

PROGRAMAÇÃO DE UM *FRONT-END* ESCALÁVEL UTILIZANDO A TECNOLOGIA *REACT* PARA A PLATAFORMA DIGITAL ARTE DE CADERNO

Beatriz do Prado Neaime¹

Douglas F. de Sousa Nunes²

Resumo

Arte de Caderno é um projeto de inclusão artística desenvolvido dentro do Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais, *campus* Poços de Caldas, cujo objetivo é incentivar a manifestação da arte enquanto evita a vandalização do espaço público. Atualmente, o processo de inscrição de obras é feito manualmente, através de serviços postais. A proposta deste trabalho consiste na codificação de uma aplicação web *front-end* utilizando a tecnologia *React*, como parte da plataforma digital Arte de Caderno, com vistas a facilitar a forma de inscrição, submissão e avaliação de obras dos concursos relacionados ao Arte de Caderno.

Palavras-chave: *React*; Arte de Caderno; *Front-end*.

1. Introdução

Com o objetivo de evitar a vandalização do espaço público em escolas, sem, ao mesmo tempo, proibir a manifestação da arte, o professor Márcio Luiz Bess, com apoio do Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais, *campus* Poços de Caldas, elaborou um projeto de extensão chamado Arte de Caderno. A iniciativa recebe, anualmente, milhares de desenhos feitos espontaneamente por alunos vinculados à rede pública de ensino de todo o país, desde o ensino infantil até a graduação. Depois de recebidas, as obras são avaliadas por uma banca de avaliadores, de forma a eleger aqueles que se destacaram de alguma forma. Além do engajamento da população com o instituto, também é promovido um estudo sócio-cultural baseado em cada desenho recebido, onde são extraídas informações do contexto dos participantes, tais como situação

¹ Graduanda em Engenharia de Computação pelo IFSULDEMINAS, *campus* Poços de Caldas.
beatriz.neaime@alunos.if sulde minas.edu.br

² Doutor em Ciências da Computação e Matemática Computacional pela Universidade de São Paulo e professor do curso da área de computação do IFSULDEMINAS, *campus* Poços de Caldas.
douglas.nunes@if sulde minas.edu.br

socioeconômica, cultura local, nível de escolaridade, entre outros aspectos importantes na formação do indivíduo.

Ainda que o Arte de Caderno tenha chegado ao âmbito nacional, os métodos utilizados para inscrição de participantes, envio de obras artísticas para os concursos, assim como para os julgamentos das artes, são bem limitados, sendo quase todos os processos realizados de forma manual. Cada participante ou professor responsável precisa realizar a impressão, preenchimento da ficha de inscrição, anexar a folha com a obra junto à ficha e, então, postar ambos os documentos em uma empresa de serviço postal para que a inscrição seja de fato realizada. Uma vez em posse da organização do projeto, o desenho é separado em sua respectiva categoria para, então, ser avaliado pela banca avaliadora. As obras classificadas são então escaneadas para um arquivo digital no intuito de ser divulgada nas mídias sociais para a fase de engajamento do público.

2. Objetivo Geral

Implementar uma ferramenta web para que o processo de inscrição, submissão de obras e avaliação das mesmas possam ser realizados todos de forma *online*, com o suporte de tecnologias computacionais.

2.1 Objetivos Específicos

1. Implementar interfaces para cadastro de alunos, professores e avaliadores.
2. Implementar interface para cadastro de obras de arte.
3. Implementar áreas de inscrição de alunos e professores participantes, além de campo para inscrição das obras.
4. Implementar interface para galeria de obras cadastradas na plataforma.
5. Otimizar a interface para diferentes dispositivos, incluindo *desktops*, *tablets* e *smartphones*.
6. Implementar interfaces para apoiar o processo de avaliação/julgamento das obras enviadas pela plataforma digital.
7. Documentar o processo de desenvolvimento para futuras manutenções.

3. Fundamentação Teórica

Segundo Lopez (2017), o *React* é uma biblioteca *JavaScript* utilizada por grandes plataformas, como *Netflix* e *Instagram*, para o desenvolvimento de interfaces de usuário, pois permite que sejam criadas aplicações de larga escala.

Neves (2023) afirma que o *React.js* oferece muitos recursos e ferramentas úteis que tornam o desenvolvimento de aplicativos web mais fácil e eficiente. Isso inclui bibliotecas de componentes prontos para uso.

Desta forma, o *React* permite a prática Não Repita a Si Mesmo, do termo em inglês *Don't Repeat Yourself (DRY)*, uma prática que incentiva a não repetição de linhas de código através da modularização de componentes, evitando arquivos extensos devido à repetição de linhas, promovendo a reutilização dos mesmos. Esta foi a principal característica estudada para a escolha desta tecnologia para codificação do portal Arte de Caderno, além de outras funcionalidades como a prática de *Routing*³ diretamente no *front-end*, permitindo o redirecionamento para outras páginas da aplicação sem a necessidade de uma requisição ao *back-end*, como o uso de ferramentas de armazenamento de estado, possibilitando o armazenamento temporário de variáveis que se alteram com o tempo e/ou interferência do usuário, entre outras.

Inicialmente, a estilização do projeto foi implementada utilizando arquivos CSS⁴. Todavia, foi observada a necessidade de outra abordagem para a estilização, uma vez que o emprego de classes semânticas sem repetição se tornava mais difícil ao longo do projeto. Desta maneira, optou-se por utilizar o *framework* TailwindCSS, uma biblioteca que fornece classes pré-estilizadas, que são adicionadas diretamente aos elementos HTML.

Criado em 2017 é um framework baseado em utilidades e tem como prioridade a facilidade de customização. Estilizar elementos com Tailwind é quase como escrever estilizações inline (escrever CSS dentro do atributo style), só que com classes. Por isso ele é um framework focado em utilidades. Classes como text-blue-600 para mudar a cor do texto de um elemento são muito utilizadas. A grande vantagem é que não precisamos seguir um padrão visual pré definido pelo framework. (ALBERTO, 2021)

³ Capacidade do gerenciamento e mudanças de páginas.

⁴ Do termo em inglês *Cascading Style Sheet*, em tradução livre: Arquivo de Estilização em Cascata.

Apesar de fácil configuração e padronização, sua utilização também resulta em arquivos extensos, uma vez que todo componente da página precisa receber cada uma de suas classes de estilização.

Desta forma, foi iniciado o estudo sobre *Styled-Components*, uma biblioteca React que permite a criação de estilos em forma de componentes, tornando possível a redução de códigos de estilos, concentrando-os em um único arquivo, tornando o projeto muito mais organizado. Além disso, a biblioteca permite a adaptação do componente de estilização baseada em interpolações e heranças, tornando-a ainda mais adequada ao objetivo do trabalho.

Outro ponto fantástico dos Styled Components é que, como a parte visual é resultante de um componente, sempre que precisarmos deletar um código vai ser fácil saber onde estão todas as suas referências, removendo tudo por completo, o que com CSS fica mais difícil pois é complicado termos um mapeamento do que está sendo trabalhando em cima de um seletor global que fazemos. (SOUTO, 2022)

Visando o maior alcance possível, a plataforma tem a necessidade de ser altamente adaptável, sendo facilmente utilizada desde computadores *desktop* a pequenos *smartphones* conectados à internet. Assim, a codificação alterna na utilização de técnicas de desenvolvimento, adaptando-se aos requisitos do sistema e do *design* estabelecido. Para telas com muitos detalhes, utilizou-se a técnica de *Mobile First*, também conhecido como *Touch-first*, onde a codificação é feita inicialmente para ser adaptável a pequenas telas, como *smartphones*.

A ideia de *Touch-first* é priorizar o código para a interatividade de *touchscreen*. Hoje em dia não apenas os *smartphones* e os *tablets* possuem tal recurso. O *touchscreen* vem ganhando espaço entre os *desktops*. [...] Uma questão importante é a otimização da área de toque: devemos estar atentos para o tamanho e a disposição dos botões, eles devem ser grandes e espaçados o suficiente para que o dedo consiga tocar. (LOPES, 2015)

Em seguida, o *layout* é adaptado para telas médias, como *tablets* e pequenos *laptops*. Por fim, mas não menos importante, o *design* é readaptado para grandes telas, tais como *laptops* de médio e grande porte, computadores *desktop*, ou ainda para monitores maiores. Desta forma, a escolha do procedimento a ser implementado depende da dificuldade técnica de cada *layout*.

4. Metodologia

4.1 Identificação do Problema

Foi verificada certa limitação e dificuldade no processo de inscrição e envio das obras artísticas para participação no projeto. A necessidade do preenchimento manual e envio físico dos documentos dificultam e limitam a participação de alunos e professores de áreas mais isoladas do país, como reservas indígenas ou vilarejos longínquos.

4.2 Sobre o referencial teórico

Nesta etapa foram conduzidos estudos a fim de definir as técnicas a serem implementadas no projeto, bem como as tecnologias a serem adotadas.

Após vários estudos, foi definido que a implementação da proposta seria realizada com *React.js*, versão 18.2.0, e *Styled-Components*, versão 6.0.7.

Para controle de versões do código, foi criado um repositório remoto na plataforma *GitHub* e, para documentação geral, como estruturação de pastas, um arquivo *README.md*. Para desenvolvimento e documentação de componentes, utilizou-se a biblioteca *Storybook*. A ferramenta de desenvolvimento utilizada foi a *Visual Studio Code*, da Microsoft.

4.3 Desenvolvimento do Trabalho

O desenvolvimento do trabalho obedeceu o roteiro a seguir:

1. Levantamento de requisitos para o sistema.
2. Aplicação de técnicas de desenvolvimento com ênfase em usabilidade, interatividade e utilização do método DRY, previamente citado neste artigo.
3. Documentação do processo.
4. Defesa deste TCC.

A realização dos procedimentos citados seguiu o cronograma a seguir:

Figura 1 - Cronograma de tarefas realizadas no primeiro semestre

		MESES				
TAREFAS	MARÇO	ABRIL	MAIO	JUNHO	JULHO	
LEVANTAMENTO DE REQUISITOS	X					
ESTUDO DE TECNOLOGIAS	X	X				
INSTALAÇÃO DAS FERRAMENTAS			X			
DESENVOLVIMENTO			X	X	X	
PRÉ-DEFESA DESTE TCC				X		

Fonte: criada pelo autor.

Figura 2 - Cronograma de tarefas realizadas no segundo semestre

		MESES			
TAREFAS	AGOSTO	SETEMBRO	OUTUBRO	NOVEMBRO	
DESENVOLVIMENTO	X	X	X		
DOCUMENTAÇÃO DO PROCESSO (STORYBOOK)	X	X	X	X	
DEFESA DESTE TCC			X		
EVENTUAIS MANUTENÇÕES			X	X	

Fonte: criado pelo autor.

4.3.1 Levantamento de Requisitos

Por ser voltada a um público de idade e escolaridade diversas, a plataforma possui a necessidade de ser altamente intuitiva, dispondo de elementos de fácil

interação. Além disso, deve contar um pouco da história deste projeto tão importante, contendo uma galeria de obras vencedoras de edições passadas.

A partir do levantamento de requisitos, foram definidas as seguintes telas:

- Cadastro de usuário.
- Cadastro de escolas/instituições de ensino.
- Cadastro de obras para a edição em andamento.
- Galeria de obras de edições passadas.
- Página contendo a história do projeto.
- *Dashboard* para acesso rápido a informações.
- Edição de dados pessoais.

Para educadores, foram disponibilizadas as opções de cadastrar alunos e vincular mais de uma escola. Desta forma, além das telas supracitadas, para este usuário, em específico, foram adicionadas as telas:

- Cadastro de alunos.
- Vincular a nova escola.
- Gerenciar alunos.
- Vincular desenhos ao aluno.

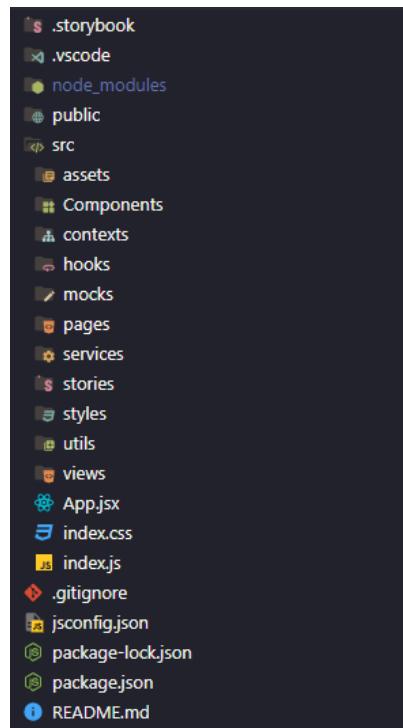
Aos avaliadores, foram atribuídas:

- *Dashboard* contendo a quantidade de desenhos:
 - avaliados,
 - aguardando avaliação.
- Tela para desclassificação de obras, sendo obrigatório o preenchimento de um formulário com justificativa.
- Tela para avaliação de desenhos, de forma anônima.

4.3.2 Desenvolvimento

Feito o levantamento de requisitos e as instalações das tecnologias necessárias, foi criado um projeto inicial em *React* e, em seguida, a criação dos diretórios básicos para o início da codificação, conforme mostra a Figura 3.

Figura 3 - Arquitetura inicial do projeto



Fonte: criado pelo autor.

A pasta *src* é o diretório central deste projeto, onde se encontram todas as demais pastas para melhor distribuição dos arquivos. Como a estilização é feita com *Styled-Components*, criou-se a pasta *styles*, que contém todos os componentes de estilos utilizados durante todo o projeto, para que fosse possível a prática de DRY. No diretório *Components* estão contidos os componentes que não são exclusivamente de estilização, como, por exemplo, o componente de Barra de Navegação. Este está localizado em sua pasta própria, dentro da pasta *Components*.

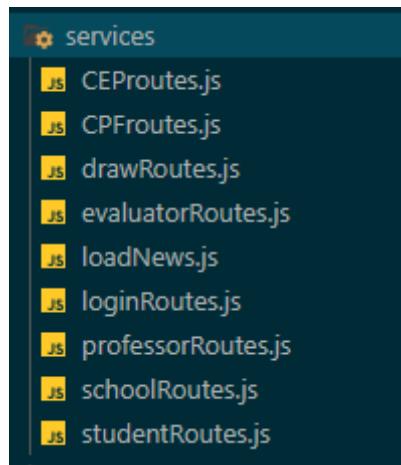
Em *assets* é possível encontrar todas as imagens utilizadas na codificação, como logos e mascotes; além de arquivos de fonte, para a garantia de que o estilo seja eficiente em todos os dispositivos.

Para garantir a responsividade do site, foi criada uma função para verificar o tamanho do dispositivo, sendo possível reconhecer dispositivos móveis, como *tablets* e *smartphones*, e *desktops*. Esta função está localizada na pasta *hooks*.

Uma maneira eficiente de realizar a comunicação com o servidor *back-end*, sem interferir no desempenho do *front-end*, é por meio da criação de arquivos que emitem requisições para o servidor. Desta forma, é possível exibir ao usuário a chamada *loading page*, do inglês, página de carregamento; para que ele saiba que a comunicação com o servidor está sendo realizada.

Para que fosse possível a implementação desta página, criou-se a pasta *services*, contendo vários arquivos, separados por finalidade, que realizam a comunicação entre o *front-end* e o *back-end*.

Figura 4 - Arquivos da pasta *services*



Fonte: criado pelo autor.

Como dito, cada arquivo possui uma finalidade diferente. Por exemplo, o arquivo *professorRoutes.js* realiza as requisições para as rotas ligadas ao educador. Desta forma, foi possível evitar a repetição de código durante a implementação do projeto e garantir a eficiência do *front-end*.

4.3.3 Documentação do Processo

Uma parte muito importante do desenvolvimento de qualquer projeto é a usabilidade oferecida ao usuário final. Para garantir uma boa experiência, cada componente exibido foi minuciosamente planejado, juntamente com a equipe de *design*, para que o usuário pudesse ter a melhor experiência ao utilizar a plataforma.

Para documentar os componentes e suas formas de utilização, empregou-se a biblioteca *Storybook*. As Figuras 5 e 6 apresentam a documentação dos Cards exibidos na página de *Dashboard*.

Figura 5 - Código da documentação do componente Card

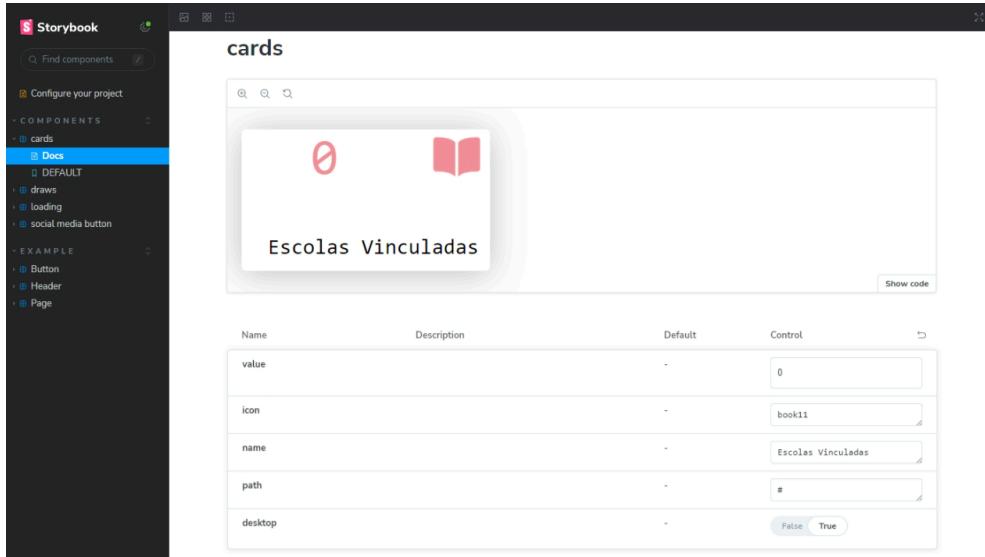
```

1 import Cards from '...'
2
3 export default {
4     title: "components/cards",
5     component: Cards,
6     tags: ['autodocs']
7 }
8
9 export const DEFAULT = {
10     args: {
11         value: 0,
12         icon: "book11",
13         name: "Escolas Vinculadas",
14         path: "#",
15         desktop: true
16     }
17 }

```

Fonte: criado pelo autor.

Figura 6 - Documentação visual do componente Card



Fonte: criado pelo autor

5. Resultados Obtidos

A partir dos problemas encontrados e soluções propostas, foi possível obter uma aplicação web intuitiva e adaptável a diferentes dispositivos, desenvolvida com as tecnologias *React* e *Styled-Components*, conforme definidos anteriormente neste artigo.

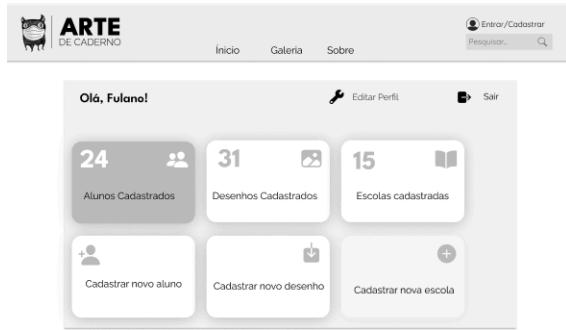
O site oferece formulários cadastrais, conforme a figura 7, devidamente integrados ao servidor *back-end*, desenvolvido pela discente Rebeca Domingos

Rosa⁵. Além de páginas de *dashboard* adaptadas para cada tipo de usuário, seja ele avaliador, educador ou estudante - Figura 8; conta ainda com página contando um pouco da história do projeto e seu vínculo com o Instituto Federal do Sul de Minas Gerais - Figura 9 - e uma galeria de obras de edições anteriores - Figura 10.

Figura 7 - Formulário de cadastro de usuário

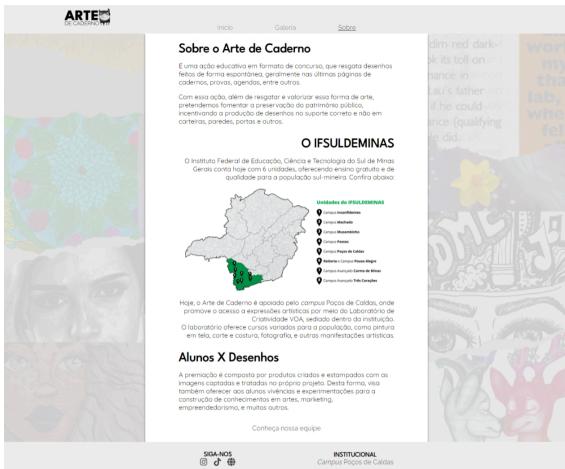
Fonte: criado pelo autor.

Figura 8 - *Dashboard* de educador



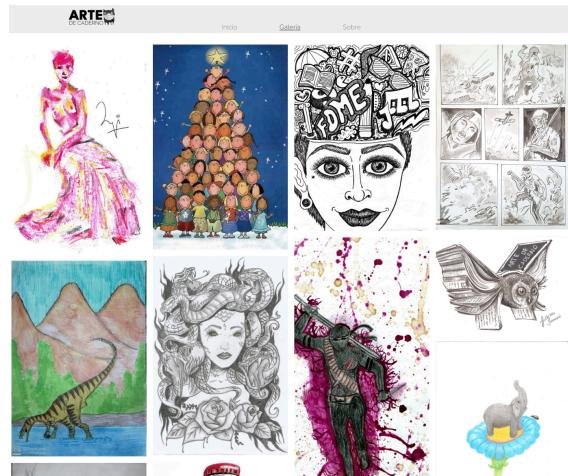
Fonte: criado pelo autor.

Figura 9 - História do projeto



Fonte: criado pelo autor.

Figura 10 - Galeria de obras



Fonte: criado pelo autor.

A plataforma conta ainda com formulários para submissão de obras, também adaptáveis ao tipo de acesso do usuário: os educadores terão, no momento da inscrição da obra, uma lista com todos os alunos vinculados a ele, Figura 11.

Para os educadores, existem telas especiais, para gerenciamento de estudantes, como formulário de inscrição de estudante, tela para visualização e

⁵ Graduanda em Engenharia de Computação pelo IFSULDEMINAS, campus Poços de Caldas. rebeca.rosa@alunos.if sulde minas.edu.br

edição dos dados de um estudante específico, figura 12, tela contendo todos os estudantes e uma tela contendo suas escolas vinculadas.

Figura 11 - Tela de cadastro de obras pelo acesso de educador

CADASTRAR UMA OBRA

Título: Autor(a): Selecionar...

Tema: Categoria: Selecionar...

ESCOLHA UM ARQUIVO

Cadastrar

Fonte: criado pelo autor.

Figura 12 - Tela de informações sobre um aluno

INFORMAÇÕES DO ALUNO:

NOME COMPLETO: Fulano de Tal
DATA DE NASCIMENTO: 10/10/2010
CPF: 111.222.333-44
TELEFONE: 11 2222-3333
E-MAIL: fulano.tal@email.com
CEP: 11111-222
CIDADE: Poços de Caldas
ESTADO: MG
ESCOLA: IFSULDEMINAS

Desenhos do aluno: < >

Fonte: criado pelo autor.

7. Conclusão

Este trabalho de conclusão de curso teve como objetivo o desenvolvimento de um *front-end* escalável e responsivo para a informatização do projeto Arte de Caderno, utilizando as tecnologias *React* e *Styled-Components*.

O desenvolvimento teve seu foco voltado para a usabilidade da plataforma, atentando-se ao nível de compreensão de cada página, de forma que pessoas menos escolarizadas possam utilizá-la sem problemas. Outro ponto importante a se destacar é a capacidade de adaptação do site a diversos tamanhos de dispositivos, para que seja possível o acesso através de *smartphones* ou mesmo computadores de mesa (*desktops*).

Como trabalho futuro, espera-se uma ampliação da plataforma, com instrumentos de acessibilidade para o usuário final, como a possibilidade de contraste e ampliação de fontes, ou ainda, legendas e transcrições de mídia, para o auxílio de pessoas com deficiência visual. Há ainda a possibilidade de adaptação da plataforma a teclados e navegação por voz, garantindo que a plataforma seja utilizável com comandos de voz, sem a necessidade de *mouse*.

8. Referência Bibliográfica

LOPEZ, Lionel. React: Quickstart Step-By-Step Guide To Learning React Javascript Library (React.js, Reactjs, Learning React JS, React Javascript, React Programming). CreateSpace Independent Publishing Platform, 2017.

NEVES, Vinicios. React: o que é, como funciona e um Guia dessa popular ferramenta JS. Alura, 2023. Disponível em: <https://www.alura.com.br/artigos/react-js#por-que-utilizar-react?>. Acesso em: 29 mai. 2023.

SOUTO, Mario. React: Componentes com Styled Components - Vantagens do Styled Components. Alura, 2022. Disponível em:
<https://www.alura.com.br/artigos/react-componentes-com-styled-components#vantagens-do-styled-components>. Acesso em: 29 mai. 2023.

LOPES, Sérgio. Touch-first: Como otimizar seu design responsivo para o mundo das touch screens. Alura, 2015. Disponível em:
<https://www.alura.com.br/artigos/touch-first-como-otimizar-seu-design-responsivo-para-o-mundo-das-touch-screens>. Acesso em: 29 mai. 2023.

ALBERTO, Matheus. Tailwind: Qual framework usar? Bootstrap vs Tailwind. Alura, 2021. Disponível em: <https://www.alura.com.br/artigos/tailwind-framework-bootstrap-tailwind>. Acesso em: 29 mai. 2023.

Storybook. Documentação do Storybook. Disponível em: <https://storybook.js.org>. Acesso em: 28 de set. 2023.

Visual Studio Code. Documentação do Visual Studio Code. Disponível em:
<https://code.visualstudio.com/docs>. Acesso em: 28 de set. de 2023.