



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

FELIPE DE AMORIM FERREIRA

REFORMULAÇÃO DE INTERFACE PARA O PROJETO SAPS

CAMPINA GRANDE - PB

2023

FELIPE DE AMORIM FERREIRA

REFORMULAÇÃO DE INTERFACE PARA O PROJETO SAPS

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de **Bacharel ou Bacharela em Ciência da Computação.**

Orientador: Professor Dr. Thiago Emmanuel Pereira

CAMPINA GRANDE - PB

2023

FELIPE DE AMORIM FERREIRA

REFORMULAÇÃO DE INTERFACE PARA O PROJETO SAPS

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de **Bacharel ou Bacharela em Ciência da Computação.**

BANCA EXAMINADORA:

Professor Dr. Thiago Emmanuel Pereira

Orientador – UASC/CEEI/UFCG

Professor Dr. Marcelo Alves Barros

Examinador – UASC/CEEI/UFCG

Professor Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 28 de junho de 2023.

CAMPINA GRANDE - PB

RESUMO (ABSTRACT)¹

O Serviço de Automação do Processamento de Imagens de Satélite (SAPS) é uma ferramenta usada para o monitoramento e previsão de fenômenos climáticos e ambientais. Este artigo aborda uma reformulação de sua interface de usuário. Esta reformulação teve o objetivo de melhorar a experiência do usuário e a eficiência da ferramenta, focando em cinco áreas-chave: acessibilidade, responsividade, consistência do design, manutenibilidade e testabilidade. Para isso, o projeto emprega o framework Next.js, trazendo uma série de vantagens, como uma experiência de usuário aprimorada, facilidade de manutenção e uma integração mais eficiente para novos colaboradores. O artigo detalha o processo de implementação, descrevendo a estrutura do projeto, a seleção de dependências e a estratégia de testes. Essas melhorias estão voltadas não só para otimizar a funcionalidade atual da ferramenta, mas também para garantir sua sustentabilidade a longo prazo, preparando o SAPS para futuras inovações e demandas.

¹The Satellite Image Processing Automation Service (SAPS) is a tool used for monitoring and forecasting climatic and environmental phenomena. This article addresses a redesign of its user interface. This redesign aimed to improve user experience and tool efficiency, focusing on five key areas: accessibility, responsiveness, design consistency, maintainability, and testability. For this, the project employs the Next.js framework, bringing a series of advantages, such as improved user experience, ease of maintenance, and more efficient integration for new contributors. The article details the implementation process, describing the project structure, the selection of dependencies, and the testing strategy. These improvements are aimed not only at optimizing the current functionality of the tool but also ensuring its long-term sustainability, preparing SAPS for future innovations and demands.

Reformulação de interface para o Projeto SAPS

Felipe de Amorim Ferreira

Universidade Federal de Campina Grande

felipe.amorim.ferreira@ccc.ufcg.edu.br

Cilas Medeiros De Farias

Marques

Universidade Federal de Campina Grande

cilas@copin.ufcg.edu.br

Thiago Emmanuel Pereira

Universidade Federal de Campina Grande

temmanuel@computacao.ufcg.edu.br

RESUMO

O Serviço de Automação do Processamento de Imagens de Satélite (SAPS) é uma ferramenta usada para o monitoramento e previsão de fenômenos climáticos e ambientais. Este artigo aborda uma reformulação de sua interface de usuário. Esta reformulação teve o objetivo de melhorar a experiência do usuário e a eficiência da ferramenta, focando em cinco áreas-chave: acessibilidade, responsividade, consistência do design, manutenibilidade e testabilidade. Para isso, o projeto emprega o framework Next.js, trazendo uma série de vantagens, como uma experiência de usuário aprimorada, facilidade de manutenção e uma integração mais eficiente para novos colaboradores. O artigo detalha o processo de implementação, descrevendo a estrutura do projeto, a seleção de dependências e a estratégia de testes. Essas melhorias estão voltadas não só para otimizar a funcionalidade atual da ferramenta, mas também para garantir sua sustentabilidade a longo prazo, preparando o SAPS para futuras inovações e demandas.

PALAVRAS-CHAVE

Interface do usuário, UFCG, Acessibilidade, Implementação, Desenvolvimento web.

1. INTRODUÇÃO

Algoritmos para a estimativa de evapotranspiração (ET) têm sido amplamente utilizados em vários países e condições climáticas, desempenhando um papel crucial no monitoramento e previsão de fenômenos climáticos e ambientais [5]. Entre essas ferramentas, o Serviço de Automação do Processamento de Imagens de Satélite (SAPS)[2], desenvolvido pela Universidade Federal de Campina Grande, destaca-se por sua capacidade de processar imagens de satélite usando o método SEBAL, automatizando todo o fluxo de trabalho e facilitando a análise dos resultados para os usuários.

No entanto, a interface atual do SAPS apresenta alguns problemas significativos. A tecnologia utilizada para o dashboard atual não está mais sendo mantida e, desde a sua concepção, a interface não seguiu os princípios de design, componentização e acessibilidade reconhecidos na comunidade de desenvolvimento web. Além disso, a manutenção do sistema tem sido prejudicada pela rápida evolução das tecnologias e a necessidade de familiarizar os novos membros da equipe com tecnologias obsoletas. A dificuldade de atrair novos colaboradores para o projeto, em particular estudantes de graduação, também se tornou uma barreira, devido ao uso de tecnologias ultrapassadas e sem suporte.

Diante desses desafios, este trabalho propõe o desenvolvimento de uma nova interface web para o SAPS. Um dashboard que visa substituir o atual, superando suas limitações. Este novo dashboard será construído utilizando tecnologias modernas e comprovadas

no campo de desenvolvimento web, tais como React. Ele será projetado com base em princípios de acessibilidade, performance, testes e design, visando fornecer uma experiência de usuário robusta e acessível.

O foco deste artigo é apresentar o design e a implementação deste novo dashboard, além de discutir as motivações para a mudança e as vantagens da nova interface. A interface proposta não apenas melhorará a experiência do usuário final, mas também simplificará a manutenção do sistema e incentivará a integração de novos membros da equipe, incluindo estudantes de graduação, mantendo o SAPS relevante e eficiente frente às demandas atuais e futuras.

Este trabalho é organizado da seguinte maneira: A Seção 2 apresenta o contexto do Serviço de Automação do Processamento de Imagens de Satélite (SAPS), descrevendo a visão arquitetural e os detalhes da interface atual. Em seguida, na Seção 3, são discutidos os princípios fundamentais que nortearam o design da nova interface, incluindo acessibilidade, responsividade, consistência de design, manutenibilidade e testabilidade. A Seção 4 detalha a implementação da nova interface, abrangendo desde a escolha do framework e das dependências, passando pela estrutura do projeto, integração e testes. Finalmente, a Seção 5 oferece agradecimentos às partes envolvidas no projeto e a Seção 6 lista as referências bibliográficas utilizadas na elaboração deste estudo.

2. SAPS

A evolução da interface do SAPS demanda uma compreensão clara da arquitetura atual do sistema e de seus desafios existentes. As seções a seguir exploram detalhadamente a visão arquitetural do SAPS e os detalhes da interface atual, proporcionando um quadro essencial para entender os obstáculos superados e os princípios adotados no processo de redesign.

2.1. Visão arquitetural do SAPS

A arquitetura do serviço automático de processamento do SEBAL (SAPS) representa uma abordagem modular e escalável para o processamento de dados de sensoriamento remoto, estruturada em torno de nove componentes essenciais que juntos proporcionam uma solução de ponta a ponta para análise de imagens de satélite.

Em uma visão mais macro, a arquitetura do SAPS pode ser descrita como um fluxo de trabalho que vai desde a recepção de uma solicitação de processamento até a entrega de dados processados ao usuário. Este fluxo de trabalho é orquestrado pelos seguintes componentes: Submission Dispatcher, Scheduler, Workers, Software Repository, Service Catalog, Temporary

Storage, Permanent Storage, Journal Archiver e a Graphical User Interface (GUI) [2].

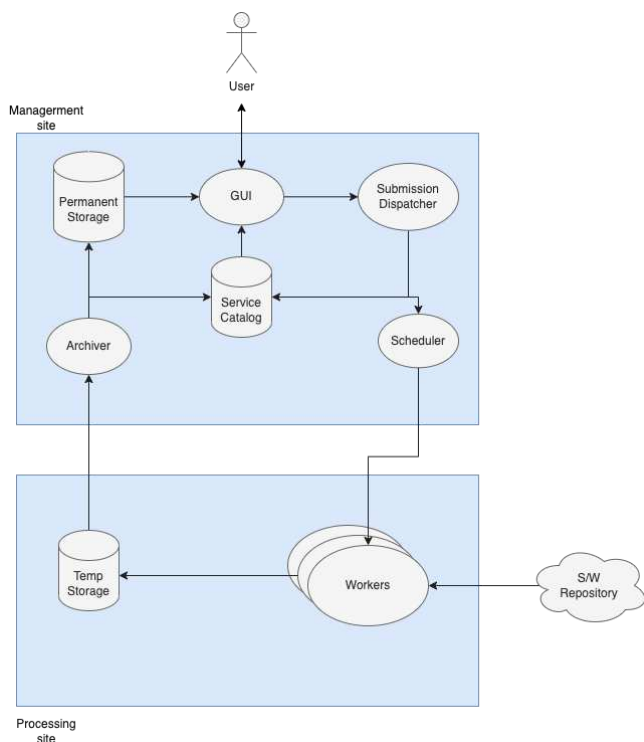


Imagem 1. Arquitetura do SAPS

O Submission Dispatcher atua como um gateway, aceitando solicitações de processamento dos usuários autenticados. Cada solicitação é então transformada em uma tarefa específica, pronta para ser processada pelos Workers. A distribuição dessas tarefas é gerenciada pelo Scheduler, que, baseado em algoritmos de escalonamento, decide qual tarefa deve ser atribuída a qual Worker.

Cada Worker é uma instância de processamento que realiza tarefas específicas de processamento de imagens. Estes Workers utilizam implementações de software obtidas do Software Repository, que age como uma biblioteca centralizada de algoritmos de processamento de imagem.

O Service Catalog é um componente fundamental que mantém um registro detalhado de todas as tarefas, rastreando o estado atual de cada tarefa, armazenando metadados de proveniência e mantendo referências para os dados de saída gerados.

Os dados gerados durante o processamento são armazenados no Temporary Storage durante a execução da tarefa, e uma vez concluída a tarefa, os dados finais são movidos para o Permanent Storage, onde permanecem disponíveis para recuperação pelo usuário.

O Journal Archiver é responsável por manter um registro de todas as atividades relacionadas às tarefas, incluindo metadados de proveniência, que são essenciais para garantir a rastreabilidade e a reprodutibilidade dos resultados gerados.

Finalmente, a Graphical User Interface (GUI) serve como o ponto principal de interação para os usuários. Através dela, os usuários podem enviar solicitações de processamento, monitorar o estado das tarefas e recuperar os dados de saída e metadados correspondentes.

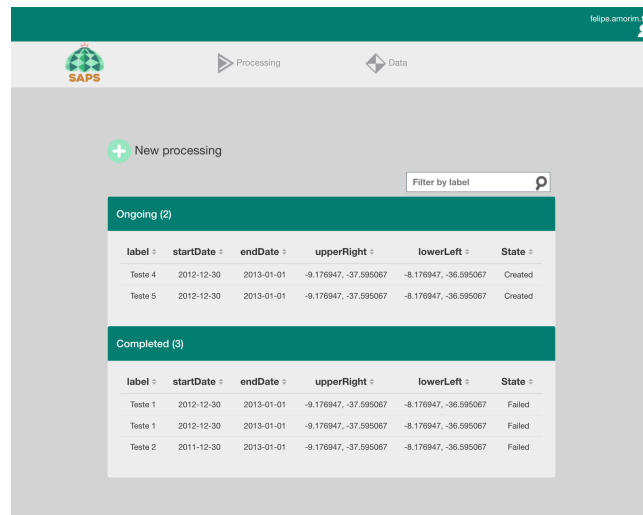


Imagem 2. Interface atual do SAPS

Essa arquitetura permite ao SAPS processar de maneira eficiente e escalável grandes volumes de dados de sensoriamento remoto, oferecendo aos usuários uma solução robusta e flexível para análises baseadas em imagens de satélite. A modularidade inerente à arquitetura também facilita a personalização e a expansão da plataforma para atender a novas necessidades e desafios no campo do sensoriamento remoto.

2.2. Detalhes da interface atual

A interface web do Serviço de Automação do Processamento de Imagens de Satélite (SAPS) ocupa uma posição central na interação do usuário com o sistema. Esta interface, desde o início, proporcionou a execução eficiente das funções principais do sistema, habilitando o processamento de imagens de satélite e facilitando a análise dos resultados.

No entanto, a interface atual enfrenta diversos desafios que necessitam de melhorias. Um dos principais obstáculos está relacionado à acessibilidade. A interface atual não adere aos padrões de acessibilidade estabelecidos, como as diretrizes do Web Content Accessibility Guidelines (WCAG) 2.1 [14]. Estas diretrizes orientam a criação de conteúdos web que sejam acessíveis a todas as pessoas, independentemente de suas habilidades físicas ou sensoriais.

Ademais, a interface atual exibe inconsistência em seu design, o que pode resultar em uma experiência do usuário fragmentada. Os problemas identificados incluem falta de consistência na tipografia e no esquema de cores, presença de elementos gráficos e controles de interface não intuitivos e falta de coerência de design entre os componentes. Essas irregularidades, em conjunto, criam um ambiente digital que pode ser confuso e inacessível para muitos usuários.

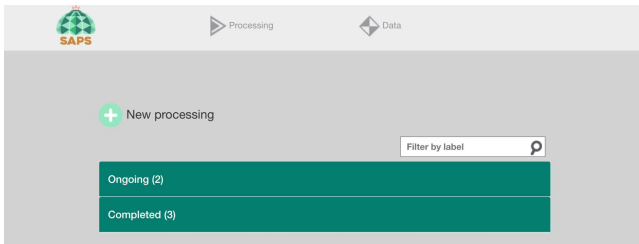


Imagem 3. Inconsistência visual Desktop

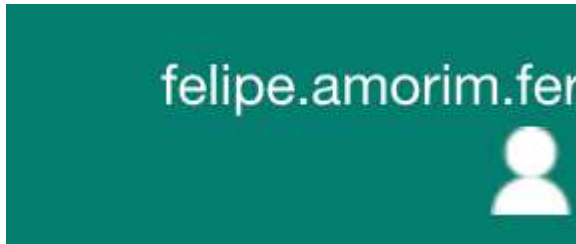


Imagem 4. Inconsistência visual Desktop

A interface web atual também não oferece suporte adequado para diferentes dispositivos, como desktop e mobile. Esta limitação restringe a capacidade dos usuários de acessarem o sistema de forma conveniente, independentemente do dispositivo que estejam utilizando.

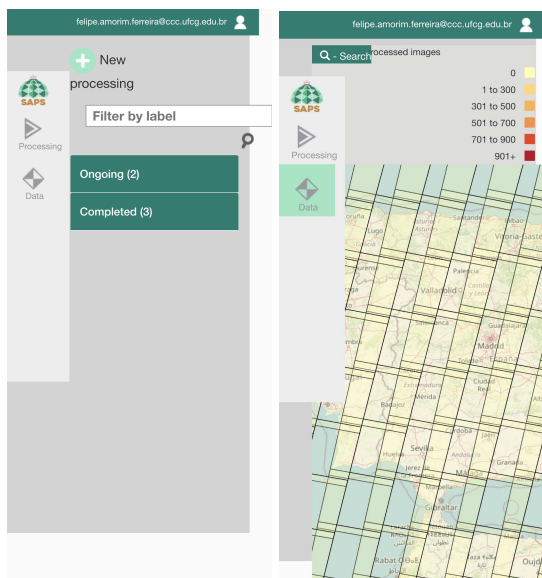


Imagem 5. inconsistência visual mobile

Outro aspecto crítico é a manutenção do sistema. A interface atual utiliza a biblioteca AngularJS, que não está mais sendo mantida. Esta realidade não só complica a manutenção e atualização da interface, como também apresenta obstáculos à integração de novos membros na equipe de desenvolvimento. Como o AngularJS é uma tecnologia antiga, muitos estudantes de graduação não estão familiarizados com ela e, frequentemente, não têm interesse em aprender uma tecnologia considerada obsoleta. Isso torna a manutenção do código existente um desafio ainda maior.

Portanto, apesar de ter sido suficiente no início, a interface atual do SAPS tem enfrentado dificuldades significativas que limitam sua funcionalidade e facilidade de uso. Estas questões enfatizam a necessidade de revisão e atualização da interface, para garantir que ela possa continuar a servir eficazmente aos usuários.

3. REQUISITOS DA SOLUÇÃO

A estratégia para reformular a interface do Serviço de Automação do Processamento de Imagens de Satélite (SAPS) contempla uma série de princípios fundamentais que garantem não apenas uma experiência aprimorada para o usuário final, mas também um sistema mais sustentável e adaptável às necessidades futuras e avanços tecnológicos. As seções que se seguem fornecem um exame detalhado desses princípios orientadores e como eles influenciaram o design e a implementação do novo sistema.

3.1. Acessibilidade

A acessibilidade digital é um componente crucial na concepção de qualquer interface web moderna [1]. A nova interface do SAPS, portanto, será projetada e implementada seguindo os princípios e diretrizes de acessibilidade estabelecidos pelo padrão a11y [13] (um numerônimo para "accessibility").

O objetivo do padrão a11y é tornar a interface web acessível a todos os usuários, independentemente de suas habilidades físicas, visuais, auditivas, cognitivas ou neurológicas. Isso significa que a interface será projetada de forma que até mesmo usuários com deficiências possam interagir com o sistema sem qualquer restrição.

A nova interface adotará algumas estratégias para garantir a conformidade com o padrão a11y. Entre elas estão a navegação por teclado, garantindo que todas as funcionalidades da interface possam ser acessadas usando apenas este dispositivo. Isso é especialmente importante para usuários com deficiências motoras que possam ter dificuldade em usar um mouse.

O contraste de cores será selecionado para garantir adequada legibilidade e usabilidade, beneficiando usuários com deficiências visuais ou daltonismo. A interface será codificada de maneira semântica para assegurar a compatibilidade com leitores de tela, ferramentas indispensáveis para usuários com deficiência visual.

Além disso, as instruções e mensagens de erro serão apresentadas de maneira clara e concisa, facilitando a compreensão dos usuários com deficiências cognitivas. A interface também permitirá o redimensionamento do texto sem que haja perda de funcionalidade ou distorção do layout, beneficiando usuários com dificuldades visuais.

A acessibilidade não é uma consideração opcional, mas um componente essencial na criação de uma web inclusiva. Deste modo, o compromisso com o padrão a11y na nova interface do SAPS é um passo importante em direção à criação de uma ferramenta verdadeiramente acessível e inclusiva.

3.2. Responsividade

A responsividade é uma característica indispensável para qualquer aplicação moderna. Isso porque o uso de dispositivos de vários tamanhos e resoluções, como smartphones, tablets e desktops, tornou-se uma norma no uso cotidiano. Portanto, a nova interface do SAPS foi projetada para ser responsiva, garantindo uma experiência de usuário consistente e de alta qualidade em todos os dispositivos.

Primeiramente, a responsividade permite que a interface seja adaptável. Isso significa que o layout e os elementos da interface se ajustam dinamicamente de acordo com o tamanho da tela do dispositivo do usuário. Ao fazer isso, o SAPS garante que todos os usuários, independentemente do dispositivo que estão usando, possam acessar e utilizar o serviço de maneira eficiente e eficaz.

Em segundo lugar, a responsividade contribui para melhorar a acessibilidade do SAPS. Ao garantir que a interface seja utilizável em vários tamanhos de tela, a nova interface do SAPS pode atender a uma gama mais ampla de usuários, incluindo aqueles que preferem usar dispositivos móveis por motivos de acessibilidade.

Além disso, a responsividade pode melhorar a velocidade e o desempenho da interface. Projetar a interface para ser responsiva pode permitir que o SAPS otimize o carregamento de conteúdo com base no dispositivo do usuário, o que pode levar a tempos de carregamento de página mais rápidos e uma experiência de usuário mais suave.

3.3. Design consistente

Na interface atual do SAPS, um dos pontos de dificuldade observados é a inconsistência no design. Este problema está associado à falta de um padrão unificado que guiaria a criação e colocação de elementos da interface. Este design inconsistente leva a uma experiência de usuário fragmentada, tornando a navegação no sistema confusa e pouco intuitiva.

No desenvolvimento da nova interface, um foco significativo foi colocado na criação de um design consistente. Para alcançar essa consistência, adotamos um sistema de design. Um sistema de design é um conjunto abrangente de componentes reutilizáveis e diretrizes de estilo que são usados em todo o produto. Isso promove a uniformidade em todas as páginas e funções, criando uma sensação de unidade e coerência.

O benefício mais evidente de adotar um sistema de design é uma melhor experiência do usuário. Com um design consistente, os usuários podem aprender rapidamente como a interface funciona e onde encontrar o que precisam. Isso reduz o esforço cognitivo, permitindo que os usuários se concentrem em suas tarefas, em vez de navegar pela interface. Além disso, a consistência cria uma experiência mais profissional e polida, aumentando a confiança dos usuários no sistema.

Para os desenvolvedores, um sistema de design simplifica o processo de desenvolvimento. Com um conjunto de componentes pré-definidos, os desenvolvedores podem construir novas páginas e funções mais rapidamente, pois não precisam criar tudo do zero. Além disso, o sistema de design facilita a manutenção do sistema. Como todos os componentes seguem o mesmo conjunto de diretrizes, é mais fácil identificar e corrigir problemas.

Finalmente, um sistema de design é um excelente recurso para a integração de novos membros da equipe. Ele atua como uma única fonte de verdade para o design do produto, permitindo que novos desenvolvedores se familiarizem rapidamente com o estilo e a estrutura da interface.

3.4. Manutenibilidade

O objetivo da nova interface do SAPS é superar as limitações da interface atual, que incluem não apenas questões de design e acessibilidade, mas também dificuldades de manutenção. Portanto, uma das metas centrais na concepção da nova interface foi aprimorar a manutenibilidade do sistema [3]. Isso foi alcançado por meio da adoção de novas tecnologias e decisões arquiteturais bem fundamentadas.

A primeira grande mudança foi a migração de AngularJS para React. AngularJS, sendo uma tecnologia mais antiga e agora descontinuada, representava um desafio significativo para a manutenção e o aprimoramento do sistema. A equipe decidiu usar React, uma biblioteca JavaScript moderna, amplamente adotada e mantida pelo Facebook. Com uma ampla comunidade de desenvolvedores e recursos robustos, React permite a criação de interfaces ricas e interativas com grande suporte para manutenibilidade.

A estrutura do projeto também foi redesenhada com a manutenibilidade em mente. A nova estrutura do projeto segue os princípios de modularidade e componentização. Cada funcionalidade do sistema é encapsulada em um módulo separado, tornando o código mais fácil de entender, testar e manter. Os componentes são reutilizáveis e independentes, o que minimiza a duplicação de código e facilita as atualizações e correções de bugs.

A adoção de um sistema de design, como discutido na seção anterior, também contribui para a manutenibilidade. Ele padroniza a aparência e o comportamento dos componentes da interface, tornando o código mais consistente e mais fácil de manter.

Finalmente, uma ênfase significativa foi colocada na documentação do projeto. A documentação clara e completa facilita a manutenção ao tornar mais fácil para os desenvolvedores entenderem o código existente e como ele deve ser modificado. Isso também facilita a integração de novos membros da equipe.

3.5. Testabilidade

A testabilidade é uma característica fundamental para o desenvolvimento de softwares robustos e confiáveis. No contexto de aplicações web, testes de unidade e integração são essenciais para garantir a funcionalidade dos componentes e a harmonia de suas interações.

Com a adoção de uma estrutura baseada em componentes para a nova interface do SAPS, o objetivo foi não apenas melhorar a manutenibilidade e a reutilização de código, mas também aprimorar a testabilidade do sistema. Cada componente da interface pode ser testado individualmente, permitindo a identificação e correção de bugs antes que eles causem problemas em um nível maior.

A estratégia de componentização é fortalecida pela adoção da injeção de dependências. Esse padrão de projeto permite que os componentes sejam independentes dos objetos ou módulos de que dependem. Na prática, isso significa que cada componente pode ser testado isoladamente, sem a necessidade de instanciar ou configurar outros componentes ou módulos. Isso simplifica os testes de unidade, já que cada componente pode ser testado em um ambiente controlado e isolado.

Os testes de integração, por outro lado, confirmam que os componentes individuais funcionam corretamente em conjunto. Com a injeção de dependências, é possível substituir facilmente dependências reais por versões fictícias ou simuladas durante os testes. Isso torna possível testar a interação dos componentes em cenários específicos, sem ser afetado pelo comportamento imprevisível ou não determinístico das dependências reais.

Em suma, a testabilidade foi um aspecto fundamental considerado na concepção da nova interface. Com a estrutura de componentes e a injeção de dependências, é possível realizar testes de unidade e integração eficientes, garantindo assim a confiabilidade do sistema e a qualidade da experiência do usuário.

4. IMPLEMENTAÇÃO

A implementação da nova interface web do SAPS é um processo cuidadoso que envolve a seleção de tecnologias adequadas, a adoção de estruturas de projeto eficazes e a realização de testes focados nas partes principais do sistema. Esta seção descreve como cada uma dessas etapas está sendo executada, detalhando as tecnologias e práticas adotadas.



Imagem 7. Tela de busca de resultados. Nova interface correspondente a atual tela apresentada na imagem 5.

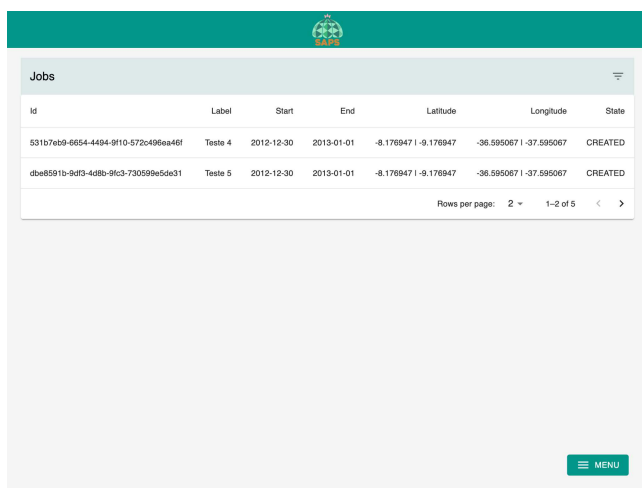


Imagem 6. Tela de listagem de processamentos. Nova interface para a atual tela apresentada na imagem 2.

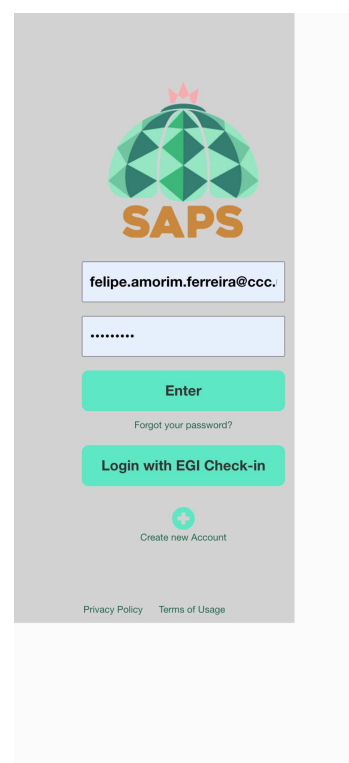


Imagem 8. Atual tela de login.

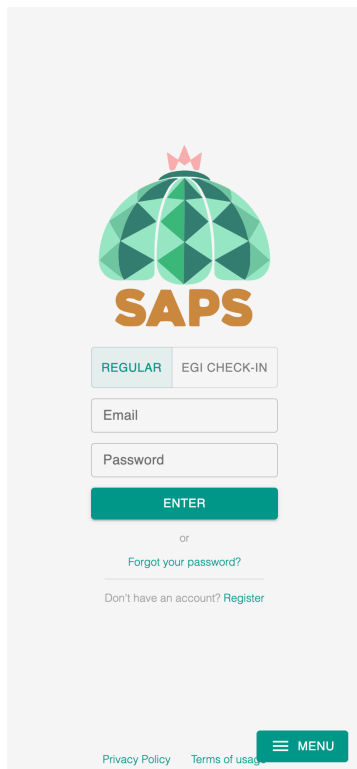


Imagem 9. Nova tela de login. Nova interface correspondente a atual tela apresentada na imagem 8.

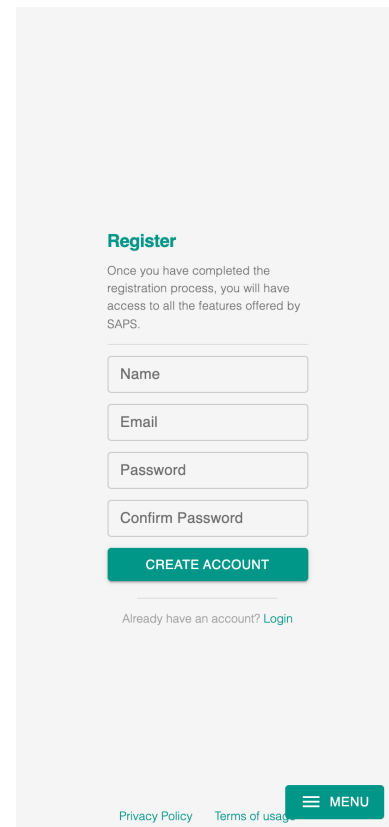


Imagem 11. Nova tela de registro. Nova interface correspondente a atual tela apresentada na imagem 10.

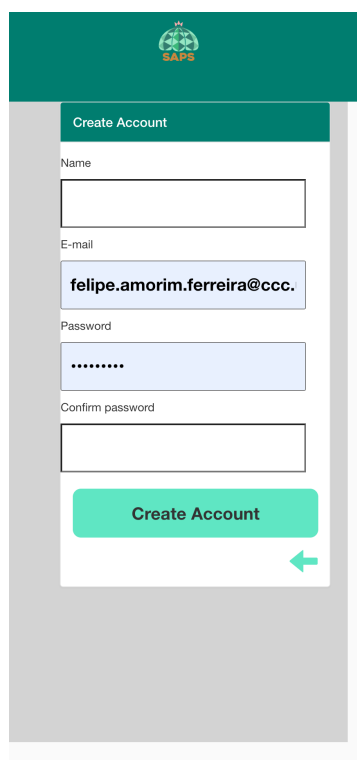


Imagem 10. Atual tela de registro.

4.1. Framework

O Next.js, um popular framework baseado em React, foi escolhido como o principal motor por trás do desenvolvimento da nova interface. Ele foi selecionado por diversas razões chave que se alinham diretamente com os objetivos do projeto.

Primeiramente, o Next.js facilita significativamente o roteamento na aplicação web. Enquanto em uma aplicação React padrão, o roteamento precisa ser configurado manualmente utilizando bibliotecas adicionais como o React Router, o Next.js fornece um sistema de roteamento integrado baseado no sistema de arquivos, que é tanto intuitivo quanto eficiente. Isso simplifica a configuração e manutenção dos caminhos da aplicação, e permite uma organização mais clara do código.

Além disso, o Next.js oferece funcionalidades avançadas que podem melhorar a performance da aplicação. Uma dessas funcionalidades é a renderização do lado servidor (SSR - Server Side Rendering), que melhora o tempo de carregamento inicial das páginas e contribui para a otimização do SEO (Search Engine Optimization), um fator crucial para a visibilidade da plataforma online.

Outra característica importante do Next.js é a geração estática de sites (SSG - Static Site Generation). Esta característica permite gerar páginas HTML em tempo de construção, o que pode aumentar a velocidade de carregamento da página e melhorar a experiência geral do usuário. Essa opção é particularmente útil

para páginas que não requerem atualizações em tempo real e cujos dados podem ser pré-carregados.

Em resumo, a adoção do Next.js representa uma decisão estratégica alinhada com as metas de otimização da performance, eficiência de desenvolvimento e melhor experiência do usuário.

4.2. Dependências

Jest

Jest é um poderoso framework de teste em JavaScript, criado pelo Facebook. Ele é conhecido por sua simplicidade e capacidade de execução rápida de testes. O Jest fornece um ambiente de teste completo e integrado, permitindo testes unitários, testes de integração e testes de snapshot [8].

Dotenv

Dotenv é uma dependência que permite ao projeto acessar variáveis de ambiente a partir de um arquivo .env [9]. Isso é especialmente útil quando se tem chaves e senhas sensíveis que não devem ser incluídas diretamente no código. O Dotenv mantém essas informações sensíveis separadas e seguras.

Styled Components

Styled-components é uma biblioteca JavaScript para estilização CSS de componentes. Ela permite escrever CSS real no JavaScript, mantendo o escopo dos estilos no componente em que foram definidos. Styled-components ajuda a manter o design consistente e a reutilização de componentes no projeto [10].

TypeScript

TypeScript é uma linguagem de programação desenvolvida e mantida pela Microsoft. É um superconjunto estrito de JavaScript, adicionando tipagem estática e objetos orientados a classes. TypeScript oferece vantagens significativas em termos de autocompletar código, validação em tempo de compilação e uma melhor legibilidade do código [11].

React Query

React-query é uma biblioteca de manipulação de dados assíncronos. Ela permite ao projeto buscar, sincronizar e atualizar dados do servidor sem precisar gerenciar todo o complexo estado de caching, sincronização e atualizações de dados manualmente. Isso ajuda a simplificar o código e torna a experiência do usuário mais fluida [12].

ally

ally é uma abreviação de "acessibilidade". No contexto de desenvolvimento web, isso geralmente se refere às práticas e ferramentas para tornar os sites e aplicativos acessíveis a todos os usuários, incluindo aqueles com deficiências [13]. No projeto, ally pode se referir à adoção de práticas de acessibilidade ou ao uso de ferramentas específicas para testar e melhorar a acessibilidade.

4.3. Estrutura de Projeto

A estrutura do projeto adota uma divisão bem definida com base nas responsabilidades, o que facilita a manutenibilidade e

compreensão do código [3][4]. A implementação foi organizada em dois módulos principais: um módulo dedicado à obtenção e tratamento dos dados e outro módulo responsável pela manipulação desses dados para apresentação ao usuário.

A estrutura de diretórios foi aprimorada a partir do padrão fornecido pelo Next.js, com novos diretórios adicionados ao diretório "src" para acomodar as necessidades específicas do projeto. Estes diretórios incluem:

components: Este diretório contém todos os componentes da aplicação. Os diretórios são nomeados de acordo com o nome do componente e cada um deles pode conter três arquivos principais: index.ts que contém a lógica do componente, style.ts que inclui os estilos específicos do componente e useHandler.ts que é um Custom hook que expõe os estados e métodos utilizados pelo componente.

services: Esse diretório aloja o fetcher, a funcionalidade responsável por requisitar e tratar os dados da API, promovendo uma separação clara entre a lógica de obtenção de dados e a lógica de apresentação de dados.

styles: Este diretório é usado para armazenar valores de estilo que são globais à aplicação.

types: Aqui estão contidas todas as declarações de tipo usadas em toda a aplicação, proporcionando um lugar centralizado para a gestão dos tipos.

A adoção dessa estrutura de diretórios facilita a orientação dentro do projeto, simplifica a manutenção do código e facilita a integração de novos membros na equipe.

4.4. Integração

Na construção do novo sistema web do SAPS, um dos aspectos críticos é a integração eficiente com a API existente. Este sistema é projetado para comunicar-se de maneira robusta e segura com os endpoints da API, o que permite uma interação fluida e dinâmica entre o usuário e a plataforma.

Três principais endpoints da API são usados para esta integração: "jobs", "tasks" e o sistema de autenticação. A interação com esses endpoints é feita através de solicitações HTTP, que permitem o acesso e a manipulação dos dados necessários para o funcionamento do sistema.

O endpoint "jobs" é utilizado para obter detalhes sobre os trabalhos de processamento de imagens de satélite. Isto inclui informações sobre o estado do trabalho, os resultados gerados e outros detalhes pertinentes.

O endpoint "tasks" oferece dados sobre as tarefas específicas associadas a cada trabalho. Esta informação é essencial para monitorar o progresso dos trabalhos e identificar qualquer problema que possa surgir durante o processamento.

Por fim, o sistema de autenticação é um componente vital que garante a segurança da plataforma. As solicitações feitas a este endpoint permitem o gerenciamento do acesso dos usuários, garantindo que apenas usuários autorizados possam acessar determinadas funções e dados.

4.5. Testes

A eficácia de qualquer plataforma web é comprovada, em grande parte, através da sua capacidade de funcionar conforme o esperado em diversos cenários de uso. Para assegurar isso, é fundamental realizar testes abrangentes e confiáveis. Na nova plataforma web do SAPS, a estratégia de teste foi estruturada para enfatizar a consistência da integração entre a interface e a API, bem como a funcionalidade dos componentes internos da plataforma.

Primeiramente, os testes de integração foram utilizados para verificar a interação correta entre a nova interface web e a API já existente. Esses testes foram centrados nos endpoints críticos para o funcionamento da plataforma, como "jobs", "tasks" e o sistema de autenticação. Ao simular solicitações para esses endpoints, foi possível assegurar que a comunicação entre a interface e a API estava operando como esperado.

Além dos testes de integração, os testes de unidade também foram implementados para avaliar componentes específicos que realizam processamento de dados de maneira independente. Isso inclui componentes que foram criados pelos desenvolvedores do projeto, sem a ajuda de bibliotecas externas ou dependências. Estes testes foram projetados para verificar se cada unidade de código estava realizando sua função corretamente, garantindo a qualidade do código e a confiabilidade do sistema.

Com essa abordagem de teste, a nova plataforma web do SAPS busca garantir a sua robustez e confiabilidade, proporcionando uma experiência de usuário consistente e segura, ao mesmo tempo que mantém a capacidade de evoluir e se adaptar a futuras necessidades e melhorias.

5. AGRADECIMENTOS

Minha gratidão a Deus, primeiro e principal alicerce da minha vida, que sempre tem me conduzido por caminhos repletos de desafios instigantes que estimulam meu crescimento contínuo. Aos meus amados Isabel, Marcelo (pai), Marcelo (irmão) e Isabela, meu agradecimento por serem os faróis que me orientam, a base do meu apoio e minha inspiração em todas as decisões e etapas do curso.

A jornada não teria sido a mesma sem a camaradagem e apoio de amigos e colegas de curso, Sobral, Arthur, Espindula, Bruno, Dayvid e João Soares. Nossos caminhos cruzados se tornaram um entrelaçamento de ajuda mútua e superação conjunta de obstáculos, a qual eu sou profundamente grato.

Nesta jornada, tive a honra de ser orientado por Thiago Emmanuel, cuja excelência e dedicação como professor e orientador fomentaram o desenvolvimento e realização deste projeto. Sua influência foi inestimável para o sucesso deste trabalho.

Por fim, agradeço a Cilas, cuja amizade e prestatividade foram cruciais para o meu crescimento acadêmico e para a realização deste projeto.

Cada um de vocês contribuiu de maneira significativa para minha trajetória e este trabalho.

6. REFERÊNCIAS

- [1] Jonathan Lazar, Alfred Dudley-Sponaugle, and Kisha-Dawn Greenidge. 2004. Improving web accessibility: a study of webmaster perceptions. In *Computers in Human Behavior*, Vol. 20. 269-288. Disponível em: <https://doi.org/10.1016/j.chb.2003.10.018>
- [2] Cunha, J., Pereira, T.E., Pereira, E., Rufino, I., Galvão, C., Valente, F., and Brasileiro, F. 2019. A high-throughput shared service to estimate evapotranspiration using Landsat imagery. *Computers & Geosciences*. Disponível em: <https://doi.org/10.1016/j.cageo.2019.104341>
- [3] Kaur, A. A Systematic Literature Review on Empirical Analysis of the Relationship Between Code Smells and Software Quality Attributes. *Arch Computat Methods Eng* 27, 1267–1296 (2020). Disponível em: <https://doi.org/10.1007/s11831-019-09348-6>
- [4] Taibi, D., Janes, A., Lenarduzzi, V. (2017). How Developers Perceive Smells in Source Code: A Replicated Study. *Information and Software Technology*, 92, 223-235. Disponível em: <https://doi.org/10.1016/j.infsof.2017.08.008>
- [5] Evapotranspiration - an overview. Disponível em: <https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/evapotranspiration>
- [6] ReactJS. (s.d.). Uma biblioteca JavaScript para criar interfaces de usuário. Recuperado de: <https://pt-br.reactjs.org/>
- [7] Next.js. (s.d.). The React Framework. Recuperado de: <https://nextjs.org/>
- [8] Jest. (s.d.). Delightful JavaScript Testing. Recuperado de: <https://jestjs.io/>
- [9] Dotenv. (s.d.). Module that loads environment variables from a .env file into process.env. Recuperado de: <https://www.npmjs.com/package/dotenv>
- [10] styled-components. (s.d.). Visual primitives for the component age. Recuperado de: <https://styled-components.com/>
- [11] TypeScript. (s.d.). JavaScript that scales. Recuperado de: <https://www.typescriptlang.org/>
- [12] react-query. (s.d.). Fetch, cache and update data in your React and React Native applications all without touching any "global state". Recuperado de: <https://react-query.tanstack.com/>
- [13] a11y. (s.d.). Web Accessibility. Recuperado de: <https://www.a11yproject.com/>
- [14] World Wide Web Consortium (W3C). 2018. Web Content Accessibility Guidelines (WCAG) 2.1. Disponível em: <https://www.w3.org/TR/WCAG21>