



UNIVERSIDADE FEDERAL DO CEARÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

FELIPE BARROS MUNIZ

SISTEMA DE AUTOAVALIAÇÃO WEB - PPGEEC/UFC

SOBRAL

2023

FELIPE BARROS MUNIZ

SISTEMA DE AUTOAVALIAÇÃO WEB - PPGEEC/UFC

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia da Computação.

Orientador: Prof. Dr. Iális Cavalcante de Paula Júnior

SOBRAL

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M935s Muniz, Felipe Barros.
SISTEMA DE AUTOAVALIAÇÃO WEB - PPGEEC/UFC SOBRAL / Felipe Barros Muniz. – 2023.
38 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Sobral,
Curso de Engenharia da Computação, Sobral, 2023.
Orientação: Prof. Dr. Iális Cavalcante de Paula Júnior.

1. Desenvolvimento Front-end. 2. Autoavaliação. 3. Design System. I. Título.

CDD 621.39

FELIPE BARROS MUNIZ

SISTEMA DE AUTOAVALIAÇÃO WEB - PPGEEC/UFC

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia da Computação.

Aprovada em: 21 de Dezembro de 2023

BANCA EXAMINADORA

Prof. Dr. Iális Cavalcante de Paula
Júnior (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. José Cláudio do Nascimento
Universidade Federal do Ceará (UFC)

Stefane Adna dos Santos
Universidade Federal do Ceará (UFC)

À minha família, por sua capacidade de acreditar em mim e investir em mim. A todos que fizeram parte da minha jornada na cidade de Sobral. A Karoline, minha namorada, que sempre se fez presente.

AGRADECIMENTOS

Ao Prof. Dr. Iális Cavalcante de Paula Júnior por me orientar em meu trabalho de conclusão de curso.

Aos meus amigos e colegas de faculdade, pelo apoio durante os estudos realizados durante a graduação.

Aos meus pais e irmãos, que nos momentos de minha ausência dedicados ao estudo superior, sempre fizeram entender que o futuro é feito a partir da constante dedicação no presente!

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

Ao aluno Thiago Nascimento do curso de ciência da computação da Universidade Estadual do Ceará que elaborou o *template* do qual este trabalho foi adaptado para Universidade Federal do Ceará.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

“O sonho é que leva a gente para frente. Se a gente for seguir a razão, fica aquietado, acomodado.”

(Ariano Suassuna)

RESUMO

Possibilitar uma coleta de *feedbacks* de participantes de um grupo específico é fundamental para melhorias e conhecimento do coletivo sobre o ambiente que eles estão presentes. Englobando assim os participantes do programa de pós-graduação, proporcionando seu sucesso e sobrevivência. Uma abordagem que melhor se enquadra nessa coleta é o sistema de Autoavaliação (AA) tanto pessoal dos integrantes quando do programa em si. Sendo assim, necessário a criação de uma plataforma que possa colher e gerenciar esses *feedbacks*, de uma forma prática e facilitada para o usuário, utilizando uma *User Interface* (UI) simples e de fácil entendimento. O objetivo deste trabalho foi desenvolver uma aplicação de um sistema *web* para coleta dos dados necessários para o desenvolvimento e manutenção do programa de pós-graduação, juntamente com uma área administrativa responsável pelo cadastro de usuário, tanto docentes quanto discentes, turmas, disciplinas, formulários e avaliações, disponibilizando uma liberdade e garantindo que os dados não passem e nem dependam de ferramentas de terceiros. Primeiramente foi criada uma estrutura de *Design System*, onde foi definido os principais pontos do *layout* e referências, possibilitando um código direcionado e eficiente com uma boa UI, sem ter a preocupação de estruturas no decorrer do desenvolvimento em código, onde se utiliza ferramentas livres de desenvolvimento, como o *Visual Studio Code*, linguagem e programação JavaScript juntamente com o TypeScript, biblioteca ReactJS e *framework* NextJS. Com isso, o principal resultado desse trabalho é o *front-end* desenvolvido para manipulação e visualização dos dados cadastrados.

Palavras-chave: UI. *Design System*. JavaScript. TypeScript. Autoavaliação. ReactJS. NextJS.

ABSTRACT

Enabling the collection of feedback from participants in a specific group is essential for improvements and knowledge of the collective about the environment in which they are present. Thus encompassing the participants of the postgraduate program, providing their success and survival. An approach that best fits this collection is the self-evaluation system, both personally for the members and for the program itself. Therefore, it is necessary to create a platform that can collect and manage this feedback, in a practical and user-friendly way, using a simple and easy-to-understand UI. The objective of this work was to develop a web system application to collect the data necessary for the development and maintenance of the postgraduate program, together with an administrative area responsible for user registration, both teachers and students, classes, disciplines, forms and evaluations, providing freedom and ensuring that data does not pass through or depend on third-party tools. Firstly, a Design System structure was created, where the main points of the layout and references were defined, enabling targeted and efficient code with a good UI, without having to worry about structures during code development, where free tools are used. development, such as Visual Studio Code, JavaScript language and programming along with TypeScript, ReactJS library and NextJS framework. Therefore, the main result of this work is the front-end developed for manipulating and visualizing registered data.

Palavras-chave: UI. Design System. JavaScript. TypeScript. Self-evaluation. ReactJS. NextJS.

LISTA DE FIGURAS

Figura 1 – Paleta de cores do protótipo	22
Figura 2 – Tipografia do protótipo	23
Figura 3 – Exemplos de imagens	23
Figura 4 – Exemplos base para protótipo	24
Figura 5 – Protótipo da página inicial	25
Figura 6 – Protótipo da área administrativa	25
Figura 7 – Participação efetuado no projeto em relação <i>commits</i>	26
Figura 8 – Diagrama de fluxo da plataforma	28
Figura 9 – Interface da página inicial	29
Figura 10 – Interface administrativa de usuários	29
Figura 11 – Interface de criação de usuários	30
Figura 12 – Interface de validação de usuários	30
Figura 13 – Interface administrativa de disciplinas	31
Figura 14 – Interface administrativa de turmas	31
Figura 15 – Interface de criação de disciplinas	32
Figura 16 – Interface de criação de turmas	32
Figura 17 – Interface administrativa de formulários	33
Figura 18 – Interface de criação de formulários	33
Figura 19 – Interface de criação de avaliações	34
Figura 20 – Interface de relação de discente com a turma	34
Figura 21 – Interface de relação de discente com avaliação	35
Figura 22 – Interface de relação de docente com avaliação	35
Figura 23 – Interface de avaliação	36

LISTA DE ABREVIATURAS E SIGLAS

AA	Autoavaliação
UI	<i>User Interface</i>
PPGEEC	Programa de Pós-graduação em Engenharia Elétrica e Computação
JS	JavaScript
TS	TypeScript
API	<i>Application Programming Interface</i>
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CNE	Conselho Nacional de Educação
HTML	<i>HyperText Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
UIs	<i>User Interfaces</i>
VDOM	<i>Virtual Document Object Model</i>
DOM	<i>Document Object Model</i>
JSX	<i>JavaScript Syntax Extension</i>
HTTP	<i>Hypertext Transfer Protocol</i>
DNS	<i>Domain Name System</i>
JWT	<i>JSON Web Token</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Justificativa	14
1.2	Objetivo Geral	14
1.3	Objetivos Específicos	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Resolução de Autoavaliação do PPGEEC	16
2.2	<i>Design System</i>	16
2.3	Desenvolvimento <i>front-end</i>	17
2.4	Linguagem JavaScript	17
2.5	Linguagem TypeScript	18
2.6	ReactJS	18
2.7	NextJS	19
2.8	Gerenciamento de pacotes	19
2.9	Github e Gitflow	20
2.9.1	<i>Github</i>	20
2.9.2	<i>GitFlow</i>	20
3	METODOLOGIA	22
3.1	Criação do <i>Design System</i>	22
3.2	Desenvolvimento do <i>Front-end</i>	26
4	RESULTADOS	28
4.1	Resultados do desenvolvimento do <i>Front-end</i>	28
5	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	37
6	TRABALHOS RELACIONADOS	38
6.1	Sistema de Autoavaliação - Programa de Pós-graduação em Engenharia Elétrica e Computação (PPGEEC)/UFC	38
6.2	Coordenador API: API REST para controle e gerenciamento de sistema de avaliação do programa de pós-graduação em engenharia elétrica e computação	38
	REFERÊNCIAS	39

1 INTRODUÇÃO

A pós-graduação no Brasil é uma modalidade de ensino superior que visa a formação de pesquisadores e profissionais qualificados para atuar em diferentes áreas do conhecimento. A pós-graduação foi instituída formalmente pelo Parecer CFE 977/1965, que conceituou e normatizou os cursos de mestrado e doutorado, com o objetivo de desenvolver o regime de pesquisa científica e tecnológica no país (BRASIL, 1965).

Segundo o Parecer CFE 977/1965, os cursos de pós-graduação devem ter as seguintes características: a) serem organizados por áreas de conhecimento; b) terem um corpo docente qualificado e experiente na área; c) exigirem dos alunos uma prova de proficiência em língua estrangeira; d) terem um programa de disciplinas teóricas e práticas; e) exigirem dos alunos a elaboração e defesa de uma dissertação ou tese original; f) terem uma duração mínima de dois anos para o mestrado e de três anos para o doutorado (BRASIL, 1965).

Os programas de pós-graduação do país tem apresentado um crescimento expressivo nas últimas décadas, tanto em termos de programas quanto de alunos. De acordo com dados da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), o número de programas de pós-graduação *stricto sensu* passou de 3.128 em 2011 para 4.650 em 2020, um aumento de 48,6%. O número de alunos matriculados nesses programas também cresceu, passando de 122.295 em 2010 para 206.609 em 2020, um aumento de 68,9%. A maior parte dos alunos está concentrada na região Sudeste (58%), seguida pela região Sul (16%), Nordeste (14%), Centro-Oeste (8%) e Norte (4%) (CAPES, 2020).

Uma maneira de garantir a qualidade básica da educação em um país de dimensões continentais como o Brasil é ter um sistema eficaz de avaliação externa. Conhecida como avaliação de desempenho, é realizada por agente externo à instituição, geralmente aplicada em larga escala. É uma ferramenta que fornece elementos para a formulação e o monitoramento de políticas públicas, bem como o redirecionamento de práticas pedagógicas. As avaliações externas têm como característica uma matriz de avaliação e o emprego de provas padronizadas, que permitem o cumprimento do direito à aprendizagem e a interpretação dos resultados para efetuar comparação entre redes e instituições (JOVENSGENIOS, 2020).

Porém, esse método tem suas limitações, pois não permite que os envolvidos no processo educativo participem da resolução dos problemas detectados. Assim, a autoavaliação se torna uma estratégia que favorece a construção da identidade, da diversidade e do engajamento dos programas avaliados, além dos padrões mínimos assegurados pela avaliação externa (CAPES,

2019).

Na educação superior, o processo autoavaliativo é definido e autogerido pela comunidade acadêmica, que tem a titularidade da avaliação. O foco e os objetivos da AA são decididos pelos protagonistas. A reflexão sobre os resultados obtidos é central ao processo e leva em conta a correção de trajetórias e de futuros projetados. É uma forma de avaliação que exige tempo, recursos e dedicação (LEITE, 2020).

O PPGEEC é o programa de pós-graduação em engenharia elétrica e computação ofertado pela Universidade Federal do Ceará. Com isso, espera-se que o desenvolvimento da autoavaliação no PPGEEC induza um processo de amadurecimento de pesquisadores e discentes no sentido de responsabilização, colaboração e engajamento na melhoria do *stricto sensu*, da qualidade da formação de pesquisadores brasileiros e, principalmente, da prática democrática (LEITE, 2020).

Dito isto, é de suma importância para a evolução de um programa como o PPGEEC, que exista um sistema que tenha a possibilidade de se autoavaliar, tanto para os docentes que transmitem o conhecimento, quanto para os discentes que o recebem. Com isso, aumentando a qualidade com que o ensino é passado/recebido pelos participantes.

Sendo assim, este trabalho propõe a criação de um sistema *web* capaz de suprir as necessidades descritas. Com isso, partindo da premissa dos participantes da pós-graduação se auto avaliarem e avaliarem aspectos do programa, temos uma aplicação digital e interativa capaz de se comunicar com o usuário permitindo sua interação de forma prática e dedutiva para uma coleta eficiente dos dados. Portanto, para o se dar início, primeiramente foi elaborado um *Design System*, que busca simplificar e padronizar o processo criativo, facilitando a comunicação, documentação e manutenção dos produtos.

Com relação o desenvolvimento da plataforma *web*, foi utilizada a linguagem de programação JavaScript (JS) com o apoio do TypeScript (TS), com a biblioteca ReactJS e o *framework* NextJS, assim disponibilizando uma aplicação atual com tecnologias difundidas no mercado e com suporte ativo na comunidade, dando uma grande vida útil ao código escrito. Sendo o consumo dos dados e envios dos mesmos, feita por uma *Application Programming Interface* (API) Rest, que proporciona uma comunicação com o banco de dados mais rápida e fácil.

O trabalho está organizado da seguinte forma: na seção dois consta uma fundamentação teórica com o objetivo de fundamentar os assuntos discutidos; na seção três é apresentado

a metodologia usada para desenvolver o sistema de autoavaliação proposto; e, por fim, são apresentados alguns resultados, considerações finais e recomendações para trabalhos futuros.

1.1 Justificativa

Para um crescente desenvolvimento e melhorias para o programa, um sistema de autoria própria e com demandas personalizadas se torna relevante e essencial. Atualmente, utiliza-se plataformas de terceiros como o do Google, que é denominada como Google Forms, para a coleta dos *feedbacks*. Que possui um intuito mais generalista na construção dos formulários e de propriedade exclusiva da empresa desenvolvedora, sendo assim, a ocorrência de um possível encerramento da ferramenta o PPGEEC perde todos os dados e a possibilidade de circular novos formulários pela plataforma.

Com base na informação apresentada, é possível notar que a tecnologia proposta traz benefícios significativos tanto para o programa que a adota quanto para os docentes e discentes que a utilizam. Isso se deve ao fato de que a tecnologia permite *feedbacks* que ajudam a melhorar o funcionamento do programa. Portanto, é de suma importância desenvolver uma aplicação que permita que docentes, discentes e técnicos administrativos avaliem tanto o programa como um todo quanto disciplinas e o desempenho dos docentes, além de se autoavaliarem. Isso contribuirá para a evolução e criação de um programa melhor para todos os envolvidos.

1.2 Objetivo Geral

O objetivo geral deste trabalho foi a inicialização do desenvolvimento de uma sistema de avaliação *web* de interface intuitiva e de fácil usabilidade proporcionando um experiência satisfatória que substitua os formulários criados na plataforma do Google Forms e que centralize todos os dados dentro da própria universidade.

1.3 Objetivos Específicos

Para que a aplicação seja bem aceita dentro da universidade, ela deve atender à todas as necessidades e funcionalidades que anteriormente se dava pela plataforma do Google Forms, como uma interface mais intuitiva, maior facilidade de uso e entendimento pelo usuário. Diante disto, um bom levantamento de requisitos é primordial para o sucesso. Sendo assim, para atender todas as partes interessadas no projeto levantaram-se os seguintes objetivos específicos:

- Desenvolver um painel administrativo capaz de cadastrar usuários do tipo docente e discente, disciplinas, turmas, formulários e avaliação.
- Desenvolver uma interface de fácil usabilidade e entendimento pelos possíveis usuários.
- Desenvolver um sistema de fácil manutenção e de futuras implementações, possibilitando uma grande vida útil ao projeto.

2 FUNDAMENTAÇÃO TEÓRICA

Para o desenvolvimento do projeto apresentado, foi aplicada um *Design System* para facilitar o desenvolvimento do projeto. Ademais, a linguagem de programação utilizada foi o javascript com auxílio do typescript e biblioteca ReactJS e do *framework* NextJS para a construção de um plataforma (*Frontend*).

2.1 Resolução de Autoavaliação do PPGEEC

A Resolução Nº 1/2021/CMEEC/CUFCSOBRAL/REITORIA, de 27 de janeiro de 2021, dispõe sobre o regulamento e o plano de ação do processo de autoavaliação do PPGEEC/UFC. Esse processo visa aprimorar a qualidade acadêmica e institucional do programa, bem como atender às exigências da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e do Conselho Nacional de Educação (CNE) (PPGEEC, 2021).

Segundo a resolução, o processo de autoavaliação é coordenado por uma comissão composta por docentes, discentes e técnicos-administrativos do PPGEEC/UFC, que deve elaborar um relatório anual com os resultados das avaliações internas e externas, as ações realizadas e as propostas de melhoria. O relatório deve ser submetido à apreciação do colegiado do programa e à homologação da reitoria da UFC (PPGEEC, 2021).

A resolução também estabelece os critérios e os indicadores de avaliação do PPGEEC/UFC, baseados nas dimensões propostas pela CAPES: proposta do programa, corpo docente, corpo discente, teses e dissertações, produção intelectual, inserção social e impacto na sociedade. Esses critérios devem orientar a elaboração do plano de ação da comissão de autoavaliação, que deve contemplar as metas, as estratégias, os recursos e os prazos para a implementação das ações (PPGEEC, 2021).

2.2 *Design System*

Design System é um conjunto de princípios, padrões e ferramentas que orientam o desenvolvimento de interfaces digitais. O mesmo visa garantir a consistência, a usabilidade e a acessibilidade dos produtos digitais, além de facilitar a colaboração entre designers e desenvolvedores. Um *Design System* pode ser visto como uma forma de pesquisa em engenharia de software, que se alinha com a ciência do *design*, uma abordagem que enfatiza a criação de artefatos e sistemas inovadores para resolver problemas complexos (WIERINGA, 2020). A

ciência do *design* também se aplica ao desenho e à condução de estudos que visam compreender a nossa metodologia e os nossos métodos de pesquisa (HEVNER *et al.*, 2004).

Um *Design System* traz diversos benefícios para a criação de produtos e serviços digitais, tais como:

- Aumento da produtividade, pois reduz o tempo gasto com tarefas repetitivas e permite o reaproveitamento de componentes e soluções já testadas.
- Melhoria da experiência do usuário, pois oferece uma interface mais coerente, intuitiva e acessível, que atende às expectativas e necessidades dos usuários.
- Fortalecimento da identidade visual, pois cria uma linguagem visual única e reconhecível, que transmite os valores e a personalidade da marca.
- Facilitação da manutenção e da evolução, pois permite a atualização e a adaptação dos componentes de forma centralizada e padronizada, garantindo a compatibilidade e a qualidade do *design*.

2.3 Desenvolvimento *front-end*

Front-end development é o processo de criar interfaces de usuário para aplicações *web*, utilizando tecnologias como *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS) e JavaScript. O desenvolvimento *front-end* envolve desafios e barreiras que exigem conhecimentos e habilidades específicas dos desenvolvedores. Alguns desses desafios são: lidar com API complexas, escolher entre diferentes *frameworks* e bibliotecas, e garantir a compatibilidade e o desempenho das interfaces em diferentes dispositivos e navegadores (SAMUDIO; LATOZA, 2022).

2.4 Linguagem JavaScript

JavaScript é uma linguagem de programação que capacita a criação de aplicações *web* dinâmicas e interativas. Essa linguagem opera no lado do cliente, ou seja, no navegador do usuário, sem depender de um servidor. Baseada em objetos, o JavaScript possui uma sintaxe simples e flexível, possibilitando a elaboração desde scripts simples até aplicações complexas (MDN, 2023).

Uma característica fundamental do JavaScript é sua natureza interpretada. Em outras palavras, o código fonte é executado diretamente pelo navegador, eliminando a necessidade de

compilação prévia. Esse aspecto agiliza o desenvolvimento, tornando-o mais eficiente, embora apresente desafios, como a ausência de tipagem estática, inconsistências entre implementações e dificuldades na depuração (WIKIPEDIA, 2023).

2.5 Linguagem TypeScript

TypeScript, uma extensão do JavaScript, incorpora tipos estáticos para capacitar desenvolvedores a identificar e corrigir erros durante a compilação. Fundamentada na sintaxe do ECMAScript, que define o padrão JavaScript, esta linguagem avançada é compatível com os principais navegadores e ambientes de execução, proporcionando recursos como classes, interfaces, módulos e decoradores para simplificar a organização e reutilização do código (TYPESCRIPT, 2023).

O sistema de tipos do TypeScript dinamicamente infere os tipos das variáveis com base em seu comportamento durante a execução. Se uma variável recebe um valor numérico, o TypeScript automaticamente a categoriza como do tipo *number*, permitindo verificações precisas sobre a validade das operações realizadas. Além disso, os desenvolvedores têm a flexibilidade de explicitar os tipos de variáveis, funções, parâmetros e retornos por meio de anotações de tipo. Estas anotações são opcionais, oferecendo aos desenvolvedores a liberdade de especificar tipos mais complexos ou personalizados (TYPESCRIPT, 2023).

2.6 ReactJS

ReactJS é uma biblioteca JavaScript para criar *User Interfaces* (UIs) baseadas em componentes reutilizáveis. Desenvolvido pelo Facebook e lançado em 2013, ReactJS é atualmente uma das tecnologias mais populares para o desenvolvimento web *front-end*. Oferecendo diversas vantagens aos desenvolvedores, como facilidade de aprendizado, alta performance, suporte à renderização no lado do servidor e integração com outras bibliotecas e *frameworks* (WIKIPEDIA, 2020).

Uma das características mais significativas do ReactJS é o uso do *Virtual Document Object Model* (VDOM), uma representação em memória do *Document Object Model* (DOM) real do navegador. O VDOM permite que o ReactJS atualize apenas as partes da UI que foram alteradas, evitando operações desnecessárias e custosas no DOM real. Essa abordagem melhora significativamente a performance da aplicação, especialmente em cenários com muitas interações

do usuário (KULKARNI *et al.*, 2019).

Outra característica distintiva do ReactJS é o uso de *JavaScript Syntax Extension* (JSX), uma extensão de sintaxe do JavaScript que permite escrever elementos HTML dentro do código JavaScript. O JSX facilita a criação de componentes reutilizáveis e a manipulação de dados e eventos dentro da UI. Adicionalmente, o JSX pode ser compilado para JavaScript puro usando ferramentas como Babel, garantindo a compatibilidade com diferentes navegadores (INDLA; PURANIK, 2021).

Além disso, o ReactJS possui um ecossistema rico e diversificado, com diversas bibliotecas e *frameworks* que complementam ou ampliam suas funcionalidades. Por exemplo, o React Router possibilita a criação de rotas dinâmicas para navegação entre componentes; o Redux permite gerenciar o estado global da aplicação de forma previsível e consistente; o NextJS permite criar aplicações web universais, que podem ser renderizadas tanto no lado do servidor quanto no lado do cliente (SÁNCHEZ *et al.*, 2020).

2.7 NextJS

O NextJS, baseado no conceito de páginas, utiliza arquivos em JavaScript ou TypeScript que exportam componentes React, correspondendo cada página a uma rota no sistema de arquivos. Adaptável às necessidades específicas, o NextJS suporta tanto geração estática quanto renderização no servidor, incluindo a capacidade de criar páginas com parâmetros variáveis (NEXT.JS, 2023).

Destacando-se pela otimização de desempenho e experiência do usuário, o NextJS emprega técnicas como pré-renderização, *code-splitting*, *prefetching* e *image optimization*. Essas práticas visam reduzir o tempo de carregamento, diminuir o tamanho dos *bundles* e aprimorar a qualidade visual, enquanto simplifica o desenvolvimento e implantação de aplicações *web*, automatizando e configura ferramentas cruciais para o React, como compilação, roteamento, *hot reloading* e *serverless functions*. Oferecendo integração com diversos provedores de hospedagem, como Vercel, Netlify e Heroku (NEXT.JS, 2023).

2.8 Gerenciamento de pacotes

O gerenciamento de pacotes é uma atividade essencial para o desenvolvimento de aplicações em Javascript, pois permite o uso e o compartilhamento de códigos prontos que

implementam diversas funcionalidades. Um dos gerenciadores de pacotes mais populares e eficientes é o Yarn, criado pelo Facebook em 2016, com a colaboração de outras empresas como o Google (YARN, 2022).

O Yarn utiliza os bancos de dados do NPM e do Bower, dois outros gerenciadores de pacotes renomados, mas oferece vantagens como maior velocidade, segurança e confiabilidade. Para instalar o Yarn, é preciso ter o Node.js instalado no sistema operacional, pois ele depende do npm para funcionar. Em seguida, basta executar o comando `npm install -g yarn` para instalar o Yarn globalmente. Após a instalação, é possível usar o Yarn para gerenciar as dependências de um projeto Javascript, criando um arquivo `package.json` que contém as informações sobre o projeto e os pacotes necessários (SOUZA, 2020).

Para adicionar um pacote ao projeto, basta usar o comando `yarn add nome-do-pacote`, que irá baixar o pacote do repositório e atualizar o arquivo `package.json`. O Yarn também cria um arquivo `yarn.lock`, que armazena as versões exatas dos pacotes instalados, garantindo a consistência e a reprodutibilidade do projeto (NICKSON, 2023).

2.9 Github e Gitflow

2.9.1 Github

GitHub é uma plataforma de hospedagem de código-fonte e colaboração para desenvolvedores de *software*. Ele permite que os usuários criem, gerenciem e compartilhem projetos de *software* usando o sistema de controle de versão Git. O mesmo também oferece recursos adicionais, como revisão de código, integração contínua, documentação e gerenciamento de projetos. É amplamente utilizado pela comunidade científica para armazenar, divulgar e reproduzir pesquisas de diferentes áreas do conhecimento. Além disso, facilita o acesso aberto à produção científica, permitindo que os usuários encontrem e consultem artigos, teses, dissertações, livros, capítulos de livro, trabalhos de conclusão de curso, artigos de conferência, dados de pesquisa e outros tipos de documentos científicos (CHACON; STRAUB, 2018).

2.9.2 GitFlow

O GitFlow é um modelo flexível e adaptável que pode ser ajustado às necessidades e preferências de cada projeto. Ele oferece uma estrutura clara e consistente para o desenvolvimento de software com o Git, facilitando a colaboração entre os desenvolvedores e a entrega de produtos

de qualidade (TERRALAB, 2019).

O fluxo de trabalho do GitFlow segue um ciclo que envolve as seguintes etapas: inicialização, criação de *feature*, finalização de *feature*, criação de *release*, finalização de *release* e criação de *hotfix*. Cada etapa corresponde a um conjunto de comandos do Git ou da extensão do GitFlow que manipulam as ramificações do repositório conforme o modelo (XPE, 2022).

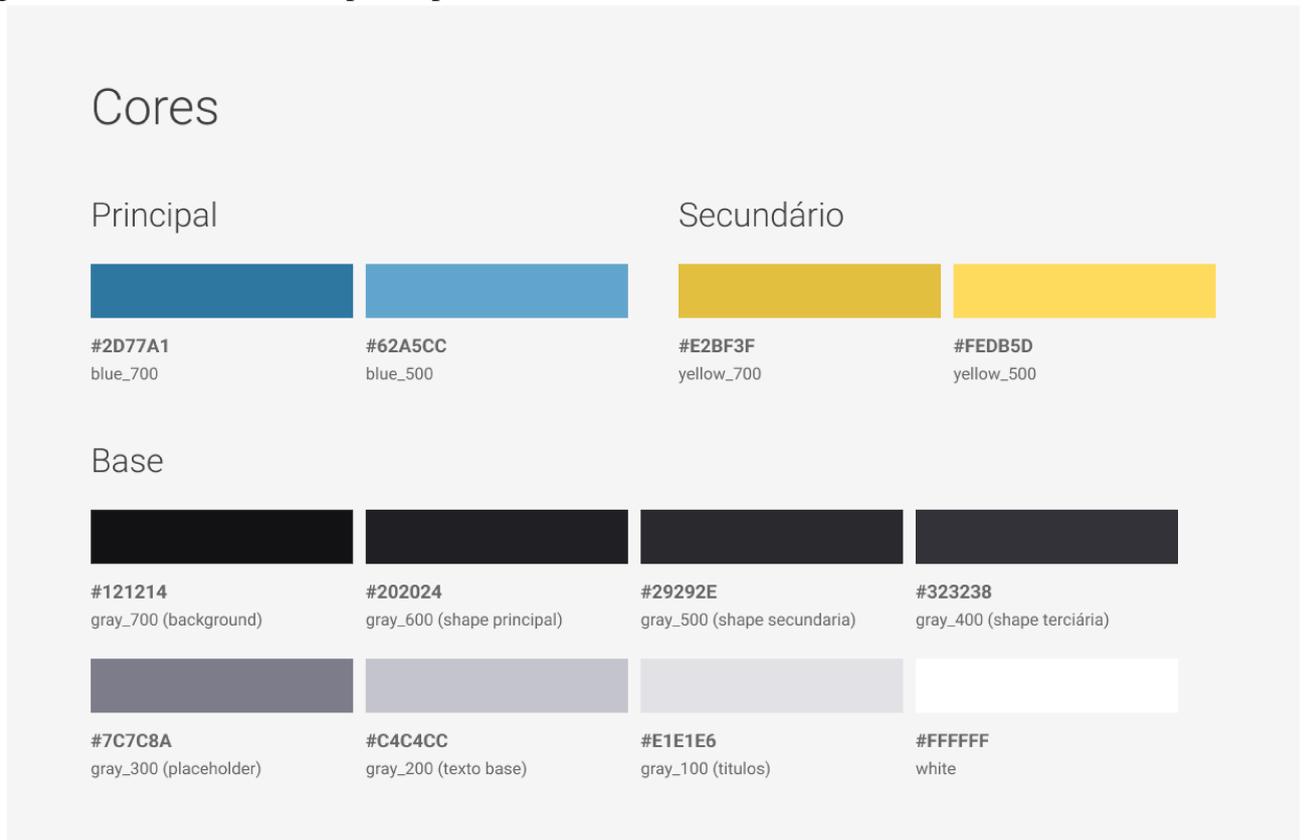
3 METODOLOGIA

Ao longo desta seção serão descritos e detalhados as etapas para o desenvolvimento da aplicação proposta.

3.1 Criação do *Design System*

Antes do desenvolvimento do código foi realizado um levantamento de requisitos para as funcionalidades e características do projeto. Com isso, foi definido um *Design System* para ser usado como base no desenvolvimento e referência, para melhor controle de tempo, consistência e usabilidade da plataforma, utilizando a ferramenta de *design* de interfaces, Figma. Primeiramente foi definido as referências em que se basearia a interface, como a paleta de cores, tipografia, imagens e exemplos de interfaces para o desenvolvimento do protótipo a ser desenvolvido, demonstrados respectivamente nas Figuras 1, 2, 3, 4.

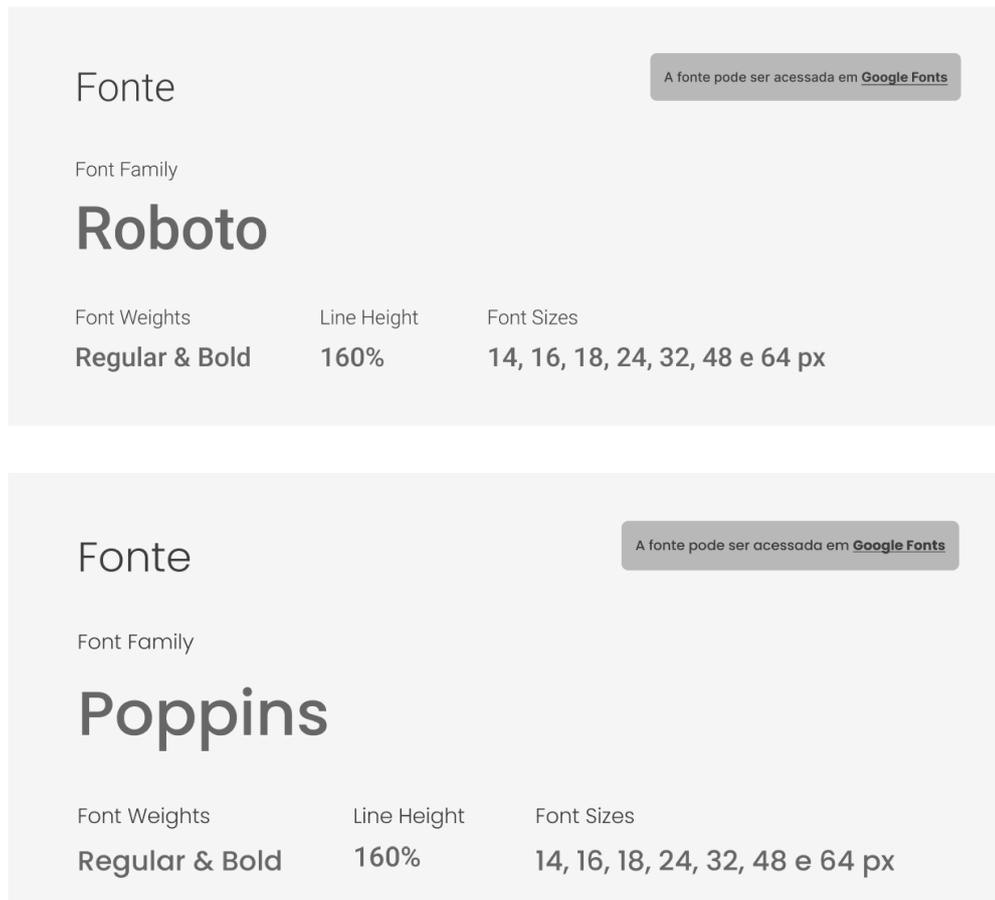
Figura 1 – Paleta de cores do protótipo



Fonte: Elaborado pelo autor.

Após a definição dos critérios anteriores foi realizado o desenvolvimento de um protótipo de interface levando em consideração os requisitos levantados anteriormente, vistos

Figura 2 – Tipografia do protótipo



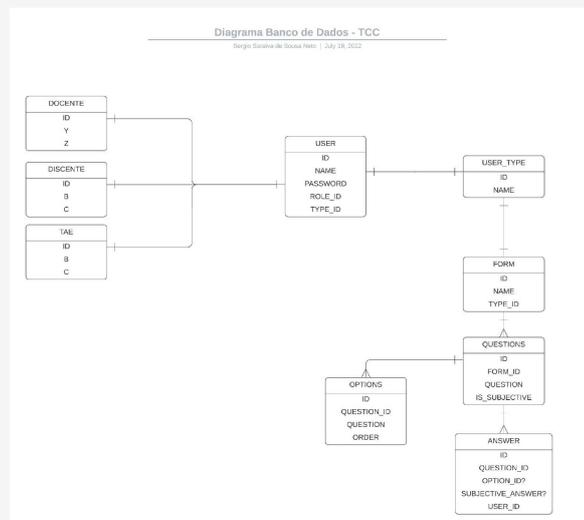
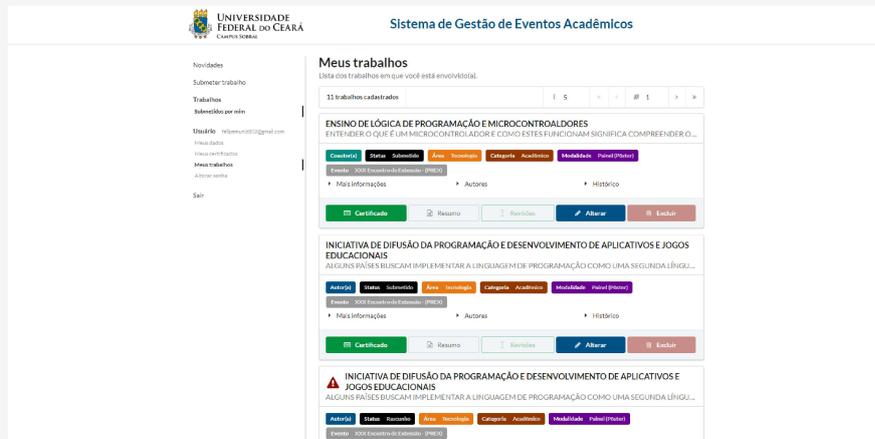
Fonte: Elaborado pelo autor.

Figura 3 – Exemplos de imagens



Fonte: Elaborado pelo autor.

Figura 4 – Exemplos base para protótipo



Fonte: Elaborado pelo autor.

nas Figuras 5, 6. Usando a característica de componentizar as partes para o desenvolvimento ser realizado de forma mais eficiente e extrair o potencial das tecnologias escolhidas.

Figura 5 – Protótipo da página inicial



Fonte: Elaborado pelo autor.

Figura 6 – Protótipo da área administrativa

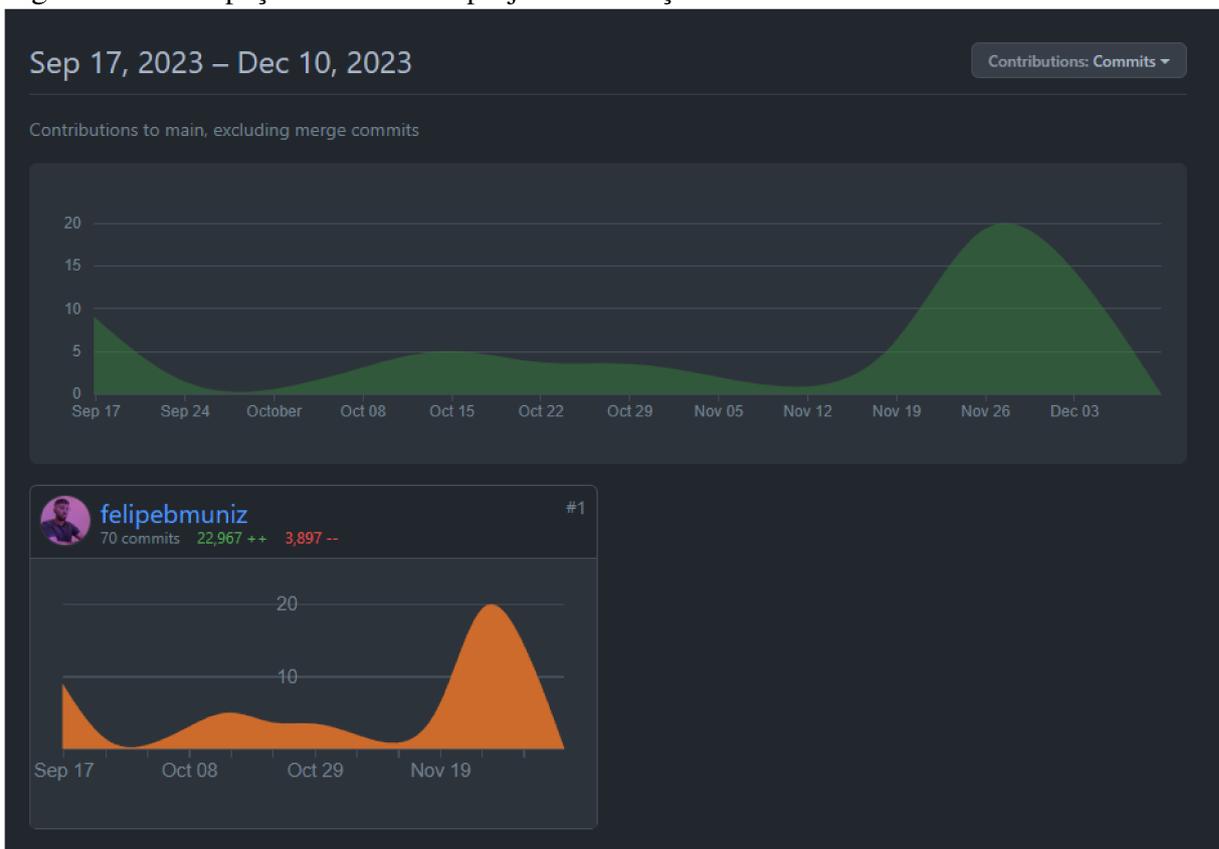


Fonte: Elaborado pelo autor.

3.2 Desenvolvimento do *Front-end*

Com a conclusão e validação do *Design System*, foi dado início ao desenvolvimento do projeto nas linguagens JavaScript em conjunto com o TypeScript, utilizando a biblioteca ReactJS e *framework* NextJS. Primeiramente foi gerado o código base do projeto usando o gerenciador de pacotes Yarn, que também foi utilizado para a instalação e gerenciamento das bibliotecas utilizadas no decorrer do projeto. A seguir o projeto foi vinculado a plataforma de gerenciamento de códigos Github, assim podendo ser feito o controle de versão e realizar a disponibilidade em nuvem do mesmo, facilitando o acesso e gerando segurança no armazenamento do código em si. As implementações de desenvolvimento seguiram um padrão de GitFlow, trazendo padronização e uma estrutura sólida para mapeamento de modificações, identificação de erros e futuras implementações. Assim proporcionando uma longevidade para o projeto. A seguinte Figura 7, mostra respectivamente a participação no Github mediante os *commits* realizados.

Figura 7 – Participação efetuado no projeto em relação *commits*



Fonte: <https://github.com/felipebmuniz/plataforma-avaliativa-mestrado-tcc/graphs/contributors>

O projeto foi desenvolvido utilizando o sistema de pasta denominada padrão como */src*, internamente dela encontrasse a pasta chamada de */pages*, onde o sistema de roteamento

entre páginas do NextJS é realizado, que é denominada uma das qualidades de se utilizar este *framework*, facilitando o desenvolvimento. Para o controle dos dados internos foi utilizado dos *hooks* da biblioteca ReactJS, em específico sendo o *useState*, para gerenciamento de valores, o *useEffect*, utilizado para observar as mudanças de estados e realizar a nova renderização da VDOM, e o *useContext*, responsável pela disponibilidade de valores e funções de forma global ou em partes específicas do projeto entre si, facilitando o acesso desses valores. Para consulta e envio dos dados foi utilizado a biblioteca Axios, que é um cliente *Hypertext Transfer Protocol* (HTTP), juntamente com ferramentas de segurança de acesso de rotas HTTP disponibilizada pelo NextJS, realizando uma máscara de acesso ocultando a verdadeira rota acessada.

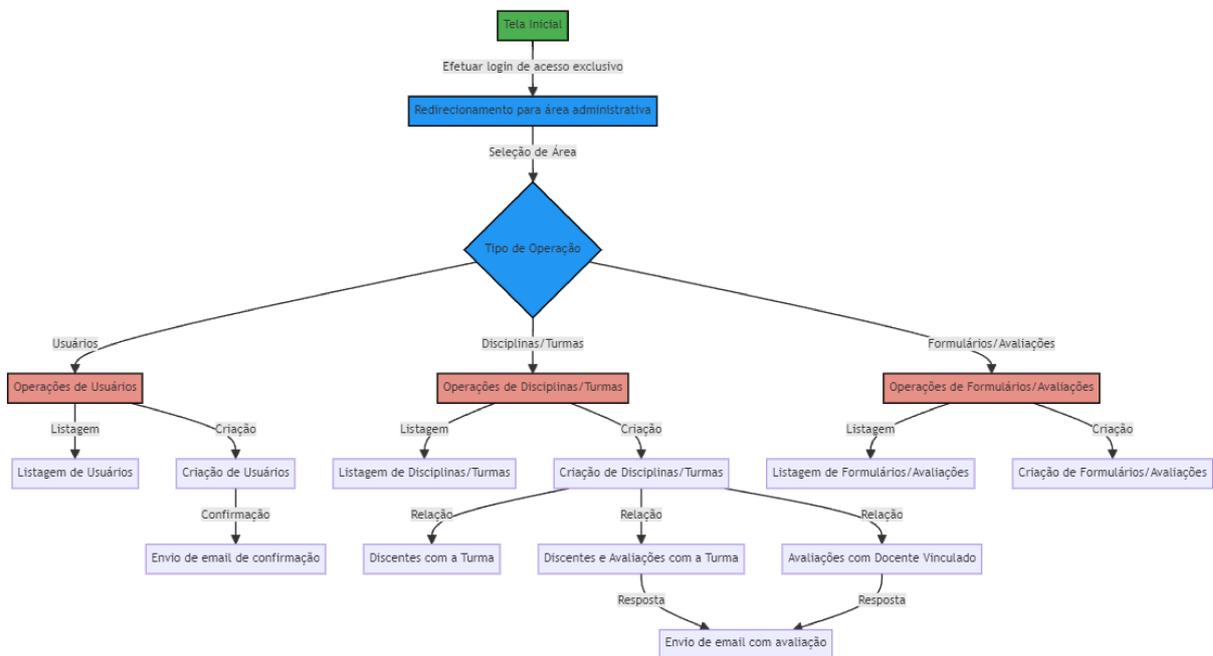
Com o projeto devidamente estruturado, foi realizada a *build* e o *deploy* na nuvem, nos servidores da empresa Vercel, criadora do *framework* NextJS. Após a conclusão da etapa anterior foi realizado o apontamento do *Domain Name System* (DNS) para o acesso se tornar disponível publicamente.

4 RESULTADOS

4.1 Resultados do desenvolvimento do *Front-end*

Como resultado do desenvolvimento do *Front-end* o seguinte diagrama visto na Figura 8, detalha os passos de acesso e rotas disponíveis dentro do sistema administrativo. Com base no fluxo demonstrado na Figura 8, é detalhado as telas desenvolvidas.

Figura 8 – Diagrama de fluxo da plataforma



Fonte: Elaborado pelo autor.

A Figura 9 mostra a interface inicial da plataforma, onde se tem a possibilidade de realizar o acesso a área restrita, que é destinada ao administrador da plataforma. Este acesso é feito pela validação das credenciais de acesso onde retornará um *JSON Web Token (JWT)* que é essencial nas futuras consultas a API, que são compostas por rotas protegidas.

Na Figura 10 podemos observar a área de listagem de usuários, que são definidos em dois tipos, sendo eles discentes e docentes, utilizando um sistema de tabela para a organização dos dados. A navegação entre os tipos é realizada por uma sub-aba de acesso onde em cada uma possui um botão para adicionar um novo usuário do tipo específico, sendo observado um

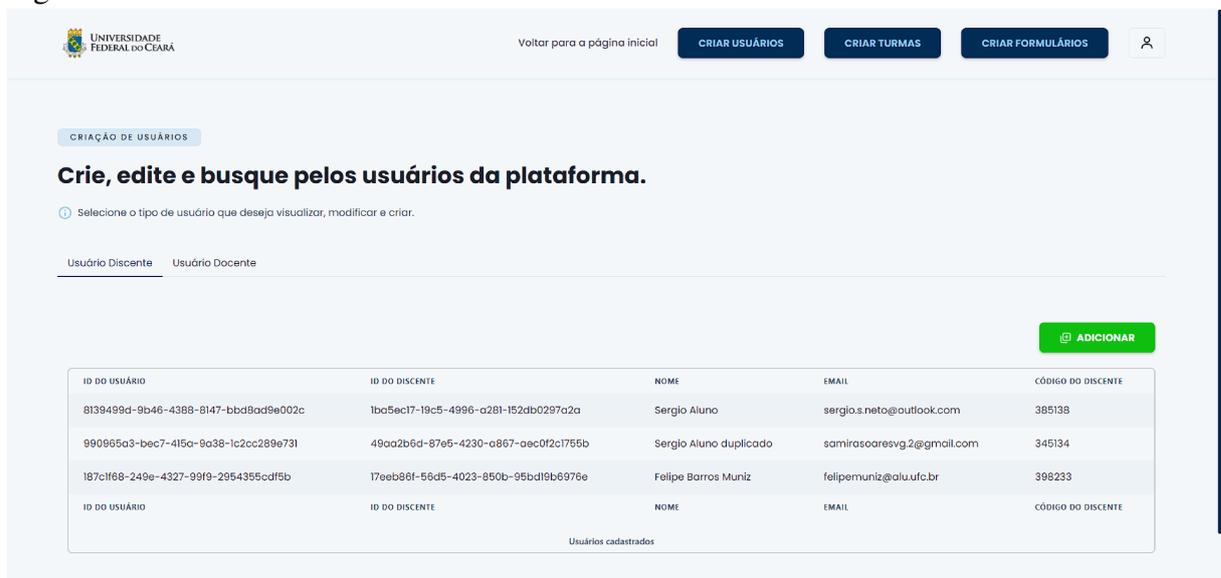
Figura 9 – Interface da página inicial



Fonte: Elaborado pelo autor.

formulário de criação, onde será pedido valores específicos para a realização das requisições para a API, vista na Figura 11. sendo necessário para os próximos passos o usuário estar devidamente validado, onde ocorrerá por meio de um *link* enviado por *email*, que será redirecionado para uma tela que efetua a validação, demonstrado na Figura 12.

Figura 10 – Interface administrativa de usuários



Fonte: Elaborado pelo autor.

Sendo assim por meio da componentização de certas áreas da plataforma que nos é permitido pelas tecnologias adotadas, os seguintes fluxos se comportam igualmente as telas de usuários, tendo uma área de listagem e de criação em suas respectivas abas de acesso. Após

Figura 11 – Interface de criação de usuários

The interface is for creating users in a system. It features a main dashboard with a table of existing users and a modal for creating new ones.

Table of Existing Users:

ID DO USUÁRIO	ID DO DISCENTE	NOME	EMAIL
8139499d-9b46-4388-8147-bbd8ad9e002c	1ba5ec17-19c5-4996-a281-152db0297a2a	Sergio Aluno	sergio.s.neto@
990955a3-bec7-415a-9a38-1c2cc289e731	49aa2b6d-87e5-4230-a867-aec0f2c1755b	Sergio Aluno duplicado	samirasoaresv
187c1f68-249e-4327-99f9-2954355cdf5b	17aeb96f-56a5-4023-850b-95bd19b6976e	Felipe Barros Muniz	felpemuniz@a
ID DO USUÁRIO	ID DO DISCENTE	NOME	EMAIL

Modal: Criar Novo Usuário

Nome:

E-mail:

Código:

Buttons: CANCELAR, CADASTRAR

Fonte: Elaborado pelo autor.

Figura 12 – Interface de validação de usuários

The interface is a validation screen for users. It features a loading indicator and a message.

VALIDAÇÃO DE USUÁRIOS

Espaço destinado a validação do usuário em nossa base de dados.

Espera pacientemente enquanto validamos seu usuário, assim que terminado você será redirecionado para a tela principal.

©2023 FELIPE BARROS MUNIZ - TODOS OS DIREITOS RESERVADOS

Fonte: Elaborado pelo autor.

criação dos usuários o fluxo se dirige para a criação de turmas e disciplinas, demonstrado nas Figuras 13 e 14, a criação dos mesmos observadas nas Figuras 15 e 16.

A criação dos formulários e avaliação e um dos últimos processos a serem realizados. A interface de listagem e navegação se mantém, demonstrados na Figura 17, porém, a criação dos formulário se dá de forma dinâmica, onde se passa o título, pergunta ou perguntas e suas respectivas opções, demonstrado na Figura 18 e a criação das avaliação é necessário datas de início e fim para o período de avaliação, demonstrado na Figura 19.

A vinculação das entidades são feitas nas turmas, onde é visto as opções na tabela de

Figura 13 – Interface administrativa de disciplinas

UNIVERSIDADE FEDERAL DO CEARÁ

Voltar para a página inicial

CRIAR USUÁRIOS

CRIAR TURMAS

CRIAR FORMULÁRIOS

ADICIONAR

CRIAÇÃO DE TURMAS E DISCIPLINAS

Crie, edite e busque pelas Turmas e Disciplinas da plataforma.

As turmas e disciplinas são essenciais para o funcionamento da plataforma. Com elas são possíveis vincular discentes e docentes para responder os formulários.

Área de Disciplinas

CÓDIGO	NOME	DATA DE CRIAÇÃO
F101	FISICA 1	2023-11-24T01:55:54.885436
F201	Fisica 2	2023-11-28T02:43:21.36041
CÓDIGO	NOME	DATA DE CRIAÇÃO

Disciplinas cadastrados

Fonte: Elaborado pelo autor.

Figura 14 – Interface administrativa de turmas

UNIVERSIDADE FEDERAL DO CEARÁ

Voltar para a página inicial

CRIAR USUÁRIOS

CRIAR TURMAS

CRIAR FORMULÁRIOS

ADICIONAR

CRIAÇÃO DE TURMAS E DISCIPLINAS

Crie, edite e busque pelas Turmas e Disciplinas da plataforma.

As turmas e disciplinas são essenciais para o funcionamento da plataforma. Com elas são possíveis vincular discentes e docentes para responder os formulários.

Área de Disciplinas

Área de Turmas

CÓDIGO	ANO DE VIGÊNCIA	DISCIPLINA	DOCENTE	DATA DE CRIAÇÃO	OPÇÕES DE VINCULAÇÕES
CLS-FIS2	2023	Fisica 2 - F201	Professor Felipe - 111111	2023-11-29T03:35:06.092432	⌵
CLS-FIS1	2023	FISICA 1 - F101	Professor Felipe - 111111	2023-11-29T02:10:46.5	<ul style="list-style-type: none"> Vincular Discentes Vincular Avaliação para os Discentes Vincular Avaliação para o Docente
CÓDIGO	ANO DE VIGÊNCIA	DISCIPLINA	DOCENTE	DATA DE CRIAÇÃO	LAÇÕES

Disciplinas cadastrados

Fonte: Elaborado pelo autor.

listagem, tendo as opções de vincular discente a turma, discente a avaliação e docente a avaliação. Vistas respectivamente nas Figuras 20, 21 e 22. Logo após a relação os usuários recebem um *link* enviado por *email*, onde se realiza a avaliação, vista na Figura 23.

Figura 15 – Interface de criação de disciplinas

UNIVERSIDADE FEDERAL DO CEARÁ

Voltar para a página inicial CRIAR USUÁRIOS CRIAR TURMAS

criação de turmas e disciplinas

Crie, edite e busque pelas Turmas e Disciplinas da plataforma.

As turmas e disciplinas são essenciais para o funcionamento da plataforma. Com elas são possíveis vincular discentes e docentes para responder os formulários

Área de Disciplinas Área de Turmas

CÓDIGO	NOME	DATA DE CRIAÇÃO
FI101	FISICA 1	2023-11-24T01:55:54.885436
FI201	Física 2	2023-11-28T02:43:21.36041
CÓDIGO	NOME	DATA DE CRIAÇÃO

Disciplinas cadastrados

Criar Nova Disciplina

Nome

Nome da Disciplina *

Deve ser informado um nome!

Código

Código da Disciplina *

Deve ser informado um código!

CANCELAR CADASTRAR

Fonte: Elaborado pelo autor.

Figura 16 – Interface de criação de turmas

UNIVERSIDADE FEDERAL DO CEARÁ

Voltar para a página inicial CRIAR USUÁRIOS CRIAR TURMAS

criação de turmas e disciplinas

Crie, edite e busque pelas Turmas e Disciplinas da plataforma.

As turmas e disciplinas são essenciais para o funcionamento da plataforma. Com elas são possíveis vincular discentes e docentes para responder os formulários

Área de Disciplinas Área de Turmas

CÓDIGO	ANO DE VIGÊNCIA	DISCIPLINA	DOCENTE	DATA DE CRIAÇÃO
CLS-FIS2	2023	Física 2 - FI201	Professor Felipe - 111111	2023-11-29T03:35:06.092432
CLS-FIS1	2023	FISICA 1 - FI101	Professor Felipe - 111111	2023-11-29T02:10:46.525487
CÓDIGO	ANO DE VIGÊNCIA	DISCIPLINA	DOCENTE	DATA DE CRIAÇÃO

Disciplinas cadastrados

Criar Nova Turma

Código

Código da Turma *

Período

Integral

Ano

2023

Estado

Aberta

Disciplina

Disciplina da Turma

Professor

Professor da Turma

CANCELAR CADASTRAR

Fonte: Elaborado pelo autor.

Figura 17 – Interface administrativa de formulários

UNIVERSIDADE FEDERAL DO CEARÁ

Voltar para a página inicial

CRIAR USUÁRIOS CRIAR TURMAS CRIAR FORMULÁRIOS

CRIAÇÃO DE FORMULÁRIOS

Crie, edite e busque pelos formulários da plataforma.

Os formulários são a base da plataforma, com eles podemos saber como se decorreu o funcionamento durante o semestre.

Área dos Formulários Área de Avaliações

ADICIONAR

ID	NOME	DATA DE CRIAÇÃO
98637148-837b-417a-b004-62321ce4113d	FORMULARIO TESTE	2023-11-24T01:57:53.307198
c9e40caf-e7d6-4be4-b857-2571e114e0ab	Teste (Plataforma)	2023-12-01T15:29:08.203284
f221f577-6dba-45bd-8224-a5c91fadf355e	Formulário via plataforma	2023-12-01T15:35:33.864355
ID	NOME	DATA DE CRIAÇÃO

Formulários cadastrados

Fonte: Elaborado pelo autor.

Figura 18 – Interface de criação de formulários

UNIVERSIDADE FEDERAL DO CEARÁ

Voltar para a página inicial

CRIAR USUÁRIOS

CRIAÇÃO DE FORMULÁRIOS

Crie, edite e busque pelos formulários da plataforma.

Os formulários são a base da plataforma, com eles podemos saber como se decorreu o funcionamento durante o semestre.

Área dos Formulários Área de Avaliações

ID	NOME
98637148-837b-417a-b004-62321ce4113d	FORMULARIO TESTE
c9e40caf-e7d6-4be4-b857-2571e114e0ab	Teste (Plataforma)
f221f577-6dba-45bd-8224-a5c91fadf355e	Formulário via plataforma
ID	NOME

Formulários cadastrados

©2023 FELIPE BARROS MUNIZ - TODOS OS DIREITOS RESERVADOS

Criar Novo Formulário

Titulo

Titulo do formulário ⚡

Perguntas

Pergunta

Titulo da pergunta ⚡

Pergunta discursiva?

Opções

Opção

Titulo da Opção ⚡

OPÇÃO

PERGUNTA

CANCELAR CADASTRAR

Fonte: Elaborado pelo autor.

Figura 19 – Interface de criação de avaliações

UNIVERSIDADE FEDERAL DO CEARÁ

Voltar para a página inicial

criar usuários

criar turmas

criação de formulários

Crie, edite e busque pelos formulários da plataforma.

Os formulários são a base da plataforma, com eles podemos saber como se decorreu o funcionamento durante o semestre.

Área dos Formulários

Área de Avaliações

TÍTULO	DATA DE INÍCIO	DATA DE ENCERAMENTO	FORMULÁRIO
Avaliação Teste	2023-11-30T12:58:00	2023-12-04T00:00:00	98637148-837b-417a-b004-6
TÍTULO	DATA DE INÍCIO	DATA DE ENCERAMENTO	FORMULÁRIO

Avaliações cadastrados

©2023 FELIPE BARROS MUNIZ - TODOS OS DIREITOS RESERVADOS

criar nova avaliação

Título

Título da Avaliação *

Início

mm/dd/yyyy --:-- --

Termino

mm/dd/yyyy --:-- --

Formulário

Formulário da Avaliação

CANCELAR

CADASTRAR

Fonte: Elaborado pelo autor.

Figura 20 – Interface de relação de discente com a turma

UNIVERSIDADE FEDERAL DO CEARÁ

Voltar para a página inicial

criar usuários

criação de turmas e disciplinas

Crie, edite e busque pelas Turmas e Disciplinas da plataforma.

As turmas e disciplinas são essenciais para o funcionamento da plataforma. Com elas são possíveis vincular discentes e docentes para resp

Área de Disciplinas

Área de Turmas

CÓDIGO	ANO DE VIGÊNCIA	DISCIPLINA	DOCENTE	DATA DE CRIAÇÃO
CLS-FIS2	2023	Física 2 - FI201	Professor Felipe - 111111	2023-11-29T03:35
CLS-FIS1	2023	FISICA 1 - FI101	Professor Felipe - 111111	2023-11-29T02:10
CÓDIGO	ANO DE VIGÊNCIA	DISCIPLINA	DOCENTE	DATA DE CRIAÇÃO

Disciplinas cadastrados

criar relação do discente com turma

Selecione os discentes a ser relacionados

Sergio Aluno - 385138

Sergio Aluno duplicado - 345134

Felipe Barros Muniz - 398233

CANCELAR

CADASTRAR

Fonte: Elaborado pelo autor.

Figura 21 – Interface de relação de discente com avaliação

The screenshot shows the 'UNIVERSIDADE FEDERAL DO CEARÁ' interface. The main page is titled 'CRIAÇÃO DE TURMAS E DISCIPLINAS' and 'Crie, edite e busque pelas Turmas e Disciplinas da plataforma.' It features a table of disciplines and a modal window for creating a relationship between a class and an evaluation.

Table: Disciplinas cadastradas

CÓDIGO	ANO DE VIGÊNCIA	DISCIPLINA	DOCENTE	DATA DE CRIAÇÃO
CLS-FIS2	2023	Física 2 - F201	Professor Felipe - 111111	2023-11-29T03:35
CLS-FIS1	2023	FISICA 1 - F101	Professor Felipe - 111111	2023-11-29T02:10

Modal: Criar Relação da Turma de Discentes com a Avaliação

Seleção a avaliação a ser relacionada

- Avaliação Teste

Buttons: CANCELAR, CADASTRAR

Fonte: Elaborado pelo autor.

Figura 22 – Interface de relação de docente com avaliação

The screenshot shows the 'UNIVERSIDADE FEDERAL DO CEARÁ' interface. The main page is identical to Figure 21. The modal window is titled 'Criar Relação do Docente da Turma com a Avaliação' and shows a different selection option for the evaluation.

Table: Disciplinas cadastradas

CÓDIGO	ANO DE VIGÊNCIA	DISCIPLINA	DOCENTE	DATA DE CRIAÇÃO
CLS-FIS2	2023	Física 2 - F201	Professor Felipe - 111111	2023-11-29T03:35
CLS-FIS1	2023	FISICA 1 - F101	Professor Felipe - 111111	2023-11-29T02:10

Modal: Criar Relação do Docente da Turma com a Avaliação

Seleção a avaliação a ser relacionada

- Avaliação Teste

Buttons: CANCELAR, CADASTRAR

Fonte: Elaborado pelo autor.

Figura 23 – Interface de avaliação

UNIVERSIDADE FEDERAL DO CEARÁ

Avalie o programa

ACESSO EXCLUSIVO

AValiação DO PROGRAMA

Avalie nosso sistema para que possamos entender as necessidades e percepção do usuário.

Leia e marque a melhor opção nas perguntas levando em consideração sua experiência na plataforma.

1. Eu acho que gostaria de usar o Sistema de Autoavaliação com frequência.*

Discordo Totalmente Discordo Neutro Concordo Concordo Totalmente

2. Eu acho o Sistema de Autoavaliação desnecessariamente complexo.*

Discordo Totalmente Discordo Neutro Concordo Concordo Totalmente

Fonte: Elaborado pelo autor.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Podemos concluir, por meio deste trabalho, que conseguimos construir uma interface funcional que atende a todos os requisitos levantados anteriormente visto no trabalho. Atendendo a proposta de realizar a Autoavaliação do Programa de Pós-Graduação em Engenharia Elétrica e Computação, proporcionando uma experiência agradável e de fácil entendimento com poucos passos e eliminando a dependência de terceiros.

Porém o projeto tem capacidades de crescimento ainda não explorados e recursos a serem desenvolvidos e implementados, para melhorar ainda mais a sua usabilidade. Partes como gráficos para melhor visualização dos dados obtidos geração de relatórios e novos tipos de usuários e campanhas de avaliação a serem integrados para um futuro desenvolvimento.

6 TRABALHOS RELACIONADOS

Nesta seção é destinada a referenciar os trabalhos que serviram como base inicial de da criação deste trabalho, assim como trabalhos que o complementam.

6.1 Sistema de Autoavaliação - PPGEEC/UFC

Como base este trabalho de conclusão de curso feito anteriormente, foi possível a realização de uma análise dos pontos de melhorias e abordagens de solução diferenciadas para a obtenção de um resultado esperado na resolução do trabalho atual.

6.2 Coordenador API: API REST para controle e gerenciamento de sistema de avaliação do programa de pós-graduação em engenharia elétrica e computação

O seguinte trabalho é um complementar direto em relação ao consumo de dados e armazenamentos dos mesmos. Tendo finalidade de proporcionar uma integração por meio de uma API REST realizando a comunicação com um banco de dados.

REFERÊNCIAS

- BRASIL. **Parecer nº 977/1965, aprovado em 3 dez. 1965.** 1965. Disponível em: <http://www.scielo.br/pdf/rbedu/n30/a14n30.pdf>. Acesso em: 21 dez. 2021.
- CAPES. Autoavaliação de programas de pós-graduação. **Capex**, Capes, v. 1, n. 1, 2019.
- CAPES. **CAPES - Plataforma Sucupira.** 2020. Disponível em: <https://sucupira.capes.gov.br/sucupira/>. Acesso em: 03 dez. 2023.
- CHACON, S.; STRAUB, B. **Pro Git.** Apress, 2018. Disponível em: <https://github.com/progit/progit2-pt-br>. Acesso em: 03 dez. 2023.
- HEVNER, A. R.; MARCH, S. T.; PARK, J.; RAM, S. Design science in information systems research. **MIS Quarterly**, JSTOR, v. 28, n. 1, p. 75–105, 2004. Disponível em: <https://doi.org/10.2307/25148625>.
- INDLA, B. V. S.; PURANIK, Y. Review on react js. **International Journal of Trend in Scientific Research and Development (IJTSRD)**, v. 5, n. 4, p. 1136–1138, 2021.
- JOVENSGENIOS. **Avaliação Interna e Avaliação externa.** 2020. Disponível em: <https://blog.jovensgenios.com/avaliacao-interna-e-externa/#:~:text=Conhecida%20como%20avalia%C3%A7%C3%A3o%20de%20desempenho,o%20redirecionamento%20de%20pr%C3%A1ticas%20pedag%C3%B3gicas>. Acesso em: 20 jan. 2022.
- KULKARNI, A. S. *et al.* Performance optimization techniques for reactjs. In: **2019 International Conference on Communication and Signal Processing (ICCSP)**. [S. l.: s. n.], 2019. p. 564–568.
- LEITE, D. e. a. **A autoavaliação na Pós-Graduação (PG) como componente do processo avaliativo CAPES.** Revista da Avaliação da Educação Superior (Campinas) [online], 2020. 339–353 p. Disponível em: <https://doi.org/10.1590/S1414-4077/S1414-40772020000200006>. Acesso em: 03 dez. 2023.
- MDN. **Introdução - JavaScript | MDN.** 2023. Disponível em: <https://developer.mozilla.org/pt-BR/docs/web/javascript/guide/introduction>. Acesso em: 03 dez. 2023.
- NEXT.JS. **Next.js Documentation.** 2023. Disponível em: <https://nextjs.org/docs>. Acesso em: 03 dez. 2023.
- NICKSON, D. **Como instalar o Yarn: um guia passo a passo.** 2023. <https://douglasnickson.com.br/2023/04/19/como-instalar-o-yarn-um-guia-passo-a-passo/>. Acesso em: 22 dez. 2023.
- PPGEEC. **Resolução N° 1/2021/CMEEC/CUFCSOBRAL/REITORIA. Dispõe sobre regulamento e plano de ação do processo de autoavaliação do Programa de Pós-graduação em Engenharia Elétrica e Computação da Universidade Federal do Ceará - Campus de Sobral (PPGEEC/UFC).** 2021. Disponível em: <https://ppgeec.ufc.br/wp-content/uploads/2021/02/sei-ppgeec-resolucao-autoavaliacao.pdf>. Acesso em: 03 dez. 2023.
- SAMUDIO, D. I.; LATOZA, T. D. Barriers in front-end web development. **IEEE Xplore**, 2022. Disponível em: <https://ieeexplore.ieee.org/document/9833127/>.

SÁNCHEZ, M. A. *et al.* humanportal – a react.js case study. In: **2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)**. [S. l.: s. n.], 2020. p. 1637–1642.

SOUZA, I. **O que é Yarn e como funciona seu gerenciamento de pacotes**. 2020. <https://rockcontent.com/br/blog/yarn/>. Acesso em: 22 dez. 2023.

TERRALAB. **Uma introdução ao Git e Gitflow**. 2019. Disponível em: <http://www2.decom.ufop.br/terralab/uma-introducao-ao-git-e-gitflow/>. Acesso em: 03 dez. 2023.

TYPESCRIPT. **TypeScript: O ponto inicial para aprender Typescript**. 2023. Disponível em: <https://www.typescriptlang.org/pt/docs/>. Acesso em: 03 dez. 2023.

WIERINGA, R. How software engineering research aligns with design science: A review. **Empirical Software Engineering**, Springer, v. 25, n. 1, p. 420–457, 2020. Disponível em: <https://doi.org/10.1007/s10664-020-09818-7>.

WIKIPEDIA. **React (JavaScript library)**. 2020. Disponível em: [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library)). Acesso em: 03 dez. 2023.

WIKIPEDIA. **JavaScript - wikipedia**. 2023. Disponível em: <https://pt.wikipedia.org/wiki/JavaScript>. Acesso em: 03 dez. 2023.

XPE. **GitFlow: o que é e para que ele serve**. 2022. Disponível em: <https://blog.xpeducacao.com.br/gitflow/>. Acesso em: 03 dez. 2023.

YARN. **Home page | Yarn**. 2022. Disponível em: <https://yarnpkg.com/>. Acesso em: 21 dez. 2023.