

Reformulação e Implementação de Laravel, PostgreSQL e Docker no site “FOCUS”

Keroly Pereira de Andrade¹, Leandro Camara Ledel²,
Guilherme Ramalho Arduini³

¹Instituto Federal de Educação, Ciência e Tecnologia – Campus Hortolândia (IFSP)
Avenida Thereza Ana Cecon Breda, No. 1896, Vila São Pedro, Hortolândia-SP

keroly.andrade@aluno.ifsp.edu.br, ledel@ifsp.edu.br, guilherme.arduini@ifsp.edu.br

Abstract. *This research focuses on the reformulation of the Focus website, which is part of the project "Catholic Congregations, Education, and the National State," aiming to improve its efficiency and usability. The new platform, developed with the Laravel framework and PostgreSQL database, integrated via Docker, provides a more intuitive interface and a robust management system. Migrating to Laravel has resulted in significant improvements in security and performance, while PostgreSQL was key for managing large datasets. These updates include a restructuring of information, smarter search functionality, and advanced filters. Comprehensive testing confirmed the effectiveness of these changes, resulting in a scalable and reliable platform. This study highlights the importance of these improvements to better meet user needs and optimize the management of data.*

Resumo. *Este trabalho apresenta a reformulação do site Focus, parte da pesquisa "Congregações Católicas, Educação e Estado Nacional", com o objetivo de aumentar sua eficiência e usabilidade. A nova plataforma, desenvolvida com o framework Laravel e o banco de dados PostgreSQL, integrados por Docker, oferece uma interface mais intuitiva e um sistema de gerenciamento robusto. A migração para o Laravel trouxe melhorias significativas em segurança e desempenho, enquanto o uso do PostgreSQL foi fundamental para lidar com grandes volumes de dados. As mudanças incluíram uma reestruturação das informações, um sistema de busca mais inteligente e filtros avançados. Testes abrangentes comprovaram a eficácia dessas modificações, resultando em uma plataforma escalável e confiável. O estudo destaca a importância dessas melhorias para melhor atender às necessidades dos usuários e otimizar a gestão de dados.*

1. Introdução

O desenvolvimento de sistemas de software tem evoluído consideravelmente nos últimos anos, impulsionado pela necessidade de entregar aplicações cada vez mais rápidas, escaláveis e eficientes. No entanto, a criação de soluções que atendam às necessidades de negócio e ofereçam uma experiência de usuário satisfatória é um desafio constante. Por trás das tecnologias modernas, como Docker, SQL, PHP, UX e UI, existe uma vasta complexidade, que muitas vezes é subestimada. Estes elementos constituem a "ponta do iceberg" no desenvolvimento de sistemas, onde problemas subjacentes de escalabilidade,

consistência de dados e usabilidade podem emergir quando não são adequadamente abordados.

Docker, por exemplo, revolucionou o processo de desenvolvimento e implantação de aplicações. Ao isolar o ambiente de desenvolvimento em contêineres, Docker garante que o código da aplicação funcione de maneira consistente em qualquer plataforma, eliminando o tradicional "funciona na minha máquina"[Poulton 2021]. Isso é crucial para sistemas que precisam escalar rapidamente ou que dependem de múltiplas equipes de desenvolvimento. No entanto, a introdução de ferramentas de contêinerização, como Docker, traz novos desafios relacionados à orquestração e ao gerenciamento de dependências entre serviços.

Por outro lado, SQL, a linguagem padrão para gerenciamento de bancos de dados relacionais, continua sendo a espinha dorsal de muitas aplicações. Bancos de dados como PostgreSQL e MySQL são amplamente utilizados em sistemas web devido à sua robustez e capacidade de lidar com grandes volumes de dados [Obe and Hsu 2015]. A eficiência na manipulação e consulta de dados é fundamental para o desempenho do sistema, mas requer uma modelagem de dados cuidadosa para evitar problemas como redundância, inconsistência ou baixa performance em consultas.

PHP mantém sua relevância no desenvolvimento web, especialmente quando combinado com frameworks modernos, como o Laravel, que proporciona uma estrutura ágil e produtiva para o desenvolvimento de aplicações complexas [Galvani 2022]. No entanto, a popularidade de PHP também se deve à sua flexibilidade, que, por outro lado, pode levar a problemas de manutenção e legibilidade do código quando não utilizado adequadamente.

Finalmente, a Experiência do Usuário e a Interface do Usuário são componentes cruciais no sucesso de qualquer aplicação. Uma interface mal desenhada ou uma experiência de usuário frustrante pode comprometer o engajamento dos usuários, independentemente da robustez técnica do sistema [Nielsen 1994]. Portanto, garantir uma boa UX/UI desde o início do desenvolvimento é tão importante quanto garantir a escalabilidade e a segurança de um sistema.

A integração eficiente de todas essas tecnologias e conceitos representa o verdadeiro desafio no desenvolvimento moderno. Este trabalho tem como objetivo explorar como Docker, SQL, PHP, UX e UI podem ser utilizados de forma integrada para criar soluções de software escaláveis, eficientes e com alta aceitação por parte dos usuários. Este projeto tem como objetivo estudar e implementar soluções que atendam às necessidades dos pesquisadores que utilizam o site Focus, com ênfase na base de dados do Projeto Temático Congregações Católicas, Educação e Estado Nacional.

2. Referencial Teórico

Esta seção apresenta os conceitos fundamentais para o desenvolvimento do projeto, abrangendo tecnologias e práticas relevantes que fundamentam a arquitetura e a implementação do sistema.

2.1. Engenharia de Software

A engenharia de software é uma disciplina essencial na construção de sistemas complexos e confiáveis, onde a aplicação de métodos, processos e ferramentas é fundamental para

assegurar a qualidade e a manutenibilidade do produto final. A adoção de práticas de engenharia de software possibilita o desenvolvimento estruturado e sistemático de software, englobando desde a elicitação de requisitos até a entrega e manutenção do sistema. Métodos ágeis, como Scrum e Kanban, têm ganhado destaque por promoverem a colaboração contínua e a adaptação rápida às mudanças, resultando em ciclos de desenvolvimento mais curtos e produtos mais alinhados às necessidades dos usuários. Além disso, o uso de padrões de projeto e a implementação de testes automatizados contribuem significativamente para a redução de erros e a garantia de um código mais robusto e sustentável [Sommerville 2011a] [Kniberg 2010a].

2.2. Docker

Docker surgiu como uma ferramenta essencial para o desenvolvimento e distribuição de software, especialmente no contexto de aplicações em larga escala. A tecnologia de contêineres introduzida por Docker permite a criação de ambientes isolados, onde o código da aplicação pode ser executado juntamente com todas as suas dependências [Poulton 2021]. Dessa forma, o uso de contêineres elimina uma das principais causas de falhas em sistemas distribuídos: a incompatibilidade entre ambientes de desenvolvimento e produção [Merkel 2014]. A portabilidade e a eficiência dos contêineres tornam o Docker uma escolha popular entre equipes de DevOps, facilitando a implementação contínua e a escalabilidade [Borkowski 2016].

A modularidade proporcionada pelos contêineres também permite que as organizações adotem uma abordagem de microserviços, onde as aplicações são divididas em componentes menores e independentes. Essa arquitetura não só melhora a manutenibilidade do código, mas também possibilita a atualização de serviços individuais sem afetar todo o sistema [Fowler 2014].

2.3. SQL

A linguagem SQL, desenvolvida por Codd [Codd 1970], continua sendo a principal escolha para gerenciamento de dados em sistemas relacionais. Sua capacidade de realizar consultas complexas e garantir a consistência dos dados é um dos motivos pelos quais grandes empresas continuam a utilizá-la em seus sistemas críticos. Postgres e MySQL, implementações populares de SQL, oferecem suporte avançado a transações e alta disponibilidade, sendo fundamentais no desenvolvimento de sistemas robustos [Murray 2019]. Além disso, a norma SQL foi expandida ao longo dos anos para incluir funcionalidades modernas, como JSON e suporte a dados não relacionais, refletindo a evolução das necessidades de armazenamento de dados [Robinson 2018].

A otimização de consultas SQL é um aspecto crucial na construção de sistemas eficientes, onde técnicas como indexação, normalização e uso de stored procedures podem impactar significativamente o desempenho da aplicação [Date 2004].

2.4. PHP

PHP permanece como uma das linguagens mais usadas para desenvolvimento web, especialmente devido à sua integração com frameworks modernos como Laravel [Galvani 2022]. Laravel, em particular, traz uma série de padrões de projeto que facilitam a organização de código, bem como ferramentas para otimizar o desenvolvimento

de aplicações web dinâmicas e escaláveis. A popularidade do PHP pode ser atribuída à sua simplicidade e ao suporte ativo da comunidade, além de ser uma escolha comum para desenvolvedores iniciantes e profissionais [W3T 2024].

Além do Laravel, o PHP também se destaca por sua capacidade de integração com diferentes bancos de dados, serviços de nuvem e APIs, tornando-o uma opção versátil para diversos tipos de projetos web [Leszczynski 2020]. A evolução do PHP 7 e 8 trouxe melhorias significativas em desempenho e recursos, solidificando ainda mais sua posição no desenvolvimento web [Roush 2021].

2.5. UX e UI

O design de UX e UI é essencial para garantir que as aplicações sejam acessíveis, intuitivas e engajantes. Jakob Nielsen [Nielsen 1994] introduziu princípios fundamentais de usabilidade que continuam a guiar o design de interfaces digitais. As diretrizes de Nielsen, como a consistência e a feedback claro, são práticas essenciais que aumentam a satisfação do usuário e a eficácia da interface.

Combinados com boas práticas de design visual (UI), os sistemas tornam-se não apenas funcionais, mas agradáveis de usar, aumentando significativamente a aceitação dos usuários e, conseqüentemente, o sucesso da aplicação [Garrett 2010]. A importância de testar a usabilidade e a experiência do usuário em diferentes estágios do desenvolvimento é ressaltada por Norman [Norman 2013], que defende que a interação com o usuário deve ser uma prioridade no processo de design.

Além disso, a implementação de princípios de design inclusivo garante que as aplicações sejam acessíveis a uma gama mais ampla de usuários, incluindo aqueles com deficiências [W3C 2021]. Este foco na acessibilidade não apenas amplia o público-alvo, mas também está alinhado com as melhores práticas de desenvolvimento sustentável e responsável.

3. Trabalhos Correlatos

O site **Focus**, da Faculdade de Educação da UNICAMP, serviu como a principal inspiração para o desenvolvimento deste trabalho. Originalmente desenvolvido em Drupal, uma plataforma robusta de gerenciamento de conteúdo, o Focus apresenta escalabilidade e modularidade, características essenciais para sistemas em crescimento [Hoffman 2014]. Recentemente, a equipe responsável pelo site revelou que está em processo de migração para as tecnologias Laravel e PostgreSQL. Essa transição demonstra uma evolução tecnológica significativa, alinhando-se às melhores práticas modernas de desenvolvimento web, como destacam Dorman et al. (2020) sobre a importância da atualização tecnológica para a eficiência organizacional.

Na Figura 1, é possível observar a interface da versão original do site, onde áreas para aprimoramento foram identificadas, especialmente em relação à otimização da interface e à experiência do usuário.

A versão do site Focus em Drupal, embora funcional, possui limitações em termos de experiência do usuário (UX) e interface (UI). Pesquisas indicam que interfaces não intuitivas podem levar a uma maior taxa de rejeição e insatisfação do usuário [Nielsen 1994]. Essa percepção motivou a escolha do Laravel como framework para o



Figura 1. Página principal do Focus

sistema proposto, juntamente com o PostgreSQL, que oferece recursos avançados de integridade de dados e suporte a transações complexas [Gonzalez 2022].

No campo onde se encontra o formulário da página identificado na Figura 2, é possível identificar os seguintes campos de pesquisa: nome da congregação, palavra-chave, família, gênero, país de fundação, países presentes, estados brasileiros onde está presente, ano de fundação e ano de chegada. Além disso, os botões 'buscar' e 'reiniciar'.

A migração para Laravel e PostgreSQL não apenas reforça a relevância dessas tecnologias, mas também responde à necessidade de melhorias em performance e usabilidade. O uso de Laravel, com suas funcionalidades de roteamento e middleware, facilita o desenvolvimento ágil, enquanto o PostgreSQL assegura uma gestão eficiente dos dados [Pos 2024].

Na Figura 3, é apresentado um dos resultados obtidos ao realizar uma pesquisa na página. Como observado, os resultados são apresentados de maneira minuciosamente detalhada, divididos em sessões específicas de cada congregação e as fontes se encontram agrupadas.

Outro exemplo relevante é a Plataforma Lattes, mantida pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). A Lattes é amplamente utilizada para gestão de currículos e produção científica no Brasil, sendo uma ferramenta vital para acadêmicos e instituições. Inicialmente, o sistema foi desenvolvido com tecnologias tradicionais e monolíticas, resultando em problemas de escalabilidade e manutenção à medida que a base de usuários crescia [Lat 2023].

Recentemente, a Plataforma Lattes passou por um processo de modernização, adotando tecnologias como PHP, PostgreSQL e Docker. Essa atualização foi motivada pela necessidade de oferecer uma experiência de usuário mais eficiente, tanto para acadêmicos quanto para administradores do sistema. A nova versão incorpora princípios de Design Responsivo e UX, melhorando a usabilidade em dispositivos móveis e diferentes resoluções de tela, conforme evidenciado por estudos recentes sobre a importância da responsividade em plataformas digitais [Smith 2018].

Nome da congregação

Palavra chave

Família
 Todos ▾

Gênero
 Todos ▾

País - Fundação
 Todos ▾

Países - Presente
 Todos ▾

Estados brasileiros onde está presente
 Todos ▾

Ano - Fundação
 a

Ano - Chegada
 a

Buscar

Reiniciar

Figura 2. Campos de formulário - FOCUS

Banco de Dados

1 Associação da Congregação das IRMÃS MISSIONÁRIAS DO SANTO NOME DE MARIA [MM(4)]
 País de fundação: Alemanha

Siglas: [MM(4)]
 Mantenedor: Não informado

FUNDAÇÃO

Nomes dos Fundadores: Arcebispo Dr. Wil
 Fundadores: 0 Feminino(s); 1 Masculino(s)
 Fundadores com Hierarquia: 0 Feminino(s)
 Fundadores sem Hierarquia: 0 Feminino(s)
 Número de Fundadores: Religiosos: 6; Le
 Santos: 0 Feminino(s); 0 Masculino(s);
 Ano - Aprovação: 1920
 Ano - Fundação: 1920
 Local de Fundação: Alemanha; Enisland;

[1] <http://www.pbmm.com.br/>
 [2] <http://www.kloster-nette.de/>
 [3] <http://www.orden-online.de/wissen/m/missionsschwestern-vom-heiligen-namen-mariens/>
 [4] <http://www.ppe.uem.br/dissertacoes/2013%20-%20Gestaine.pdf>
 [5] https://it.wikipedia.org/wiki/Suore_missionarie_del_Santo_Nome_di_Maria
 [6] <http://www.pbmm.com.br/quem-somos/historico/>
 [7] <http://www.pbmm.com.br/missoes/>
 [8] <http://www.kloster-nette.de/de/apostolat/apostolat-kommunitaeten.html>

[2] [1]
 [1]

► CONGREGAÇÃO NO BRASIL

► CONGREGAÇÃO NO MUNDO

► PUBLICAÇÕES

Figura 3. Resultados do formulário - FOCUS

A transição da Plataforma Lattes mostrada na Figura 4 inspirou diretamente este trabalho, que busca aprimorar não apenas a estrutura de back-end, mas também a interface e a experiência do usuário. A escolha de Laravel e PostgreSQL, juntamente com o uso de containers via Docker, proporciona maior flexibilidade e eficiência no desenvolvimento e manutenção de sistemas em larga escala, conforme discutido por Diogo et al. (2021).



Figura 4. Interface da Plataforma Lattes após a modernização

Esse tipo de modernização, adotado por sistemas como a Lattes, demonstra como a evolução tecnológica é essencial para garantir a longevidade e o desempenho de sistemas críticos, servindo como modelo para a abordagem utilizada neste trabalho. As experiências da Lattes e do Focus destacam a necessidade de adaptação às tecnologias emergentes, que são fundamentais para manter a relevância e a eficiência em um ambiente digital em rápida evolução.

4. Metodologia

A metodologia desta pesquisa foi organizada em diversas etapas para estruturar o processo de desenvolvimento do site. Foram realizados encontros quinzenais, de forma remota, com a equipe de pesquisadores para discutir dificuldades no uso do site antigo e possíveis melhorias. Durante esses encontros, utilizamos ferramentas de videoconferência e colaboração online, como Google Meet, para facilitar a comunicação e o acompanhamento das atividades.

A análise inicial focou no tempo necessário para o desenvolvimento e nas informações das congregações que deveriam ser destacadas no site, aplicando os princípios da Engenharia de Software, como levantamento de requisitos [Sommerville 2011b]. Para isso, foram realizadas entrevistas com usuários finais e stakeholders, coletando dados qualitativos e quantitativos que ajudaram a identificar as principais funcionalidades desejadas e as deficiências do sistema atual.

As entrevistas em grupo realizadas entre o orientador, a orientada e os pesquisadores da base de dados consistiram em perguntas abertas, realizadas quinzenalmente através do Google Meet. Em média, as reuniões duravam de 1 a 2 horas, totalizando aproximadamente 10 reuniões com este grupo específico.

Utilizou-se a metodologia ágil Kanban [Kniberg 2010b] para monitorar o progresso de tarefas. Essa abordagem visual e iterativa garantiu que as etapas de desenvolvimento fossem concluídas com eficiência, com algumas partes envolvidas acompanhando o projeto em tempo real. As tarefas foram organizadas em cartões no quadro Kanban, permitindo que a equipe visse o status das atividades de forma clara e intuitiva.

Para a organização do desenvolvimento e produção, foram definidos internamente os seguintes quadros: 'em planejamento', 'em desenvolvimento', 'em fase de teste' e 'realizado'. Todos os cartões passaram por todas ou quase todas as etapas, garantindo uma evolução estruturada do projeto. Além disso, a utilização desses quadros permitiu monitorar o progresso e identificar rapidamente quaisquer obstáculos que surgissem ao longo do caminho. Dessa forma, é possível ajustar o planejamento e a execução conforme necessário, aumentando a eficiência e a qualidade do trabalho final.

Além disso, foi adotado um ciclo de feedback contínuo, onde cada iteração incluía revisões e testes das funcionalidades implementadas. A interação constante com os usuários permitiu ajustes rápidos e eficazes, garantindo que o produto final atendesse às expectativas e necessidades do público-alvo.

Para assegurar a qualidade do software, implementamos testes automatizados, que foram realizados a cada nova versão do site. Esses testes incluíram tanto testes funcionais, para verificar se cada funcionalidade estava operando corretamente, quanto testes de integração, assegurando que as diferentes partes do sistema funcionassem bem em conjunto. Essa abordagem proativa ajudou a identificar e corrigir bugs antes do lançamento final.

Também foi investido tempo na documentação do processo, incluindo decisões de design, protótipos de interface e manuais de uso, o que facilitou a transferência de conhecimento e a manutenção futura do sistema. A documentação foi mantida em um repositório compartilhado, utilizando o Git, garantindo acessibilidade e facilidade de atualização.

5. Desenvolvimento

5.1. Levantamento de Requisitos

Durante as reuniões com os stakeholders, foram levantados os requisitos funcionais e não funcionais necessários para o novo software. Esses requisitos foram fundamentais para direcionar o desenvolvimento e garantir que as necessidades dos usuários fossem atendidas adequadamente.

Segundo Sommerville (2011), o levantamento de requisitos é um dos passos mais críticos no desenvolvimento de software, pois define as expectativas do cliente e a direção do projeto. Além disso, é essencial que todos os requisitos sejam documentados de forma clara e compreensível, evitando ambiguidades que possam gerar retrabalho.

É importante mencionar que, conforme afirmado por Wiegers e Beatty (2013), a comunicação constante com os stakeholders durante essa fase é vital. Essa interação contínua ajuda a esclarecer as necessidades dos usuários e a ajustar os requisitos conforme necessário. Para facilitar o acompanhamento dos envolvidos e garantir uma comunicação clara sobre o progresso do projeto, foi elaborado um diagrama que ilustra as etapas do desenvolvimento. Esse diagrama serve como um guia visual, permitindo que todos os

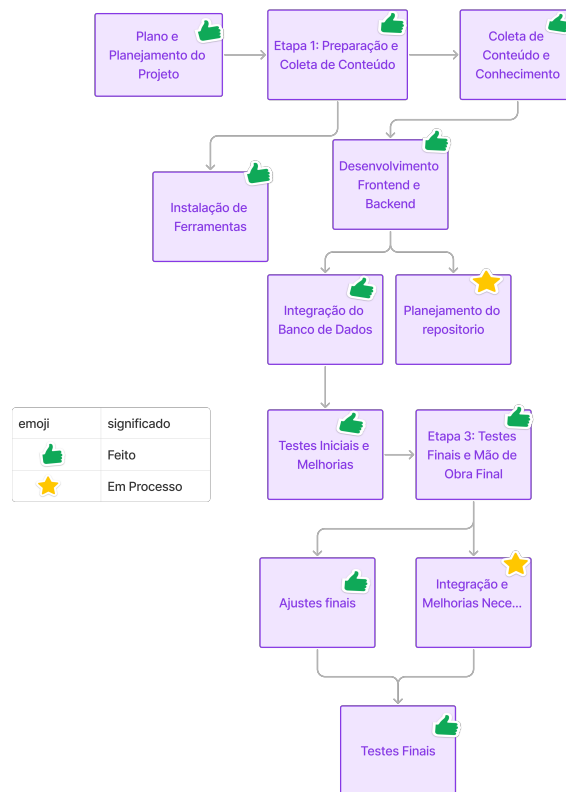


Figura 5. Diagrama de desenvolvimento

participantes do projeto compreendam suas responsabilidades e o status das tarefas em andamento.

A Figura 5 apresenta algumas das etapas enfrentadas no projeto. Algumas fases foram concluídas com sucesso, enquanto outras ainda estão em desenvolvimento.

Durante o processo de desenvolvimento, o diagrama ficou disponível em tempo real, permitindo que o orientador acompanhasse o que já havia sido realizado, o que ainda estava em andamento e o que ainda não havia sido desenvolvido. Essa transparência no fluxo de trabalho facilitou a comunicação e a identificação de possíveis impedimentos, contribuindo para uma gestão mais eficiente do projeto.

Como discutido por Kniberg (2010), a visualização do progresso do desenvolvimento é uma prática recomendada em métodos ágeis, pois promove a transparência e a colaboração entre a equipe. O diagrama em questão possibilitou que as partes interessadas tivessem uma visão geral do status do projeto e facilitou a identificação de áreas que necessitavam de atenção.

Como pode ser observado na Tabela 1, os requisitos funcionais e não funcionais foram discutidos. Até o final deste texto, será evidente que nem tudo foi realizado, pois, devido à limitação de tempo, não foi possível implementar todos os requisitos. No entanto, a maioria deles foi implantada com sucesso.

Requisitos Funcionais	Requisitos Não Funcionais
Divisão de seções claras	Segurança
Destacar informações importantes	LGPD
Organização em tabela ou lista	Manutenção
Exibição de previsões de pesquisa	
Links rápidos para informações específicas (fontes)	
Filtros de pesquisa avançada	
Integração com mapas interativos	
Padronização de dados	
Visualizações gráficas de dados	
Repositório de documentos	

Tabela 1. Requisitos Funcionais e Não Funcionais

5.2. Front-end

A Figura 6 mostra a página inicial da nova plataforma, onde é possível realizar pesquisas interativas sobre as congregações por meio de formulários. No formulário, também é possível limpar a pesquisa e selecionar um intervalo de anos de fundação dessas congregações. Nesta página inicial, é possível navegar por todas as 657 congregações e consultar informações como gênero de fundação, fontes, carisma, chegada ao Brasil, membros e dados gerais, além de todas as fontes mencionadas.

Para garantir que a busca funcione corretamente, algumas funções em JavaScript foram implementadas, como:

A imagem mostra a interface de usuário do novo site. No topo, há um cabeçalho com o texto "Congregações Católicas, Educação e Estado Nacional" e links para "Banco", "Sobre" e "Equipe". Abaixo, o formulário de pesquisa é intitulado "PESQUISA DE CONGREGAÇÕES". Ele contém campos para "Nome da Congregação", "Nomes Alternativos", "Siglas" e "Familia Final". Abaixo do formulário, há uma lista de resultados de busca com três itens:

1. CONGREGAÇÃO BENEDITINA DO BRASIL - OSB - ABADIA DA IMACULADA CONCEIÇÃO DA BEM-AVENTURADA VIRGEM MARIA
Gênero: Não Informado País de Fundação: Brasil
2. CARMELITAS MISSIONÁRIAS TERCEIRAS DESCALÇAS
Gênero: Não Informado País de Fundação: Espanha
3. CÔNEGAS REGULARES DAS CINCO CHAGAS
Gênero: Não Informado País de Fundação: N/E

Figura 6. Página inicial do novo site

- `initializeAutocomplete`: autocompletar no formulário;
- `initializeCache`: limpar o cache;
- `initializeDetailsToggle`: expandir detalhes das congregações nos resultados da busca;
- `initializeFormValidation`: validação do formulário.

Essas implementações contribuem significativamente para a experiência do usuário, pois permitem uma interação mais dinâmica e responsiva com a plataforma. De acordo com Nielsen (1993), a usabilidade é um fator crucial em sistemas interativos, e a

implementação de funcionalidades como autocompletar e validação de formulários ajuda a aumentar a eficiência e a satisfação do usuário. Além disso, conforme a pesquisa de Garrett (2010), um design intuitivo e uma navegação clara são fundamentais para garantir que os usuários consigam acessar as informações desejadas com facilidade.

```
resources > js > JS congregations.js > ready() callback > initializeAutocomplete > source > debounce() callback
6
7   clearTimeout(timeout);
8   timeout = setTimeout(() => func.apply(this, args), delay);
9
10  });
11
12  // Função para inicializar o autocomplete
13  const initializeAutocomplete = () => {
14    $('#nome_congregacao').autocomplete({
15      source: debounce((request, response) => [
16        $.ajax({
17          url: '/congregations/autocomplete',
18          dataType: 'json',
19          data: { query: request.term },
20          success: data => response(data),
21          error: () => console.error('Erro ao buscar sugestões de autocomplete.')
22        }
23      ]), 300),
24      minLength: 2
25    });
26
27    // Função para inicializar o cache de sugestões
28    const initializeCache = () => {
29      const cache = {};
30
31      $('#nome_congregacao').on('input', debounce(function() {
32        const query = $(this).val();
33        if (cache[query]) {
34          $('#suggestions').html(cache[query]).show();
35        } else {
36          $.ajax({
37            url: '/congregations/search-suggestions',
38            dataType: 'json',
39            data: { query },
40            success: data => {
41              const suggestions = data.map(item => `<li>${item}</li>`).join('');
42              $('#suggestions').html(suggestions).show();
43              cache[query] = suggestions;
44            },
45            error: () => console.error('Erro ao buscar sugestões de pesquisa.')
46          });
47        }
48      }), 300));
```

Figura 7. Funções em JavaScript para o formulário de pesquisa

O código apresentado na Figura 7 é um trecho de JavaScript que possui duas funções principais: `initializeAutocomplete` e `initializeCache`. A primeira função é responsável por configurar um sistema de autocompletar em um campo de entrada, enquanto a segunda gerencia um cache para sugestões de pesquisa.

Na função `initializeAutocomplete`, a biblioteca jQuery é utilizada para ativar o autocompletar em um elemento identificado por `home congregacao`. Ela realiza uma chamada AJAX para um endpoint específico, que retorna sugestões com base no texto inserido pelo usuário. O uso da função `debounce` é notável aqui, pois evita chamadas excessivas ao servidor durante a digitação, permitindo que a função AJAX seja chamada apenas após um atraso de 300 milissegundos e quando o usuário tiver digitado pelo menos dois caracteres.

A chamada AJAX usa o método POST para enviar uma solicitação ao URL `congregations autocomplete`, esperando uma resposta no formato JSON. No caso de sucesso,

os dados recebidos são manipulados para serem apresentados; caso contrário, um erro é registrado no console.

A segunda função, `initializeCache`, inicializa um sistema de cache para sugestões de pesquisa. Semelhante à primeira função, ela usa um evento de entrada (`input`) que aciona uma chamada `debounce`. Quando o usuário digita, a função verifica se o termo de pesquisa já está presente no cache. Se estiver, ela exibe as sugestões armazenadas; caso contrário, uma nova chamada AJAX é feita ao endpoint `/congregations/search-suggestions`. Quando os dados são recebidos, eles são processados e armazenados no cache para futuras referências, e se ocorrer um erro, ele também é impresso no console.

Essas funções proporcionam uma experiência de usuário mais rápida e eficiente em termos de comunicação com o servidor, ao mesmo tempo em que gerenciam os dados localmente com o uso de cache.

5.3. Seeder

Para garantir que todas as congregações e as informações desejadas sejam integradas de maneira ágil no Laravel, que é uma plataforma que atua como back-end também, criamos um seeder. Esse arquivo é responsável pela migração para o banco de dados e foi anexado como um arquivo CSV na raiz do código do projeto.

Durante o processo de migração, enfrentamos um grande desafio: alguns campos não estavam reagindo conforme o esperado. Em colunas onde se aguardava números, havia alguns caracteres indesejados, o que resultou em erros, pois o código estava esperando um número inteiro. Portanto, esse processo de migração demorou mais do que o esperado devido à necessidade de uma atenção detalhista para resolver esses problemas.

```
try {
    while (($data = fgetcsv($file, 0, ',')) !== false) {
        $rowNumber++;
        $row = array_combine($header, $data);

        Log::info('Processing row: ' . $rowNumber . ' - Data: ' . json_encode($row));

        // Corrigir valores inválidos para a data
        if (isset($row['Data de fundação'])) {
            if (preg_match('/^\d{4}$/', $row['Data de fundação'])) {
                $row['Data de fundação'] .= '-01-01';
            } elseif (preg_match('/^\d{4}-\d{2}-\d{2}$/', $row['Data de fundação']) === 0) {
                $row['Data de fundação'] = null; // Corrigir para NULL se não for uma data válida
            }
        }

        // Corrigir outros campos inválidos
        foreach ($row as $key => $value) {
            if (trim($value) === '') {
                $row[$key] = null;
            }
        }
    }
}
```

Figura 8. Tratativa dos campos invalidos no seeder

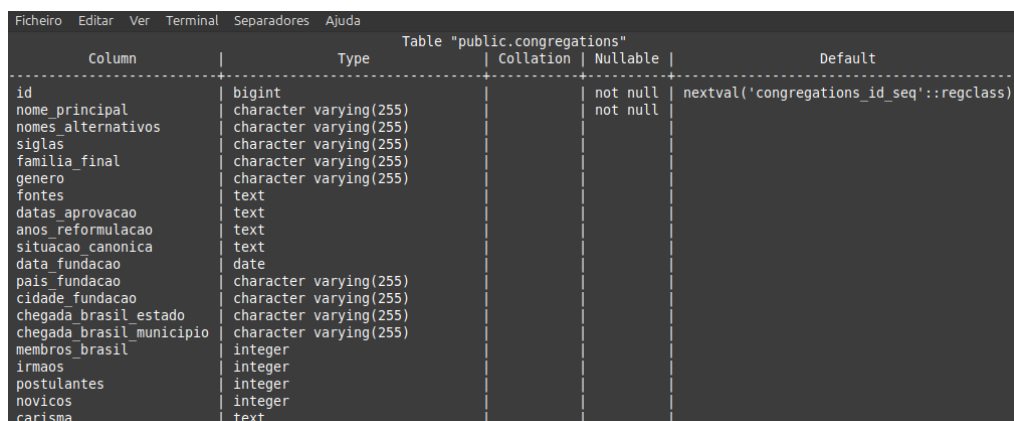
A fim de corrigir esse problema e adiantar o processo, foram desenvolvidas algumas funções onde a data que na tabela original era tratada com o ano foi ajustada para puxar apenas o ano. Os campos que não correspondiam ao que desejávamos foram tratados na função mostrada na Figura 9. Futuramente, foi realizado um estudo de caso mais

detalhado sobre o banco, e, juntamente com um script em Python, propôs-se uma análise mais descritiva sobre a estrutura e a integridade dos dados (Codd, 1970; Date, 2004).

Além disso, a utilização de bancos de dados relacionais como o PostgreSQL, que é integrado ao Laravel, proporciona um gerenciamento eficaz dos dados. Segundo Date (2004), o design do banco de dados é crucial para garantir a integridade e a eficiência das operações de CRUD (Create, Read, Update, Delete), e a utilização de práticas recomendadas, como normalização, contribui para um sistema mais robusto [Codd 1970].

O PostgreSQL, por sua vez, é amplamente reconhecido por suas funcionalidades avançadas e confiabilidade, sendo uma escolha popular para aplicações que requerem robustez e desempenho [Gonzalez 2022]. A integração do Laravel com PostgreSQL também facilita a implementação de migrations, que ajudam a manter a estrutura do banco de dados sob controle e a realizar alterações de forma segura e eficiente [Otwell].

Com isso, a implementação do banco de dados não só viabilizou a estruturação adequada das informações, mas também garantiu que o sistema fosse escalável e mantivesse a performance necessária para atender um volume crescente de dados.



Column	Type	Collation	Nullable	Default
id	bigint		not null	nextval('congregations_id_seq'::regclass)
nome principal	character varying(255)		not null	
nomes alternativos	character varying(255)			
siglas	character varying(255)			
familia_final	character varying(255)			
genero	character varying(255)			
fontes	text			
datas aprovacao	text			
anos reformulacao	text			
situacao canonica	text			
data fundacao	date			
pais fundacao	character varying(255)			
cidade fundacao	character varying(255)			
chegada brasil_estado	character varying(255)			
chegada brasil_municipio	character varying(255)			
membros brasil	integer			
irmaos	integer			
postulantes	integer			
novicos	integer			
carisma	text			

Figura 9. Tabela congregações criada através do seeder

Esta fase do desenvolvimento foi marcada por aprendizados significativos, especialmente no que diz respeito à interação entre front-end e back-end, à comunicação entre a equipe e ao manejo de requisitos. Os desafios enfrentados, como a migração de dados e a implementação de funcionalidades, reforçaram a importância de um planejamento bem estruturado e da colaboração entre os membros da equipe.

Ademais, a documentação contínua do progresso e a adaptação às necessidades emergentes dos usuários foram essenciais para garantir que o produto final atendesse às expectativas e oferecesse uma experiência satisfatória. Com essas bases sólidas, o próximo passo será a fase de testes e validação, onde o software será submetido a rigorosos testes para garantir sua robustez e confiabilidade.

5.4. Containers e serviços

A base da imagem utilizada no Dockerfile é o PHP 8.2 com Apache, uma configuração popular para executar aplicações web [Group b]. Essa imagem serve como um ambiente isolado que contém todas as dependências necessárias para rodar uma aplicação PHP, incluindo extensões específicas como 'pdo' e 'pgsql' para suporte ao banco de dados Post-

```
postgres_db | 2024-10-03 23:40:13.139 UTC [1] LOG: starting PostgreSQL 13.16 (Debian 13.16-1.pgdg120+1)
postgres_db | on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
postgres_db | 2024-10-03 23:40:13.140 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
postgres_db | 2024-10-03 23:40:13.140 UTC [1] LOG: listening on IPv6 address ":::", port 5432
postgres_db | 2024-10-03 23:40:13.162 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
postgres_db | 2024-10-03 23:40:13.202 UTC [25] LOG: database system was shut down at 2024-10-03 23:40:00 UTC
postgres_db | 2024-10-03 23:40:13.286 UTC [1] LOG: database system is ready to accept connections
laravel_app | INFO Nothing to migrate.
laravel_app | INFO Seeding database.
laravel_app | Database\Seeders\CongregationsTableSeeder ..... RUNNING
laravel_app | Database\Seeders\CongregationsTableSeeder ..... 39,341 ms DONE
laravel_app | [Thu Oct 03 23:40:56.709917 2024] [mpm_prefork:notice] [pid 15:tid 15] AH00163: Apache/2.4.62 (Debian) PHP/8.2.24 configured -- resuming normal operations
laravel_app | [Thu Oct 03 23:40:56.710098 2024] [core:notice] [pid 15:tid 15] AH00094: Command line: 'apache2 -D FOREGROUND'
```

Figura 10. Rodando o sedder dentro do container automaticamente

```
root@keroly-275E4E-275E5E:/home/keroly/projeto-de-pesquisa# docker-compose up -d
[+] Running 2/2
 ✓ Container postgres_db Started 12.8s
 ✓ Container laravel_app Started 13.1s
```

Figura 11. Serviços do docker inicializados

greSQL [Group a]. O Dockerfile automatiza a instalação do Composer, gerenciador de dependências do PHP [Potencier], e configura o servidor Apache com uma configuração personalizada. Além disso, ele garante que os arquivos da aplicação sejam copiados para o contêiner, define as permissões adequadas para o funcionamento correto da aplicação e gera uma chave de segurança essencial para aplicações Laravel [Otwell]. Ao final, a imagem resultante proporciona um ambiente pronto para o desenvolvimento e a execução de aplicações PHP de forma consistente e escalável, facilitando o processo de deployment e manutenção [Docker b].

O Docker Compose é uma ferramenta que permite definir e executar aplicações Docker com múltiplos contêineres a partir de um arquivo de configuração, chamado 'docker-compose.yml' [Docker a]. No projeto, estão definidos dois serviços principais: 'app' e 'db'. O serviço 'app' constrói a imagem da aplicação Laravel a partir do Dockerfile especificado e define variáveis de ambiente essenciais, como configurações de ambiente, chave da aplicação e detalhes da conexão com o banco de dados PostgreSQL. Além disso, ele mapeia a porta 80 do contêiner para a porta 8081 do host, permitindo que a aplicação seja acessada através do navegador.

O serviço 'db' utiliza a imagem oficial do PostgreSQL na versão 13 e configura o banco de dados com variáveis de ambiente que definem o nome do banco, usuário e senha, com dados provenientes de variáveis definidas no ambiente do sistema [Group a]. Um volume persistente, chamado 'postgres data', é criado para armazenar os dados do banco de dados, garantindo que eles não sejam perdidos quando o contêiner for reiniciado. Ambos os serviços são conectados à mesma rede chamada 'laravel', permitindo comunicação entre eles. Essa estrutura simplifica o gerenciamento e a execução da aplicação Laravel e do banco de dados PostgreSQL, facilitando o processo de desenvolvimento e deployment.

6. Conclusão

É importante destacar que o projeto ainda está em fase de implementação, portanto, o resultado final ainda não está disponível para a comunidade. No entanto, em breve, o site estará acessível. Algumas partes do projeto, como os mapas e o repositório de imagens e conteúdos valiosos para nosso público-alvo, ainda não foram implementadas. É interessante notar que este projeto é amplo e, devido ao tempo, muitas melhorias ainda podem ser realizadas. Contudo, é preciso lembrar que o processo tem sido desafiador.

Está sendo implementando testes para identificar partes do sistema que ainda não foram programadas. Essa abordagem nos permitirá detectar e corrigir eventuais lacunas, garantindo que todas as funcionalidades planejadas sejam integradas de forma eficaz. A realização de testes sistemáticos é fundamental para assegurar a qualidade do produto final e proporcionar uma experiência de usuário ainda mais robusta e confiável.

A reformulação do site Focus, com a adoção de tecnologias modernas como Laravel, PostgreSQL e Docker, trouxe melhorias substanciais em termos de segurança, desempenho e escalabilidade. A implementação de funcionalidades aprimoradas, como busca inteligente e filtros avançados, não apenas garante uma experiência de usuário mais eficiente, mas também atende melhor às necessidades de pesquisa acadêmica, permitindo que os usuários acessem informações de forma rápida e intuitiva [Nielsen 1993].

Além disso, o uso de contêineres Docker assegurou um ambiente de desenvolvimento e produção estável e de fácil manutenção, facilitando a implementação de atualizações e a gestão de dependências [Docker b] e [Docker a]. Essa abordagem também permite a replicação do ambiente em diferentes plataformas, reduzindo problemas relacionados à configuração [Merkel 2014].

O estudo demonstra a relevância dessas melhorias na disseminação do conhecimento sobre as congregações católicas no Brasil, promovendo um acesso mais fácil e eficiente a informações que podem ser vitais para pesquisadores, acadêmicos e interessados no tema. Como afirma Date (2004), um design de banco de dados adequado é crucial para garantir a integridade e a eficiência das operações, e essas inovações asseguram que o site não só se adapte às demandas atuais, mas também se posicione para futuras expansões e melhorias, assegurando sua relevância no cenário digital [Date 2004].

Além disso, é fundamental ressaltar o papel colaborativo da equipe envolvida no projeto. A diversidade de habilidades e experiências dos membros tem sido um fator crucial para o sucesso da reformulação do site. A troca constante de ideias e feedbacks tem permitido não apenas a identificação de melhorias, mas também a criação de soluções inovadoras que atendem às necessidades do nosso público-alvo. Essa colaboração contínua não só fortalece o desenvolvimento do site, mas também promove um ambiente de aprendizado e crescimento profissional, essencial para enfrentar os desafios que surgem ao longo do processo de implementação.

Referências

- (2021). Accessibility guidelines for web content.
- (2023). Plataforma lattes: História e evolução.
- (2024). Postgresql documentation.

- (2024). Usage statistics and market share of server-side programming languages for web sites. *W3Techs*.
- Borkowski, S. (2016). Scaling applications with docker. *DevOps Magazine*.
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*.
- Date, C. J. (2004). *Database System: A Practical Approach to Design, Implementation, and Management*. Addison-Wesley.
- Docker, I. Define and run multi-container applications with docker. Acessado em: 2024-10-01.
- Docker, I. Get started with docker. Acessado em: 2024-10-01.
- Fowler, M. (2014). *Microservices: A Definition of This New Architectural Term*. Martin Fowler.
- Galvani, F. (2022). *Laravel: The definitive guide*. O'Reilly Media.
- Garrett, J. J. (2010). *The Elements of User Experience: User-Centered Design for the Web*. New Riders.
- Gonzalez, M. (2022). PostgreSQL: Features and capabilities. *Database Journal*.
- Group, P. G. D. PostgreSQL: The world's most advanced open source relational database. Acessado em: 2024-10-01.
- Group, T. P. *PHP: Hypertext Preprocessor*. Acessado em: 2024-10-01.
- Hoffman, M. (2014). Drupal as a content management system for educational institutions. *International Journal of Educational Technology*.
- Kniberg, H. (2010a). *Kanban: Mudando a Cultura de Desenvolvimento de Software*. Lulu Press.
- Kniberg, H. (2010b). *Kanban: Mudando a Cultura de Desenvolvimento de Software*. Lulu Press.
- Leszczynski, A. (2020). *Php: The good parts*. O'Reilly Media.
- Merkel, D. (2014). *Docker: Lightweight Linux Containers for Consistent Development and Deployment*.
- Murray, B. (2019). PostgreSQL performance tuning. O'Reilly Media.
- Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann.
- Nielsen, J. (1994). *Usability engineering*.
- Norman, D. A. (2013). *The Design of Everyday Things*. Basic Books.
- Obe, A. and Hsu, L. (2015). PostgreSQL: Up and running. O'Reilly Media.
- Otwell, T. The php framework for web artisans. Acessado em: 2024-10-01.
- Potencier, F. Dependency manager for php. Acessado em: 2024-10-01.
- Poulton, R. (2021). *Docker: Up & Running*. O'Reilly Media.

- Robinson, J. (2018). The evolution of sql: From relational to nosql. *ACM Computing Surveys*.
- Roush, W. (2021). Php 8: The next generation. *Web Development Magazine*.
- Smith, J. (2018). Responsive design: Principles and best practices. *Web Design Journal*.
- Sommerville, I. (2011a). *Engenharia de Software*. Addison-Wesley.
- Sommerville, I. (2011b). *Engenharia de Software*. Addison-Wesley.

Documento Digitalizado Público

Artigo Final de TCC da Keroly Pereira de Andrade

Assunto: Artigo Final de TCC da Keroly Pereira de Andrade
Assinado por: Leandro Ledel
Tipo do Documento: Estudo Técnico
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- Leandro Camara Ledel, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 06/03/2025 13:59:55.

Este documento foi armazenado no SUAP em 06/03/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1957099

Código de Autenticação: 889bd65f02

