

Modernização e Reestilização da Interface do Usuário de um Sistema para SEAPA: Uma Abordagem Centrada no Usuário

Victor Samuel Levindo Mont’Mor

Orientador: Ismael Santana Silva

Coorientador: Edson Marchetti da Silva

2 de julho de 2026

Resumo

Este trabalho teve como objetivo refatorar a arquitetura e a interface do *front-end* do Sistema de Gestão de Editais (SIGET), utilizado pela Secretaria de Estado de Agricultura, Pecuária e Abastecimento de Minas Gerais (SEAPA), aplicando os princípios do Design Centrado no Usuário. Para tanto, adotou-se uma estratégia de refatoração contínua e in loco, realizada diretamente sobre o repositório principal em produção, de modo a mitigar a dívida técnica acumulada sem interromper a operação do sistema. A intervenção envolveu a implementação de uma arquitetura modular baseada em componentes, utilizando React, Tailwind CSS e Shadcn UI, bem como a definição de um *Design System* voltado à padronização visual e à acessibilidade. Complementarmente, os fluxos de interação foram reestruturados por meio de formulários declarativos com React Hook Form e Zod, centralizando as regras de validação e fornecendo feedback em tempo real aos usuários. A avaliação da solução foi conduzida por meio de inspeções heurísticas com especialistas e de testes empíricos de usabilidade com usuários finais, utilizando o questionário *System Usability Scale* (SUS). Os resultados indicam que 67% dos módulos do sistema foram migrados para a nova arquitetura, com eliminação de conflitos de estilo em cascata e redução expressiva da complexidade ciclomática dos arquivos de página. Na avaliação com especialistas, a severidade dos problemas de usabilidade caiu de níveis críticos (graus 3–4) para cosméticos (graus 0–1), e 100% dos participantes da pesquisa SUS declararam preferência pela nova interface para uso cotidiano. As contribuições deste trabalho evidenciam a viabilidade da evolução incremental de sistemas governamentais em operação, demonstrando que a refatoração progressiva constitui uma alternativa eficaz para a modernização de aplicações legadas em ambientes institucionais ativos.

Palavras-chaves: Interação Humano-Computador. Refatoração. Dívida Técnica. React. Design System.

1 Introdução

A regularização fundiária é um processo essencial para a justiça social e o desenvolvimento rural. No Estado de Minas Gerais, a Secretaria de Estado de Agricultura, Pecuária e Abastecimento (SEAPA) é responsável por gerenciar as demandas, credenciamentos e adesões de municípios e empresas a programas governamentais da área. Conforme documentado no plano de trabalho do projeto de extensão entre a SEAPA e o CEFET-MG [Silva e Marchetti \(2024\)](#), a gestão desses processos enfrentava gargalos operacionais significativos decorrentes do uso intensivo de ferramentas manuais, como planilhas e formulários impressos. A dependência de processos manuais em atividades críticas compromete a confiabilidade dos dados e limita a capacidade de gestão eficiente [Sommerville \(2011\)](#), além de dificultar o controle dos editais e a integração com sistemas oficiais do estado.

Em resposta à demanda por modernização e otimização desses fluxos, estabeleceu-se uma parceria entre a SEAPA e o Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG), materializada em um projeto de extensão [Silva e Marchetti \(2024\)](#). O objetivo desse projeto foi a concepção e o desenvolvimento do Sistema de Gestão de Editais (SIGET), visando informatizar todo o ciclo de vida dos editais, desde a publicação até o encerramento. A relevância e a complexidade arquitetural do SIGET decorrem da diversidade de atores institucionais que compõem o seu ecossistema de usuários finais [Lopes \(2024\)](#). Esses perfis englobam:

- *Servidores da SEAPA-MG*: analistas estaduais responsáveis pela publicação de editais, revisão de documentos de regularização fundiária e condução interna das titulações;
- *Representantes de Prefeituras*: agentes municipais encarregados de formalizar a adesão de seus municípios aos editais do Estado por meio do preenchimento de formulários densos;
- *Técnicos da EMATER e Empresas Credenciadas*: operadores externos de campo (frequentemente em zonas rurais com baixa infraestrutura de conectividade) que submetem laudos técnicos, cadastros de beneficiários e dados de georreferenciamento.

A viabilização técnica para integrar essa malha de usuários foi impulsionada pelo trabalho de [Lopes \(2024\)](#), que desenvolveu a camada de integração de *back-end* do SIGET com o Sistema Eletrônico de Informações (SEI-MG) por meio de um *gateway* REST. Essa base automatizou os trâmites processuais nos bastidores e estabeleceu a arquitetura sobre a qual o *front-end* — escopo do presente trabalho — pôde ser posteriormente estruturado.

A primeira implementação do SIGET cumpriu seu papel primordial de validar os requisitos e entregar a funcionalidade básica para a Secretaria. Contudo, por ter sido desenvolvida progressivamente por uma equipe rotativa de voluntários com níveis técnicos distintos, o sistema em constante evolução acumulou uma dívida técnica. A interface, embora funcional, foi construída de forma utilitária, sem a adoção de padrões de usabilidade [Nielsen \(1993\)](#), consistência visual ou acessibilidade, o que, na avaliação do autor deste trabalho, impunha uma carga cognitiva aos servidores que operavam o sistema diariamente. Nesse cenário, o autor do presente trabalho ingressou no projeto de extensão em outubro de 2025, com o propósito específico de conduzir a intervenção de modernização do front-end do SIGET como trabalho de conclusão de curso, permanecendo vinculado ao projeto pelo período de um ano.

Diante desse contexto, o objetivo deste trabalho foi propor e implementar uma reestilização completa e a reestruturação arquitetural do *front-end* do sistema SIGET da SEAPA, aplicando os princípios do Design Centrado no Usuário (DCU) [International Organization for Standardization \(2019\)](#), por meio de uma refatoração progressiva no repositório principal ativo da aplicação. Tal objetivo implica não apenas a substituição de componentes visuais isolados, mas a reestruturação sistêmica da camada de apresentação, contemplando a adoção de um *Design System* [Suarez et al. \(2019\)](#) coeso, a padronização dos fluxos de interação e a introdução de mecanismos de validação que forneçam feedback contínuo ao operador. Dessa forma, busca-se demonstrar a viabilidade de refatorar um sistema governamental em operação sem a necessidade de interromper suas atividades ou comprometer a integridade dos dados em produção.

Para o alcance do objetivo geral, definiram-se os seguintes objetivos específicos:

- Mitigar a dívida técnica do *front-end* refatorando o código in loco e implementando uma arquitetura modular baseada em componentes (em substituição à abordagem *greenfield* prévia).
- Integrar um *Design System* sustentado por tecnologias atualizadas (React, Tailwind CSS, Shadcn UI) para garantir consistência visual e acessibilidade em todas as rotas do sistema.
- Substituir o modelo imperativo de formulários por validações declarativas que proveem feedback contínuo e instantâneo ao operador.
- Avaliar empiricamente as melhorias de usabilidade e qualidade técnica por meio de uma abordagem multi-método (inspeção heurística, métricas de código e percepção de usuários).

A modernização progressiva do SIGET justifica-se pela otimização direta do tempo e do esforço empregado pelos servidores e parceiros da SEAPA na operação de fluxos de trabalho essenciais. Uma interface pautada pelas heurísticas de usabilidade diminui substancialmente o treinamento necessário para a adoção do sistema e previne erros operacionais críticos [Nielsen \(1993\)](#).

Sob a ótica da Engenharia de Software, a refatoração contínua de um sistema em produção documenta e explora um desafio clássico da indústria de tecnologia: o gerenciamento de dívida técnica em sistemas ativos [Fowler \(2018\)](#). A intervenção direta na *codebase* principal resgata a manutenibilidade do código real da aplicação e valida um fluxo de trabalho realista. O escopo deste trabalho foca-se estritamente na camada de apresentação (*front-end*), demonstrando como é possível elevar o padrão de qualidade de uma aplicação em andamento sem paralisar seu desenvolvimento ou exigir a paralisação da equipe (*feature freeze*).

Para atingir os objetivos propostos, adotou-se uma estratégia de refatoração contínua e in loco, atuando diretamente no repositório principal em produção do SIGET. A intervenção fundamentou-se na implantação de um *Design System* modular sustentado por React, Tailwind CSS e Shadcn UI, na substituição de formulários imperativos por validações declarativas com React Hook Form e Zod, e na migração gradativa dos módulos antigos sem interromper as entregas do sistema. A avaliação foi conduzida por meio de inspeções heurísticas com especialistas, análise de métricas de código e aplicação do questionário *System Usability Scale* (SUS) com usuários finais. Os resultados indicam que

67% dos módulos foram migrados para a nova arquitetura, a severidade dos problemas de usabilidade reduziu de níveis críticos para cosméticos na avaliação dos especialistas, e 100% dos participantes da pesquisa SUS declararam preferência pela nova interface para uso cotidiano. As principais contribuições residem na validação empírica de uma metodologia de refatoração progressiva aplicável a sistemas governamentais em operação, e na entrega à SEAPA de uma ferramenta com interface padronizada, acessível e preparada para escalar tecnicamente.

O presente documento está organizado em sete seções. A seção 2 revisa a fundamentação teórica necessária, abordando conceitos de Interação Humano-Computador, usabilidade, dívida técnica e arquitetura baseada em componentes. A seção 3 apresenta os trabalhos relacionados, situando a presente proposta em relação à literatura de *redesign* de interfaces. A seção 4 detalha o diagnóstico do sistema antigo do SIGET e define a metodologia de intervenção adotada. Na seção 5, discute-se o processo de desenvolvimento e implementação prática, expondo as decisões tomadas durante a refatoração do repositório principal. A seção 6 apresenta a metodologia de avaliação adotada e analisa detalhadamente os resultados quantitativos e qualitativos, validando o impacto do projeto. Por fim, a seção 7 tece as considerações finais, ressaltando as contribuições obtidas e apontando direções para trabalhos futuros.

2 Referencial Teórico

Esta seção apresenta a base conceitual que sustenta as decisões técnicas e metodológicas tomadas durante a modernização do sistema SIGET. São abordados temas transversais que unem o design de interfaces e a engenharia de software, partindo dos fundamentos da usabilidade, transitando pelos desafios de arquitetura e dívida técnica, e culminando na descrição das tecnologias escolhidas para viabilizar a refatoração.

2.1 Interação Humano-Computador e Usabilidade

A disciplina de Interação Humano-Computador (IHC) dedica-se ao design, avaliação e implementação de sistemas computacionais interativos para o uso humano, focando em como os indivíduos interagem com a tecnologia [Hewett et al. \(1992\)](#). No contexto de sistemas governamentais complexos, a IHC é fundamental para garantir que o software não seja apenas funcional, mas também operável de maneira eficiente.

O conceito de *Usabilidade* é um dos pilares da IHC. Segundo a norma [International Organization for Standardization \(2018\)](#), a usabilidade é a capacidade de um sistema ser utilizado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um determinado contexto de uso. Para balizar o design de interfaces e facilitar sua avaliação, Jakob Nielsen e Robert Molich propuseram as 10 Heurísticas de Usabilidade [Nielsen \(1993\)](#), princípios gerais que orientam o desenvolvimento centrado no usuário. Dentre essas heurísticas, destacam-se no contexto deste trabalho:

- *Visibilidade do status do sistema*: o usuário deve sempre saber o que está acontecendo por meio de feedback adequado e em tempo hábil.
- *Prevenção de erros*: mais valioso do que boas mensagens de erro é um design cuidadoso que impeça o problema de ocorrer.

- *Consistência e padrões*: os usuários não devem ter que se perguntar se palavras, situações ou ações diferentes significam a mesma coisa.
- *Estética e design minimalista*: diálogos não devem conter informações irrelevantes ou raramente necessárias, pois cada informação extra compete com as unidades relevantes de informação e diminui sua visibilidade relativa.

Sistemas que gerenciam densas camadas de dados, como o fluxo de adesões do SIGET, dependem intrinsecamente dessas heurísticas para mitigar falhas cognitivas dos operadores que lidam com dezenas de processos diariamente. Nesse sentido, [Garrett \(2011\)](#) propõe um modelo de experiência do usuário estruturado em camadas, desde a estratégia até a superfície da interface, amplamente utilizado como base para projetos de melhoria de usabilidade. Complementarmente, [Norman \(2013\)](#) destaca a importância de conceitos como *affordances*, feedback e mapeamento na criação de sistemas mais intuitivos e alinhados ao comportamento humano.

2.2 Engenharia de Software e Dívida Técnica

Sob a óptica da Engenharia de Software, o desenvolvimento acelerado ou sem padronização cobra um preço estrutural a longo prazo. O termo *Dívida Técnica*, cunhado por [Cunningham \(1992\)](#), descreve o custo implícito de retrabalho causado pela escolha de uma solução de implementação fácil ou rápida no presente em detrimento de uma abordagem estruturada que levaria mais tempo.

Conforme [Fowler \(2018\)](#) detalha, à medida que a dívida técnica acumula, o código torna-se frágil; pequenos ajustes em um módulo provocam falhas imprevisíveis em componentes não relacionados, criando uma barreira para a evolução do sistema. Esse fenômeno é comumente exemplificado no *front-end* com o “*CSS Bleeding*”, quando regras de estilo globais vazam e afetam páginas inesperadas.

Ao confrontar um sistema com alta dívida técnica, as equipes de desenvolvimento frequentemente enfrentam o dilema entre refatorar o código existente progressivamente ou abandoná-lo para construir uma nova versão a partir do zero (abordagem *Greenfield*). Embora a ideia de “começar do zero” pareça sedutora por oferecer uma arquitetura supostamente limpa, trata-se de uma estratégia de alto risco. Em sistemas ativos e de alta complexidade, a reescrita sofre com a síndrome do “alvo em movimento” (*moving target*). Enquanto a nova versão é desenvolvida, a versão legada em produção continua recebendo correções de bugs, novas regras de negócios (frequentemente não documentadas) e atualizações de API. Manter as duas bases sincronizadas é custoso e falho. Portanto, a literatura moderna de refatoração, defendida por [Fowler \(2018\)](#), privilegia a transformação progressiva e *in loco*: substitui-se os componentes antigos gradualmente em produção, garantindo que o software permaneça funcional e evitando a perda de conhecimento intrínseco já codificado. Essa foi exatamente a transição metodológica adotada entre as fases 1 e 2 deste projeto.

2.3 Arquitetura Baseada em Componentes e *Clean Architecture*

Para sanar os problemas estruturais do *front-end*, recorreu-se a princípios de arquitetura de software limpa adaptados para a camada visual. [Martin \(2018\)](#), em “*Clean Architecture*”, postula que sistemas devem aplicar rigorosamente o Princípio da Responsabilidade Única (SRP) — segundo o qual cada módulo deve ter uma, e apenas uma,

razão para mudar — e o Princípio da Inversão de Dependência (DIP), que estabelece que módulos de alto nível não devem depender de módulos de baixo nível, mas ambos devem depender de abstrações.

No desenvolvimento web moderno, esses princípios materializam-se na *Arquitetura Baseada em Componentes*. A interface é fracionada em blocos isolados, coesos e reutilizáveis. Um componente de Tabela de Dados, por exemplo, não deve ser responsável por buscar os dados no servidor, nem por estipular como um botão específico é renderizado, mas apenas por orquestrar a exibição das linhas e colunas. Essa separação de interesses (*Separation of Concerns*) Martin (2018) garante que a lógica de negócio e os modelos de dados não fiquem excessivamente acoplados às regras de CSS ou *tags* HTML primitivas, viabilizando testes isolados e manutenções seguras.

2.4 Tecnologias e Ferramentas Empregadas

Para viabilizar a modernização do SIGET com base nos preceitos teóricos apresentados, selecionou-se um ecossistema de tecnologias maduras e interoperáveis, descritas a seguir.

O React¹ é uma biblioteca JavaScript para a construção de interfaces de usuário reativas. Diferente de abordagens imperativas de manipulação direta do *Document Object Model* (DOM), o React utiliza um DOM Virtual e uma abordagem declarativa: o desenvolvedor descreve o estado desejado da interface, e a biblioteca encarrega-se de atualizar e renderizar eficientemente apenas os componentes cujos dados mudaram. A escolha do React justifica-se pela sua vasta adoção pelo mercado, pela robusta comunidade de suporte e, sobretudo, pela sua capacidade inata de promover a Arquitetura Baseada em Componentes, característica essencial para estancar a fragmentação de interfaces e a dívida técnica diagnosticadas na versão legada do SIGET.

Para a estilização, adotou-se o Tailwind CSS², um *framework* CSS com abordagem *utility-first* — isto é, uma filosofia de estilização que substitui classes CSS semânticas e globais por classes atômicas de propósito único —, para criar o design do sistema juntando várias classes CSS prontas e de função única diretamente na marcação do código. Em vez de depender de grandes arquivos globais de estilo com classes semânticas (que frequentemente causam o citado *CSS Bleeding*, descrito na Seção 2.2), o Tailwind fornece classes de baixo nível rigorosamente tipadas (ex: `flex`, `pt-4`, `text-center`) que são compostas diretamente na marcação do componente. A adoção desta tecnologia foi motivada pela necessidade crítica de eliminar o acoplamento global herdado da primeira versão do SIGET. Ao confinar o estilo à camada do próprio componente, o Tailwind permite iterações ágeis de leiaute e garante que a modernização visual de uma página não quebre acidentalmente a renderização visual das rotas adjacentes.

Como biblioteca de componentes, utilizou-se o Shadcn UI³, que, diferente de bibliotecas tradicionais de componentes monolíticos importados via NPM⁴ (como Bootstrap ou Material-UI), é uma coleção de componentes modulares copiados diretamente para o repositório do projeto. Eles combinam a estilização previsível do Tailwind CSS com a

¹ <<https://react.dev/>>

² <<https://tailwindcss.com/>>

³ <<https://ui.shadcn.com/>>

⁴ NPM (*Node Package Manager*): gerenciador de pacotes padrão do ecossistema JavaScript, utilizado para instalar e gerenciar dependências de projetos.

infraestrutura lógica do *Radix UI*⁵, este último responsável por lidar com todas as complexidades nativas de acessibilidade web. A escolha do Shadcn, que representou a maior mudança de paradigma arquitetural do SIGET, justifica-se particularmente pelo seu modelo de interface autônoma (*Headless UI*). Ao injetar o código-fonte dos componentes para dentro do repositório em vez de engessá-lo em uma dependência inatingível (`node_modules`), assegurou-se à equipe de tecnologia do Estado o controle absoluto para adaptar a estética visual à marca corporativa da instituição e para implementar regras de negócios atípicas exigidas pelo setor público, restrições estas frequentemente intransponíveis em *frameworks* rígidos. Além disso, a acessibilidade nativa embutida atende instantaneamente às rigorosas conformidades governamentais de navegação por teclado e compatibilidade com leitores de tela.

Por fim, para o gerenciamento e a validação de formulários densos, empregou-se o *React Hook Form*⁶ em conjunto com o *Zod*⁷. O React Hook Form isola o estado das entradas (inputs) do ciclo de renderização principal do React, elevando de forma expressiva a performance em páginas com dezenas de campos. O Zod, por sua vez, atua como uma ferramenta de validação de *schema* rigorosa. O emprego conjunto dessas tecnologias foi uma decisão estratégica para transferir a severa carga de validação, antes dependente exclusivamente de requisições dispendiosas ao *back-end*, para o próprio navegador do usuário. Essa mudança na arquitetura do formulário alivia a infraestrutura do servidor da SEAPA em períodos de alto pico de acessos (como a abertura de grandes editais) e transforma o nível do feedback heurístico do operador, que agora é corrigido em tempo real, mitigando envios errôneos de dados sensíveis à base governamental.

3 Trabalhos Relacionados

Esta seção apresenta trabalhos recentes que abordam o *redesign* e a reestruturação de interfaces de sistemas existentes, permitindo situar o presente trabalho em relação à literatura e evidenciar suas contribuições específicas.

O trabalho de Oliveira, Santos e Marques (2024) apresenta o *redesign* da ferramenta USINN Modeler, uma aplicação web voltada à modelagem de interação e navegação. Os autores conduziram uma avaliação heurística da versão original, identificaram problemas de usabilidade e acessibilidade e propuseram uma nova interface baseada em princípios de design centrado no usuário. A validação foi realizada por meio de testes com usuários, demonstrando melhorias na eficácia e na satisfação. No entanto, a intervenção focou-se em uma ferramenta de escopo acadêmico e com baixa complexidade de regras de negócio, sem enfrentar os desafios típicos de sistemas governamentais em operação contínua, como a necessidade de manter compatibilidade com fluxos burocráticos em andamento.

Em uma linha complementar, Borges e Araújo (2024) relatam as experiências e os desafios de um processo de *redesign* conduzido com o suporte de um assistente de inteligência artificial em uma plataforma educacional. O estudo documenta limitações técnicas enfrentadas durante a reestruturação, como a dificuldade de manter a consistência visual entre módulos e os entraves organizacionais decorrentes da resistência à mudança. Embora o trabalho contribua com reflexões relevantes sobre o processo de modernização, a avaliação concentra-se na perspectiva do desenvolvedor e não inclui validações empíricas

⁵ <<https://www.radix-ui.com/>>

⁶ <<https://react-hook-form.com/>>

⁷ <<https://zod.dev/>>

com os usuários finais da plataforma.

O presente trabalho diferencia-se dos mencionados em três aspectos. Primeiro, opera sobre um sistema governamental real e em produção (SIGET/SEAPA), enfrentando diretamente os desafios de refatoração in loco em um repositório ativo com evolução contínua de regras de negócio, cenário não abordado pelos trabalhos anteriores. Segundo, completa o ciclo de desenvolvimento que vai desde o diagnóstico da dívida técnica até a implementação concreta em código (React, Tailwind CSS, Shadcn UI), enquanto trabalhos como o de Oliveira, Santos e Marques (2024) concentram-se na fase de prototipagem e validação. Terceiro, adota uma abordagem de avaliação multi-método que combina inspeção heurística com especialistas e pesquisa de percepção com usuários finais (SUS), fornecendo evidências tanto qualitativas quanto quantitativas sobre o impacto da modernização.

4 Metodologia

Esta seção detalha o percurso metodológico adotado para a reestruturação do sistema SIGET. A pesquisa caracterizou-se como um estudo de intervenção técnica, estruturada em quatro fases principais: diagnóstico inicial, definição da estratégia de melhorias, desenvolvimento da proposta de modernização e, por fim, avaliação da solução. A seguir, detalha-se cada uma dessas etapas.

4.1 Fase 1: Diagnóstico do Sistema Antigo

A primeira versão do SIGET foi desenvolvida com foco estrito na entrega de funcionalidades, sem o estabelecimento de um sistema de design robusto ou padrões consistentes de codificação de interface. O resultado foi o acúmulo de uma dívida técnica substancial, manifestada tanto no código-fonte quanto na experiência do usuário.

Sob a ótica do código, a aplicação sofria de um forte acoplamento estrutural. O estilo visual dependia de arquivos CSS globais, frequentemente gerando o fenômeno de *CSS bleeding*, onde a alteração da cor de um botão em uma página causava efeitos colaterais imprevisíveis em componentes de outras rotas. Estruturas primitivas de HTML, como `<table>`, `<thead>` e `<tbody>`, eram espalhadas repetitivamente pelos arquivos, tornando a manutenção e a inserção de novos recursos excessivamente trabalhosas e violando a separação de interesses.

No que tange à usabilidade, o sistema apresentava deficiências graves nas *affordances* e no feedback visual. Formulários críticos, como a aprovação de processos ou o controle de editais, apoiavam-se em elementos nativos do navegador que não ofereciam clareza de uso. O uso de `radio buttons` diminutos para ações de aprovação ou rejeição de processos com alta responsabilidade aumentava a probabilidade de erros por cliques acidentais (*misclicks*). Além disso, a validação de formulários era majoritariamente processada e validada apenas após a submissão, frustrando os operadores que só eram alertados sobre inconsistências quando toda a página precisava ser recarregada.

4.2 Fase 2: Definição de Melhorias e Estratégia de Intervenção

A definição das melhorias a serem implementadas fundamentou-se em uma abordagem analítica e empírica, prescindindo de entrevistas formais iniciais com os usuários finais devido à natureza técnica do problema radicular. O escopo da intervenção foi delimitado a partir de três vetores principais: (1) uma inspeção heurística preliminar conduzida

pelo autor, que mapeou violações evidentes de usabilidade e acessibilidade na interface legada; (2) a análise quantitativa e estrutural da dívida técnica acumulada, focada em sanar a fragilidade do código de apresentação; e (3) o alinhamento contínuo com os gestores da SEAPA, absorvendo as limitações reportadas rotineiramente pela equipe de campo durante a fase de operação piloto do sistema.

Diante do diagnóstico, a metodologia testada na fase 1 deste projeto foi adotar a abordagem *greenfield*. Criou-se um projeto inteiramente novo, visando reimplementar a interface em um repositório apartado. No entanto, conforme apontado pela literatura [Fowler \(2018\)](#), a reescrita total de um sistema ativo esbarra na síndrome do “alvo em movimento” (*moving target*). Enquanto a nova interface era projetada isoladamente, a equipe do SIGET continuou evoluindo o sistema em produção, alterando endpoints (pontos de acesso remotos expostos pela API do servidor para comunicação com o front-end) e regras de negócio essenciais.

Sincronizar o projeto *greenfield* com essas mudanças orgânicas revelou-se um esforço inviável. Essa experiência empírica forneceu a principal diretriz da Fase 2: exigir a paralisação do desenvolvimento de funcionalidades (*feature freeze*) de uma ferramenta vital de gestão não seria sustentável. Como consequência, a estratégia do projeto pivotou definitivamente para a refatoração progressiva e *in loco*.

4.3 Fase 3: Proposta de Reestruturação e Implementação

Com a estratégia *in loco* definida, estabeleceu-se o plano de ação metodológico para modernizar a camada visual do SIGET, atuando diretamente no repositório ativo sem interromper suas entregas. A proposta técnica pautou-se nos seguintes pilares:

1. *Adoção de um Design System Modular*: utilização de abordagem arquitetural *utility-first* para isolar a estilização no escopo de cada componente, mitigando regressões visuais em páginas legadas adjacentes.
2. *Integração de Componentes Acessíveis*: substituição progressiva de elementos HTML nativos e obsoletos por blocos modulares modernos aderentes a diretrizes estritas de acessibilidade web.
3. *Formulários Declarativos e Reativos*: migração da validação transacional centralizada no *back-end* para uma abordagem de gestão de estado no navegador, provendo feedback imediato e preventivo ao operador.
4. *Substituição Gradativa*: implementação não destrutiva, onde as rotas críticas de cadastro foram refatoradas prioritariamente, permitindo que os novos componentes coexistissem em paralelo com a interface legada.

O desenvolvimento prático e detalhado desta proposta de reestruturação é descrito integralmente na seção 5.

4.4 Fase 4: Avaliação da Solução

Para validar a eficácia da modernização arquitetural e da interface proposta, estipulou-se uma metodologia de avaliação multi-método. A qualidade técnica e ergonômica foi inspecionada qualitativamente por meio de Avaliação Heurística conduzida por especialistas da área de IHC. De forma complementar, a aceitação pragmática e a

usabilidade em contexto real foram mensuradas quantitativamente mediante testes empíricos com uma amostra de usuários finais, aplicando-se a ferramenta padronizada *System Usability Scale* (SUS). Os protocolos específicos, os perfis dos participantes e a análise pormenorizada dos resultados alcançados nesta fase são discutidos na seção 6.

5 Desenvolvimento e Implementação Prática

A transição metodológica descrita na seção anterior — a opção pela refatoração contínua sobre a reescrita total — exigiu uma execução meticulosa. Esta seção detalha o processo prático de modernização das interfaces do SIGET, evidenciando as principais intervenções arquiteturais que transformaram formulários densos e engessados em componentes interativos e eficientes. Exemplos das interfaces da versão inicial e da versão refatorada do sistema são mostrados na Seção 5.2.

5.1 Construção da Base e do Design System

A primeira etapa do desenvolvimento focou na reestruturação da fundação visual da aplicação. O código anterior possuía um leiaute global rigidamente acoplado, com barras laterais renderizadas imperativamente.

A fundação legada de CSS foi expurgada em favor da versão 4 do *Tailwind CSS*. Essa consolidação das variáveis de estilo estancou definitivamente os problemas de conflito em cascata (*CSS bleeding*). Com o ambiente CSS estabilizado, implementou-se o pacote de interface *Shadcn UI*, que viabilizou a substituição completa do esqueleto da aplicação.

A antiga barra de navegação foi reescrita utilizando o componente *Sidebar* do *Shadcn*, que provê responsividade e recolhimento nativo (*collapse*). A gestão de perfis de usuário, outrora relegada ao rodapé inferior com comportamentos inconsistentes, foi isolada estruturalmente e movida para o cabeçalho (*Header*) superior direito. Esta decisão alinha a aplicação à Lei de Fitts — um modelo preditivo da interação humano-motora que postula que o tempo para alcançar um alvo é função da distância até ele e de seu tamanho — e aos padrões mentais propostos por Nielsen (1993), diminuindo o tempo de busca visual do operador, uma vez que a maioria dos sistemas na web posiciona o acesso à conta neste quadrante. Os seletores de perfil (*Perfil* e *Vínculo*), frequentemente manipulados pelos servidores estaduais, ganharam acessibilidade de alto nível utilizando contêineres modulares nativos (*DropdownMenu*).

5.2 Padronização e Refatoração de Componentes Estruturais

Com o *Design System* assentado, o foco direcionou-se ao núcleo da aplicação: as listagens e modais operacionais.

5.2.1 Migração de Listagens para DataTable

A ampla maioria das rotas do SIGET (tais como a listagem de *Modelos de E-mail* ou a *Publicação de Editais*) baseava-se na construção manual de *tags* HTML primitivas (`<table>`, `<tr>`, `<td>`), montadas estaticamente na camada de renderização e sem suporte à filtragem ou paginação interna.

A arquitetura dessas páginas foi substancialmente modificada aplicando-se o componente centralizado *DataTable*, construído sobre a engine reativa da biblioteca *react-table*.

A alteração consolidou o Princípio da Responsabilidade Única (SRP) [Martin \(2018\)](#): arquivos centrais (`page.tsx`) deixaram de descrever regras de interação visual e um novo arquivo de suporte (`columns.tsx`) passou a gerir exclusivamente a tipagem e os mapeadores visuais (como a injeção automática de *Badges* coloridos de Status). Essa orquestração elevou de forma expressiva a eficiência de uso, permitindo que a busca unificada comprimissem ações complexas, como edição ou delegação de itens, em menus suspensos acionados por ícones (`DropdownMenu`).

5.2.2 Modais e *Dialogs* Nativos

Historicamente, as sobreposições flutuantes (*popups*) eram projetadas como `<div>` artesanais opacas, carecendo de recursos mínimos de acessibilidade, como confinamento de foco do teclado (*focus trap*) ou bloqueio da rolagem do fundo (*scroll lock*). Em rotas críticas de inserção, como o cadastro de Meso-regiões ou edição de Padrões de Edital no sistema, toda essa camada arcaica foi envelopada no conceito de *Dialog* do Shadcn. Tal transição automática garante, por exemplo, o fechamento via teclado (tecla ESC) e cliques externos em áreas vazias. Os antigos e disformes botões HTML deram lugar ao contêiner `Button`, agora emparelhado com a iconografia semântica limpa do pacote `lucide-react`.

5.3 Fluxos Complexos Modernizados

As refatorações granulares atingiram sua máxima expressão funcional ao remodelar os fluxos mais burocráticos e complexos de operação estadual. Esta seção descreve como os fluxos de interação complexos foram refatorados.

5.3.1 Gestão Reativa de Formulários Públicos

Nos cadastros abertos de usuários (prefeituras, EMATER e empresas), a gerência de dados dependia unicamente de `useState` manuais e chamadas sequenciais para *endpoints* não rastreáveis do *back-end*.

Para sanar a ausência de comunicação com o usuário, a nova arquitetura foi baseada em *React Hook Form* e *Zod*. As regras de formatação (tamanhos de texto, formatos exatos de E-mail ou CNPJ) foram movidas para *schemas* *Zod* puramente testáveis e desacoplados da lógica do componente visual. Isso viabilizou a validação *inline*: assim que o usuário transita entre campos, mensagens contextuais vermelhas indicam as correções antes mesmo de haver uma tentativa inútil de envio ao servidor. Ao transformar a submissão de dados em um processo que dá feedback imediato e positivo de digitação contínua (com a adoção de estilos como o atributo `aria-invalid`), atingiram-se em grande nível a visibilidade do status do sistema e o controle da prevenção de falhas do operador.

5.3.2 Redesenho Card-Based (Adesões e Validações)

A página de “*Aderir ao Edital*”, responsável pela recepção massiva de anexos dos municípios, apresentava status textuais com *backgrounds* sem contraste e misturava toda a árvore de envio sem hierarquia visual de aprovação. O módulo sofreu reengenharia mediante o desenvolvimento de um escopo *Card-Based*. Os documentos deixaram de ser itens amontoados, transformando-se em pequenos cartões interativos englobados por painéis limpos, com margens visuais arredondadas marcadas com indicadores (esmeralda para “Validado”, âmbar para “Criando”). Adicionou-se ainda o componente *Progress*, traduzindo instantaneamente ao servidor as porcentagens pendentes de envio.

O fluxo complementar a este, a “*Validação de Processo de Adesão*”, tela onde o analista de estado recusa ou aceita documentos, sofreu evolução semelhante. Substituíram-se antigos *radio buttons* minúsculos em favor de botões interativos maiores que assumem a semântica da ação do perito (Verde para Validação, Vermelho Destrutivo para Recusa). Essa progressão estruturada ajuda a evitar que o funcionário recuse algo por clique indevido, elevando consideravelmente o design emocional e o controle preventivo em rotas operacionais tão críticas e sensíveis às prefeituras estaduais.

6 Resultados e Avaliação

Esta seção apresenta os resultados práticos da refatoração e modernização do *front-end* do sistema SIGET. A avaliação foi estruturada em três eixos principais: métricas de código (demonstrando a mitigação da dívida técnica), comparativo visual das melhorias de usabilidade e interface, e avaliação empírica com especialistas e usuários finais, validando as escolhas arquiteturais.

6.1 Evolução Arquitetural e Redução da Dívida Técnica

A transição de uma arquitetura acoplada e baseada em elementos primários de HTML para um *Design System* modular gerou impactos mensuráveis na estrutura do projeto. A análise do repositório revela que, dos aproximadamente 30 módulos principais da aplicação, 20 (67%) foram integralmente migrados para a nova arquitetura baseada em React e componentes Shadcn UI.

Para alcançar este resultado, foram incorporados 26 novos componentes maduros do ecossistema Shadcn (como `Button`, `Dialog`, `Input`, `DataTable` e `Badge`), estabelecendo uma base de reutilização. Um dos maiores indícios de mitigação da dívida técnica foi a segregação de responsabilidades: em todos os módulos reestruturados, a renderização passiva de tabelas (`<table>` bruto) foi substituída pelo componente reativo `DataTable`, acompanhado da extração de um arquivo exclusivo (`columns.tsx`) para definir as tipagens e mapeadores visuais das colunas.

6.1.1 Análise de Acoplamento e Tamanho (Estudo de Caso)

Para tangibilizar a redução da dívida técnica exigida pela evolução do projeto, conduziu-se um estudo de caso comparativo focado no módulo de “Listagem de Editais”. A análise avaliou as métricas de estrutura de código antes e após a intervenção arquitetural:

- *Tamanho (Linhas de Código - LOC)*: na versão inicial, a tela de editais concentrava toda a lógica de paginação, busca e renderização visual de linhas em um único arquivo, que excedia 800 linhas de código (LOC). Com a nova arquitetura, a página principal foi reduzida para menos de 150 linhas, delegando a responsabilidade de renderização para o componente `<DataTable>`.
- *Separação de Responsabilidades (SRP)*: a configuração de cabeçalhos e emblemas de status (*Badges*) foi extraída para um arquivo isolado estrito (`columns.tsx`), eliminando lógicas condicionais complexas (`if/else`) de dentro do fluxo de renderização (JSX).

Cabe ressaltar que a migração arquitetural do SIGET adotou o *Padrão de Estrangulamento* Fowler (2018), que preconiza a substituição gradual do sistema antigo. Em

virtude do tamanho e do uso contínuo da aplicação pela SEAPA, o sistema ainda convive temporariamente com módulos residuais antigos que aguardam refatoração em iterações futuras e pequenas instabilidades (bugs) inerentes à transição, um cenário comum em refatorações progressivas de software em produção Fowler (2018).

6.1.2 Análise Quantitativa da Refatoração

A fim de contornar a subjetividade de análises puramente visuais e aferir concretamente o impacto da refatoração na manutenibilidade do repositório, foi realizado um levantamento quantitativo das estruturas de código antes e após a adoção do *Design System*. Essa verificação objetiva os reflexos da nova arquitetura na padronização dos componentes e na mitigação da dívida técnica.

O levantamento revelou que, do escopo total aproximado de 30 módulos CRUD mapeados no SIGET, 20 módulos (67%) foram integralmente migrados para a nova arquitetura orientada ao Shaden UI. Entre os 10 módulos remanescentes (33%), preservou-se a estrutura anterior aguardando iterações futuras, respeitando a premissa de transição gradual ditada pelo Padrão de Estrangulamento.

Para suportar essa migração estrutural, 26 novos componentes maduros do ecossistema Shaden foram incorporados à base de código (como `Button`, `Dialog`, `DataTable`, `Input` e `Badge`). A adoção em massa desses elementos propiciou um alto índice de reaproveitamento, extinguindo a proliferação de estilizações paralelas. A título de exemplo, o novo componente padronizado de botão (`<Button>`) atingiu 47 ocorrências de importação ao longo dos módulos refatorados, enquanto o componente genérico de janelas modais (`<Dialog>`) foi instanciado 17 vezes, consolidando a consistência visual em dezenas de interfaces.

Outro marco arquitetural extraído das métricas diz respeito ao Princípio de Responsabilidade Única (SRP). Em 100% dos módulos migrados que demandam listagem de dados, a marcação primitiva de tabelas (`<table>`) foi substituída pelo `<DataTable>`. Este movimento foi acompanhado pela extração sistemática das lógicas de colunas, formatação e emblemas de status para arquivos isolados (`columns.tsx`), o que culminou na redução da complexidade dos arquivos principais de página.

6.2 O (Re)Design da Solução SIGET

Para tangibilizar o impacto do novo design, destacam-se a seguir as interfaces principais do sistema que sofreram intervenção arquitetural e estética.

Aviso sobre Imagens: *Com o intuito de facilitar a leitura visual das diferenças descritas a seguir (sobretudo em versões impressas ou com aproximação de 100%), as Figuras apresentarão as interfaces lado a lado ou com foco estrito nos componentes alterados, mitigando o problema de ilegibilidade presente em capturas de telas inteiras.*

6.2.1 Design System como Contribuição Estrutural

A principal contribuição arquitetural desta intervenção foi a concepção e implementação do *Design System* baseado em Shaden UI e Tailwind CSS. Antes da intervenção, a ausência de um sistema de design exigia que cada desenvolvedor reinventasse soluções de interface, gerando inconsistência visual acentuada. O novo *Design System* operou como a fundação de todo o (re)design subsequente, oferecendo um catálogo previsível, acessível

e estritamente tipado de blocos construtivos (botões, caixas de diálogo, tabelas). Isso não apenas unificou a estética governamental da SEAPA, como reduziu de forma expressiva o tempo necessário para criar ou manter novas telas.

6.2.2 Reestruturação Global e Lógica de Navegação

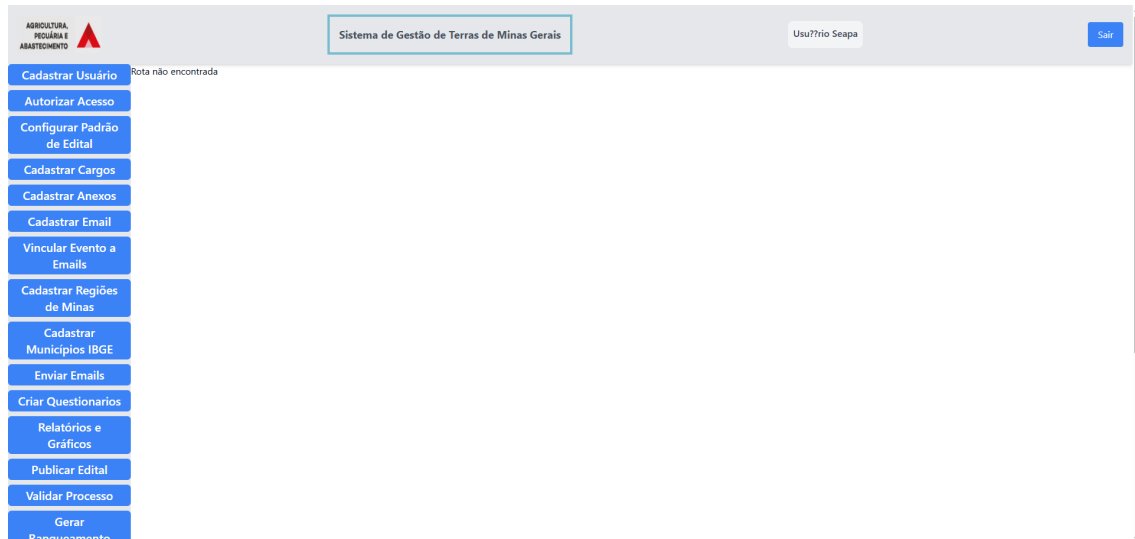
No que diz respeito à estrutura global da aplicação, o leiaute antigo apresentava caixas desconfiguradas e menus listados de forma engessada, comprometendo a navegabilidade. A aplicação adotou a *Sidebar Responsiva* do Shadcn. Mais do que uma mudança estética, a lógica de organização do menu principal foi fundamentalmente alterada: os itens deixaram de ser exibidos em longas listas genéricas para serem agrupados semanticamente por contexto de operação (ex: "Administração", "Editais", "Avaliações"). Esse agrupamento reduz a sobrecarga cognitiva, pois o usuário não precisa varrer visualmente todas as opções, direcionando sua atenção apenas ao módulo que atende ao seu modelo mental de trabalho (Figuras 1 e 2).

6.2.3 Linguagem, Acessibilidade e Formulários

A tela de login também passou por reformulação. A interface antiga apresentava um design básico, ao passo que a nova versão exibe uma identidade modernizada com destaque para a marca institucional. Uma mudança sutil, porém significativa para a usabilidade e padronização, foi a adoção do termo "Entrar" em vez de "Login" no botão de ação principal. Optou-se pelo verbo de ação claro na língua materna para diminuir a barreira cognitiva, alinhando-se a diretrizes de linguagem simples em serviços governamentais (Figuras 3 e 4).

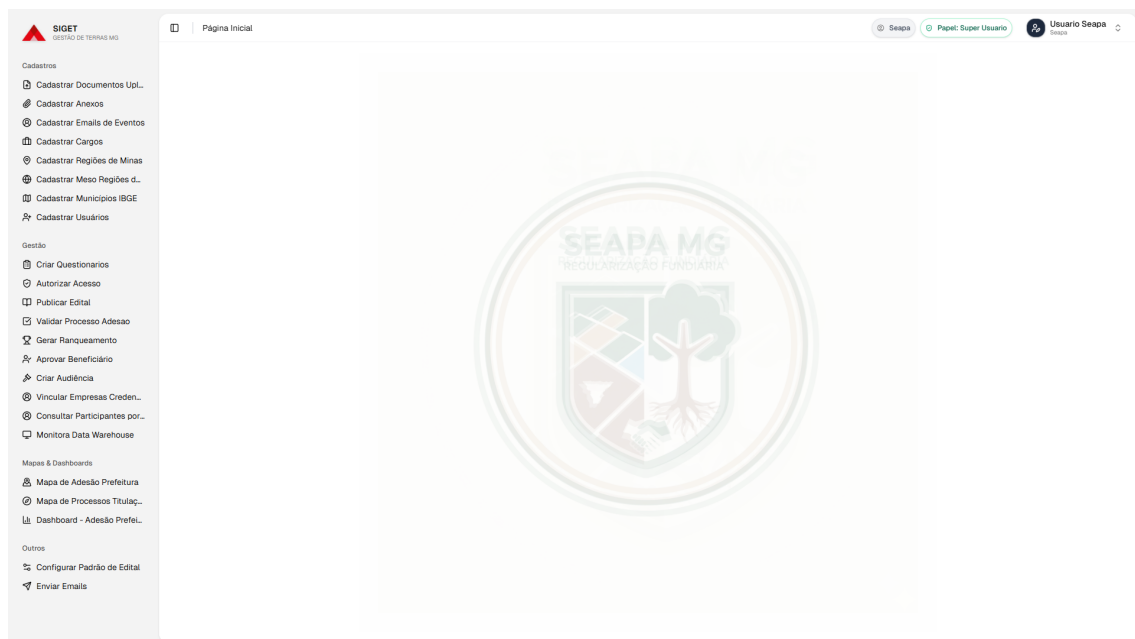
Em relação aos formulários de cadastro, adotou-se o compromisso de manter a interface a mais limpa possível. Nomes de campos extensos e com espaços foram padronizados para evitar contrações e abreviações confusas que prejudicassem a leitura dinâmica. Além disso, embora a indicação clássica de obrigatoriedade (asteriscos) seja omitida no estado de repouso por escolha de design minimalista da biblioteca adotada, a validação reativa *inline* acusa imediatamente a obrigatoriedade daquele preenchimento assim que o usuário tenta pular o campo ou submeter a requisição, fornecendo a instrução no exato momento da infração.

Figura 1 – Leiaute Global Antigo



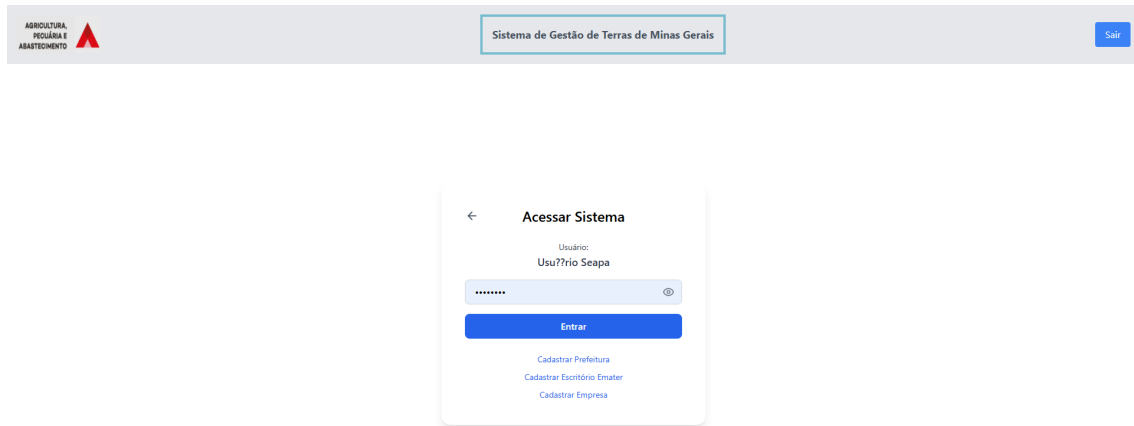
Fonte: Elaborado pelo autor.

Figura 2 – Leiaute Global Novo



Fonte: Elaborado pelo autor.

Figura 3 – Tela de Login Antiga



Fonte: Elaborado pelo autor.

Figura 4 – Tela de Login Nova



Fonte: Elaborado pelo autor.

6.2.4 Padronização e Trade-offs em Listagens

No âmbito da padronização, as antigas sobreposições flutuantes (*popups*), projetadas artesanalmente em `<div>` opacas, foram substituídas pelo componente `Dialog` nativo (Figuras 5, 6 e 7), assegurando o fechamento via teclado e o travamento de fundo.

As listagens de dados (*Cadastrar E-mail*, *Editais*), antes construídas em HTML estático e com poluição de botões visíveis para cada linha, foram convertidas para `DataTable` reativas. Ocultar opções de manipulação (como Editar e Excluir) dentro de menus suspensos (`DropDownMenu`) sob um ícone representou um *trade-off* intencional de design.

Embora exija um clique extra do usuário (diminuindo levemente a *discoverability* ou "descobrimento"), essa escolha elimina a severa poluição visual provocada por dezenas de botões repetidos ao longo de uma tela com dezenas de registros, aderindo à heurística de Estética e Design Minimalista. Para mitigar o custo do clique extra, a reestruturação da arquitetura previu a futura implementação de "ações em lote" (*batch actions*), permitindo a exclusão de múltiplos itens simultaneamente através de seleção por *checkboxes* (Figuras 8, 9 e 10).

Figura 5 – Modal Antigo (Cadastro de Região)

Cadastrar Regiões de Minas

— Voltar

Código da região	Nome da região	Ações
I	Central	 
II	Mata	 
III	Sul de Minas	 
IV	Triângulo	 
V	Alto Paranaíba	 
VI	Centro-Oeste	 
VII	Noroeste	 
VIII	Norte	 
IX	Jequitinhonha-Mucuri	 
X	Rio Doce	 

Inserir Nova Região

Nova Região

Código da Região:

Nome da Região:

Cancelar

Salvar

Fonte: Elaborado pelo autor.

Figura 6 – Dialog Novo (Cadastro de Município)

Novo Município

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Nome

UF

Pop. Urbana

Pop. Rural

Tot. Urbana

IBGE

Região

Meso Região

IBGE

Fonte: Elaborado pelo autor.

Figura 7 – Dialog Novo (Edição de Edital)

Novo Edital

Código: Ano: Descrição:

Link para o Edital:

Publicação DOE:

Carregando critérios...

Prazos e Datas (Base: Publicação DOE)

Qtd. Dias Inscrição: Qtd. Dias Pub. Lista:

Qtd. Dias Recurso: Qtd. Dias Vigência:

☒ Edital Ativo

Cancelar

Fonte: Elaborado pelo autor.

Figura 8 – Tabela de Dados Antiga

AGRICULTURA, Pecuária e Abastecimento

Sistema de Gestão de Terras de Minas Gerais

Usu77rio Seapa

Cadastrar Usuário

CPF	Nome	E-mail	Administrador	Funcionário	Gestor	Ativo	Ações
12121212121	test12	test12@email.com	✗	✓	✓	✗	
31313131313	test31	test31@email.com	✗	✓	✗	✓	
87878787878	test87	test87@email.com	✓	✓	✓	✓	
90909090909	test90	test90@email.com	✓	✓	✓	✗	
40028922222	Som do yudi	valdarps2@email.com	✓	✗	✓	✓	
09809809090	test908	test90-8@email.com	✓	✓	✓	✓	
22222222222	Usu77rio Seapa	emarchettisilva@gmail.com	✓	✓	✓	✓	
11122233344	Victor Mont Mor	vicvicvic@email	✓	✓	✓	✓	
99999999999	test9	test9@email.com	✓	✓	✓	✓	
88888888888	test8	test1@email.com	✓	✓	✗	✓	
77777777777	test7	test7@email.com	✓	✗	✓	✗	
66666666666	test6	test6@email.com	✓	✓	✓	✓	
55555555555	test5	test5@email.com	✗	✓	✓	✓	

Fonte: Elaborado pelo autor.

Figura 9 – DataTable Reativa Nova

IBGE	Nome	IDH	Pop. Urbana	Pop. Rural	% Urbana	IDENE	Região	Meso Região
3100104	Abadia dos Dourados	0.689	4392	1880	70.03%	×	Uberlândia	Triângulo Mineiro/Alto Paranaíba
3100203	Abatã	0.698	20260	2415	89.35%	×	Divinópolis	Central Mineira
3100302	Abre Campo	0.654	8495	5432	61.00%	×	Juiz de Fora	Zona da Mata
3100401	Acaiaçu	0.630	2834	1075	72.50%	×	Juiz de Fora	Zona da Mata
3100500	Agucena	0.610	4521	4422	50.55%	✓	Ipatinga	Vale do Rio Doce
3100609	Água Boa	0.576	6742	5847	53.55%	✓	Teófilo Otoni	Vale do Rio Doce
3100708	Água Comprida	0.675	1651	457	78.32%	×	Uberaba	Triângulo Mineiro/Alto Paranaíba
3100807	Aguanil	0.663	3407	950	78.20%	×	Varginha	Oeste de Minas
3100906	Águas Formosas	0.645	14496	3952	78.58%	✓	Teófilo Otoni	Vale do Mucuri
3101003	Águas Vermelhas	0.601	10340	3697	73.66%	✓	Teófilo Otoni	Norte de Minas

Fonte: Elaborado pelo autor.

Figura 10 – Filtragem e Personalização de Colunas na Nova Tabela

CPF	Nome	Status	Permissões
1111111101	Adm Seapa	Ativo	Administrador Seapa
1111111111	Admin Demo	Inativo	Papel inválido

Fonte: Elaborado pelo autor.

No tocante à validação de formulários, os fluxos de cadastro abertos a entidades externas (Prefeituras, Empresas) dependiam exclusivamente de validação no servidor, frustrando os operadores com erros tardios identificados somente após a submissão completa do formulário. Com a adoção do **React Hook Form** e do **Zod**, os campos passaram

a fornecer feedback em tempo real, por meio de bordas vermelhas semânticas exibidas instantaneamente ao sair de um campo preenchido de forma incorreta. Essa abordagem transferiu a carga de validação para o navegador, permitindo que o usuário corrija inconsistências antes mesmo de tentar o envio ao servidor (Figuras 11 e 12).

Figura 11 – Erro de Cadastro sem Feedback Visual Prévio (Antigo)

A imagem mostra a interface de usuário do sistema SIGET, especificamente a tela de 'CADASTRO PREFEITURA'. O formulário está dividido em três etapas: 1. PREFEITURA, 2. CONTATOS e 3. ADMINISTRADOR. A etapa atual é a 3. ADMINISTRADOR, onde o usuário precisa fornecer 'Dados do Usuário Administrador'. Os campos incluem CPF, Nome e E-mail. Abaixo do formulário, há uma lista de mensagens de erro em português, indicando problemas como CPF inválido, nome obrigatório e e-mail inválido. O formulário não possui bordas vermelhas ou outros indicadores visuais de erro antes do envio.

ABRIGAMENTO, PECUÁRIA E ABASTECIMENTO

SIGET
SISTEMA DE GESTÃO DE TERREIRAS DE ÁREAS GRANDES

CADASTRO PREFEITURA

1. PREFEITURA 2. CONTATOS 3. ADMINISTRADOR

Dados do Usuário Administrador

CPF * Nome *
E-mail *

Voltar Finalizar

- Código do Município tem de ser um valor numérico inteiro - invalid literal for int() with base 10: "
- Quantidade de dígitos do CNPJ diferente de 14!
- Bairro é obrigatório e limitado a 50 caracteres!
- Quantidade de dígitos do CPF diferente de 11!
- Erro ao validar dados da prefeitura -> invalid literal for int() with base 10: "
- Pelo menos um contato tem de ser informado
- Quantidade de dígitos do CPF diferente de 11!
- E-mail do usuário é obrigatório!
- A chave 'documentos' não foi encontrada no envio.

Fonte: Elaborado pelo autor.

Figura 12 – Validação em Tempo Real com *Feedback* Visual (Novo)

A imagem mostra a interface de usuário do sistema SIGET, especificamente a tela de 'PREFEITURA'. O formulário está dividido em quatro etapas: 1. PREFEITURA, 2. CONTATOS, 3. ADMINISTRADOR e 4. REVISÃO. A etapa atual é a 3. ADMINISTRADOR, onde o usuário precisa fornecer 'Dados do Usuário Administrador'. Os campos incluem CPF, Nome e E-mail. Abaixo do formulário, há uma seção 'Documentação Anexa' com campos para upload de arquivos. O formulário possui bordas vermelhas e mensagens de erro em português, indicando problemas como CPF inválido, nome obrigatório e e-mail inválido. O formulário também possui botões para 'Voltar' e 'Finalizar'.

ABRIGAMENTO, PECUÁRIA E ABASTECIMENTO

SIGET
SISTEMA DE GESTÃO DE TERREIRAS DE ÁREAS GRANDES

PREFEITURA

1. PREFEITURA 2. CONTATOS 3. ADMINISTRADOR 4. REVISÃO

Dados do Usuário Administrador

CPF * Nome *
E-mail *

CPF inválido. Nome é obrigatório.
E-mail inválido.

Documentação Anexa

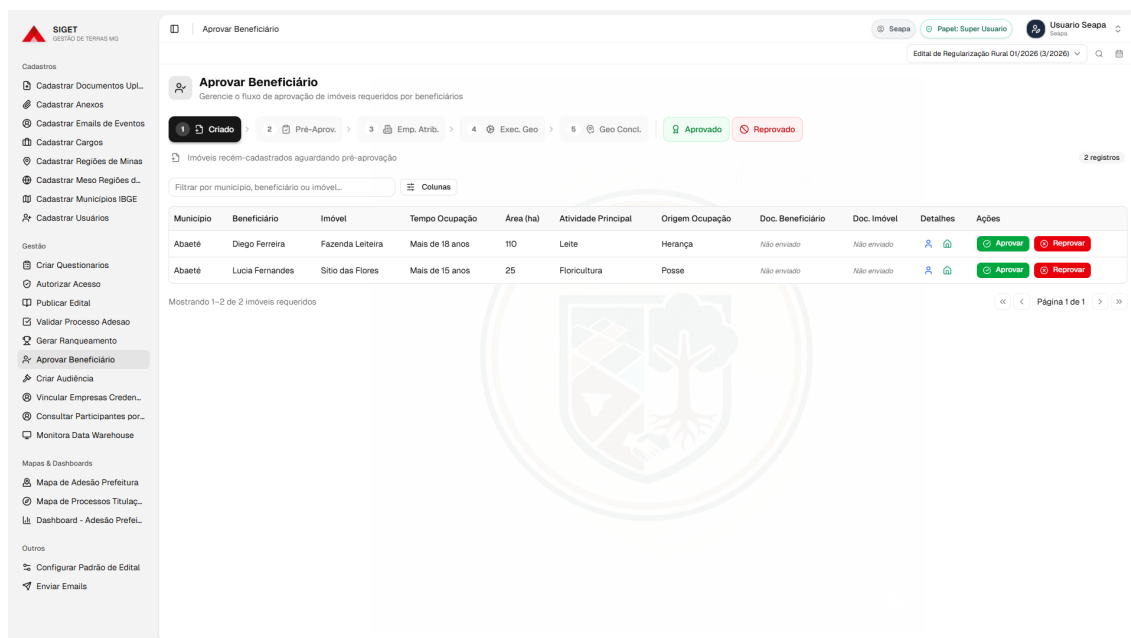
Comprovante de Vínculo com a Prefeitura
Documento de Identidade
Comprovante de vínculo prefeitura
Carteira de trabalho

Confirmar Dados e Documentos

Fonte: Elaborado pelo autor.

Por fim, no que concerne à clareza e à acessibilidade, duas intervenções merecem destaque. Em páginas de alta complexidade e com múltiplas etapas, introduziram-se descrições textuais e indicadores visuais de progresso — representações gráficas que exibem dinamicamente na interface as etapas sequenciais do fluxo de trabalho e o ponto em que o usuário se encontra — que auxiliam o entendimento de novos usuários e reduzem a necessidade de memorização de funcionalidades. No fluxo de cadastro de beneficiários, por exemplo, as etapas de aprovação foram mapeadas visualmente adotando um padrão semântico de cores: verde para aprovação, vermelho para reprovação e amarelo para alertas ou pendências (Figura 13). Ademais, a interface passou a oferecer suporte a atalhos de teclado, recurso antes inexistente e que limitava a acessibilidade do sistema. A nova implementação introduziu uma navegação dinâmica compatível com padrões web convencionais, como a tecla ESC para fechar modais e a tecla TAB para transitar ordenadamente entre opções, conferindo maior agilidade e ergonomia à operação do sistema.

Figura 13 – Fluxograma Visual Dinâmico no Cadastro de Beneficiário (Novo)



Fonte: Elaborado pelo autor.

6.3 Avaliação da Solução Proposta na Perspectiva de Especialistas

Para avaliar a proposta de reestruturação do *front-end* do SIGET, foi realizada uma Inspeção Heurística conduzida por dois especialistas ($n=2$) com perfis sêniores e ampla vivência na construção e pesquisa de interfaces corporativas e governamentais: o Especialista A atua como desenvolvedor *front-end* sênior, lidando diariamente com os desafios arquiteturais de migração de legados; o Especialista B é prestador de serviços em software com foco metodológico em usabilidade. Eles submeteram as interfaces antiga e nova aos mesmos fluxos de tarefa, pontuando problemas conforme o método consagrado de avaliação e as 10 heurísticas de usabilidade propostas por Nielsen (1993), com uma escala de severidade variando de 0 (sem problema) a 4 (catástrofe de usabilidade). O Quadro 1 apresenta as dez heurísticas utilizadas como critérios de avaliação neste estudo.

Tabela 1 – As 10 Heurísticas de Usabilidade de Nielsen

Código	Heurística
H1	Visibilidade do status do sistema
H2	Correspondência entre o sistema e o mundo real
H3	Controle e liberdade do usuário
H4	Consistência e padrões
H5	Prevenção de erros
H6	Reconhecimento em vez de memorização
H7	Flexibilidade e eficiência de uso
H8	Estética e design minimalista
H9	Ajuda aos usuários para reconhecer, diagnosticar e recuperar erros
H10	Ajuda e documentação

Fonte: Adaptado de Nielsen (1993).

Os especialistas identificaram múltiplas infrações severas de usabilidade na versão antiga. As heurísticas de *Estética e Design Minimalista* (H8), *Flexibilidade e Eficiência de Uso* (H7) e *Consistência e Padrões* (H4) receberam as piores avaliações (graus 3 e 4, indicando problemas maiores a catástrofes). Dentre as principais infrações apontadas nos relatórios qualitativos dos avaliadores, destacam-se:

- *Inconsistência Visual e Estrutural* (H4): o uso irregular de fontes, estilos e elementos de interface (como modais e formulários construídos de maneiras diferentes ao longo do sistema) comprometeu a familiarização do usuário;
- *Deficiência de Feedback e Prevenção de Erros* (H1 e H5): a ausência de validações de formulário em tempo real (mensagens de erro *inline*) e a falta de *toasts* ou confirmações claras deixavam o usuário em dúvida sobre o sucesso ou falha de suas requisições;
- *Sobrecarga Cognitiva em Listagens* (H7 e H8): páginas contendo tabelas com altos volumes de dados careciam de opções de filtragem ou paginação eficiente, exigindo esforço visual demasiado e dificultando a localização de informações específicas;
- *Barreiras de Acessibilidade*: foi constatada a impossibilidade de operar certos fluxos cruciais apenas pelo teclado, além de modais inacessíveis e navegação de menus confusa.

Os mesmos especialistas constataram uma redução dos problemas nas rotas re-estilizadas. A severidade dos gargalos avaliados caiu para níveis entre 0 e 1 (problemas ausentes ou apenas cosméticos). A melhoria mais apontada ocorreu nas heurísticas de Consistência e Estética, resultante principalmente da introdução de paginação reativa, da acessibilidade nativa via teclado (ESC em modais) e do feedback instantâneo de erros nos novos formulários. Na etapa de consolidação dos pareceres, em que os avaliadores contrastaram seus relatórios individuais sobre as duas versões, ambos os especialistas classificaram a nova interface como perceptivelmente superior e consideraram-na mais organizada, consistente e acessível, corroborando a hipótese de que a padronização modular e o respeito às heurísticas de usabilidade produzem ganhos perceptíveis de manutenibilidade e qualidade de software.

6.4 Avaliação da Solução Proposta na Perspectiva dos Usuários

Para a avaliação quantitativa da usabilidade percebida, utilizou-se a *System Usability Scale* (SUS), um questionário padronizado composto por 10 afirmações em escala Likert, desenvolvido por Brooke (1996). O instrumento é amplamente adotado na literatura de IHC como uma métrica rápida e confiável para mensurar a satisfação subjetiva do usuário com um sistema interativo. O Quadro 2 apresenta as afirmações que compõem o questionário SUS, conforme adaptação para o português.

Tabela 2 – Afirmações do Questionário SUS (*System Usability Scale*)

N.º	Afirmação
1	Eu acho que gostaria de usar esse sistema com frequência.
2	Eu achei o sistema desnecessariamente complexo.
3	Eu achei o sistema fácil de usar.
4	Eu acho que precisaria de apoio de um técnico para usar esse sistema.
5	Eu achei que as várias funções do sistema estão muito bem integradas.
6	Eu achei que havia muita inconsistência neste sistema.
7	Eu imagino que a maioria das pessoas aprenderia a usar esse sistema rapidamente.
8	Eu achei o sistema muito complicado de usar.
9	Eu me senti muito confiante usando o sistema.
10	Eu precisei aprender várias coisas antes de conseguir usar o sistema.

Fonte: Adaptado de Brooke (1996).

O questionário foi aplicado a um grupo de 9 indivíduos, selecionados com o objetivo de compor uma amostra diversificada, que englobasse tanto a alta capacidade técnica e acadêmica quanto a perspectiva do usuário final sem vivência tecnológica. O perfil dos participantes dividiu-se em três categorias:

- *Especialista Acadêmico (1 participante)*: profissional com doutorado e mestrado em Ciências da Computação pela Universidade Federal de Minas Gerais (UFMG) e graduação em Sistemas de Informação pela Pontifícia Universidade Católica de Minas Gerais (PUC Minas). O profissional possui atuação nos temas de desenvolvimento de software, *data mining* e *machine learning*. Sua avaliação proveu um olhar arquitetural robusto e embasado.
- *Usuários Inexperientes (2 participantes)*: indivíduos com faixa etária superior a 50 anos e sem qualquer familiaridade com o uso de tecnologia. A participação desse grupo foi importante para simular o comportamento empírico de um usuário leigo da ponta do ecossistema do SIGET (por exemplo, um produtor de zonas rurais submetendo uma requisição técnica), testando os limites da acessibilidade e da clareza da nova interface.
- *Avaliadores com Formação Técnica (6 participantes)*: graduandos concluintes do curso de Engenharia de Computação, com idades entre 20 e 30 anos e elevada fluência digital. A pertinência desse grupo justifica-se não apenas pelo conhecimento intrínseco de software, mas fundamentalmente por já terem cursado a disciplina de Interação Humano-Computador (IHC), conferindo-lhes propriedade teórica para julgar a usabilidade e a estética visual das interfaces apresentadas.

Os participantes assistiram a vídeos executando fluxos reais de trabalho nas duas versões do sistema, sendo em seguida indagados acerca de sua confiança, necessidade de suporte e satisfação geral com as interfaces.

As respostas ao formulário SUS refletiram de maneira unânime a percepção de evolução qualitativa. Indagados sobre as versões:

- 100% dos respondentes afirmaram que a *Interface Nova* “transmite uma impressão mais profissional e moderna”.
- 100% atestaram que, caso tivessem que usar o SIGET em sua rotina diária de trabalho, “prefeririam a interface nova”.

A antiga arquitetura despertava incertezas nas avaliações sobre confiabilidade e complexidade desnecessária. Já a interface refatorada registrou picos de concordância máxima no quesito “facilidade de integração das funções”, demonstrando que os novos *Cards* padronizados e formulários tipados conferem segurança à operação visual.

6.5 Discussão dos Resultados

Os dados levantados pelas métricas de repositório, atrelados ao feedback empírico (seja de experts ou de usuários inexperientes), indicam que a abordagem de refatoração in loco mitigou satisfatoriamente a dívida técnica sem invalidar o fluxo diário de negócios da SEAPA, provando o sucesso prático do *Design System* como ferramenta saneadora de código. Do lado interativo, ao incorporar bibliotecas acessíveis (*Shadcn UI*) e validações em tempo real, atingiu-se as metas de usabilidade almejadas, transformando um sistema de gestão historicamente rígido em uma ferramenta aderente e ergonômica para o corpo estadual.

6.5.1 Limitações do Trabalho

A despeito dos avanços relatados, este trabalho convive com limitações metodológicas intrínsecas ao seu escopo e tempo de execução, que merecem ser ponderadas para uma correta interpretação dos resultados:

- *Ausência de Participação do Usuário na Fase Inicial:* A definição das melhorias foi conduzida sem a realização preliminar de entrevistas ou sessões de *brainstorming* formais com os usuários finais. Esta restrição foi uma decisão tática e emergencial: a dívida técnica acumulada era tamanha que as violações estruturais da interface (como botões minúsculos inoperáveis e formulários sem feedback) eram barreiras fundamentais que precediam qualquer refinamento comportamental mais subjetivo, além de se apoiar nas demandas já sabidas via operação do projeto piloto na SEAPA. Contudo, essa ausência configura uma limitação estrita aos preceitos clássicos do Design Centrado no Usuário, os quais pressupõem a cocriação ativa desde a concepção.
- *Interação Simulada na Avaliação SUS:* A aplicação da ferramenta SUS é majoritariamente indicada após a interação prolongada e real do usuário no cotidiano da ferramenta. Neste projeto, devido a restrições de tempo de desenvolvimento e cronogramas de implantação em produção pelo governo do Estado, os participantes do SUS avaliaram a nova interface observando a execução guiada de fluxos funcionais

representativos. O fato de os usuários não terem experimentado livremente o estresse do uso autônomo e de resolução de erros em produção interfere nos resultados; o índice de satisfação capturado pela pesquisa atesta primordialmente uma “preferência estética e estrutural” baseada em cenários otimizados, devendo ser observada com cautela em relação a uma validação de usabilidade sob o rigor diário da burocracia.

7 Considerações Finais

Este trabalho apresentou o processo de modernização arquitetural e visual da interface do Sistema de Gestão de Editais (SIGET), ferramenta essencial para o fluxo operacional da Secretaria de Estado de Agricultura, Pecuária e Abastecimento de Minas Gerais (SEAPA). Ao longo do desenvolvimento, demonstrou-se na prática como a união de princípios de Interação Humano-Computador (IHC) e padrões modernos de Engenharia de Software pode revitalizar as interfaces de um sistema e reduzir o acúmulo de dívida técnica.

O principal êxito técnico do projeto residiu na consolidação de uma metodologia viável de refatoração contínua (*in loco*). Atuando diretamente no repositório em produção, a intervenção alcançou a migração integral de 67% dos módulos do sistema para a nova arquitetura baseada em React e componentes do Shadcn UI. O expurgo do estilo global despadronizado e a adoção do Tailwind CSS garantiram o isolamento estético. Paralelamente, fluxos críticos foram enriquecidos com validação de dados em tempo real (*React Hook Form* e *Zod*) e acessibilidade nativa.

7.1 Contribuições do Trabalho

As contribuições deste estudo transcendem o ganho operacional da SEAPA, oferecendo relevância direta tanto para a academia quanto para a indústria. Para pesquisadores da área de IHC e Engenharia de Software, o estudo provê evidências empíricas sobre a eficácia do Padrão de Estrangulamento aplicado estritamente na camada visual, demonstrando como a síndrome do "alvo em movimento" em monolitos governamentais pode ser mitigada com a injeção paralela de um *Design System* utilitário.

Para os profissionais do mercado de tecnologia, o relato fornece um roteiro pragmático e validado de modernização progressiva. A documentação da transição — do fracasso da abordagem *greenfield* (reescrita total) ao sucesso da refatoração fragmentada — atua como um guia de melhores práticas para times que necessitam sanear dívidas técnicas massivas sem paralisar o desenvolvimento de novas funcionalidades (*feature freeze*) de sistemas corporativos em plena operação.

7.2 Trabalhos Futuros

Apesar dos resultados evidenciarem um avanço no *front-end* do SIGET, a natureza contínua do desenvolvimento de software determina que a modernização não se encerra nesta etapa. Para a continuidade prática e acadêmica deste trabalho, sugerem-se como trabalhos futuros:

- *Migração do código remanescente*: finalizar a refatoração dos cerca de 33% de módulos antigos e residuais, como os fluxos de Ranqueamento e Aceite de Requisições, padronizando 100% a camada visual da aplicação e permitindo a remoção definitiva dos arquivos `.css` antigos.

- *Implementação de testes automatizados (E2E)*: com a interface estabilizada em componentes previsíveis, torna-se oportuna a adoção de testes ponta a ponta (com ferramentas como *Cypress* ou *Playwright*) para garantir que futuras atualizações não provoquem regressões na validação de formulários.
- *Ampliação da acessibilidade*: aprofundar as auditorias de acessibilidade seguindo as *Web Content Accessibility Guidelines* (WCAG) 2.1, conjunto de recomendações internacionais publicadas pelo *World Wide Web Consortium* (W3C) para tornar conteúdos web acessíveis a pessoas com deficiência, assegurando alto contraste e total compatibilidade com tecnologias assistivas.
- *Validação SUS longitudinal*: replicar o questionário SUS após um semestre de uso orgânico da nova interface em ambiente de produção, comparando os novos índices de usabilidade com os índices obtidos pela interação observada do presente estudo.

Em suma, o projeto atendeu de forma satisfatória aos objetivos propostos, entregando à SEAPA e à sociedade civil uma ferramenta governamental mais ágil, robusta, esteticamente agradável e preparada para escalar tecnicamente nos próximos anos.

Referências

BORGES, J. M.; ARAÚJO, R. D. Experiences and challenges of a redesign process with the support of an ai assistant on an educational platform. In: *Anais do XXIII Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC)*. Brasília, DF: SBC, 2024. p. 98–109. Disponível em: <<https://sol.sbc.org.br/index.php/ihc/article/view/32862>>. Acesso em: 02 jul. 2026. Citado na página 7.

BROOKE, J. Sus: A ‘quick and dirty’ usability scale. In: JORDAN, P. W. et al. (Ed.). *Usability Evaluation in Industry*. London: Taylor & Francis, 1996. p. 189–194. ISBN 978-0748404605. Disponível em: <<https://doi.org/10.1201/9781498710411-35>>. Acesso em: 02 jul. 2026. Citado na página 23.

CUNNINGHAM, W. The wycash portfolio management system. *ACM SIGPLAN OOPS Messenger*, ACM, v. 4, n. 2, p. 29–30, 1992. Disponível em: <<https://dl.acm.org/doi/10.1145/157709.157715>>. Acesso em: 06 jun. 2026. Citado na página 5.

FOWLER, M. *Refactoring: Improving the Design of Existing Code*. 2. ed. Boston: Addison-Wesley Professional, 2018. ISBN 978-0134757599. Disponível em: <<https://martinfowler.com/books/refactoring.html>>. Acesso em: 02 jul. 2026. Citado 5 vezes nas páginas 3, 5, 9, 12 e 13.

GARRETT, J. J. *The Elements of User Experience: User-Centered Design for the Web and Beyond*. 2. ed. Berkeley, CA: New Riders, 2011. ISBN 978-0321683687. Disponível em: <<https://www.peachpit.com/store/elements-of-user-experience-user-centered-design-for-the-9780321683687>>. Acesso em: 02 jul. 2026. Citado na página 5.

HEWETT, T. T. et al. *ACM SIGCHI Curricula for Human-Computer Interaction*. New York: ACM, 1992. Disponível em: <<https://dl.acm.org/doi/book/10.1145/2594128>>. Acesso em: 02 jul. 2026. Citado na página 4.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO 9241-11: Ergonomics of human-system interaction – Part 11: Usability: Definitions and concepts*. Geneva, Switzerland, 2018. Disponível em: <<https://www.iso.org/standard/63500.html>>. Acesso em: 06 jun. 2026. Citado na página 4.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO 9241-210: Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems*. Geneva, Switzerland, 2019. Disponível em: <<https://www.iso.org/standard/77520.html>>. Acesso em: 23 jun. 2026. Citado na página 3.

LOPES, L. F. P. *Integração do Sistema de Gestão de Terras de Minas Gerais (SIGET-MG) com o SEI-MG*. Monografia (Trabalho de Conclusão de Curso) — Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2024. Citado na página 2.

MARTIN, R. C. *Arquitetura Limpa: O Guia do Artesão para Estrutura e Design de Software*. Rio de Janeiro: Alta Books, 2018. ISBN 978-8550804606. Disponível em: <<https://altabooks.com.br/produto/arquitetura-limpa/>>. Acesso em: 02 jul. 2026. Citado 3 vezes nas páginas 5, 6 e 11.

NIELSEN, J. *Usability Engineering*. San Francisco, CA: Morgan Kaufmann, 1993. ISBN 978-0125184069. Disponível em: <<https://www.nngroup.com/books/usability-engineering/>>. Acesso em: 02 jul. 2026. Citado 6 vezes nas páginas 2, 3, 4, 10, 21 e 22.

NORMAN, D. A. *The Design of Everyday Things*. Revised and expanded. New York: Basic Books, 2013. ISBN 978-0465050659. Disponível em: <<https://www.hachettebookgroup.com/titles/don-norman/the-design-of-everyday-things/9780465050659/>>. Acesso em: 02 jul. 2026. Citado na página 5.

OLIVEIRA, A. C. L. de; SANTOS, P. H. B.; MARQUES, A. B. Redesign da usinn modeler para solução de problemas de usabilidade e acessibilidade. In: *Anais Estendidos do XXIII Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC)*. Brasília, DF: SBC, 2024. p. 149–153. Disponível em: <https://sol.sbc.org.br/index.php/ihc_estendido/article/view/30654>. Acesso em: 02 jul. 2026. Citado 2 vezes nas páginas 7 e 8.

SILVA, I. S.; MARCHETTI, E. *Plano de Trabalho — Edital N.º 006/2024 HUBMG GOV*. Belo Horizonte: [s.n.], 2024. Documento interno do projeto de extensão CEFET-MG/SEAPA-MG. Citado na página 2.

SOMMERVILLE, I. *Software Engineering*. Pearson, 2011. (International Computer Science Series). ISBN 9780137053469. Disponível em: <<https://books.google.com.br/books?id=l0egcQAACAAJ>>. Citado na página 2.

SUAREZ, M. et al. *Design Systems Handbook*. DesignBetter by InVision, 2019. Disponível em: <<https://web.archive.org/web/20230220212232/https://www.designbetter.co/design-systems-handbook>>. Acesso em: 02 jul. 2026. Citado na página 3.



MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS
GERAIS
DEPARTAMENTO DE COMPUTAÇÃO - NG



FOLHA DE APROVAÇÃO DE TCC Nº 7 / 2026 - DECOM (11.56.03)

Nº do Protocolo: 23062.031708/2026-21

Belo Horizonte-MG, 19 de junho de 2026.

VICTOR SAMUEL LEVINDO MONT'MOR

**MODERNIZAÇÃO E REESTILIZAÇÃO DA INTERFACE DO USUÁRIO DE UM
SISTEMA PARA SEAPA: UMA ABORDAGEM CENTRADA NO USUÁRIO**

Trabalho de Conclusão de Curso apresentado
ao Curso de Engenharia de Computação do
Centro Federal de Educação Tecnológica de
Minas Gerais, do Campus Nova Gameleira,
como parte do requisito para obtenção do grau
de Engenheiro(a) de Computação.

Aprovado em 19 de junho de 2026.

Banca Examinadora:

Prof. (D.Sc.) Ismael Santana Silva - Orientador
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. (D.Sc.) Edson Marchetti da Silva - Coorientador
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. (D.Sc.) Glívia Angélica Rodrigues Barbosa
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. (D.Sc.) Kecia Aline Marques Ferreira
Centro Federal de Educação Tecnológica de Minas Gerais

Belo Horizonte 2026

(Assinado digitalmente em 25/06/2026 12:35)

EDSON MARCHETTI DA SILVA
PROFESSOR ENS BASICO TECN TECNOLOGICO
DECOM (11.56.03)
Matrícula: 1509224

(Assinado digitalmente em 25/06/2026 14:08)

GLIVIA ANGELICA RODRIGUES BARBOSA
PROFESSOR ENS BASICO TECN TECNOLOGICO
DECOM (11.56.03)
Matrícula: 1058249

(Assinado digitalmente em 25/06/2026 15:38)

ISMAEL SANTANA SILVA
PROFESSOR ENS BASICO TECN TECNOLOGICO
DECOM (11.56.03)
Matrícula: 1028758

(Assinado digitalmente em 24/06/2026 17:32)

KECIA ALINE MARQUES FERREIRA
PROFESSOR ENS BASICO TECN TECNOLOGICO
DECOM (11.56.03)
Matrícula: 1670931

Visualize o documento original em <https://sig.cefetmg.br/public/documentos/index.jsp>
informando seu número: **7**, ano: **2026**, tipo: **FOLHA DE APROVAÇÃO DE TCC**, data de emissão:
19/06/2026 e o código de verificação: **f1e083f078**