



PROGRAMAÇÃO DE UM *FRONT-END* ESCALÁVEL UTILIZANDO A TECNOLOGIA *REACT* PARA A PLATAFORMA DIGITAL ARTE DE CADERNO¹

Beatriz do Prado Neaime ²
Caroline Ferreira de Melo ³
Douglas Fabiano de Sousa Nunes ⁴
Giselle Cristina Cardoso⁵
Márcio Luiz Bess⁶

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. A metodologia adotada compreendeu a identificação do problema, uma revisão teórica abrangente e o desenvolvimento da plataforma utilizando tecnologias modernas como *React* e *Styled-Components*. A escolha do *React* foi fundamentada em sua capacidade de criar interfaces escaláveis, enquanto o *Styled-Components* foi empregado para garantir a estilização sem repetição de código. A estruturação da plataforma seguiu um cronograma detalhado, abrangendo desde a definição de requisitos até a implementação e documentação do processo. Os resultados obtidos foram um *front-end* intuitivo e responsivo, integrado ao servidor *back-end*, com funcionalidades específicas para educadores e avaliadores. Como perspectiva futura, sugere-se a implementação de recursos de acessibilidade e adaptação para comandos de voz, visando tornar a plataforma ainda mais inclusiva e acessível. Em síntese, o desenvolvimento desta plataforma visa promover a expressão artística de forma organizada e inclusiva, contribuindo para o alcance dos objetivos do projeto Arte de Caderno.

Palavras-chave: *React*, Arte de Caderno, *Front-end*, Escalável, Responsivo.

¹Projeto fomentado pelo EDITAL PROEX 22/2023 - APOIO A PROJETOS DE ARTE E CULTURA do Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais - IFSULDEMINAS e realizado pelo Laboratório de Criatividade VOA IFSULDEMINAS *campus* Poços de Caldas.

²Bacharela em Engenharia de Computação pelo IFSULDEMINAS *campus* Poços de Caldas. beatrizneaime@outlook.com;

³Discente no curso de Engenharia de Computação no IFSULDEMINAS *campus* Poços de Caldas. caroline.melo@alunos.ifsuldeminas.edu.br;

⁴Doutor em Ciências da Computação e Matemática Computacional pela Universidade de São Paulo, USP. douglas.nunes@ifsuldeminas.edu.br;

⁵Mestre em Engenharia Elétrica pela Universidade Estadual de Campinas, UNICAMP. giselle.cardoso@ifsuldeminas.edu.br;

⁶Doutor em Desenvolvimento Humano e Tecnologias pela Universidade Estadual Paulista Júlio de Mesquita Filho, UNESP. marcio.bess@ifsuldeminas.edu.br.



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 Bess⁷, 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 analisadas 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 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, no 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 analisado 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.

Este trabalho teve como objetivo geral implementar uma ferramenta web para que o processo de inscrição, submissão de obras e avaliação das mesmas pudessem ser realizados todos de forma *online*, com o suporte de tecnologias computacionais.

METODOLOGIA

O primeiro passo para a implementação do projeto foi a identificação dos problemas enfrentados. Foi verificada certa limitação e dificuldade, por parte dos estudantes em geral, no processo de inscrição e envio das obras artísticas, de modo a garantir suas participações no projeto. A necessidade do preenchimento manual e envio físico dos documentos dificultavam

⁷ Lattes: <http://lattes.cnpq.br/2142214809343643>



e limitavam a participação de alunos e professores de áreas mais isoladas do país, como reservas indígenas ou vilarejos longínquos.

Logo após, foram conduzidos profundos estudos com o objetivo de definir a melhor tecnologia a ser utilizada para a construção da aplicação. 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, por exemplo, a estruturação de pastas e o arquivo *README.md*. Para desenvolvimento e documentação de componentes, utilizou-se a biblioteca *Storybook*⁸. A ferramenta de desenvolvimento utilizada foi a Visual Studio Code, da Microsoft.

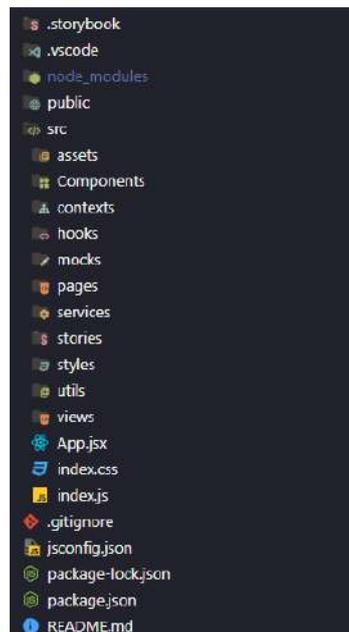
O desenvolvimento do trabalho obedeceu o roteiro a seguir:

1. Levantamento de requisitos para o sistema: nesta etapa, foram levantados os requisitos mínimos para uma melhor interação do usuário com a plataforma. São apontados dados, tais como nome, endereço, idade e escolaridade dos participantes. Desta forma, foi possível definir o conteúdo de cada tela.
2. Aplicação de técnicas de desenvolvimento com ênfase em usabilidade, interatividade e utilização do método *Don't Repeat Yourself (DRY)*⁹, que consiste na reutilização de códigos, a fim de evitar sua redefinição. Após o levantamento de requisitos, criou um projeto inicial em *React* e a instalação das tecnologias necessárias. A Figura 1 mostra a organização inicial da estrutura de diretórios usada no projeto.

⁸ Vide referencial teórico.

⁹ Vide referencial teórico.

Figura 1 - Estrutura de diretórios 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 com os códigos. 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*. Essa característica incorporada na plataforma é importante para que a aplicação possa se ajustar, da melhor forma possível, às especificidades de interatividade e visualização dos dados apresentados por cada dispositivo.

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.

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 2 e 3 apresentam a documentação dos *Cards* exibidos na página de *Dashboard*.

Figura 2 - Código da documentação do componente Card

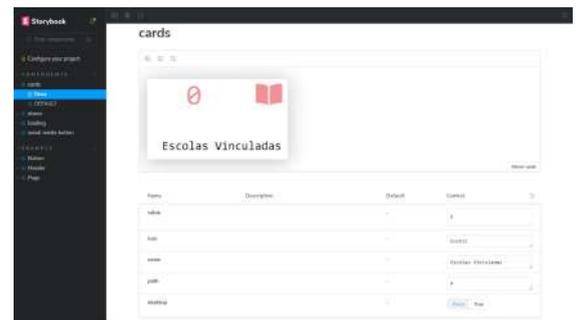
```

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 3 - Documentação visual do componente Card



Fonte: criado pelo autor.

REFERENCIAL TEÓRICO

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 (*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 que motivou a escolha desta tecnologia para codificação do portal Arte de Caderno, além de outras, tais como a prática de *Routing*¹⁰ diretamente no *front-end*, permitindo o redirecionamento para outras páginas da aplicação sem a necessidade de uma requisição ao *back-end*, e 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.

Inicialmente, a estilização do projeto foi implementada utilizando arquivos CSS¹¹. 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)

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 trabalhado 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

¹⁰ Capacidade do gerenciamento e mudanças de páginas.

¹¹ Do termo em inglês *Cascading Style Sheet*, em tradução livre: Arquivo de Estilização em Cascata.



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*.

RESULTADOS E DISCUSSÃO

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 4, devidamente integrados ao servidor *back-end*, desenvolvido pela equipe de desenvolvimento *back-end*. Além de páginas de *dashboard* adaptadas para cada tipo de usuário, seja ele avaliador, educador ou estudante - Figura 5, a plataforma conta ainda com uma página retratando um pouco da história do projeto e seu vínculo com o Instituto Federal do Sul de Minas Gerais - Figura 6 - e uma galeria de obras de edições anteriores - Figura 7.

Figura 4 - Formulário de cadastro de usuário



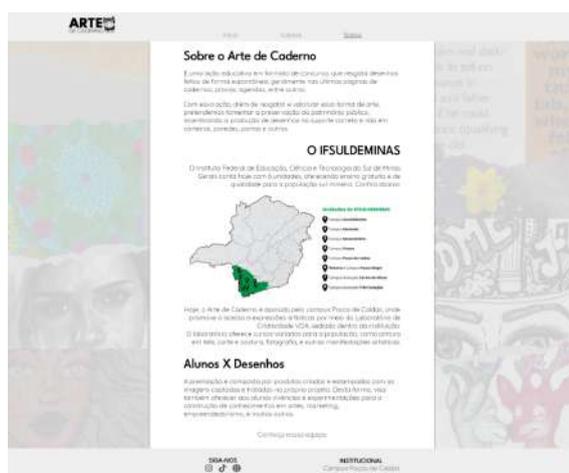
Fonte: criado pelo autor.

Figura 5 - Dashboard de educador



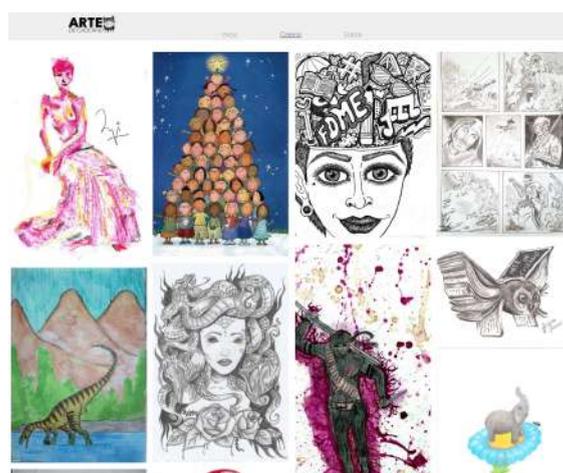
Fonte: criado pelo autor.

Figura 6 - História do projeto



Fonte: criado pelo autor.

Figura 7 - Galeria de obras



Fonte: criado pelo autor.

A plataforma conta também com formulários para submissão de obras, que são 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 8.

Para os educadores, existem telas especiais, para gerenciamento de estudantes, como formulário de inscrição de estudante, tela para visualização e edição dos dados de um estudante específico, Figura 9, tela contendo todos os estudantes e uma tela contendo suas escolas vinculadas.

Figura 8 - Tela de cadastro de obras pelo acesso de educador



Fonte: criado pelo autor.

Figura 9 - Tela de informações sobre um aluno



Fonte: criado pelo autor.

CONSIDERAÇÕES FINAIS

Este projeto 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*). Em resumo, a criação desta plataforma busca fomentar a expressão artística de maneira estruturada e abrangente, colaborando para a concretização dos propósitos delineados pelo projeto Arte de Caderno. A plataforma se destaca como uma solução sólida e flexível para a submissão de desenhos de instituições públicas no concurso promovido pelo projeto Arte de Caderno, e seu potencial para aprimoramentos contínuos promete expandir sua capacidade e efetividade.

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 uso do *mouse*.

REFERÊNCIAS

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, Vinícios. 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: 02 fev. 2024.

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: 05 fev. 2024.

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: 05 fev. 2024.

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: 02 fev. 2024.

Storybook. Documentação do Storybook. Disponível em: <https://storybook.js.org>. Acesso em: 10 fev. 2024.

Visual Studio Code. Documentação do Visual Studio Code. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 10 fev. 2024.