



INSTITUTO DO EMPREGO E FORMAÇÃO PROFISSIONAL, I.P.
CENTRO DE EMPREGO E FORMAÇÃO PROFISSIONAL DO PINHAL INTERIOR NORTE
SERVIÇO DE FORMAÇÃO PROFISSIONAL DE ARGANIL
C-EFPI – J+D – 2024 – *Linguagens de Programação – Programação Web*

Projeto de Tecnologias e Programação de Sistemas de Informação (UFCD 5425)

Portifólio de Ilustração e Animação



INSTITUTO DO EMPREGO E FORMAÇÃO PROFISSIONAL, I.P.
CENTRO DE EMPREGO E FORMAÇÃO PROFISSIONAL DO PINHAL INTERIOR NORTE
SERVIÇO DE FORMAÇÃO PROFISSIONAL DE ARGANIL
C-EFPI – J+D – 2024 – *Linguagens de Programação – Programação Web*

RENAN ABA DA SILVA

Linguagens de Programação – Programação Web

(Programa J+D - 1/2024)

Arganil, 04/11/2024

Índice de Conteúdos

Índice de Conteúdos	i
1. Introdução.....	1
1.1. Apresentação	1
1.2. Motivação	1
1.3. Contexto	1
1.4. Público-Alvo.....	2
1.5. Viabilidade	2
2. Objetivos.....	3
2.1. Objetivo geral	3
2.2. Objetivos específicos.....	3
3. Metodologia	4
3.1. Levantamento de requisitos.....	4
3.2. Protótipo	4
3.3. Definição das tecnologias a serem empregadas.....	4
3.4. Desenvolvimento do código.....	5
4. Protótipo	6
4.1. Página inicial	6
4.2. Seção de projetos.....	7
4.3. Página sobre	8
4.4. Responsividade	9
5. Tecnologias e recursos	10
5.1. Figma	10
5.2. Vite	10
5.3. React.....	11
5.3.1. React Router Dom	11
5.4. Sass	11
5.5. Typescript	12
6. Desenvolvimento	13
6.1. Estruturação do Projeto	13
6.2. Desenvolvimento da interface com React.....	15
6.3. Estilização com Sass.....	16
6.4. Tipagem com TypeScript	17
6.5. Implementação das funcionalidades.....	18
7. Conclusão e Trabalhos Futuros	22

7.1.	Conclusão	22
7.2.	Trabalhos Futuros.....	22
8.	Referências	24

1. Introdução

1.1. Apresentação

A intenção do projeto é desenvolver um site de portfólio para um artista de ilustração e animação, com o objetivo de mostrar seu trabalho de maneira profissional e acessível. O artista terá a oportunidade de mostrar suas obras, divulgar detalhes de seus projetos e se comunicar com possíveis clientes ou colaboradores através de um portfólio online.

1.2. Motivação

Criar um site de portfólio é essencial para artistas que buscam expandir sua visibilidade no mercado. Com o avanço das tecnologias digitais e o aumento do uso de plataformas online, ter um lugar exclusivo para o artista mostrar suas obras e métodos de criação é essencial para divulgar seu trabalho e atrair clientes. Além do mais, a plataforma oferece independência para que o artista possa controlar sua imagem e divulgar suas obras com eficiência.

1.3. Contexto

Nos últimos anos, a indústria da ilustração e animação tem crescido devido ao aumento da procura por conteúdo visual em setores como publicidade, entretenimento e design digital, principalmente para ser utilizado nas plataformas digitais. Possuir um portfólio digital é fundamental para chamar a atenção e disponibilizar aos clientes uma forma conveniente de examinar o trabalho do artista. O projeto do site será elaborado com referência às técnicas mais eficazes de design e usabilidade, ao empregar recursos de interface intuitiva e responsiva.

1.4. Público-Alvo

O público-alvo do website inclui: agências de publicidade e design à procura de colaboradores para trabalhos visuais; produtoras de animação em busca de artistas com portfólios de animações 2D e 3D; clientes individuais que desejam contratar o artista para trabalhos personalizados, como ilustrações, concept art ou animações para projetos audiovisuais; fãs e seguidores do artista que desejam acompanhar suas atualizações e novos trabalhos.

1.5. Viabilidade

O projeto é tecnicamente viável, pois emprega ferramentas comuns e amplamente utilizadas para o desenvolvimento de websites, como Vite React, HTML5, CSS3 e Typescript. Também será utilizada a plataforma Figma para realizar a prototipagem.

O custo de desenvolvimento é moderado, e o artista poderá autogerir o site após a conclusão deste, sem a necessidade de suporte técnico constante. O design responsivo garantirá que o portfólio seja acessível tanto em dispositivos móveis quanto em desktops.

2. Objetivos

2.1. Objetivo geral

O objetivo geral deste trabalho é desenvolver um website de portfólio profissional para um artista de ilustração e animação, utilizando as melhores práticas de desenvolvimento web e design, permitindo a exibição eficiente dos trabalhos do artista e promovendo sua presença online.

2.2. Objetivos específicos

Os objetivos específicos do projeto incluem planejar e implementar a estrutura do website com foco em responsividade, garantindo uma experiência de navegação fluida em diferentes dispositivos (desktop, tablet e mobile); desenvolver um sistema de galeria interativa para exibir as ilustrações e animações do artista de forma otimizada e visualmente atraente; implementar uma interface de usuário (UI) intuitiva e simples, focada na facilidade de navegação e usabilidade; integrar um formulário de contato funcional que permita a comunicação direta com potenciais clientes ou parceiros; garantir a otimização do desempenho do site, aplicando técnicas de compressão de imagens e scripts para reduzir o tempo de carregamento e incorporar práticas de SEO, assegurando que o site tenha boa visibilidade nos motores de busca.

3. Metodologia

O desenvolvimento do website será realizado em três etapas, utilizando uma abordagem iterativa e incremental. Cada fase será acompanhada por testes e ajustes, garantindo que o projeto atenda aos requisitos técnicos e de design estabelecidos no planejamento. A seguir, detalham-se as etapas previstas para o projeto.

3.1. Levantamento de requisitos

O levantamento de requisitos inicia-se pela coleta de informações sobre o conteúdo a ser exibido no website (ilustrações, animações, biografia, etc.). Depois, será feita a definição das funcionalidades essenciais (galeria de imagens, sistema de contato, biografia). Por fim, o design do website proposto pelo artista será analisado sob uma ótica funcional, para garantir que siga os preceitos de usabilidade e que sua implementação seja viável.

3.2. Protótipo

Em um segundo momento, será desenvolvido um protótipo de média fidelidade para validar o layout e os elementos de design utilizando a plataforma Figma. Serão também realizadas reuniões de feedback com o artista para ajustes no design e na funcionalidade do website, garantindo a satisfação do cliente.

3.3. Definição das tecnologias a serem empregadas

Após as etapas previamente descritas, será realizada uma pesquisa de tecnologias empregadas atualmente no desenvolvimento web e, a seguir, serão definidas quais destas tecnologias se adequam melhor ao desenvolvimento deste projeto.

3.4. Desenvolvimento do código

Por fim, utilizando as tecnologias definidas anteriormente, será desenvolvido o código do website. Ao longo do processo, serão realizados testes de usabilidade e reponsividade para garantir a qualidade do código.

4. Protótipo

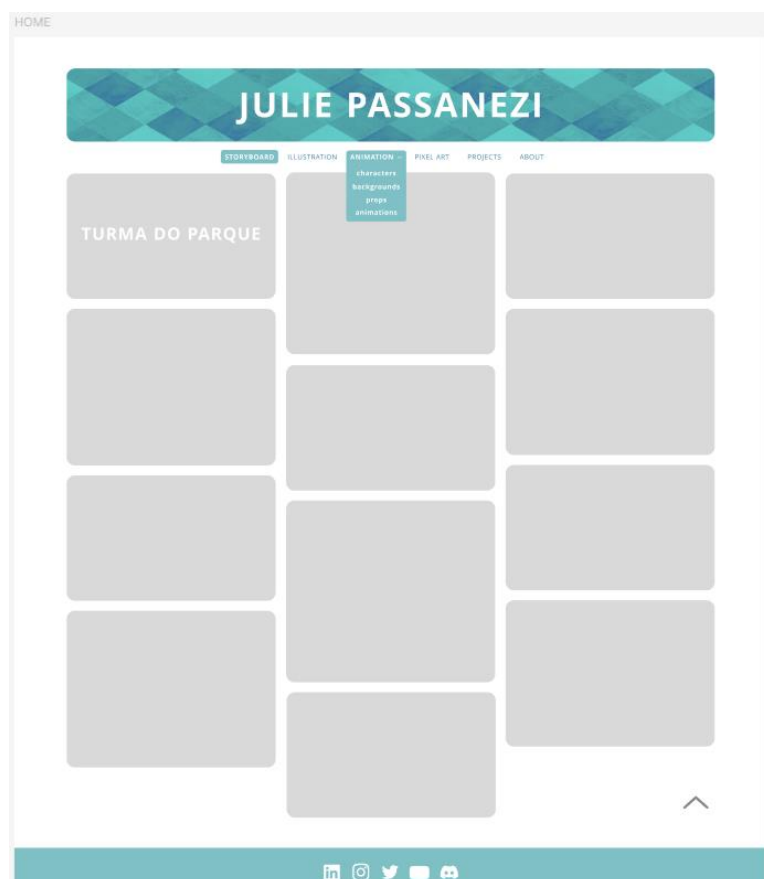
O protótipo de alta fidelidade foi desenvolvido utilizando a plataforma Figma, garantindo que o design do website de portfólio fosse visualmente consistente com os objetivos propostos. Esse protótipo oferece uma visão clara de como o website funcionará e quais serão seus principais componentes visuais, como tipografia, paleta de cores e layout.

4.1. Página inicial

A página inicial foi projetada para causar impacto visual, apresentando as ilustrações e animações em destaque, além de links rápidos para as seções de *storyboard*, *illustration*, *animation*, *pixel art*, *projects* e *about* (**Figura 1**).

Figura 1

Protótipo do Figma para a página inicial

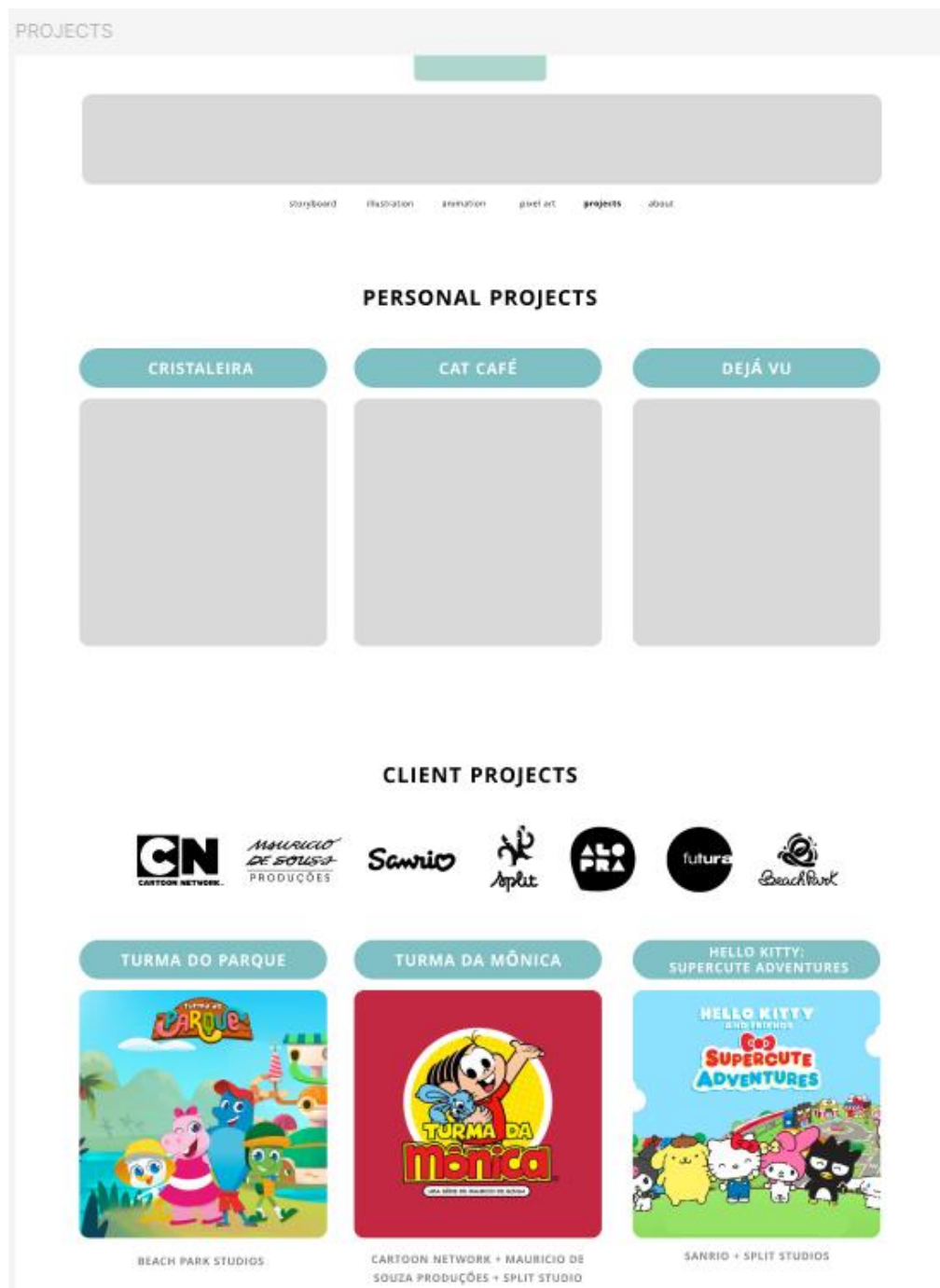


4.2. Seção de projetos

A seção de projetos (*projects*) exibe os trabalhos do artista em uma galeria interativa. Cada projeto pode ser clicado para expandir e visualizar mais detalhes, como descrições, imagens adicionais e vídeos (**Figura 2**).

Figura 2

Protótipo do Figma para a página de projetos

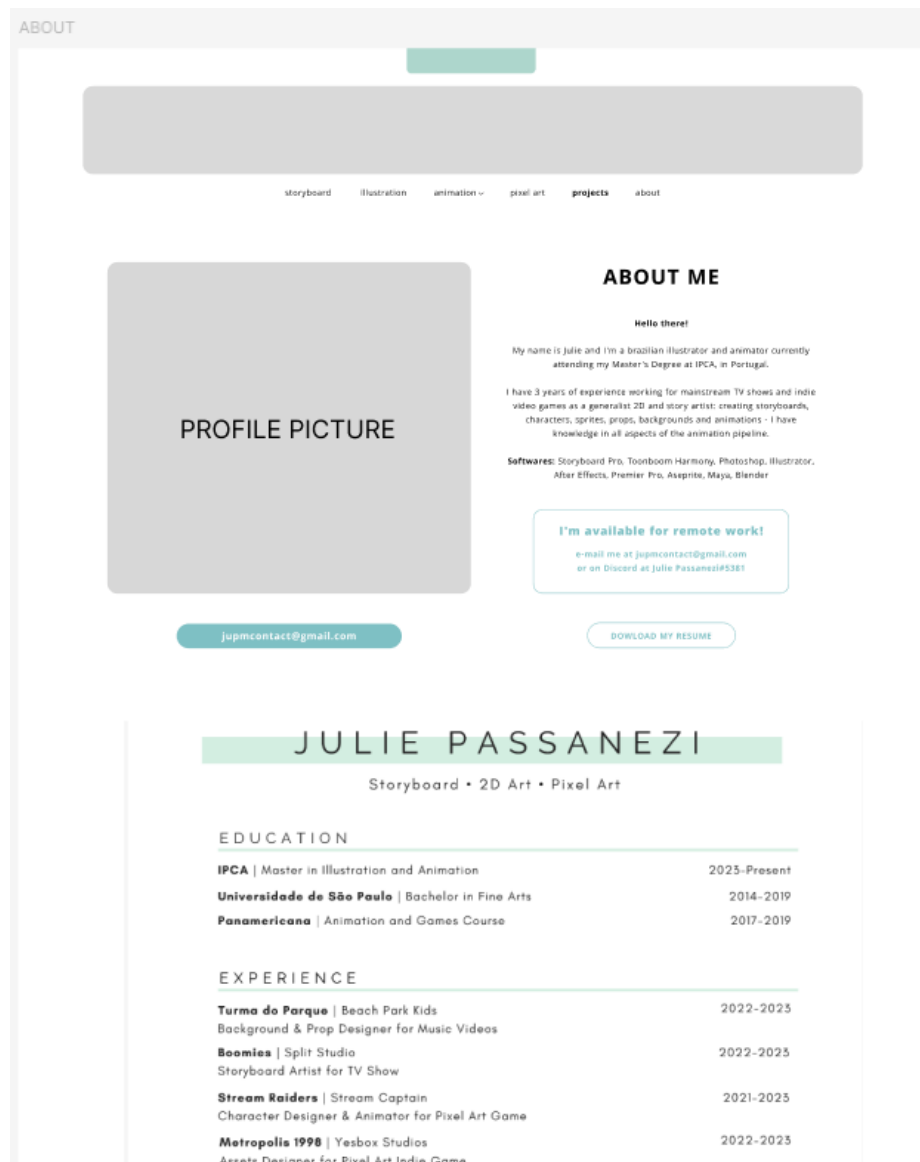


4.3. Página sobre

A página *about* (sobre) combina a biografia e o contato em uma única sessão. Ela apresenta uma breve narrativa sobre a carreira e conquistas do artista, bem como um formulário de contato simples para facilitar a comunicação com clientes e parceiros. Além disso, inclui um formulário para comunicação **(Figura 3)**.

Figura 3

Protótipo do Figma para a página sobre

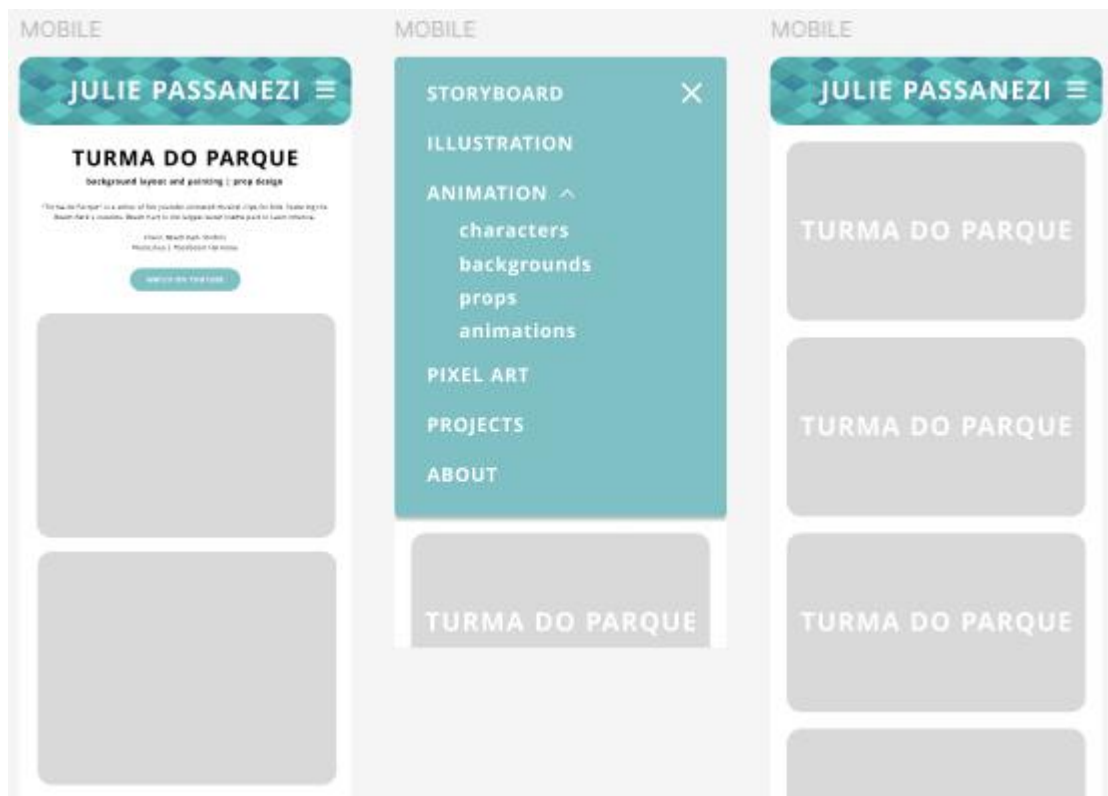


4.4. Responsividade

O protótipo foi desenvolvido com foco em uma interface responsiva, garantindo que o website funcione de maneira otimizada em diferentes dispositivos, como smartphones e tablets **(Figura 4)**.

Figura 4

Protótipo do Figma para as páginas mobile



5. Tecnologias e recursos

O desenvolvimento do website de portfólio contará com um conjunto de tecnologias modernas, que oferecem eficiência e flexibilidade tanto no design quanto na implementação. A seguir estão as ferramentas que serão utilizadas e suas respectivas funções no projeto.

5.1. Figma

O Figma será utilizado para a criação de protótipos e wireframes, permitindo o design colaborativo e a definição da estrutura visual do website. Como uma ferramenta de design de interface, o Figma facilita a criação de layouts responsivos, que poderão ser testados e ajustados antes de serem implementados.

- **Funções:** Criação de protótipos de alta fidelidade, colaboração em design, layouts responsivos.
- **Justificativa:** O Figma permite visualizações claras do design e navegação do site, facilitando a comunicação com o artista para ajustes visuais.

5.2. Vite

Vite será utilizado como o bundler para o desenvolvimento do projeto, proporcionando um ambiente de desenvolvimento rápido e moderno, com suporte a hot module replacement (HMR). Essa ferramenta acelera o processo de desenvolvimento ao permitir compilações mais rápidas e eficientes.

- **Funções:** Bundler de projeto, ambiente de desenvolvimento rápido com suporte a módulos ECMAScript.
- **Justificativa:** Vite oferece melhorias significativas no tempo de desenvolvimento e na experiência do desenvolvedor, otimizando o processo de construção da aplicação.

5.3. React

React será o framework principal utilizado no desenvolvimento da interface do website. Com sua abordagem baseada em componentes, React facilita a criação de uma interface dinâmica e interativa, permitindo a reutilização de elementos e melhorando a organização do código.

- **Funções:** Biblioteca JavaScript para construção da interface de usuário (UI) baseada em componentes.
- **Justificativa:** A escolha de React se deve à sua popularidade e flexibilidade, além da capacidade de criar interfaces interativas e altamente escaláveis.

5.3.1. React Router Dom

React Router Dom será responsável pelo gerenciamento das rotas e navegação dentro do website. Ele permitirá a criação de uma aplicação de página única (SPA), garantindo transições suaves entre as páginas do portfólio e uma experiência de navegação rápida e fluida para os usuários.

- **Funções:** Gerenciamento de rotas para navegação entre as páginas do site.
- **Justificativa:** React Router Dom facilita o desenvolvimento de uma SPA, onde o carregamento de novas páginas acontece sem interrupções na experiência do usuário.

5.4. Sass

Sass será utilizado para o gerenciamento de estilos do website. Como um pré-processador CSS, o Sass permite a utilização de funcionalidades avançadas, como variáveis, aninhamento e mixins, facilitando a organização e manutenção do código de estilos.

- **Funções:** Pré-processador CSS para melhorar a organização e escrita de estilos.

- **Justificativa:** Sass oferece maior controle e flexibilidade sobre os estilos do projeto, possibilitando um desenvolvimento mais rápido e com menos redundância de código.

5.5. Typescript

TypeScript será adotado no desenvolvimento para adicionar tipagem estática ao JavaScript, proporcionando maior segurança e previsibilidade no código. Isso facilitará a detecção de erros durante o desenvolvimento, além de melhorar a manutenção e escalabilidade do projeto.

- **Funções:** Superset do JavaScript com tipagem estática para detecção precoce de erros.

- **Justificativa:** O uso de TypeScript reduz a ocorrência de bugs e facilita a manutenção do projeto, sendo ideal para projetos de médio a grande porte.

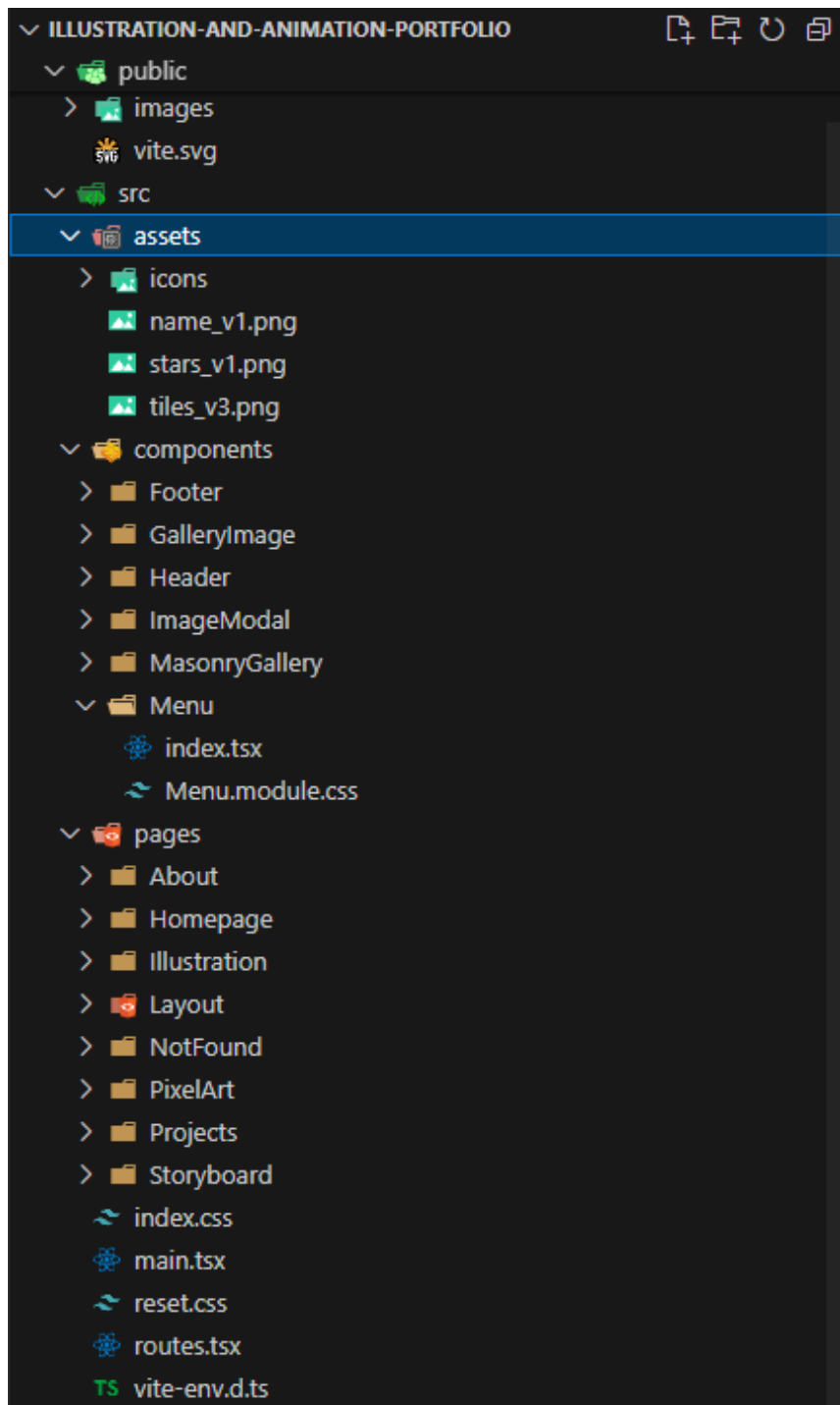
6. Desenvolvimento

O desenvolvimento do website foi estruturado em etapas, desde o planejamento até a implementação das funcionalidades e componentes de design. Cada etapa do desenvolvimento focou em garantir a eficiência e a qualidade final do produto, utilizando as tecnologias definidas na metodologia.

6.1. Estruturação do Projeto

A estrutura do projeto foi criada utilizando Vite como ambiente de desenvolvimento, que fornece uma configuração otimizada para integrar tecnologias modernas, como React, TypeScript e Sass. O diretório foi organizado de forma modular, com pastas dedicadas a componentes, estilos, rotas e assets (imagens e vídeos) (**Figura 5**).

Figura 5
Organização dos diretórios



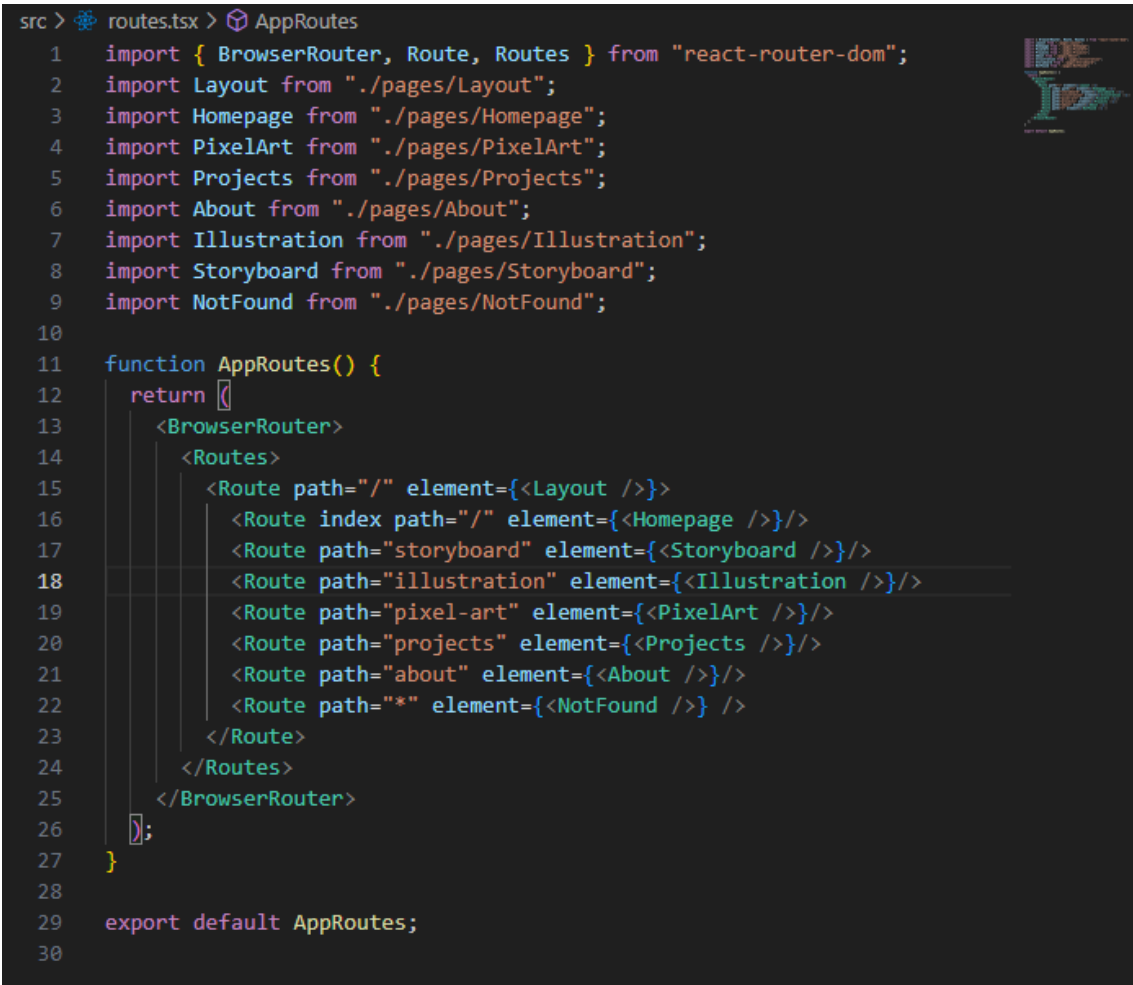
6.2. Desenvolvimento da interface com React

O React foi utilizado para criar uma interface modular, permitindo o desenvolvimento de componentes reutilizáveis e organizados. Cada componente foi projetado para poder ser reutilizado em outras partes do site, podendo ser compartilhados por mais de uma página.

React Router Dom foi integrado para gerenciar a navegação entre as páginas, garantindo uma experiência de usuário fluida em uma aplicação de página única (SPA) (**Figura 6**).

Figura 6

Gerenciamento da navegação



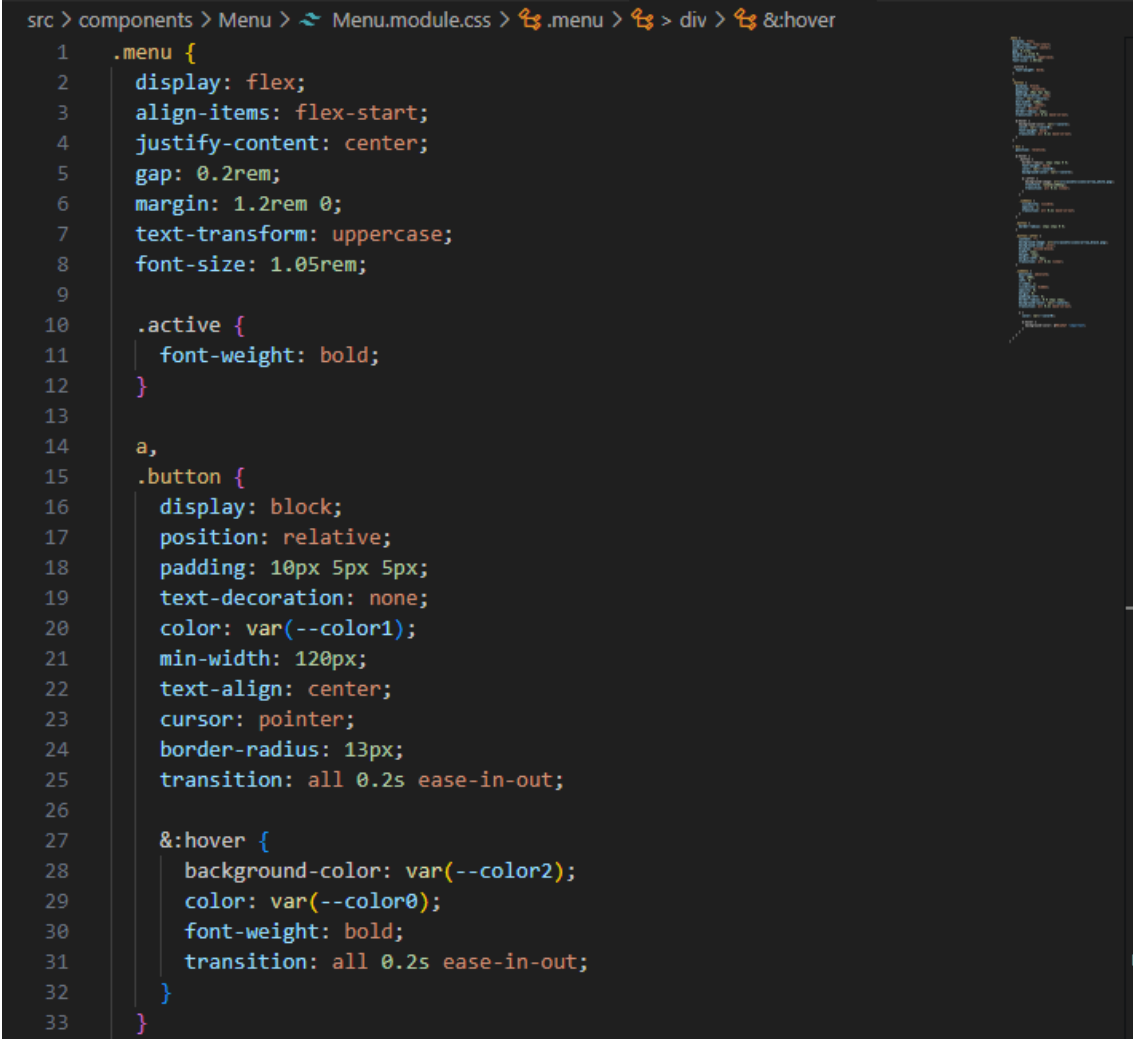
```
src > routes.tsx > AppRoutes
1  import { BrowserRouter, Route, Routes } from "react-router-dom";
2  import Layout from "./pages/Layout";
3  import Homepage from "./pages/Hompage";
4  import PixelArt from "./pages/PixelArt";
5  import Projects from "./pages/Projects";
6  import About from "./pages/About";
7  import Illustration from "./pages/Illustration";
8  import Storyboard from "./pages/Storyboard";
9  import NotFound from "./pages/NotFound";
10
11 function AppRoutes() {
12   return (
13     <BrowserRouter>
14       <Routes>
15         <Route path="/" element={<Layout />}>
16           <Route index path="/" element={<Homepage />} />
17           <Route path="storyboard" element={<Storyboard />} />
18           <Route path="illustration" element={<Illustration />} />
19           <Route path="pixel-art" element={<PixelArt />} />
20           <Route path="projects" element={<Projects />} />
21           <Route path="about" element={<About />} />
22           <Route path="*" element={<NotFound />} />
23         </Route>
24       </Routes>
25     </BrowserRouter>
26   );
27 }
28
29 export default AppRoutes;
30
```

6.3. Estilização com Sass

Os estilos foram gerenciados com Sass, aproveitando variáveis e aninhamento para uma organização eficiente do CSS (**Figura 7**).

Figura 7

Exemplo de aninhamento utilizando SASS



```
src > components > Menu > Menu.module.css > .menu > > div > &:hover
1  .menu {
2    display: flex;
3    align-items: flex-start;
4    justify-content: center;
5    gap: 0.2rem;
6    margin: 1.2rem 0;
7    text-transform: uppercase;
8    font-size: 1.05rem;
9
10   .active {
11     font-weight: bold;
12   }
13
14   a,
15   .button {
16     display: block;
17     position: relative;
18     padding: 10px 5px 5px;
19     text-decoration: none;
20     color: var(--color1);
21     min-width: 120px;
22     text-align: center;
23     cursor: pointer;
24     border-radius: 13px;
25     transition: all 0.2s ease-in-out;
26
27     &:hover {
28       background-color: var(--color2);
29       color: var(--color0);
30       font-weight: bold;
31       transition: all 0.2s ease-in-out;
32     }
33   }
```

A paleta de cores, a tipografia e os espaçamentos foram definidos globalmente, seguindo a identidade visual do artista, e aplicados de forma consistente em todas as páginas do site (**Figura 8**).

Figura 8

Variáveis globais

```
1  @charset "UTF-8";
2
3  @import url('https://fonts.googleapis.com/css2?family=Khula:wght@400;
4  500;600;700&display=swap');
5
6  :root {
7    --font0: 'Khula', Arial, Helvetica, sans-serif;
8    --color0: #ffffff;
9    --color1: #000000;
10   --color2: #7EC0C4;
11   --color3: #878787;
12 }
13
14 body, html {
15   background-color: var(--color0);
16   font-family: var(--font0);
17   scroll-behavior: smooth !important;
18 }
```

6.4. Tipagem com TypeScript

O uso de TypeScript garantiu uma codificação mais segura e menos suscetível a erros, proporcionando tipagem estática para variáveis, propriedades e funções (**Figura 9**). Essa abordagem ajudou a prevenir possíveis erros durante o desenvolvimento e contribuiu para a escalabilidade do projeto, facilitando manutenções futuras.

Figura 9

Tipagem estática de variáveis utilizando Typescript

```
src > components > Menu > index.tsx > Menu > pages.map() callback > p
1  import { NavLink } from "react-router-dom";
2  import styles from "../Menu.module.css"
3
4  const pages: Array<{
5    name: string;
6    path?: string;
7    submenu?: Array<{
8      name: string;
9      path: string;
10     }>;
11 > }> = [ ...
57 ];
58
59 const Menu = () => {
60   return(
61     <nav className={styles.menu}>
62       {pages.map((page) => {
63         if (page.submenu !== undefined) {
64           return (
65             <div key={page.name}>
66               <div
67                 tabIndex={0}
68                 role="button"
69                 className={styles.button}
70               >
71                 {page.name}
```

6.5. Implementação das funcionalidades

A seção de projetos (*projects*) inclui uma galeria interativa (**Figura 10**), onde pode-se clicar em cada projeto para exibir detalhes adicionais em um modal de visualização ampliada (**Figura 11**). Este modal permite que o usuário navegue pelo array da galeria de imagens, avançando e retrocedendo entre elas com auxílio do hook `useState` do React (**Figura 12**), proporcionando uma experiência de visualização fluida.

Figura 10

Galeria principal no estilo masonry

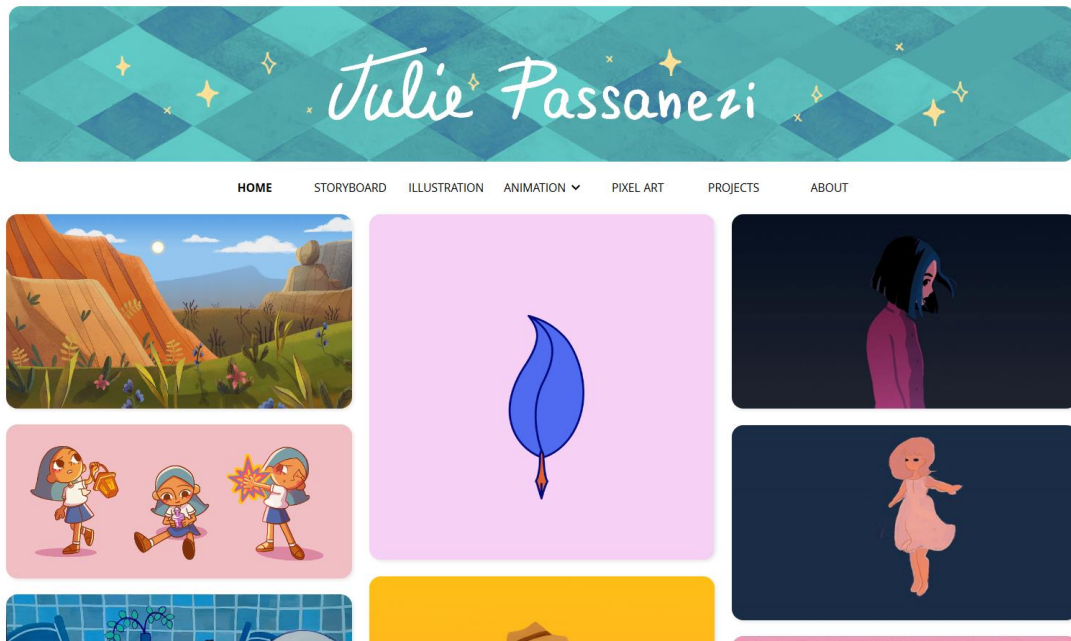


Figura 11

Modal com navegação pela galeria

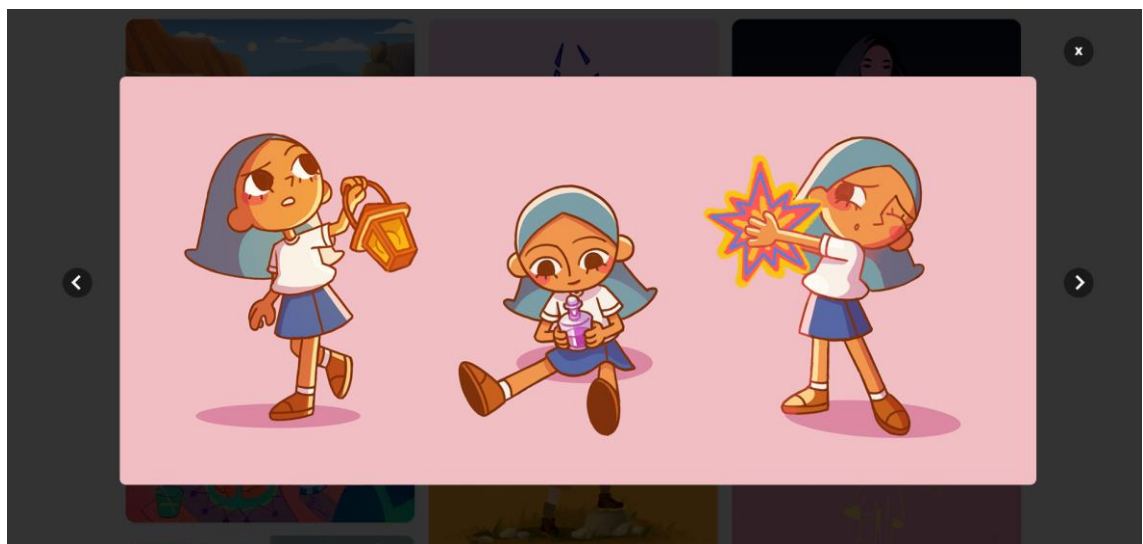


Figura 12

Utilização do hook `useState` para gerenciamento do Modal

```
src > components > MasonryGallery > index.tsx > [MasonryGallery]
84  const MasonryGallery = () => {
85    const [currentImageModal, setCurrentImageModal] = useState<number |
      null>(null);
86
87    const openModal = (index: number) => {
88      setCurrentImageModal(index);
89    };
90
91    const closeModal = () => {
92      setCurrentImageModal(null);
93    };
94
95    const nextImage = () => {
96      if (currentImageModal !== null){
97        setCurrentImageModal((currentImageModal + 1) % homeGallery.
          length);
98      }
99    };
100
101    const prevImage = () => {
102      if (currentImageModal !== null){
103        setCurrentImageModal((currentImageModal - 1 + homeGallery.
          length) % homeGallery.length);
104      }
105    };
106
107    return [
108      <div>
109        <div className={styles.masonryGallery}>
110          {homeGallery.map((galleryImg, index) => {
111            return (
112              <GalleryImage
113                key={galleryImg.path} ...
114                openModal={() => openModal(index)}
115              />
116            );
117          })}
118        </div>
119      ]
120    );
121
122    {currentImageModal !== null && (
123      <ImageModal
124        src={homeGallery[currentImageModal].path}
125        alt={homeGallery[currentImageModal].alt}
126        onClose={closeModal}
127        onNext={nextImage}
128        onPrev={prevImage}
129      />
130    )}
131  ]
132 }
```


Como previsto, todo o site foi desenvolvido com uma abordagem responsiva, assegurando que o conteúdo seja acessível e visualmente agradável em dispositivos móveis, tablets e desktops (**Figura13**).

Figura 13

Exemplo de utilização das Media Queries na Galeria Masonry

```
src > components > MasonryGallery > ↻ MasonryGallery.module.css > {}  
1  ∨ .masonryGallery {  
2    column-gap: 1.5em;  
3    column-count: 1;  
4    margin: 2em 1em;  
5    text-align: center;  
6  }  
7  
8  ∨ @media screen and (min-width: 768px) {  
9    ∨ .masonryGallery {  
10     column-count: 2;  
11     margin: 0 auto;  
12     padding: 0 2em;  
13   }  
14 }  
15  
16 ∨ @media screen and (min-width: 992px) {  
17 ∨ .masonryGallery {  
18   column-count: 3;  
19   padding: 0 3em;  
20   max-width: 1500px;  
21 }  
22 }  
23
```

7. Conclusão e Trabalhos Futuros

7.1. Conclusão

O desenvolvimento do website foi realizado com foco na criação de uma página inicial completa e funcional, que inclui uma galeria interativa para exibição dos projetos do artista, além de componentes essenciais, como header, footer e modal. A escolha cuidadosa das tecnologias e ferramentas, aliada a uma abordagem modular e organizada, garantiu que a página inicial oferecesse uma experiência de usuário fluida, responsiva e visualmente atraente, alinhada com a identidade do artista. Embora o escopo inicial tenha sido concluído com sucesso, o desenvolvimento das demais páginas do site permanece uma tarefa futura.

7.2. Trabalhos Futuros

Para que o website alcance seu potencial máximo, o próximo passo é a implementação das páginas adicionais que complementarão a página inicial e proporcionarão uma experiência completa para o usuário. Nomeadamente, é necessário completar a Seção de Projetos e desenvolver a página About. Além disso, devem ainda ser realizadas otimizações de SEO.

A implementação da Seção de Projetos de forma completa implica desenvolver uma página dedicada para cada projeto. Isso permitirá uma descrição detalhada e personalizada para cada trabalho do artista, com imagens adicionais e insights sobre o processo criativo.

O desenvolvimento da página About envolve a criação de uma página informativa com a biografia do artista, formulário de contato e links para redes sociais, permitindo que os visitantes do website conheçam mais sobre o artista e possam contatá-lo diretamente.

Por fim, ainda é necessário realizar ajustes de SEO para as novas páginas e otimizações adicionais no desempenho e carregamento de imagens, para que se assegure uma experiência de navegação rápida e de alta qualidade em todas as páginas.

Esses trabalhos futuros permitirão que o website torne-se mais completo e aprofundado, expandindo a presença digital do artista e oferecendo uma navegação rápida e rica em conteúdo para os visitantes.

8. Referências

Figma. (n.d.). *Figma help center*. <https://help.figma.com/hc/en-us>

Vite. (n.d.). *Vite documentation* . <https://pt.vite.dev/guide/>

React. (n.d.). *Getting started with React*. <https://reactjs.org/docs/getting-started.html>

React Router. (n.d.). *React Router documentation*.
<https://reactrouter.com/en/main/start/overview>

Sass. (n.d.). *Sass documentation*. <https://sass-lang.com/documentation>

TypeScript. (n.d.). *TypeScript documentation*. <https://www.typescriptlang.org/docs/>

W3Schools. (n.d.). *HTML tutorial*. <https://www.w3schools.com/html/default.asp>

W3Schools. (n.d.). *CSS tutorial*. <https://www.w3schools.com/css/default.asp>