



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

SHEILLA DA SILVA

REESTRUTURAÇÃO DA FERRAMENTA DO COMPENSAR

CAMPINA GRANDE - PB

2024

SHEILLA DA SILVA

REESTRUTURAÇÃO DA FERRAMENTA DO COMPENSAR

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 em Ciência da Computação.

Orientadora : Dra. Livia Maria Rodrigues Sampaio Campos

CAMPINA GRANDE - PB

2024

SHEILLA DA SILVA

REESTRUTURAÇÃO DA FERRAMENTA DO COMPENSAR

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 em Ciência da Computação.

BANCA EXAMINADORA:

Nome do Orientador(a)

Orientador – UASC/CEEI/UFCG

Nome do(a) Examinador(a)

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: XX de NOME DO MÊS de 20XX.

CAMPINA GRANDE - PB

RESUMO

O Pensamento Computacional fundamenta-se nas competências adquiridas por meio da Ciência da Computação. Com base nisso, surgiu o ComPensar, uma ferramenta web voltada para o desenvolvimento do pensamento computacional em conjunto com a matemática, direcionada à educação básica. Sua estrutura baseia-se na classificação das questões matemáticas conforme as competências do pensamento computacional que podem ser estimuladas durante o processo de resolução. No entanto, diversos fatores contribuíram para sua transformação em um sistema legado e inoperante. A ausência de contribuições ao longo de mais de 4 anos resultou na obsolescência das tecnologias empregadas tanto no frontend quanto no backend da ferramenta. Além disso, a inatividade do serviço de hospedagem do ComPensar, devido à ausência de requisições por parte dos usuários, culminou na perda do banco de dados devido à falta de consultas. A identificação desses problemas ressaltou a necessidade de uma reengenharia de software em conjunto com a engenharia reversa, visando reconstruir o sistema com tecnologias mais recentes. Esse processo foi dividido em etapas que culminaram no desenvolvimento de uma nova versão do ComPensar, agora disponível para acesso público. Em uma dessas etapas, foram realizadas avaliações para compreender o nível de satisfação do cliente com o software entregue, bem como análises de desempenho e indexação em motores de busca. Os resultados obtidos foram positivos. Na análise de desempenho, o ComPensar recebeu boas avaliações em categorias como acessibilidade, práticas recomendadas e SEO (Search Engine Optimization). Esses resultados contribuíram para a visibilidade da ferramenta nos motores de busca, os quais utilizam dados dessas categorias, como o carregamento de conteúdo e a acessibilidade, na atribuição de pontuação do ComPensar nos motores de busca.

REESTRUTURAÇÃO DA FERRAMENTA DO COMPENSAR

ABSTRACT

Computational Thinking is based on skills acquired through Computer Science. From this foundation emerged ComPensar, a web tool aimed at developing computational thinking alongside mathematics, targeted at basic education. Its structure is based on classifying mathematical questions according to the computational thinking skills that can be stimulated during the problem-solving process. However, various factors led to its transformation into a legacy and inoperative system. The absence of contributions over more than four years resulted in the obsolescence of the technologies used in both the frontend and backend of the tool. Additionally, the inactivity of the ComPensar hosting service, due to the lack of requests from users, led to the loss of the database due to a lack of queries. Identifying these problems highlighted the need for software reengineering along with reverse engineering, aiming to rebuild the system with more recent technologies. This process was divided into stages that culminated in the development of a new version of ComPensar, now available for public access. In one of these stages, evaluations were conducted to understand the level of customer satisfaction with the delivered software, as well as performance and indexing analyses in search engines. The results were positive. In the performance analysis, ComPensar received good ratings in categories such as accessibility, best practices, and SEO (Search Engine Optimization). These results contributed to the tool's visibility in search engines, which use data from these categories, such as content loading and accessibility, in assigning scores to ComPensar in search engines.

REESTRUTURAÇÃO DA FERRAMENTA DO COMPENSAR

Trabalho de Conclusão de Curso

Aluna: Sheilla da Silva

Universidade Federal de Campina Grande Campina
Grande, Paraíba, Brasil

sheilla.silva@ccc.ufcg.edu.br

Orientadora: Lívia Maria Rodrigues

Sampaio Campos

Universidade Federal de Campina Grande Campina
Grande, Paraíba, Brasil

livia@computacao.ufcg.edu.br

RESUMO

O Pensamento Computacional fundamenta-se nas competências adquiridas por meio da Ciência da Computação. Com base nisso, surgiu o ComPensar, uma ferramenta web voltada para o desenvolvimento do pensamento computacional em conjunto com a matemática, direcionada à educação básica. Sua estrutura baseia-se na classificação das questões matemáticas conforme as competências do pensamento computacional que podem ser estimuladas durante o processo de resolução. No entanto, diversos fatores contribuíram para sua transformação em um sistema legado e inoperante. A ausência de contribuições ao longo de mais de 4 anos resultou na obsolescência das tecnologias empregadas tanto no frontend quanto no backend da ferramenta. Além disso, a inatividade do serviço de hospedagem do ComPensar, devido à ausência de requisições por parte dos usuários, culminou na perda do banco de dados devido à falta de consultas. A identificação desses problemas ressaltou a necessidade de uma reengenharia de software em conjunto com a engenharia reversa, visando reconstruir o sistema com tecnologias mais recentes. Esse processo foi dividido em etapas que culminaram no desenvolvimento de uma nova versão do ComPensar, agora disponível para acesso público. Em uma dessas etapas, foram realizadas avaliações para compreender o nível de satisfação do cliente com o software entregue, bem como análises de desempenho e indexação em motores de busca. Os resultados obtidos foram positivos. Na análise de desempenho, o ComPensar recebeu boas avaliações em categorias como acessibilidade, práticas recomendadas e SEO (Search Engine Optimization). Esses resultados contribuíram para a visibilidade da ferramenta nos motores de busca, os quais utilizam dados dessas categorias, como o carregamento de conteúdo e a acessibilidade, na atribuição de pontuação do ComPensar nos motores de busca.

PALAVRAS-CHAVE

pensamento computacional; matemática, educação básica;

REPOSITÓRIOS

<https://github.com/compensar-UFCG/compensar-frontend>

<https://github.com/compensar-UFCG/compensar-backend>

INTRODUÇÃO

O Pensamento Computacional se baseia nas competências adquiridas através da Ciência da Computação, não apenas como ferramenta, mas como uma forma de pensar de maneira organizada e capaz de explorar as potencialidades provenientes das tecnologias da informação e comunicação [1]. A partir de estudos de pesquisa voltados para a integração do pensamento computacional em disciplinas da educação básica, exemplificado pela tese de doutorado *Uma Estratégia Metodológica para Integração do Pensamento Computacional ao Ensino de Matemática* [1], que analisou mais de 300 questões matemáticas em relação às competências do pensamento computacional que elas desenvolviam, surgiu a ferramenta ComPensar [2]. Essa ferramenta foi desenvolvida com o propósito de oferecer um meio para avaliar e disponibilizar questões matemáticas associadas às competências do pensamento computacional.

Problemas do ComPensar

Entretanto, alguns fatores levaram o projeto a se tornar legado e inoperante. Por ser um projeto em um repositório privado e não possuir manutenção ou acompanhamento da saúde do projeto por mais de 4 anos, os módulos e pacotes utilizados ficaram desatualizados e obsoletos.

As tecnologias do front-end encontravam-se com mais de 90% dos pacotes utilizados obsoletos [3][4]. O AngularJS, framework com o qual o ComPensar foi desenvolvido, foi descontinuado e demandava uma migração para a versão mais recente do Angular. Contudo, os problemas não se limitavam a isso. A estrutura do projeto apresentava problemas que comprometiam sua manutenibilidade. Por exemplo, o projeto não fazia uso do Angular CLI, ferramenta

padrão para a execução de projetos em Angular. A equipe anterior optou por utilizar o live-serve, uma biblioteca com 13 dependências e um tamanho considerável, o que tornava o processo de desenvolvimento menos eficiente e mais propenso a falhas.

As tecnologias do backend enfrentavam problemas semelhantes de desatualização e falta de manutenção. O projeto foi desenvolvido em Java, porém, não foi possível encontrar uma versão do Java que não apresentasse conflitos com pacotes importantes do projeto, como o *Lombok*, e os testes unitários não puderam ser executados com sucesso.

O Heroku, serviço de implantação onde o ComPensar estava hospedado, ficou inativo devido à falta de requisições de usuários, o que resultou na perda do banco de dados do MongoDB Atlas. O uso do plano gratuito resultou na interrupção do serviço devido à falta de consultas por mais de 30 dias consecutivos.

A identificação desses problemas destacou a necessidade de uma reengenharia de software, que consiste na análise e alteração de um sistema para reconstituí-lo em uma nova forma e subsequente implementação da nova versão [5]. Para isso, uma das práticas desse processo envolve a engenharia reversa, que é o processo de análise de um sistema para identificar componentes e suas inter-relações, criando representações do sistema em outra forma ou em um nível superior de abstração [5].

METODOLOGIA

Dado isso, foram planejadas etapas para a reengenharia de software do projeto legado. Sendo elas:

Análise e Avaliação inicial: análise dos requisitos funcionais e não-funcionais do sistema atual, coleta dos requisitos do novo sistema e análise de potenciais riscos.

Planejamento: definição de objetivos e escopo, estratégia de modernização, recursos utilizados e construção de um cronograma.

Desenvolvimento e Implantação: modelagem do novo sistema, desenvolvimento dos novos componentes, ferramentas de CI/CD (Continuous Integration e Continuous Deployment), implantação em um servidor e monitoramento do sistema.

Migração: migração dos dados para um novo banco de dados integrado ao novo sistema.

Avaliação: avaliação de satisfação do cliente, análise de desempenho e análise de indexação nos motores de busca.

Análise e Avaliação inicial

Análise dos Requisitos Funcionais e Não-Funcionais do Antigo ComPensar

A ferramenta legada não possuía documentação sobre os requisitos não-funcionais estabelecidos para o sistema. Além disso, apresentava um conjunto de 22 requisitos funcionais, que incluíam: página inicial, funcionalidade de login e autenticação de usuário, operações CRUD (Create, Read, Update, Delete) para usuários, questões, avaliações de questões, listas de questões e download de listas de questões. A análise completa encontra-se no documento *Funcionalidades do Antigo ComPensar* [17].

Requisitos do Novo ComPensar

Os requisitos não-funcionais definidos, foram: desempenho, usabilidade, escalabilidade, manutenibilidade, portabilidade e implantação.

Dado os requisitos funcionais da ferramenta antiga, foi realizado um processo de priorização para amenizar potenciais riscos para o prazo de entrega e ter um foco maior em garantir os requisitos não-funcionais estabelecidos. Sendo assim, foi definido que seriam desenvolvidos os requisitos: página inicial; login e autenticação de usuário; CRUD (Create, Read, Update, Delete) de usuário e questão, lista de questões e download de lista de questões, foram replanejadas para caber no escopo.

Planejamento

Para garantir que a nova ferramenta não reproduzisse os mesmos problemas que levaram às falhas do antigo ComPensar, foi estabelecido um critério fundamental para a seleção das novas tecnologias, além dos requisitos não funcionais. Esse critério consistia em escolher tecnologias que contassem com uma comunidade ativa de desenvolvedores, com uma projeção de interesse dos novos desenvolvedores sobre essas tecnologias.

Sobre o ponto de inoperância do sistema pela falta de atividade dos usuários, seriam aplicadas técnicas de SEO (Search Engine Optimization) para melhorar a indexação do novo ComPensar nos motores de busca. Isso visa, permitir que a nova ferramenta possua mais visibilidade dentre as buscas pelo tema de pensamento computacional.

Dado esses pontos, foi dado início ao planejamento da metodologia de desenvolvimento, definição das tecnologias, padronização de boas práticas para experiência do usuário,

definição de padrões de projeto e a definição do processo de avaliação do ComPensar.

Metodologia de Desenvolvimento

Para assegurar agilidade no desenvolvimento, optamos por adotar o Scrum como metodologia ágil e conduzir sprints curtas de uma semana. Isso nos permite realizar um planejamento mais dinâmico e efetuar ajustes de rota com rapidez ao longo do processo de desenvolvimento.

Definição das Tecnologias

Com base nos requisitos não-funcionais foi realizada uma avaliação sobre as opções disponíveis no mercado atual. Tomando como base os resultados de pesquisas realizadas pelo Stackoverflow no ano de 2023 nas categorias de *Frameworks e Tecnologias mais utilizados* e *Quais as Tecnologias mais trabalhadas vs Quais quero trabalhar* [6].

O Next.js e o Node.js tiveram um destaque em potencial. Dado os resultados, observou-se que o Next.js possui um apoio crescente na comunidade de desenvolvedores, além de possuir suporte em todos os requisitos não-funcionais definidos para a nova ferramenta, como por exemplo: uma boa eficiência e desempenho devido ao Server Side Rendering (SSR) que é a renderização da aplicação do lado do servidor; Altamente escalável, sendo capaz de lidar com grandes volumes de tráfego e demanda de forma mais rápida; É construído com base no React, conhecido por sua simplicidade e clareza de código, facilitando a manutenção do sistema e o tornando altamente portátil; Facilidade de implantação pelos serviços de hospedagem da Vercel, que oferecem integração contínua (CI) e implantação contínua (CD).

O Node.js, não ficou atrás em suas análises. Ele se destaca por sua eficiência devido ao modelo E/S não bloqueante que lhe permite lidar com várias operações de forma simultânea sem bloquear outras tarefas. Sua portabilidade também é uma vantagem, já que é baseado em JavaScript, uma linguagem de programação executada em praticamente qualquer plataforma.

Ambas as tecnologias selecionadas tinham uma projeção de interesse da comunidade de desenvolvedores de forma crescente entre profissionais jovens e os experientes.

Padronização de Boas Práticas para Experiência do Usuário

Um grande desafio foi a busca de uma nova biblioteca para interface do projeto, que nos garantisse alguns pontos de desempenho e usabilidade através de boas práticas em experiência do usuário, como: simplicidade e clareza, consistência dos elementos, feedback visual, facilidade de navegação, design responsivo, hierarquia dos textos, acessibilidade, por isso optamos por seguir com o Material UI [7], que é uma biblioteca open-source de componentes React que é implementada pelo Google Material Design.

Definição de Padrões de Projeto

O desenvolvimento do frontend e do backend seguiu uma abordagem modular e uma arquitetura limpa, incorporando padrões de codificação bem definidos e testes de unidade, com base nos princípios descritos nos livros *Código Limpo* [8] e *Arquitetura Limpa* [9]. Adotamos sistemas de controle de versão conforme as diretrizes do Gitflow [10] e configuramos integração contínua e entrega contínua (CI/CD) em cada repositório. Para a estrutura do projeto frontend, combinamos a organização de pastas do Next.js [11] com os princípios do Atomic Design [12], uma metodologia para organizar componentes de interface. No backend, seguimos o padrão de projeto do Node.js [13], organizando em modelos, rotas, middlewares e testes.

Definição de um Cronograma

Com base na metodologia de desenvolvimento adotada e nos requisitos funcionais e não-funcionais, foi elaborado um backlog de atividades e histórias de usuários. A partir desse backlog, foi estabelecido um cronograma composto por quatro sprints de uma semana cada, no qual foram planejadas as histórias e atividades a serem desenvolvidas em cada sprint, visando garantir o cumprimento do cronograma de entrega.

Migração

O mapeamento das questões matemáticas em relação às competências do pensamento computacional que elas desenvolviam realizadas por COSTA, E. J. F em sua tese de doutorado [1] se encontravam em uma planilha no Excel, possibilitando a recuperação desses dados para transferência a um novo banco de dados.

Avaliação

O processo de avaliação do ComPensar inclui as seguintes etapas:

Avaliação de Satisfação do Cliente: pesquisa de avaliação de satisfação do cliente, com o produto desenvolvido. COSTA, E. J. F. foi o cliente responsável pelo acompanhamento do desenvolvimento da nova ferramenta e por sua avaliação.

Análise de Desempenho: análise de desempenho de páginas web através do PageSpeed Insights [14], uma ferramenta gratuita oferecida pela Google. Essa análise aborda diversos aspectos, como velocidade de carregamento, otimização de recursos, tempo de resposta do servidor e renderização da página. Ela é dividida e pontuada nas seguintes categorias: Desempenho, Acessibilidade, Práticas Recomendadas e SEO.

Análise de Indexação nos Motores de Busca: análise de indexação das páginas nos motores de busca, consiste em verificar como as páginas de um site estão sendo indexadas e exibidas nos resultados de pesquisa. Para esta avaliação, contamos com o uso do Search Console [15], uma ferramenta fornecida pelo Google. Além disso, buscamos a colaboração de usuários para realizar pesquisas específicas e responder a um formulário. Isso contribuiu para análises das palavras-chave e avaliar a apresentação do site em diferentes navegadores.

DESENVOLVIMENTO

Concluída a etapa de planejamento, iniciou-se o processo de reengenharia com a modelagem do novo sistema e desenvolvimento dos novos componentes. Inicialmente, definimos a modelagem do banco de dados e arquitetura do novo ComPensar, após isso demos início a criação de repositórios públicos para dispor os novos projetos e suas configurações. No decorrer do processo de desenvolvimento, foram inseridas algumas etapas de teste e aprimoramentos no processo de CI/CD.

Definição da Modelagem do Banco de Dados

Para a estrutura do banco de dados, decidimos utilizar um modelo relacional devido à identificação de relacionamentos entre entidades, além da necessidade de manter entidades isoladas. Essa escolha nos permite realizar consultas entre entidades e lidar com relações complexas, atendendo às necessidades presentes e futuras da ferramenta. Na fase inicial do desenvolvimento, identificamos três entidades principais: Usuário, Competência e Questão, além de uma

relação entre Questão e Competência [16]. Essa abordagem proporciona uma base sólida para a implementação e expansão futura da ferramenta.

Arquitetura do ComPensar

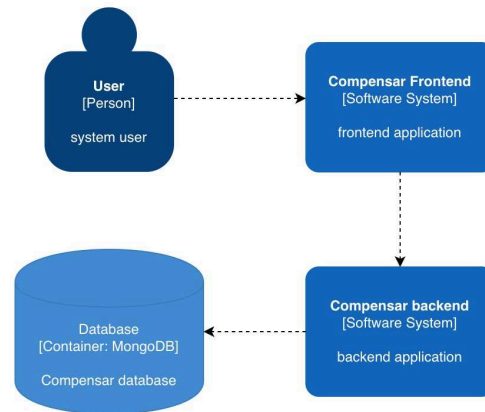


Figura 01: documentação arquitetural do ComPensar

A ilustração acima é uma parte da documentação arquitetural da ferramenta [17], na visão de contexto do C4 Model. Nela, temos a interação do usuário com o frontend, que ocorre por meio de requisições HTTPS, onde o usuário tem acesso às funcionalidades de cadastro, login, visualização de questões, criação de listas de questões e a possibilidade de baixar essas listas em formato de pdf. O usuário pode acessar via web ou em aplicativos móveis.

O frontend se comunica com o backend via API RESTful. O backend oferece get, post, put e delete, para as entidades de usuário, questão e competência do pensamento computacional. Essa API é acessada por meio de requisições HTTPS.

Para armazenar e gerenciar os dados da aplicação, utilizamos o banco de dados MongoDB. O backend se comunica com o MongoDB para realizar operações de persistência e recuperação de dados.

Criação de Novos Repositórios

O projeto anterior do ComPensar, o backend e o frontend compartilhavam o mesmo repositório privado no Bitbucket, vinculado à conta pessoal de um de seus antigos contribuidores. Para melhorar o processo de desenvolvimento, o primeiro passo foi a criação de novos repositórios públicos para armazenar cada lógica de forma isolada. Essa abordagem permitiu o isolamento no processo de deploy dos projetos. Além disso, essa separação proporcionou uma redução significativa no processo de

rollback, caso apenas uma das partes apresentasse alguma instabilidade ou problema.

Criação dos Projetos e Configuração

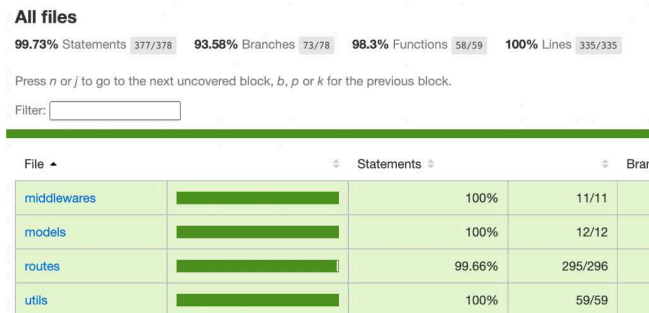
Ambos os projetos foram criados em Typescript (TS). Ele oferece vantagens como: tipagem do código, o que permite detecção de erros em tempo de compilação, melhor suporte à refatoração de código, maior legibilidade e manutenibilidade do código e aumento da segurança. Isso permite alinhar os tipos de dados que serão trafegados entre as comunicações entre o backend e frontend. Após criados os projetos, foram instalados algumas bibliotecas fundamentais para cada escopo e realizadas as devidas configurações. Ambos os projetos dispõem do ESLint, que foi incluído no processo de CI/CD para garantir padrões de projeto, localizar impurezas no código e assim garantir qualidade do código desenvolvido.

Projeto do Backend

Para o desenvolvimento do backend do projeto, aplicamos princípios do SOLID e arquitetura REST em nossos modelos, rotas, utilitários e demais componentes. Essa abordagem resultou em uma arquitetura robusta e flexível, com responsabilidades bem definidas e independentes em cada parte do sistema. As entidades Competência, Questão e Usuário foram tratadas de forma isolada, promovendo coesão e legibilidade.

Adotamos a prática de separação de lógicas entre modelos e rotas, o que garantiu ainda mais clareza e independência nas responsabilidades de cada componente do sistema. Os endpoints da API do Compensar foram projetados com URLs significativas para cada entidade, seguindo os padrões REST para operações de GET, POST, PUT e DELETE. Os recursos são representados em formato JSON, assegurando consistência e acessibilidade dos dados. Além disso, alguns recursos são protegidos por autenticação via token, enquanto a busca por competências do pensamento computacional está disponível livremente, visando facilitar o acesso a informações úteis para outros sistemas e comunidades.

Um destaque importante foi a cobertura de testes de unidade do backend, que superou 99%, como mostra a figura 02. Realizamos testes para validação de APIs, verificação de códigos de status, sanitização de dados recebidos e autenticação de usuários. Foram desenvolvidos 243 testes de unidade, resultando em uma cobertura de 99.73% de statements, 93.58% de branches, 98.3% de funções e 100% de linhas testadas. Esses resultados garantem alta confiabilidade e qualidade do software, promovendo a robustez e eficácia da ferramenta.



File	Statements	Branches	Functions	Lines
middlewares	100%	11/11		
models	100%	12/12		
routes	99.66%	295/296		
utils	100%	59/59		

Figura 02: relatório dos testes unitários

Projeto do Frontend

O desenvolvimento do frontend do projeto utilizou o Next.js, aproveitando a estrutura básica [11] em conjunto com os princípios do Atomic Design [12]. A estrutura do Atomic Design foi adotada para organizar componentes reutilizáveis. Esta técnica permite organizar componentes de forma modular, facilitando a leitura, reutilização e manutenção ao longo do ciclo de vida do projeto. A organização do projeto seguiu a segunda opção proposta pelo Next.js, conforme ilustrado na Figura 03.



Figura 03: estrutura de pastas recomendadas pelo Next.Js

Na estrutura do projeto, os arquivos de rotas foram alocados na pasta "pages", incluindo a pasta "api" destinada à API interna, o que melhora a segurança, pois os endpoints da aplicação não são expostos diretamente ao navegador. Dentro da pasta "app", os elementos globais do projeto foram organizados em subpastas: "components", "contexts", "hooks", "interfaces", "styles" e "utils". Esta organização facilita a manutenção e acessibilidade dos diferentes aspectos do projeto.

Foram implementadas boas práticas de desenvolvimento baseadas nos princípios do SOLID [9], com exceção do Princípio da Substituição de Liskov (LSP), que não se aplicou ao caso específico. Práticas extraídas do Clean Code [8] foram adotadas para melhorar a legibilidade do projeto, incluindo o uso de nomes significativos, boas práticas de escrita de funções, além de boas práticas para comentários, formatação e tratamento de erros.

Esta abordagem garantiu uma arquitetura bem-estruturada, segura e de fácil manutenção, promovendo a robustez e a eficiência do desenvolvimento frontend do projeto.

Principais Funcionalidades Desenvolvidas



Figura 04: página inicial do novo Compensar

Listagem de questões e criação de lista personalizada

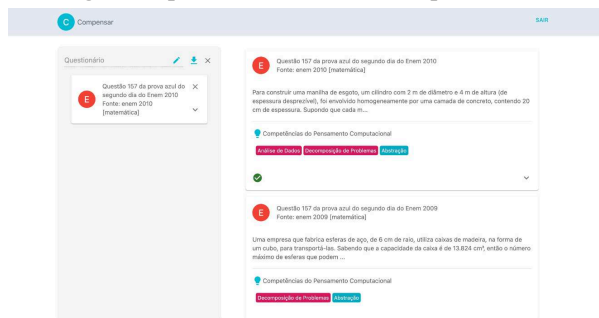


Figura 05: página de listagem de questões do novo Compensar e criação de lista de questões

Detalhes da Questão

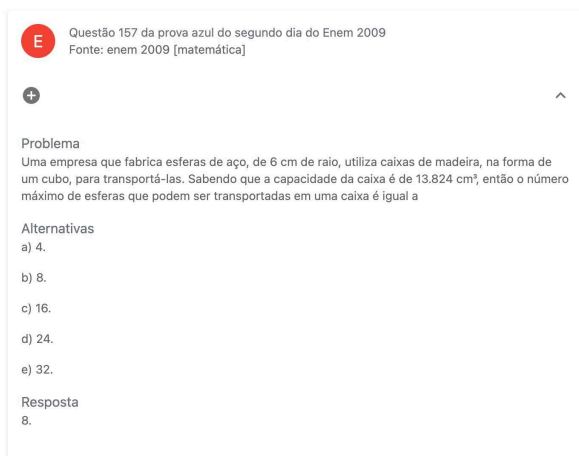


Figura 06: componente de questão do novo Compensar

Implantação dos Projetos

Ambos os projetos foram implantados na Vercel, o que proporcionou benefícios para o processo CI/CD integrado ao GitHub. O fluxo de desenvolvimento foi organizado da seguinte maneira: ao abrir um pull request para a branch principal, o GitHub Actions executa validações de lint e testes de unidade. Em seguida, ele se integra ao serviço da Vercel para realizar o deploy de uma instância do projeto para testes. Isso permite a validação do serviço com testes end-to-end antes de implantá-lo em produção. Cada push na branch principal aciona automaticamente a integração com a Vercel, que realiza o deploy em produção em questão de segundos. Além disso, a Vercel fornece um conjunto de ferramentas para monitorar a saúde da aplicação e logs de erro, garantindo uma supervisão contínua e detalhada do ambiente de produção.

Migração dos Dados

Concluída a etapa de desenvolvimento, foi realizado um processo para disponibilizar os dados sobre as questões matemáticas, juntamente com a avaliação das competências do pensamento computacional, em uma nova base de dados no MongoDB Atlas. Além disso, foi criado um script para popular a base de dados em caso de futuras perdas.

AVALIAÇÃO

Avaliação de Satisfação do Cliente

O projeto foi apresentado a COSTA, E. J. F., seguido pela entrega de um formulário para que ele fornecesse feedback. Este formulário continha nove perguntas obrigatórias, destinadas a avaliar sua satisfação com a qualidade do produto entregue, sua eficiência, usabilidade e considerações gerais.

Com base nas respostas do cliente, registradas no documento de *Análise de Satisfação do Cliente* [17], constatou-se que o índice de satisfação com a entrega foi satisfatório. Em relação à usabilidade e ao layout, obteve-se um retorno mediano, enquanto o desempenho da ferramenta recebeu uma avaliação aceitável. As avaliações sobre a experiência geral de desenvolvimento e planejamento foram igualmente positivas. Portanto, é possível concluir que o trabalho obteve um retorno positivo e dentro das expectativas do cliente.

Análise de Desempenho

O *PageSpeed Insights* [14] realizou uma análise do Compensar para dispositivos móveis e Desktop, onde cada

categoria (desempenho, acessibilidade, práticas recomendadas e SEO) são avaliadas e pontuadas em uma escala de 0 a 100 em cada dispositivo.

Na categoria de *Desempenho* são avaliadas as seguintes métricas: **FCP (First Contentful Paint)**, **SI (Speed Index)**, **LCP (Largest Contentful Paint)**, **TBT (Total Blocking Time)** e **CLS (Cumulative Layout Shift)**. Essas métricas, de forma resumida, avaliam o tempo que o navegador leva para renderização o primeiro conteúdo, a rapidez que esse conteúdo visível é preenchido na tela durante o carregamento, o tempo de carregamento do maior elemento visual, tempo total de bloqueio da página sem receber interações do usuário e por fim, mede a estabilidade visual da página durante seu carregamento.

A categoria de *Acessibilidade* é uma média ponderada de todas as auditorias de acessibilidade [18], dentre elas temos a auditoria de **Os botões não têm um nome acessível**, que é avaliada da seguinte maneira, se alguns botões em uma página tiverem nomes acessíveis, mas outros não, a página receberá um 0 para essa auditoria. Na referência de Pontuação de acessibilidade do Lighthouse [18] é possível ver a lista completa de autoria.

A categoria de *Práticas Recomendadas* avalia se a página segue as práticas recomendadas em páginas web. Ela possui uma lista de auditorias, que vão desde **análise se a página faz uso de HTTPs**, até **se exibe imagem com resolução adequada**. Essa categoria também possui uma lista extensa de auditoria, que é disponibilizada no relatório gerado no documento *Relatório de Análise de Desempenho do ComPensar* [17].

Por fim, temos a categoria de SEO, que realiza verificações para analisar se a página segue as orientações básicas para otimização de mecanismos de pesquisa. Ela também dispõe de uma listagem de auditoria disponibilizada no relatório gerado [17].

De acordo com a avaliação da ferramenta, nosso sistema ficou com notas 100, em quase todas as categorias: *Desempenho*, *Acessibilidade*, *Práticas Recomendadas* e *SEO*. Tivemos notas abaixo de 100, apenas em *Desempenho* para dispositivos móveis, que ficamos com 95 e em *Acessibilidade* para ambos, que ficamos com 91 [17].

As notas positivas nessas categorias impactam diretamente na visibilidade da ferramenta nos motores de busca, uma vez que eles utilizam dados dessas categorias, como carregamento de conteúdo (desempenho), acessibilidade e SEO, na pontuação do score do ComPensar.

Análise de Indexação nos Motores de Busca

Para validar a indexação das páginas do ComPensar nos motores de busca foi utilizada a ferramenta Search Console do Google [15], que realiza diferentes análises nos sites, dentre elas, temos a indexação. Entretanto, para a ferramenta realizar a análise de dados, é preciso ter dados das pesquisas sobre os temas do site nos navegadores de busca.

Sendo assim, elaboramos um questionário que contou com a participação de 20 voluntários para realizar pesquisas em seus navegadores por palavras pré-selecionadas e dispomos de um campo em aberto para eles descreverem outras palavras-chaves que eles fizeram uso.

O intuito era não apenas gerar dados para o Search Console do Google analisar, como também validar se os resultados de pesquisa seriam equivalentes em cada navegador. O formulário [19] foi disponibilizado em algumas redes sociais e ficou disponível por 15 dias. A seguir, temos os resultados do primeiro cruzamento de pesquisas realizadas com a palavra-chave "pensamento computacional compensar". Já podemos descartar que a maioria das pesquisas ocorreu no navegador do Google e que em todas tivemos o retorno da ferramenta compensar nas três primeiras respostas. Os demais navegadores, apesar do baixo número de pessoas, podemos ver que a maioria teve pelo menos um retorno nas três primeiras respostas.

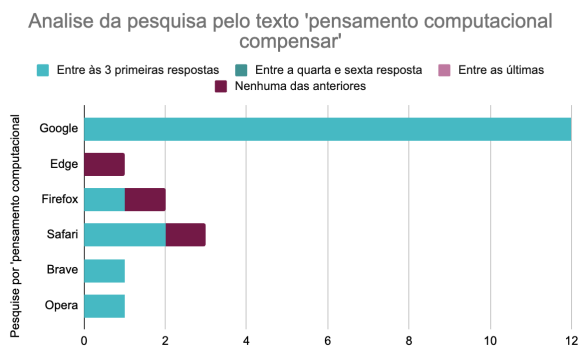


Gráfico 01: análise da pesquisa 'pensamento computacional ComPensar' por navegador

Nas próximas pesquisas, já vemos um impacto diferente. O uso das palavras-chaves "pensamento computacional" e "compensar" não teve muitos retornos positivos. Contudo, podemos ver que tivemos duas pessoas no navegador do Google que conseguiram ter um retorno do nosso site entre as respostas em ambas as palavras. Essas pessoas realizaram pesquisa no penúltimo dia de disponibilidade do formulário, talvez isso tenha sido um retorno satisfatório em relação a nossa pontuação. Todavia, não é possível inferir apenas com esses resultados.

Análise da pesquisa pelo texto 'pensamento computacional'

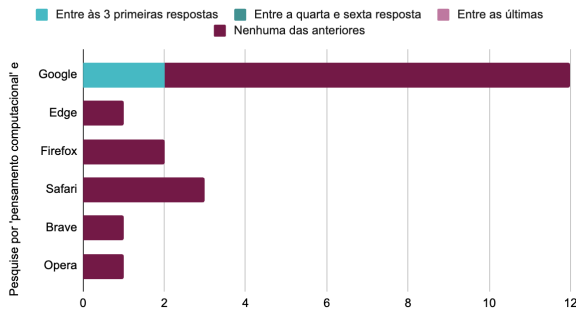


Gráfico 02: análise da pesquisa 'pensamento computacional' por navegador

Análise da pesquisa pelo texto 'compensar'

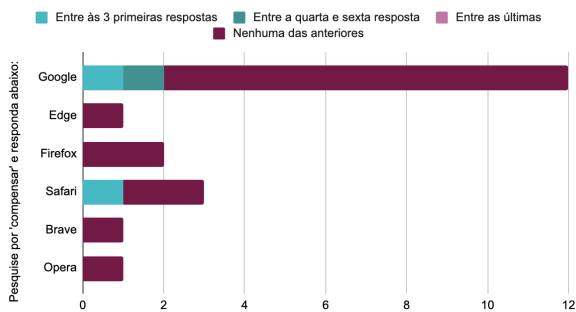


Gráfico 03: análise da pesquisa 'ComPensar' por navegador

Contudo, podemos observar que o Google foi o navegador com os melhores resultados em todo o processo, isso também se dá ao fato do número significativo de respostas ao nosso questionário que fez uso do navegador, totalizando 12 respostas de 20. Considerando os dados do *StatCounter* [20], que oferece estatísticas detalhadas sobre o uso de navegadores com base em dados de tráfego da web global, o Google é o navegador mais utilizado, o que nos sugere que o ComPensar está alcançando resultados promissores com o navegador mais utilizado. Vale ressaltar, que esse dado foi analisado com base na data de criação deste trabalho.

Realizamos um monitoramento completo da ferramenta pela ferramenta Search Console do Google [15], que indicou que o ComPensar ficou entre os cinco principais resultados de pesquisa. Na ferramenta também podemos ver que o título que nos fez exercer essa posição foi o *pensamento computacional compensar*. O relatório completo desta análise, está disponível no documento *Relatório de Indexação* [17].

EXPERIÊNCIAS E LIÇÕES APRENDIDAS

Restaurar o Projeto Antigo ou Criar um Novo?

O planejamento inicial envolvia a implementação de melhorias nos projetos antigos. No entanto, após a identificação dos problemas, essa abordagem se mostrou inviável. Isso ressalta a importância de realizar uma análise prévia da saúde de um projeto de software antes de iniciar qualquer planejamento.

Novo Design x Antigo Design

Devido à necessidade de um processo de desenvolvimento ágil, decidimos adotar o Material UI para garantir agilidade. Essa escolha impactou no layout da ferramenta, o que resultou em alguns pontos de melhoria na experiência apontados pelo cliente.

Mudanças no Decorrer do Processo

Inicialmente, o planejamento previa a utilização do Google Authenticator para o cadastro e avaliação dos usuários na ferramenta do ComPensar. No entanto, devido ao processo de avaliação do Google, que exigia um período de 90 dias para o uso desse recurso, foi necessário reavaliar o processo de login dentro do prazo estipulado. Diante dessa situação, optamos por implementar um acesso mais simplificado para viabilizar a liberação do MVP dentro do prazo previsto. Apesar dos desafios encontrados, o cliente demonstrou compreensão e concordou com as mudanças propostas, destacando a possibilidade de realização dessa melhoria em trabalhos futuros.

TRABALHOS FUTUROS

Desenvolvimento das Funcionalidades Faltantes

Como mencionado anteriormente, extraímos um MVP da antiga ferramenta. Com isso, algumas funcionalidades que existiam não foram implementadas. Então, realizar o desenvolvimento das funcionalidades faltantes é crucial.

Melhorias na Indexação em Outros Navegadores

A discrepância de usuários por navegador sobre a indexação do ComPensar destaca a importância de continuar monitorando e coletando dados sobre o uso da ferramenta em diferentes navegadores. Com um conjunto de dados mais robusto para cada navegador é possível coletar análises mais precisas e tomar medidas para melhorar a indexação do usuário em todos os navegadores suportados.

CONCLUSÃO

O processo de reengenharia do ComPensar atendeu aos requisitos funcionais e não-funcionais definidos na etapa de análise e planejamento inicial. Algumas mudanças de planejamento ocorreram durante o desenvolvimento, mas sem dificuldades, graças ao uso de uma metodologia ágil. Isso permitiu ajustar o processo sem comprometer a qualidade da entrega final ou os prazos estipulados. Na etapa de avaliação, o projeto teve bons retornos, com notas altas na análise de desempenho, que contribuíram na visibilidade da ferramenta nos motores de busca. Dado que, esses motores de busca utilizam dados de categorias como carregamento de conteúdo (desempenho), acessibilidade e SEO para pontuar. Embora a ferramenta tenha obtido mais retornos positivos para apenas uma palavra-chave e uma quantidade maior de usuários para um navegador, esse resultado já é promissor, indicando que a ferramenta recém-lançada já estava alcançando o público e entre as três primeiras respostas de pesquisa, para a palavra-chave *pensamento computacional compensar*. No entanto, há uma quantidade limitada de dados disponíveis para navegadores que não sejam o Google, o que dificulta uma inferência sobre a ferramenta, uma vez que a amostra é muito pequena. Enquanto o Google recebeu contribuições de doze usuários, outros navegadores apresentaram uma mediana de apenas uma resposta, com o Safari sendo o segundo navegador mais respondido, com três respostas. Isso impossibilita determinar se a ferramenta tem um retorno positivo ou negativo para esses navegadores. Todavia, o trabalho realizado possibilitou a disponibilização do ComPensar ao público e agora permitindo a contribuição de desenvolvedores, uma vez que não é mais um projeto privado, dando um potencial de evolução em trabalhos futuros.

REFERÊNCIAS

[1] COSTA, E. J. F. Uma Estratégia Metodológica para Integração do Pensamento Computacional ao Ensino de Matemática. 2022. Tese (Doutorado) - Universidade Federal de Campina Grande, Campina Grande, Paraíba.

- [2] COSTA, E. J. F.; VITORINO, M. G. S. Q.; MEDEIROS, J. M. L.; CAMPELO, C. E. C.; CAMPOS, L. M. R. S. A digital application to assist basic education teachers in the interdisciplinary development of computational thinking skills on the math discipline in Brazilian learning context. In: 13th INTERNATIONAL CONFERENCE ON COMPUTER SUPPORTED EDUCATION, 2021. p. 475-482. DOI: 10.5220/0010452004750482.
- [3] Análise de projeto do backend. Disponível em: https://docs.google.com/document/d/1IkiQi_m1_ocQVe-f_oi_DG5z3IUUpKi3g4V6Dmj9sQLE/edit?usp=drive_link
- [4] Análise de projeto do frontend. Disponível em: https://docs.google.com/document/d/11mFeeKpmF-nwjsxhAhdMs14-KEO5DgMVpTyRJY9O4Wc/edit?usp=drive_link
- [5] RUHL, Melissa; GUNN, M. T. Software Reengineering: A Case Study and Lessons Learned. 1991. Disponível em: <https://doi.org/10.6028/nist.sp.500-193>.
- [6] Stack Overflow Developer Survey 2023. Disponível em: <https://insights.stackoverflow.com/survey/2023>
- [7] Material-UI Documentation. Disponível em: <https://mui.com/>
- [8] Martin, Robert C. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall, 2008.
- [9] Martin, Robert C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2017.
- [10] Gitflow Documentation. Disponível em: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>
- [11] Next.js Documentation. Disponível em: <https://nextjs.org/docs>
- [12] Frost, Brad. Atomic Design. Disponível em: <https://bradfrost.com/blog/post/atomic-web-design/>
- [13] Node.js Documentation. Disponível em: <https://nodejs.org/en/docs/>
- [14] PageSpeed Insights: Ferramenta de análise de desempenho. Disponível em: <https://pagespeed.web.dev>.
- [15] Search Console do Google: Ferramenta de análise de indexação. Disponível em: search.google.com/search-console.
- [16] Modelagem do banco de dados. Disponível em: https://lucid.app/lucidchart/cd31e18b-ebae-4a05-bf9a-8e8423112e00/edit?viewport_loc=112%2C-69%2C2387%2C1308%2C0_0&invitationId=inv_74323efc-341b-45b2-8dd9-c98a7a172877
- [17] Documentação do ComPensar. Disponível em: <https://github.com/compensar-UFPG/documentation>
- [18] Pontuação de acessibilidade do Lighthouse. Disponível em: https://developer.chrome.com/docs/lighthouse/accessibility/scoring?utm_source=lighthouse&utm_medium=lr&hl=pt-br
- [19] Formulário de Pesquisa de Indexação do ComPensar. Disponível em: <https://forms.gle/tM5ztSRqtLkigucc7>
- [20] GS StatCounter. Disponível em: <https://gs.statcounter.com>