barra de tempo, a função “keyDownTimeBar” é responsável por ativar determinada ação neste elemento de acordo com a tecla utilizada pelo usuário e bloquear funcionalidades nativas dessas teclas na página se o elemento esteja em foco, como por exemplo rolar para o final da página quando se é pressionada a tecla *end*. Caso a faixa de áudio esteja sendo reproduzida, ao acionar as teclas para cima ou para direita: a faixa de áudio avança em 5 segundos, para baixo ou esquerda: a faixa retrocede em 5 segundos, *home*: a faixa retorna ao início e *end*: a reprodução é encerrada, conforme. Se a tecla não corresponde a nenhuma das já mencionadas, nada ocorre.

Semelhantemente na barra de volume as teclas direita e cima aumentam o volume do áudio em 1%, baixo e esquerda diminuem em 1%, *home* torna o volume em 0 e *end* aplica o volume máximo. Esse trabalho é realizado pela função chamada de “keyDownVolumeBar”.

Como ocorrem alterações dos valores WAI-ARIA referentes ao elementos da barra de tempo ou de volume, esses novos valores são relatados através do leitor de tela e para que o

usuário consiga perceber essa informação de maneira clara, a reprodução do áudio é pausada por 8 segundos, esta função nomeada como “enhanceAudioTips”, conforme pode ser visto na figura 27.

Figura 27 – Conteúdo da função enhanceAudioTips

```
//pauses the audiobook when an audioplayer's elements is focused or activated to improve usability
function enhanceAudioTips(audioBook) {
  if (!audioBook.paused && !audioBook.muted && !audioBook.ended) {
    audioBook.pause();

    //plays the audiobook after 8 seconds
    setTimeout(function () {
      audioBook.play();
    }, 8000);
  } else return;
}
```

Fonte: Autoria Própria

Nesta seção foram abordados os aspectos mais importantes do que foi desenvolvido no *front end* do site Reverbera, como os elementos de acessibilidade e funcionalidades, bem como as páginas estáticas. Agora, na próxima seção será apresentado o desenvolvimento do *back end* e seu detalhamento.

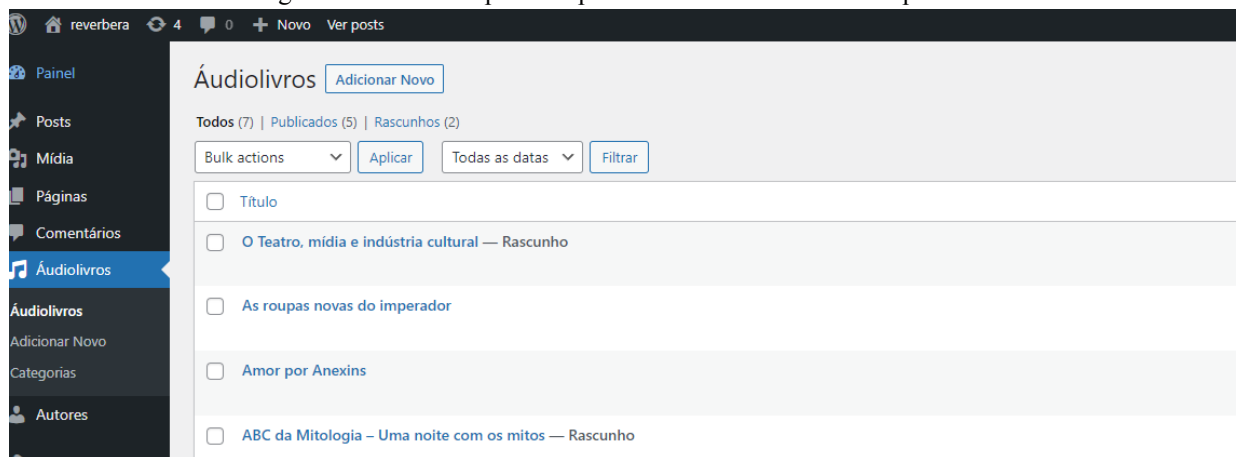
2.3.3. DESENVOLVIMENTO DO BACK END

O *back end* se refere a camada do sistema que é regida por estruturas que não são perceptíveis ao usuário mas, que são importantes para o funcionamento do *software* como um todo. Além disso, diferentemente do *front end*, em aplicações *WEB* as funções presentes no código fonte são executadas pelo servidor e o seu resultado convertido em código processável pelo navegador, como em linguagem HTML por exemplo. No caso deste trabalho, foram usadas como ferramentas de *back end* a linguagem PHP - Pré Processador de Hipertexto (*Hypertext Preprocessor*), que será referida como PHP de agora em diante no trabalho, e o CMS - Sistema Gerenciador de Conteúdo (*Content Management System*), referido aqui como CMS, Wordpress.

A linguagem PHP no âmbito da *WEB* atua no pré-processamento de dados, como os de banco de dados e de funções diversas. Esse pré-processamento funciona como uma ponte entre os recursos do *back end* e do *front end*, fazendo uma conversão de informações do servidor (*server side*) para o navegador (*client side*).

Um CMS tem o propósito de possibilitar o gerenciamento de conteúdo de um *site*, como criação, edição e exclusão. No Wordpress, CMS mundialmente utilizado, esse gerenciamento se dá através de um painel administrativo, figura 28, no qual é possível a gestão de páginas, postagens, mídias, usuários entre outros. Os itens criados por meio do painel são salvos no banco de dados que é criado através do próprio Wordpress e acessados através de suas estruturas internas.

Figura 28 – Visão de parte do painel administrativo do Wordpress



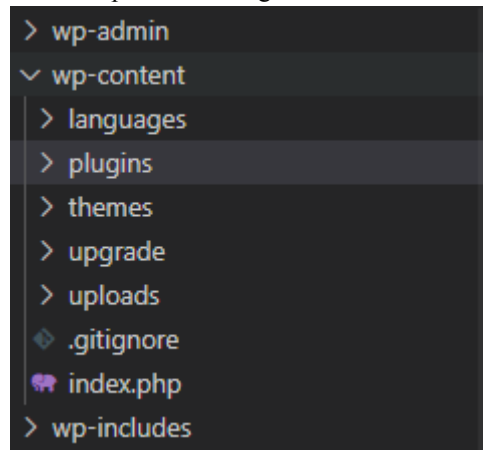
Fonte: Autoria Própria

Neste trabalho, o início do desenvolvimento do *back end* se deu primeiramente com a criação dos elementos básicos do Wordpress e posteriormente sua integração com o *front end*, para isso foi necessário ser feita a instalação do CMS na pasta do projeto. O arquivo de instalação é obtido do próprio *site* do fornecedor. Essa instalação é realizada movendo a pasta baixada do Wordpress para o diretório raiz do *site* e em seguida, seguindo o passo a passo de instalação exibido ao se acessar o *site* localmente pelo navegador. Esse passo a passo irá definir o nome do banco de dados gerado, nome e senha do usuário administrador e prefixo das tabelas do banco de dados.

O Wordpress possui uma estrutura básica de diretórios, conforme figura 29, e diversos arquivos que compõem o seu núcleo de funcionamento. Esses arquivos são escritos em linguagens como PHP, javascript, CSS, HTML entre outras. A próxima etapa para integrar é mover a pasta do *site*, que foi criada com o nome “reverbera”, para dentro da pasta “themes” localizada no interior da “wp-content” que é responsável por conter os arquivos referentes ao *site* em si e não ao núcleo do Wordpress. Após fazer isso, é necessário criar um arquivo chamado “style.css” com quatro campos, vide figura 30, e no primeiro adicionar o nome do

tema¹¹. acessar o painel administrativo do *site*, navegar para “aparência” e depois “temas” e definir o tema para aquele com o nome que foi definido, assim o conteúdo do tema será referenciado para o CMS.

Figura 29 – Visão superficial de alguns dos diretórios do Wordpress



Fonte: Autoria Própria

Na próxima seção será abordada como foi feita a criação da estrutura básica de componentes do Wordpress e a integração das páginas e de outros elementos de maneira detalhada

Figura 30 – Conteúdo do arquivo “style.css”

```
/*
Theme Name: Reverbera
Description: Site disponibilizador de audiolivros
Author: Elizeu
Version: 1.0.3
*/
```

Fonte: Autoria Própria

2.3.3.1. Criação das estruturas do CMS

O Wordpress, em sua estrutura de CMS, conta com diversos componentes que juntos compõem uma estrutura de posts e páginas. Com relação aos componentes de posts, existem os *post types*, taxonomias e termos. *Post types* são os tipos de posts, no Wordpress por padrão, o único *post type* já existente é o “post” e é necessário a criação de outros *post types* através

¹¹ Temas no Wordpress são um compilado de arquivos responsáveis pelo visual do *site* e suas funcionalidades específicas.

de código ou *plugins*¹², caso o projeto demande. Taxonomias são categorias para os *post types* e termos são os membros das taxonomias. Além destes componentes existem vários outros, como os campos, que são espaços para inserção de informações no painel e que ficam associados ao componente no qual estão listados no painel, essas informações também ficam salvas no banco de dados. Agora serão apresentados como foram criados e estruturados esses elementos.

No Reverbera, o *post type* padrão não foi usado por conter campos que não seriam utilizados. Então, criou-se inicialmente um *post type* para os audiolivros. Essa criação foi feita na “functions.php” que é um arquivo padrão do Wordpress localizado dentro do tema e tem a função de englobar as funções personalizadas. A função para a criação é chamada “register_post_type” que recebe por parâmetro o nome do *post type* e uma lista de argumentos que definem várias coisas como: nome dos elementos no painel, ícone de menu entre outros. Também foi registrada através de função, chamada “register_taxonomy”, a taxonomia para o *post type* “audiolivro” chamada “categorias”, conforme figura 31. Também foi criado outro *post type* para os autores dos audiolivros denominado “Autor”, sem taxonomia associada.

Figura 31 – Grupo de campos do *post type* Audiolivro

```
//Creates the taxonomy categorias to the posttype audiolivro
register_taxonomy(
    "categorias",
    "audiolivro",
    array(
        "label" => "Categorias",
        "singular_label" => "Categoria",
        "rewrite" => true,
        "hierarchical" => true,
        'has_archive' => true
    )
);
```

Fonte: Autoria Própria

Ambos os dois *post types* personalizados possuem elementos definidos na função “register_post_type” que habilitam a inserção de uma imagem, título e um campo de texto através do painel, cuja utilização será explicada no capítulo referente à integração.

¹² *Plugins* são aplicações de terceiros ou não e que podem ser acopladas no Wordpress através do painel administrativos e que adicionam funcionalidades extras ao *site*.

Complementando os elementos mencionados, foram criados campos personalizados com o auxílio do *plugin* chamado “*Advanced Custom Fields - Campos Avançados Customizados*, que será abordado como ACF”. No menu desse *plugin* foram criados 2 itens: audiolivro e autor, cada um com seus campos que possuem referências para uso no código-fonte, vide figura 32. No item audiolivro, foram criados os campos: audiolivro, autor, locutor, editora e descrição da capa do livro. O campo audiolivro comporta um arquivo do tipo mp3, é de preenchimento obrigatório e tem o nome de referência “audiofile”. O campo autor é do tipo objeto de *post*, que faz referência a um item de autor, o nome de código do campo é “author”. O campo locutor é não obrigatório e tem nome de referência “speaker”. O campo editora também é não obrigatório e o nome de referência é “publishingCompany”. O campo descrição da capa do livro é obrigatório, possui o nome de referência “coverAlt”.

Figura 32 – Conteúdo da função responsável pelo registro da taxonomia

Editar Grupo de Campos [Adicionar Novo](#)

Audiolivro

Campos

Ordem	Rótulo	Nome	Tipo
1	Audiolivro *	audioFile	Arquivo
2	Autor *	author	Objeto do Post
3	Locutor	speaker	Texto
4	Editora	publishingCompany	Texto
5	Descrição da capa do livro *	coverAlt	Texto

+ Adicionar Campo

Fonte: Autoria Própria

Foi mencionada a criação dos elementos referentes a estrutura de *post* e agora serão pontuados os aspectos da estrutura de página deste trabalho. A única página criada foi a de Ajuda de Acessibilidade e seu conteúdo no painel é vazio, tem o propósito apenas para que o CMS reconheça sua existência e a alimente com o conteúdo estabelecido em código e que será tratado na seção de integração, que será descrita a seguir.

2.3.3.2. Integração

O primeiro passo da integração foi a criação de uma subpasta chamada “partials”, que contém os arquivos com trechos de código que se repetem, como cabeçalhos e rodapés, por exemplo. O arquivo “header.php” foi o primeiro criado nessa pasta e nele foi inserido o trecho de código HTML relacionado ao cabeçalho da *index*, que foi criado na etapa de desenvolvimento do *front end*.

Como o *site* neste ponto teve sua base instaurada no Wordpress, modificações foram feitas para se adequar a nova estruturação. No arquivo “header.php”, a referência do endereço dos arquivos, como imagens por exemplo, tiveram alterações: a indicação da pasta raiz foi substituída para uma função do Wordpress chamada “get_template_directory_uri()” que indica a pasta raiz do tema.

Outra modificação foi na *tag title*, onde foi inserida uma condicional para modificar o título da página conforme a sua origem: na *index*, exibir “Reverbera - Audiolivros Gratuitos”, em outras exibir o nome do *site* concatenado com o nome da página, isso ocorreu com a utilização das funções do Wordpress “is_home()” que retorna verdadeiro caso esteja na página *index*, “wp_title” que insere o título de página definido no painel administrativo e “bloginfo” para exibir o nome da página Wordpress, que será explicada posteriormente. Após o fechamento da *tag head* foi inserida uma função também do Wordpress, chamada “wp_head()” que insere a chamada dos arquivos de *script* definidos no arquivo “functions.php”, que será detalhado posteriormente. Parte das alterações do arquivo “header.php” podem ser vistas na figura 33.

Figura 33 – Trechos da integração do arquivo “header.php”

```
<link rel="mask-icon" href="<?= get_template_directory_uri(); ?>/assets/images/safari-pinned-tab.svg"
      color="#5bbad5">

<title>Reverbera - Áudiolivros Gratuitos</title>
<?php
if (is_home()) {
    echo "Reverbera - Áudiolivros Gratuitos";
} else {
    //shows site name with the current page name
    wp_title('|', true, 'right');
    bloginfo('name');
}
?>
</title>
</head>

<?php wp_head(); ?>
```

Fonte: Autoria Própria

O segundo arquivo da pasta “partials” criado foi o “side-menu.php” que englobou o trecho referente ao menu lateral de categoria de audiolivros. Nele foi removida a parte previamente estática dos *links* fictícios e inserido um laço de repetição “foreach” que varre todos os itens de categoria do *post type* audiolivro e insere para cada item um *link* para a *archive*¹³ correspondente e adiciona um parâmetro na url com o nome da categoria, conforme figura 34. O conteúdo das *archives* será tratado posteriormente.

Figura 34 – Trechos demonstrando a inserção de *links* através do laço de repetição

```
<!-- Renders the category list -->
<?php $categories = get_categories('taxonomy=categorias&post_type=audiolivro');
foreach ($categories as $category): ?>
    <li class="c-category-menu__link c__transition300">
        <!-- insert the category name in the url -->
        <a href="<?= get_post_type_archive_link('audiolivro') ?>?categoria=<?= $category->name; ?>
            <?= $category->name; ?>
        </a>
    </li>
<?php endforeach; ?>
</ul>
```

Fonte: Autoria Própria

No terceiro arquivo da “partials”, o “audiobook-item.php”, foi criada uma variável denominada “attachment_id” que recebe o valor do campo de audiolivro provindo do plugin ACF, recebido como ID. Esse valor é processado por uma função do Wordpress chamada “wp_get_attachment_metadata”, que retorna os dados de um arquivo anexo em forma de vetor, no caso o arquivo mp3, o retorno é aplicado na variável “metadata”. Foi criada outra

¹³ *Archive* é um tipo de página do Wordpress que é referente a um item de taxonomia e tem o seu conteúdo estruturado via codificação.

variável chamada de “size” que recebe o valor do tamanho do arquivo mp3 pelo acesso a chave do vetor. Mais uma variável foi criada e chamada de “author” e recebe valor do campo autor.

Além das variáveis mencionadas, o *link* do audiolivro é recebido pela função do Wordpress “the_permalink”, a origem da imagem, o atributo src, é referenciada pela função “get_the_post_thumbnail_url()” e o atributo alt recebe o valor concatenado com do campo da capa do livro. O título do audiolivro é recebido pela função “the_title”, o nome do autor é referenciado pelo “post_title” e o tamanho do arquivo é informado pela função “size_format”. A informação do tamanho do arquivo é uma das recomendações da WCAG 2.0. Além disso, é incluído o arquivo também da pasta “partials” chamado “audioplayer.php”, que será mencionado abaixo.

No “audioplayer.php”, foi inserido o trecho HTML referente ao audioplayer. Nele o atributo “aria-label” recebe uma concatenação com o título do audiolivro que é obtido através da função do Wordpress “the_title”. Na *tag* de *audio* a origem do arquivo, atributo src, recebeu como valor a função do Wordpress “wp_get_attachment_url” que buscou o caminho do arquivo de áudio. O mesmo foi feito no *link* de *download* do audiolivro.

O quinto arquivo da “partials” foi o “breadcrumb.php” e o seu propósito é estruturar os caminhos de pão, também conhecidos como *breadcrumbs*, que são indicadores de uma rota para uma página, conforme a figura 35. Essa estruturação foi feita com o auxílio de um *plugin* do Wordpress chamado de “Breadcrumb NavXT” através do painel administrativo na lista de configurações deste *plugin*.

Figura 35 – Exemplo de um *breadcrumb* so site Reverbera

Você está aqui:
[reverbera](#) > [Conto](#) > [As roupas novas do imperador](#)

Fonte: Autoria Própria

O último arquivo da pasta “partials” é o “footer.php” e contém os trechos do rodapé, que já foram mencionados. A modificação nesse ponto foi a adição da função do Wordpress chamada “wp_footer”, que faz a chamada dos arquivos inseridos na “functions.php” que será abordada posteriormente.

Na *index* e em outros arquivos de página, a inserção dos arquivos da “partials” é feita através da função do PHP chamada “include”, que insere através da passagem de parâmetro do caminho da pasta do arquivo. Agora serão discutidas as nuances das criações das *singles* dos *post types* criados e da *archive* de audiolivros.

Singles são páginas padrão do Wordpress referentes a um *post type* e servem para exibir o seu conteúdo singular, ou seja, cada item do *post type* terá uma página alimentada pelo que foi inserido no painel administrativo referente a ele. No código-fonte o arquivo da *single* deve ser referenciado com o prefixo “single-” e o nome do *post type* com extensão php para que o Wordpress a reconheça.

A primeira *single* criada foi a do *post type* “audiolivro”, na qual foram criadas algumas variáveis: “audiobook” que recebe o objeto de *post* da *single*, “author” que recebe o valor do campo de autor, “attachment_id” para capturar o valor arquivo de áudio, “metadata” para armazenar o ID dos dados do arquivo, “size” que registra o tamanho do arquivo e “duration” para a duração formatada do arquivo, “category” para armazenar as categorias do *post* e “categoryName” que recebe o nome das categorias. As variáveis foram dispostas conforme a figura 36.

Figura 36 – Variáveis da *single* de audiolivro

```
single-audiolivro.php
1  <?php include('partials/header.php');
2
3  //gets the audiobook of the single page
4  $audiobook = get_post();
5
6  //gets the author of audiobook of the single page
7  $author= get_field('author');
8
9
10 //gets the audiobook file
11 $attachment_id = get_field('audioFile');
12 $metadata = wp_get_attachment_metadata($attachment_id);
13 $size = $metadata['filesize'];
14 $duration = $metadata['length_formatted'];
15
16 $categoryName;
17 $category = get_the_terms(get_post()->ID, 'categorias');
18 foreach ($category as $term) {
19     $categoryName = $term->name;
20 }
21
22 ?>
```

Fonte: Autoria Própria

Estruturalmente, foi criada uma seção para comportar o título da página e que contém um atributo ARIA, o “aria-labeledby” que faz o leitor de tela anunciar o título do elemento com o valor do ID contido neste atributo. Uma seção secundária foi estabelecida para comportar o conteúdo da página, sendo dividido no menu lateral que contém a lista de categorias de audiolivros e o conteúdo principal. O conteúdo principal começa com uma *tag* “main” que possui um atributo de “accesskey” destinado a fornecer um atalho de navegação para o elemento em questão. Dentro da “main” existe o título do *post*, obtido pela função do Wordpress “the_title”, a capa do audiolivro, inserida pela *tag* “img” cuja origem é provida pela função “get_the_post_thumbnail_url” que retorna a imagem de capa do *post*. Em sequência, o resumo do audiolivro, o autor, categoria do post, editora, locutor, duração e tamanho são obtidos através das variáveis do arquivo da *single* que foram mencionadas no parágrafo anterior. Além disso, é inserido o *audioplayer* por meio da função “include”, o resultado da execução do trecho de código da *main* pode ser visualizado na figura 37.

Figura 37 – Parte da *single* de audiolivro

Você está aqui:
[reverbera](#) > [Conto](#) > As roupas novas do imperador

As roupas novas do imperador



Resumo: "Há muitos e muitos anos havia um imperador tão apaixonado por suas roupas novas que gastava com elas todo o dinheiro que possuía"... é assim que começa a estória divertida e interessante do imperador que amava suas roupas a ponto de "ter uma para cada hora do dia". Fazendo um contrato com dois vigaristas que fingiam ser tecelões que podiam fazer uma roupa que somente os competentes e inteligentes podiam vê-la. "Devem ser trajes magníficos - pensou o imperador" e toda a cidade estava empolgada (pois todos já sabiam das "qualidades maravilhosas do tecido em questão") para saber se seu vizinho ou amigo é um tolo.

Autor: [Hans Christian Andersen](#)

Categoria: Conto

Editora: Universidade Falada

Locutor(a): Leonardo Rossi

Duração: 11 minutos e 27 segundos

Tamanho: 8 MB



Fonte: Autoria Própria

A *single* de autor teve nela criadas duas variáveis para auxiliar a renderização da página: “author”, que recebe a objeto de *post* e “title”, que armazena o título do objeto de *post*. Com o estabelecimento dessas variáveis, foram criadas as estruturas de conteúdo. A primeira seção foi a do título principal, que da mesma forma que na *single* de audiolivro, possui o atributo “aria-labeledby” para que o leitor de tela ao dar o foco no elemento “section” anuncie o mesmo valor do título h1 que possui o ID correspondente. A segunda seção teve a estruturação de forma semelhante à *single* de audiolivro, tendo como diferença o conteúdo da tag “main”: nome do autor como título h2, foto do autor, um resumo sobre e por fim, uma lista com os audiolivros referentes a este autor. O resultado da *single* de autor pode ser visualizado na figura 38.

Figura 38 – Parte da *single* de autor

[reverbera](#) > [Autor](#) > Machado de Assis

Machado de Assis



Machado de Assis (1839-1908) foi um escritor brasileiro, um dos nomes mais importantes da literatura brasileira do século XIX. Destacou-se principalmente no romance e no conto, embora tenha escrito crônicas, poesias, crítica literária e peças de teatro. Machado de Assis escreveu nove romances. Os primeiros – Ressurreição, A Mão e a Luva, Helena e Iaiá Garcia –, apresentam alguns traços românticos na caracterização dos personagens. A partir de Memórias Póstumas de Brás Cubas, teve início sua fase propriamente realista quando revelou seu incrível talento na análise do comportamento humano, descobrindo, por trás dos atos bons e honestos, a vaidade, o egoísmo e a hipocrisia.

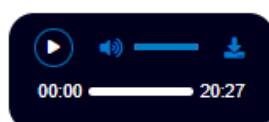
Audiolivros do autor



A igreja do diabo

Autor: Machado de Assis

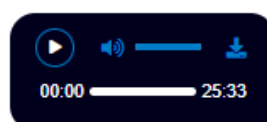
Tamanho: 5 MB



A Cartomante

Autor: Machado de Assis

Tamanho: 6 MB



Fonte: Autoria Própria

Na *archive* de audiolivros, foi criada uma variável de apoio chamada “categoryName” que recebe o valor de parâmetro da url da página com a chave “categoria” através da função do php “\$_GET”. Estruturalmente, foi criada uma seção de título da página que exibe o nome da categoria e uma seção de conteúdo que exibe a listagem de audiolivros referentes a essa categoria, conforme a figura 39.

Figura 39 – Parte da *archive* de audiolivro

Você está aqui:

[reverbera](#) > Áudiolivros

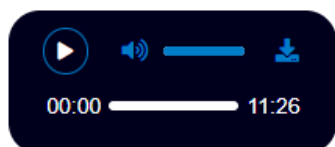
Listagem de audiolivros



As roupas novas do imperador

Autor: Hans Christian Andersen

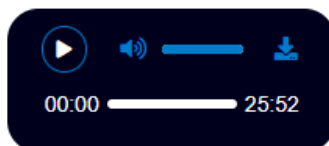
Tamanho: 8 MB



Amor por Anexins

Autor: Arthur Azevedo

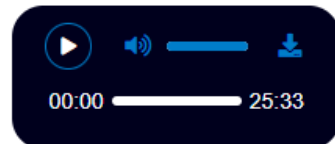
Tamanho: 7 MB



A Cartomante

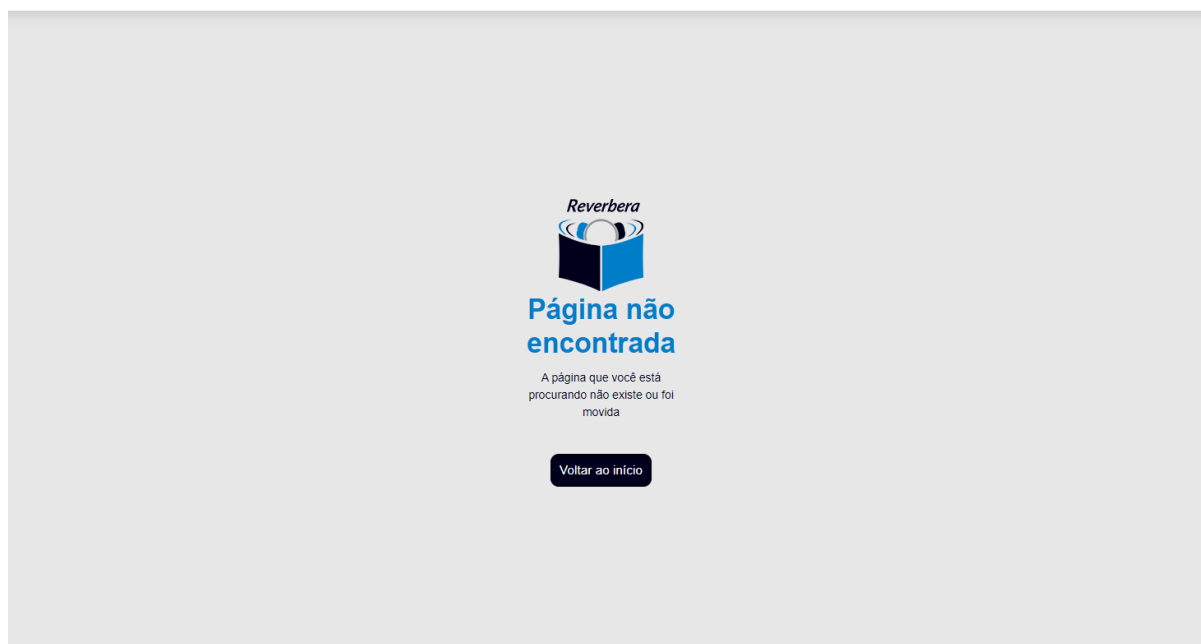
Autor: Machado de Assis

Tamanho: 6 MB



Fonte: Autoria Própria

Outro tipo de página criada foi a 404, para qual o usuário é redirecionado ao acessar uma *url* indisponível no *site*. O Wordpress faz o reconhecimento desta página através do arquivo “404.php” criado dentro do tema. Nela, foram inseridos através da função “include”, como em todas as outras páginas já descritas neste trabalho, o cabeçalho e o rodapé e além disso um texto informado que a página buscada não existe e um link para redirecionar a página inicial. O resultado visual da página pode ser observado na figura 40.



Fonte: Autoria Própria

Agora será abordado o “functions.php”, um arquivo do tema do Wordpress destinado a armazenar funções personalizadas para o tema na qual está inserido. A primeira função inserida tem o propósito de ocultar a barra do painel administrativo que aparece nas páginas não-administrativas. Essa função é chamada através de um filtro, que é um recurso do Wordpress que atua antes ou depois de um fluxo específico do processamento da página, conforme figura 41.

Figura 41 – Função para ocultar a barra administrativa

```
//removes wordpress admin bar  
add_filter('show_admin_bar', '__return_false');
```

Fonte: Autoria Própria

A segunda função criada foi denominada de “registerScripts” e serve para inserir os códigos de Javascript nas páginas. Nela foi estabelecida uma variável chamada de “path” que recebe o trecho de caminho comum aos arquivos de Javascript personalizados do tema. E, para cada arquivo foi inserida uma função para registrar o *script* e outra para inserir na página. Esta função de inserção, a “wp_enqueue_script” tem como último parâmetro a indicação se o *script* será inserido no rodapé ou não, e neste caso recebeu o valor *true*. A

“registerScripts” é chamada através de uma *action*, um recurso do Wordpress onde é acoplada função e chamada por meio do primeiro parâmetro, conforme figura 42.

Figura 42 – Função registerScripts

```
//Register and enqueue the scripts
function registerScripts()
{
    $path = get_template_directory_uri() . '/assets/js/';
    //Removes the default wordpress jquery script
    wp_deregister_script('jquery');

    wp_register_script('jquery', $path . 'jquery-3.4.1.min.js', array(), '3.4.1', true);
    wp_enqueue_script('jquery');

    wp_register_script('scripts', $path . 'scripts.js', array(), '', true);
    wp_enqueue_script('scripts');

    wp_register_script('audioplayer', $path . 'audioplayer.js', array(), '', true);
    wp_enqueue_script('audioplayer');

    wp_register_script('fontSize', $path . 'fontSize.js', array(), '', true);
    wp_enqueue_script('fontSize');

    wp_register_script('contrast', $path . 'contrast.js', array(), '', false);
    wp_enqueue_script('contrast');
}

add_action('wp_enqueue_scripts', 'registerScripts');
```

Fonte: Autoria Própria

A terceira função é semelhante à “*registerScripts*” porém, serve para registrar os arquivos CSS na página e foi denominada de “*registerStyles*”. Tendo como maior diferença nos trechos da “*wp_enqueue_script*” que foi trocada pela função ““*wp_enqueue_style*” que sempre aloca o arquivo no cabeçalho.

A quarta função foi adicionada para dar suporte a imagens de capa para os *posts*. E as outras funções que ainda não foram mencionadas serviram para formatar o valor da duração de tempo dos audiolivros e modificar a exibição dos *breadcrumbs*.

Essas foram as funções desenvolvidas no arquivo “*functions.php*” e alguns de seus detalhes aqui mencionados.

3. CONSIDERAÇÕES FINAIS

A dificuldade das pessoas com deficiências visuais em relação ao consumo de conteúdo literário é um problema preocupante e que prejudica o seu senso crítico e formação intelectual como um todo. Por isso, é necessária maior atuação e investimento de recursos para que o acesso à leitura seja feito de maneira igualitária e integrativa e isso não pode ser diferente no meio digital, que têm sido o principal ambiente de acesso à informação nos dias atuais.

A existência dos audiolivros é muito importante para que a população cega consiga ter acesso à literatura, já que a produção de livros em *braille* é muito custosa e não fornece variedade de exemplares. Os audiolivros proporcionam maior interesse na leitura, pois são narrados de forma expressiva, o que gera mais interesse, assim como as ilustrações presentes nos livros físicos. Além disso, podem ser facilmente distribuídos no meio digital e isso têm ocorrido, porém o verdadeiro problema é que os *sites* privados que fazem esta distribuição não possuem acessibilidade o suficiente porque o seu foco não são os deficientes visuais.

Foram criadas leis no Brasil e em outros países que obrigam o emprego da acessibilidade em *sites* e portais de órgãos governamentais. Porém, infelizmente a maior parte dos *sites* existentes é de origem privada e majoritariamente não existe o interesse em se investir para que suas páginas tenham acessibilidade como um todo, nem ao menos para usuários de leitores de tela.

Existe a alegação que a inclusão da acessibilidade na *web* como um requisito no desenvolvimento dos *sites* é algo trabalhoso e que não gera retorno o suficiente para se justificar o custo extra para a sua realização, o que resulta na privação dos usuários cegos até mesmo da aquisição de produtos no meio digital. Essa privação ocorre porque esses usuários não conseguem ter uma boa navegação utilizando seus *softwares* leitores de tela pela falta de semântica e de *tags* próprias no código destas páginas e até mesmo por uma navegação por meio de tabulação que não foi pensada no momento do desenvolvimento, o que resulta na não compreensão do conteúdo destes *sites*.

Tendo em vista o que foi dito, a importância de produtos tecnológicos como o Reverbera é de extrema relevância e exemplo para que os usuários com deficiências visuais não sejam marginalizados no meio digital. Seria sensato e ético, os desenvolvedores, as empresas e os cursos de formação incluírem em suas grades e produtos estratégias e recursos que garantam acessibilidade ao público como um todo.

O término do desenvolvimento do Reverbera resultou na aplicação de recursos assistivos importantes como: controle de tamanho textual do conteúdo, reproduutor de áudio semântico e acessível, modo de alto contraste para usuários com baixa visão, além de todo corpo do *site* voltado ao usuário final com semântica para utilização de leitores de tela. Além de um painel no qual o conteúdo dos audiolivros e seus autores é totalmente gerenciável e conta com inserções de dados que aumentam a compreensão de seu conteúdo.

A metodologia *Personal Scrum* garantiu o desenvolvimento do produto de maneira ágil e organizada, contribuindo para que dificuldades durante o desenvolvimento como o aprendizado de recursos do Wordpress e sua aplicação pudessem ser resolvidas sem comprometer o restante da resolução das tarefas.

Em relação ao reproduutor de áudio, houve uma dificuldade para implementação da acessibilidade do mesmo quando a navegação ocorre por meio de dispositivos móveis, um ponto importante para trabalhos futuros explorarem a solução deste problema, que se deu pelos leitores de tela deste dispositivos não aceitaram toda a se semântica inserida no código, principalmente nos controles deslizantes de volume e faixa de reprodução.

Outra questão interessante que não foi abordada foi a mensuração através de coletas de dados acerca da usabilidade e acessibilidade da plataforma com usuários reais que se encaixam no público-alvo da plataforma. Esta análise, pode inclusive abrir portas para implementação de novos recursos que possam sanar necessidades que não foram percebidas e que favoreçam a navegação do público alvo na busca por audiolivros em plataformas *online*.

Como alternativa à mensuração por vias humanas, foi realizado um teste automatizado utilizando uma plataforma disponibilizada pelo governo, intitulada ASES - Avaliador e Simulador de Acessibilidade em Sítios, que tem como critérios diretrizes formuladas pelo governo e que levam em conta a WCAG 2.0 e outros documentos de

recomendações assistivas. Os detalhes do teste estão localizados no Apêndice F - Avaliação de Acessibilidade e o mesmo apresentou alguns erros que se mostram como pontos interessantes a serem comentados.

O primeiro ponto é um erro de marcação, no qual foi recomendado inserir o *link* de ir para o conteúdo principal como o primeiro item da listagem de navegação, essa melhoria se mostra bastante relevante e pode ser implementada futuramente para aprimorar a usabilidade do *site*.

O segundo ponto é um erro classificado pelo ASES como comportamental no qual foi recomendado que os *scripts* inseridos no site possuem ausência de NOScript, esse problema demanda de uma investigação futura sobre o termo para melhoria da plataforma nesse critério.

O último ponto foi um critério de conteúdo/informação no qual foi apontada a mesma descrição textual para *links* diferentes. Isso destaca a importância dos testes humanos para a acessibilidade pois o ASES não consegue reconhecer que os *links* mencionados são do botão de *download* contido nos *players* de áudio e que possuem valores de referência diferentes justamente por se tratarem de arquivos distintos relativos aos seus audiolivros. Caso esse critério tivesse sido avaliado da forma correta, provavelmente o Reverbera teria obtido uma nota de 95% de acessibilidade, mas ficou com o resultado de quase 92% e isso demonstra que a plataforma ainda consegue melhorar em vários aspectos, mas, o seu objetivo que foi o embasamento na acessibilidade focada no uso de leitores de tela foi alcançado.

O esforço e pesquisa para o desenvolvimento e melhoria de produtos e recursos assistivos não deve parar mas sim, andar de mãos dadas com os avanços tecnológicos para que a discrepância do acesso à informação entre pessoas com deficiência e videntes, pessoas que não deficientes visuais, seja cada vez menor e assim cumprindo o propósito da tecnologia que é a evolução da humanidade.

REFERÊNCIAS

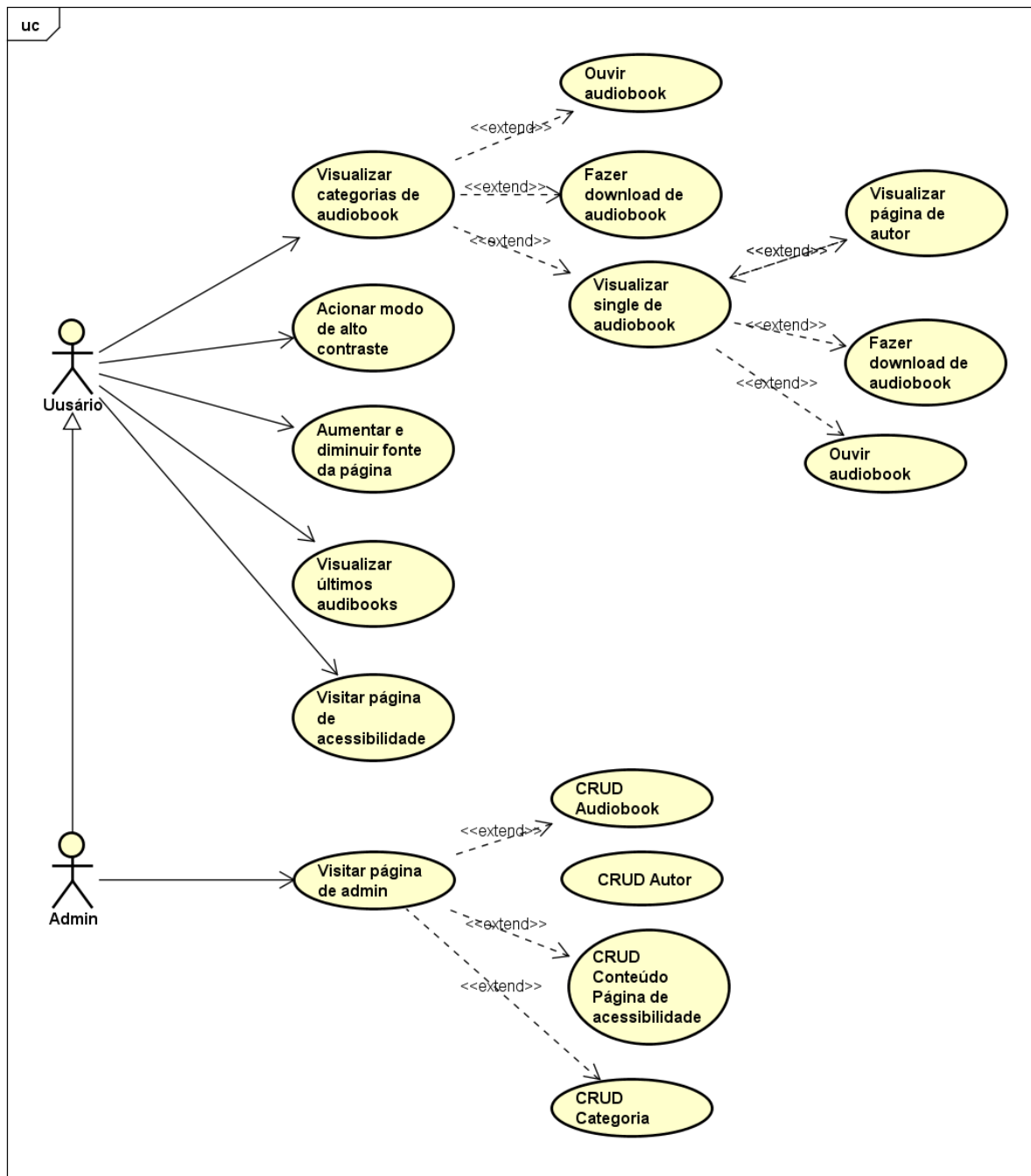
- CENSO DEMOGRÁFICO. **Características gerais da população, religião e pessoas com deficiência.** Disponível em: ftp://ftp.ibge.gov.br/Censos/Censo_Demografico_2010/Caracteristicas_Gerais_Religiao_Deficiencia/caracteristicas_religiao_deficiencia.pdf. Acesso em: 14 out. 2019.
- COSTA, K. R. N. Personal Scrum: uma alternativa ágil para desenvolvimento de indie games. 30 nov. 2016.
- INSTITUTO BENJAMIN CONSTANT. **Os conceitos de deficiência** – As diversas definições. Disponível em: http://www.ibc.gov.br/images/conteudo/AREAS_ESPECIAIS/CEGUEIRA_E_BAIXA_VISAO/ORIENTACOES-BASICAS/tipos-de-deficiencia.pdf. Acesso em: 14 out. 2019.
- KRUG, F. **A importância da leitura na formação do leitor.** 01 dez. 2015. Disponível em: <https://docplayer.com.br/39574602-A-importancia-da-leitura-na-formacao-do-leitor.html>. Acesso em: 15 out. 2019.
- MENEZES, N. C.; RIBEIRO, S. F. AUDIOLIVRO: UMA IMPORTANTE CONTRIBUIÇÃO TECNOLÓGICA PARA OS DEFICIENTES VISUAIS. **Ponto de Acesso**, v. 2, n. 3, p. 58–72, 9 dez. 2008.
- OLIVEIRA, M. DE et al. AUDIOBOOKS COMO FERRAMENTA PEDAGÓGICA NA EDUCAÇÃO INCLUSIVA DE DEFICIENTES VISUAIS. **Revista Eletrônica em Gestão, Educação e Tecnologia Ambiental**, v. 20, n. 1, p. 254–265, 24 fev. 2016.
- PAGOTTO, T. et al. **Scrum solo: Software process for individual development.** 2016 11th Iberian Conference on Information Systems and Technologies (CISTI). **Anais...** In: 2016 11TH IBERIAN CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGIES (CISTI). Gran Canaria, Spain: IEEE, jun. 2016b. Disponível em: <http://ieeexplore.ieee.org/document/7521555/>. Acesso em: 13 set. 2019
- PANSANATO, L. T. E.; RODRIGUES, L.; SILVA, C. E. Inclusão de estudante cego em curso de Análise e Desenvolvimento de Sistemas de uma instituição pública de ensino superior: um estudo de caso. **Revista Educação Especial**, v. 29, n. 55, p. 471–486, 9 ago. 2016.
- SANTOS, J. P. *et al.* TECNOLOGIA ASSISTIVA: um estudo sobre o uso de aplicativos para deficientes visuais. **Brasil Para Todos - Revista Internacional**, v. 4, n. 1, p. 59–69, 27 jul. 2017.
- SOARES, M. DOS S. Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software. **Revista Eletrônica de Sistemas de Informação**, v. 3, n. 1, 2004.
- W3C BRASIL. **Cartilha Acessibilidade na Web.** Disponível em: <https://ceweb.br/cartilhas/cartilha-w3cbr-acessibilidade-web-fasciculo-I.html>. Acesso em: 02 nov 2022.

APÊNDICE A – MEMORIAL DESCRITIVO

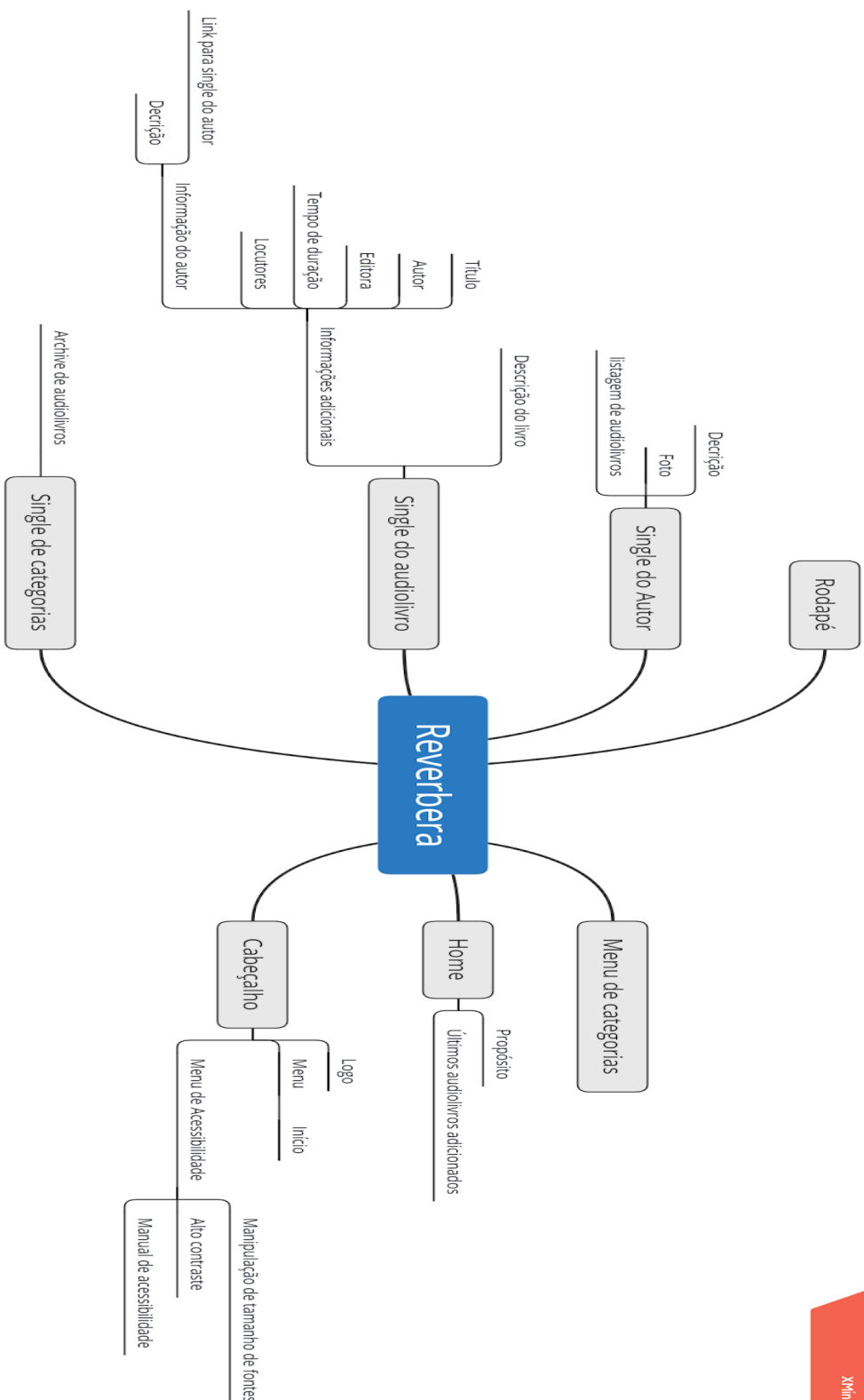
Memorial Descritivo

Deverá ser desenvolvido um site com acessibilidade e responsividade com o intuito de disponibilização de audiolivros. Será necessária a inclusão de recursos assistivos (**controle do tamanho de fontes, ativação de alto contraste e link para a página de acessibilidade**). O usuário poderá encontrar os audiolivros por meio de um **menu de categorias** e também pela página de autor, podendo fazer o *download* do arquivo ou ouvi-lo *online*. Com relação ao conteúdo, serão disponibilizadas as seguintes informações: **acessibilidade do site, descrição dos autores dos *audiobooks*, títulos dos *audiobooks*, resumos, nomes dos locutores, durações, editoras**. Haverá também um painel administrativo para gerenciar os seguintes itens: **audiolivro, autor, categoria e conteúdo da página de acessibilidade**. Somente o usuário administrador terá a capacidade de acessar este painel e a forma na qual os itens cadastrados poderão ser alterados seguirá o padrão disponibilizado pelo sistema de CMS utilizado. O usuário comum não necessitará ter uma conta para utilizar o *site* e poderá fazer a navegação sem nenhuma ferramenta extra, exceto caso seja um **usuário deficiente visual que deverá utilizar um leitor de tela na navegação**.

APÊNDICE B – DIAGRAMA DE CASOS DE USO



APÊNDICE C – MIND MAP

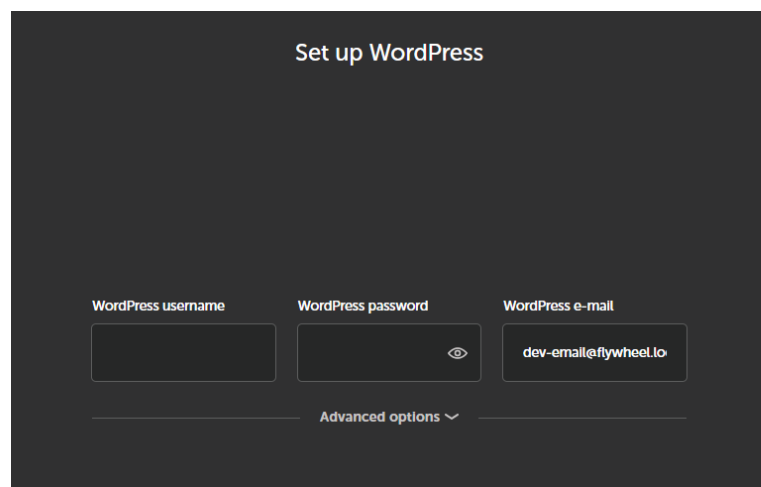
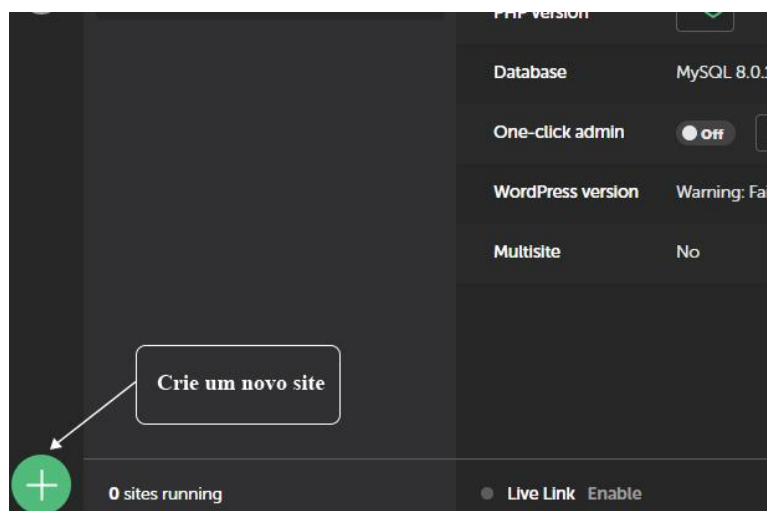


APÊNDICE D – MANUAL DE INSTALAÇÃO

1. O Wordpress tem como requisito um servidor que suporte PHP, MySQL e Apache/Nginx para funcionar. Por isso é necessário ter instalado na máquina *softwares* que forneçam esses serviços. Existem produtos que já proporcionam um ambiente ideal que atende essas necessidades, é recomendado utilizar o Local WP. Baixe este *software* através do link: <https://localwp.com>.



2. Execute o Local WP e crie um novo site. Siga os passos descritos do *software*, selecione a opção “Wordpress Site”, selecione um nome para o *site*, e defina um nome de usuário, senha e e-mail para a criação das credenciais de *login*.



3. Realize um clone (cópia) do repositório de código da plataforma no Bitbucket pelo link: <https://bitbucket.org/elizeuO/reverbera/src/master/> através da opção “Download repository” conforme a figura.

Reverbera

Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, [add a description to](#)

master Files Filter files

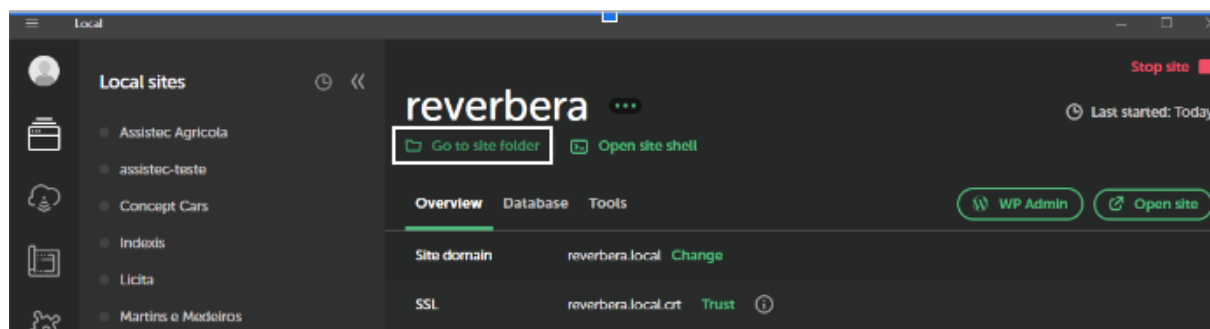
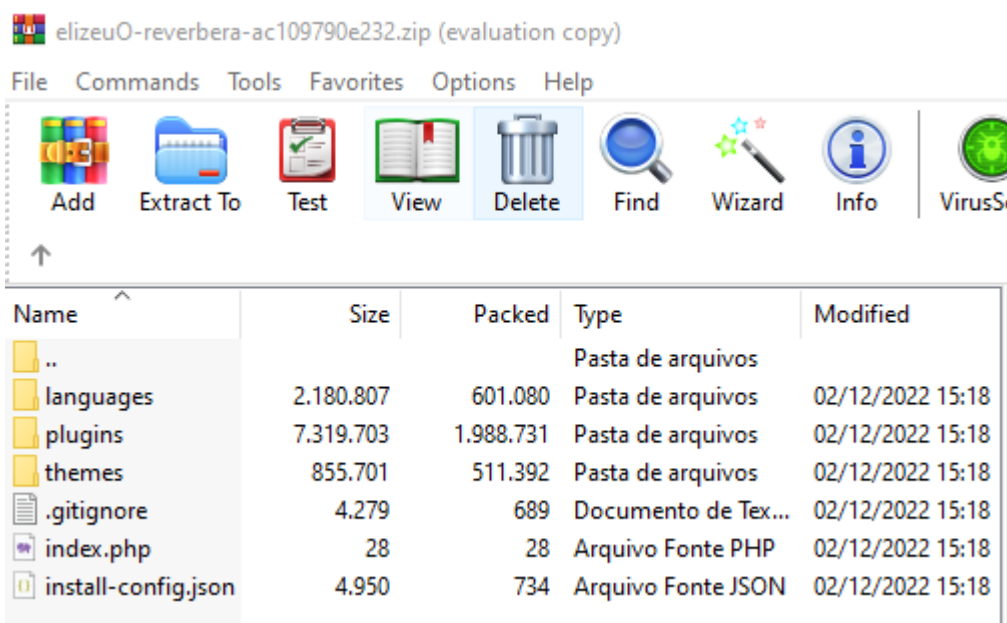
/

Name Size Last commit Message

Invite Clone

Fork this repository
Compare branches or tags
Add file
Manage notifications
Download repository

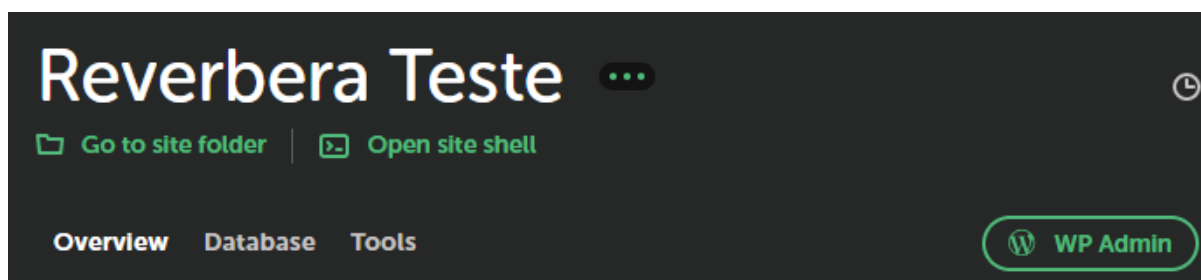
4. Será feito um *download* de um arquivo no formato *zip* com o conteúdo necessário para instalar o *site* por completo. Faça a cópia do conteúdo da pasta interna do arquivo *zip* e substitua o existente na pasta *wp-content* navegando pelo botão “*Go to site folder*” seguindo o caminho de pastas: “*\app\public\wp-content*”.



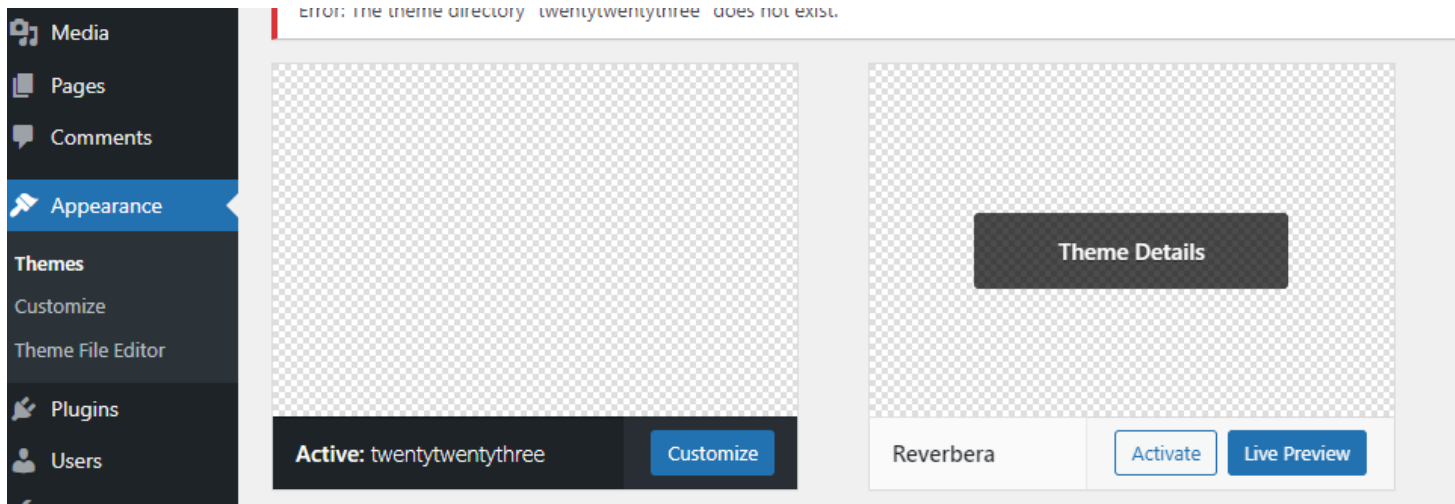
Elizeu Oliveira da Silva > Local Sites > reverbera-teste > app > public > wp-content

Nome	Data de modificação	Tipo	Tamanho
languages	02/12/2022 15:18	Pasta de arquivos	
plugins	02/12/2022 15:18	Pasta de arquivos	
themes	02/12/2022 15:18	Pasta de arquivos	
.gitignore	02/12/2022 15:18	Documento de Te...	5 KB
index	02/12/2022 15:18	Arquivo Fonte PHP	1 KB
install-config	02/12/2022 15:18	Arquivo Fonte JSON	5 KB

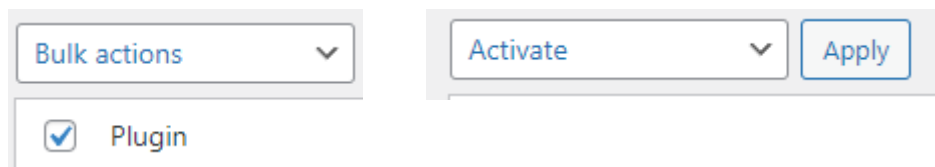
- Abra o painel de administrador através do botão “WP Admin” no Local WP, preencha com suas credenciais criadas (usuário e senha) e depois clique em “Log In”.



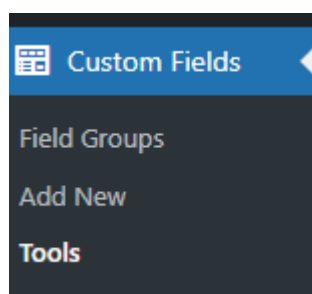
- Já no painel, no menu vá em clique em “Appearance”, passe o mouse na área com o nome “Reverbera” e clique em “Activate”. Agora o tema do *site* está instalado.



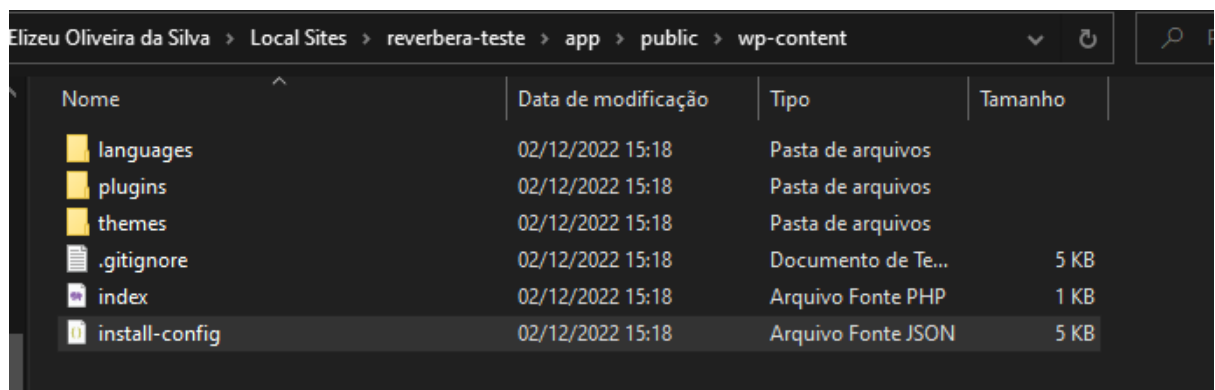
7. Agora, no meu vá em “*Plugins*”, marque a caixa na coluna “*Plugin*”, depois clique em “*Bulk actions*”, depois em “*Activate*” e por fim no botão “*Apply*”.



8. Passe o *mouse* em “*Custom Fields*” e clique em “*Tools*”. Já nessa página, você perceberá uma área de importação intitulada “*Select File*” dentro da área “*Import Field Groups*”. Vá novamente no botão “*Go to site folder*” do Local WP e navegue até a pasta “wp-content” seguindo o caminho: “\app\public\wp-content”. Agora arraste o arquivo “install-config.json” contido nesta pasta para a área de importação “*Select File*”. Clique em “*Import File*” para finalizar a importação dos campos de cadastro.



9. Novamente no menu do painel, cliquem em “Pages” e depois no botão “Add New”.



Import Field Groups

Select the Advanced Custom Fields JSON file you would like to import. When you click the import button below, ACF will import the field groups.

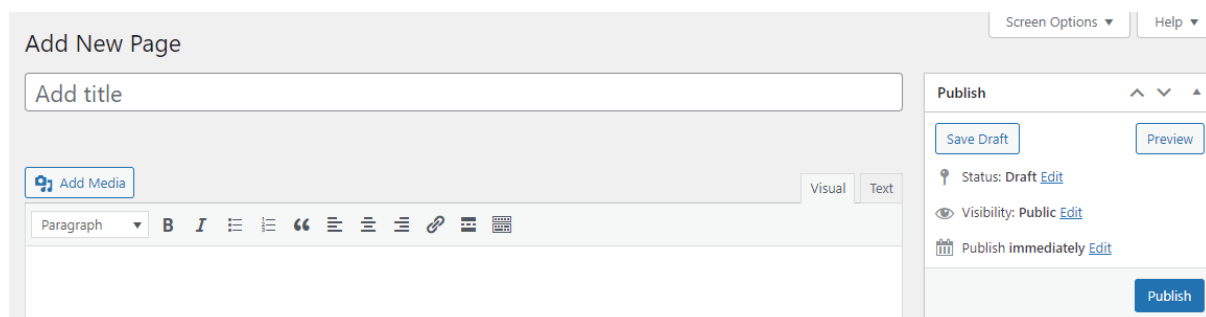
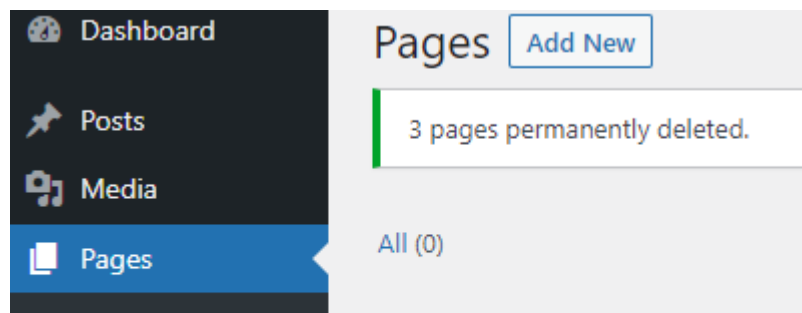
Select File

[Escolher arquivo](#) Nenhum arquivo escolhido

[Import File](#)

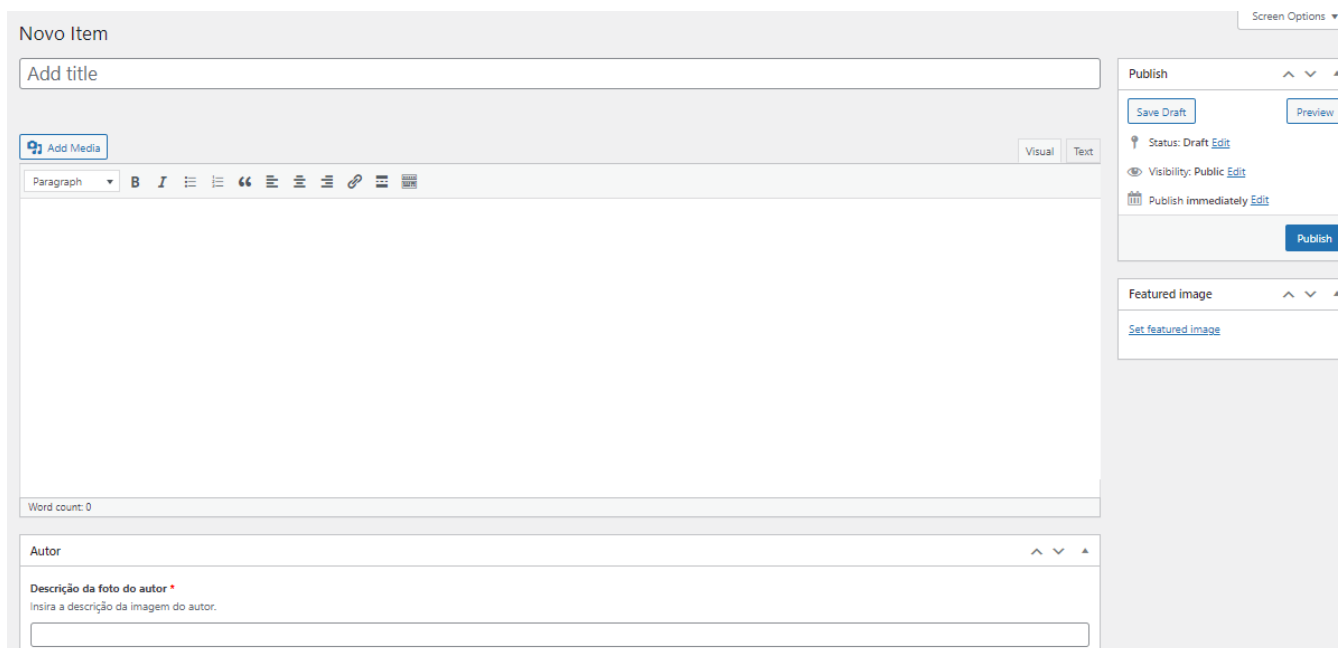
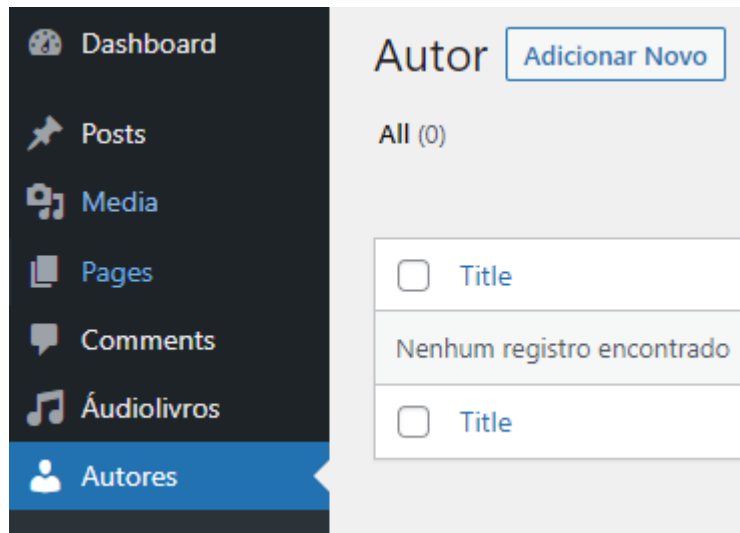
No campo “Add News Page”, digite “Ajuda de acessibilidade” e clique em “Publish” para registrar a página de ajuda de acessibilidade na instalação.

10. Realize o cadastro dos autores dos audiolivros, no menu administrativo clique no item



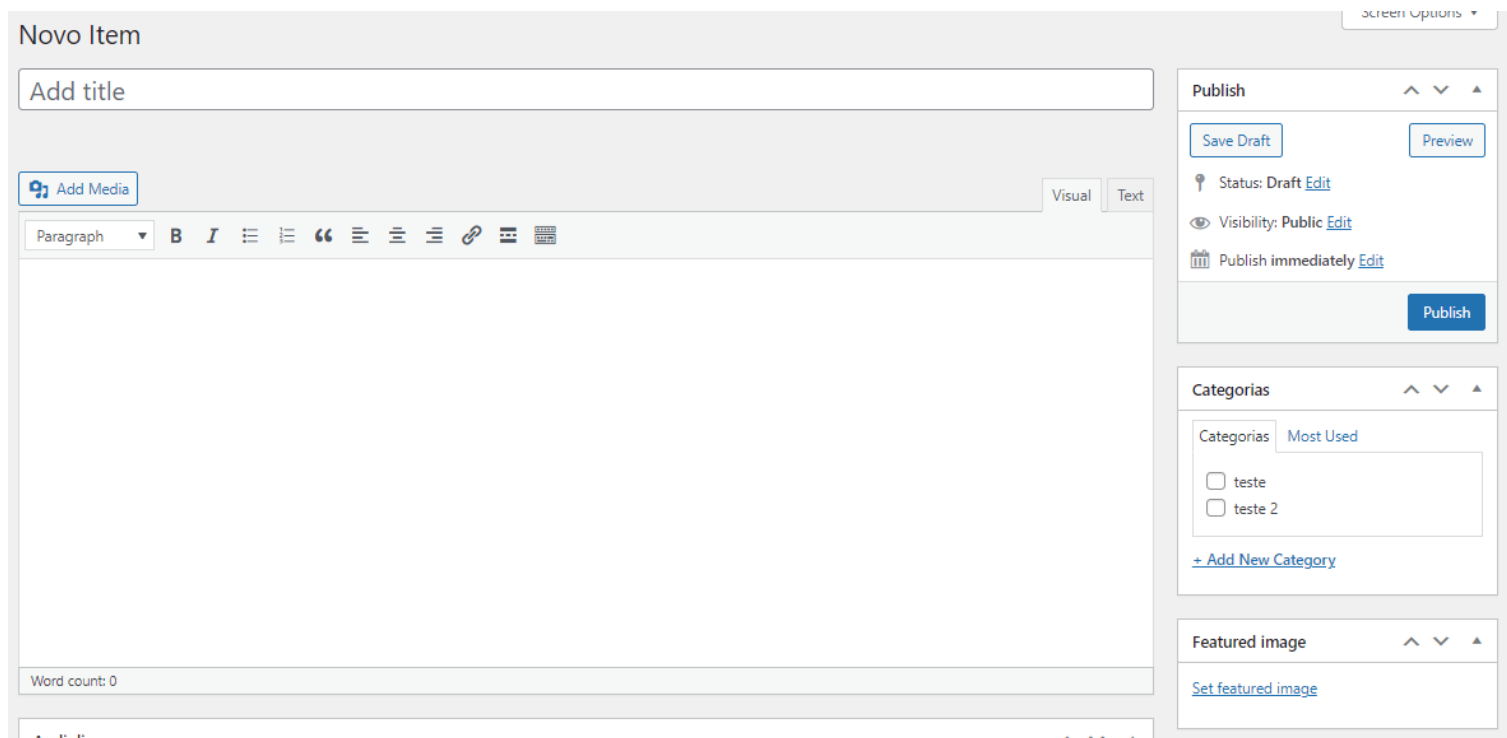
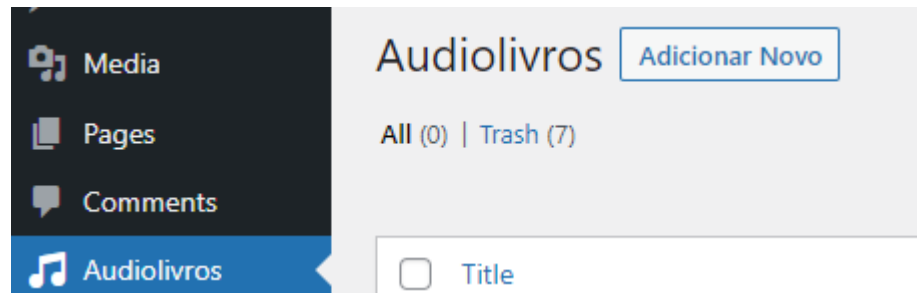
“Autores” e posteriormente no botão “Adicionar novo”. No campo “*Add title*”, insira o nome do autor. No próximo campo insira a biografia do autor. Em “Descrição da foto do autor”, insira uma breve frase descrevendo a imagem de capa do autor para os usuários de leitores de tela.

Em “*Featured image*”, insira a foto do autor. Para publicar, clique em “*Publish*”. Você pode alterar os autores cadastrados clicando no item desejado e alterando os dados já mencionados.



11. O próximo passo é cadastrar os audiolivros, no menu administrativo clique no item "Audiolivros" e em seguida em “Adicionar Novo”. No Campo “*Add title*”, insira o título do audiolivro. No campo abaixo, escreva a descrição do conteúdo. No campo “Audiolivro”, clique no botão e insira o arquivo no formato de áudio do audiolivro. Em “Autor”, selecione o autor cadastrado. Em “Locutor” e em “Editora” preencha

com os dados correspondentes, caso existam. No campo “Descrição da capa do Livro”, insira uma breve descrição da imagem da capa do livro para os usuários de leitores de tela. Em categorias defina as categorias do Audiolivro, nesta opção você pode utilizar categorias já criadas ou adicionar novas. Em “*Featured image*”, insira a imagem de capa do audiolivro. Para publicar, clique em “*publish*”. Você pode editar um audiolivro já cadastrado, clicando no desejado e alterando os dados que foram descritos.



Audiolivro

Audiolivro *

Insira o arquivo em formato mp3

No file selected

Add File

Autor *

Select

Locutor

Editora

Descrição da capa do livro *

Insira a descrição da capa

APÊNDICE E – MANUAL DE UTILIZAÇÃO

1. Para usuários com deficiência visual que necessitem a utilização de *softwares* leitores de tela, segue algumas sugestões de ferramentas: ChromeVox (extensão exclusiva do navegador Chrome), Jaws, NVDA, YeoSoft Text, VoiceOver.
2. Para acionar ou desligar a opção de alto contraste utilize o botão de contraste localizado no cabeçalho da página no canto direito.



Reverbera

[Ir para o conteúdo \[1\]](#) [Ir para o menu \[2\]](#) [Ajuda de acessibilidade \[3\]](#)

A+ A-

Contraste 

3. Para aumentar ou diminuir a fonte do conteúdo utilize os botões “A+” e “A-” respectivamente, localizados também no cabeçalho da página.



Reverbera

[Ir para o conteúdo \[1\]](#) [Ir para o menu \[2\]](#) [Ajuda de acessibilidade \[3\]](#)

A+ A-

Contraste 

4. A página inicial do *site* contém informações sobre o propósito da plataforma e uma listagem com os últimos audiolivros disponibilizados na plataforma.
5. Para filtrar por categoria os audiolivros, utilize a listagem lateral de categorias e acesse o *link* da categoria de sua preferência.

Categorias de audiolivros

[Comédia](#)

[Conto](#)

[Infantil](#)

6. Ao clicar no audiolivro, você será redirecionado para a página do mesmo, onde haverá descrito um resumo do audiolivro, o autor, categoria, editora, locutores, duração do arquivo e tamanho.
7. No *link* do autor haverá um resumo de sua história e os audiolivros cadastrados relacionados a ele.
8. O *player* de áudio possui os seguintes atalhos quando focados em suas barras de controles: “para cima” (avançar ou aumentar), “para baixo” (retroceder ou diminuir), “home” (início ou mutar), “end” (finalizar ou volume máximo).

9. Para mais informações sobre atalhos específicos para navegadores e informações de



Reverbera

[Ir para o conteúdo \[1\]](#)

[Ir para o menu \[2\]](#)

[Ajuda de acessibilidade \[3\]](#)

A+ A-

Contraste

leitores de tela, vá no cabeçalho e clique na opção “Ajuda de acessibilidade”.

APÊNDICE F – AVALIAÇÃO DE ACESSIBILIDADE

Teste de acessibilidade feito na plataforma ASES, link de acesso:

<https://asesweb.governoeletronico.gov.br/>

Página Avaliada

Página: <https://reverberatcc.000webhostapp.com/>

Título: Reverbera - Áudiolivros Gratuitos

Tamanho: 46062 Bytes

Data/Hora: 23/03/2023 13:15:18

Nota e Resumo da Avaliação de Acessibilidade



Legenda	
■	>= 95%
■	>= 85% < 95%
■	>= 70% < 85%
■	< 70%

Resumo de Acessibilidade por Seção eMAG

Seção	✖ Erro(s)	⚠ Aviso(s)
Marcação	1	59
Comportamento	1	4
Conteúdo/Informação	5	1
Apresentação / Design	0	1
Multimídia	0	10
Formulários	0	0
Total	7	75

Marcação	Comportamento	Conteúdo/Informação	Apresentação / Design	Multimídia	Formulários
✖ Erros da seção marcação					
Recomendação		Quantidade	Linha(s) do código fonte		
1.5 Fornecer âncoras para ir direto a um bloco de conteúdo.(link para um novo site)		1	73		
Marcação	Comportamento	Conteúdo/Informação	Apresentação / Design	Multimídia	Formulários
✖ Erros da seção comportamento					
Recomendação		Quantidade	Linha(s) do código fonte		
2.2 Garantir que os objetos programáveis sejam acessíveis.(link para um novo site)		1	1		
Marcação	Comportamento	Conteúdo/Informação	Apresentação / Design	Multimídia	Formulários
✖ Erros da seção conteúdo/informação					
Recomendação		Quantidade	Linha(s) do código fonte		
3.5 Descrever links clara e sucintamente.(link para um novo site)		5	253, 319, 385, 517, 451		