



**INSTITUTO
FEDERAL**
Santa Catarina

**INSTITUTO FEDERAL DE SANTA CATARINA
CURSO TÉCNICO DESENVOLVIMENTO DE SISTEMAS
PROJETO DE SOFTWARE II - PRS301003**

**SISTEMA GERENCIADOR DE POSTAGENS
DOCUMENTAÇÃO, REQUISITOS E MODELAGEM UML**

**AUTOR:
ANDRÉ GUILHERME BARRETO DE FARIAS**

Matrícula: 202111701842
Andre.gbf@aluno.ifsc.edu.br

HISTÓRICO

Data	Descrição	Autor
20 de março de 2024	Criação de modelo original	André Guilherme Barreto de Farias

ÍNDICE

Sumário

Sumário

HISTÓRICO	2
ÍNDICE	3
Sistema de Gerenciamento de Postagens	4
2. Descrição geral.....	6
3. Restrições.....	7
Licença	8
4. Casos de Uso	9
5. Requisitos Funcionais.....	13
6. Requisitos Não-Funcionais	17
7. Plugins Recomendados	19
8. Tecnologias e Recursos Utilizados	23
9. Glossário	24

Sistema de Gerenciamento de Postagens

1. Introdução

Introdução do Sistema Cherry CMS (Content Managment System)

1.1. Objetivo

Este projeto tem como objetivo desenvolver uma aplicação web... As principais metas do sistema incluem:

- Implementar um sistema de gerenciamento de conteúdo eficiente.
- Garantir a segurança e integridade dos dados.
- Prover uma interface intuitiva e amigável para os usuários.
- Facilitar a manutenção e escalabilidade do sistema.

Este projeto será desenvolvido utilizando tecnologias como PHP, MySQL, Laravel, Composer, NodeJS e WampServer.

1.2. Visão Geral

Em termos gerais este documento fornece um roteiro abrangente e detalhado para o desenvolvimento do sistema de avaliação de locais proposto. Está dividido em capítulos, cada um com um propósito específico.

- Capítulo 1: Descrição geral do projeto, abordando sua finalidade, escopo e objetivos.
- Capítulo 2: Características do sistema, destacando suas funcionalidades principais e os atores envolvidos.
- Capítulo 3: Requisitos funcionais e não funcionais do sistema, delineando as necessidades e restrições que devem ser consideradas durante o desenvolvimento.
- Capítulo 4: Casos de uso, descrevendo as interações entre os usuários e o sistema.
- Capítulo 5: Tecnologias e recursos utilizados no desenvolvimento, incluindo linguagens de programação, frameworks e ferramentas necessárias para a implementação bem-sucedida do sistema.

Este documento servirá como um guia fundamental para orientar o desenvolvimento do sistema, garantindo que todos os aspectos importantes sejam considerados e implementados de acordo com as necessidades e requisitos estabelecidos.

2. Descrição geral

2.1. Função do Produto

2.1.1. Descrição: O sistema/produto deverá ser capaz de gerenciar postagens, categorias e banners, permitindo aos usuários criar, visualizar, editar e excluir conteúdo de maneira eficiente.

Os atores envolvidos serão:

- **Usuários:** Pessoas que interagem com o sistema para criar, editar, visualizar e excluir postagens.
- **Postagens:** Conteúdos criados pelos usuários que serão gerenciados pelo sistema.
- **Banners:** Imagens ou anúncios exibidos no site.
- **Categorias:** Classificações das postagens para organização e facilidade de acesso.

2.2. Características

O sistema deverá possuir as seguintes características:

- **Visualização de Informações Básicas:** Usuários podem visualizar informações essenciais das postagens.
- **Listagem:** Exibição de todas as postagens disponíveis no sistema.
- **Edição:** Permitir a modificação dos conteúdos das postagens.
- **Exclusão:** Remover postagens do sistema.
- **Funcionamento Local:** O sistema será executado localmente no Campus do Instituto Federal de Santa Catarina de Florianópolis.

Como se trata de um protótipo educacional, este sistema será local e rodará apenas na máquina local do Campus do Instituto Federal de Santa Catarina de Florianópolis. O sistema funcionará com as linguagens de programação PHP, HTML integrado ao banco de dados MySQL, utilizando o ecossistema Laravel com a pilha TALL Stack (Tailwind CSS, Alpine.js, Laravel, Livewire).

3. Restrições

O projeto está sujeito às seguintes restrições

- 3.1. **Restrições Orçamentárias:** Devido ao caráter acadêmico e avaliativo do projeto, este será executado exclusivamente em ambiente local. A hospedagem em um banco de dados real não será viável. Portanto, o sistema funcionará adequadamente apenas na máquina que estiver executando a aplicação WampServer, simulando uma aplicação web PHP.
- 3.2. **Delimitação de Tempo:** Para garantir a realização das funções propostas, que incluem documentação, prototipagem, diagramação, redação e testes, o autor deste documento, André, optou por descartar funcionalidades mais complexas, como integrações externas, autenticação avançada e funcionalidades de busca sofisticada. Portanto, como desenvolvedor estudante, comprometo-me a implementar apenas as operações básicas CRUD (Create, Read, Update, Delete). Essa decisão visa garantir a conformidade com as convenções e normas estabelecidas. O foco estará em demonstrar domínio e uso correto das ferramentas e tecnologias ensinadas durante o curso, dentro das limitações de tempo e recursos disponíveis aos estudantes no componente curricular PRS II.

Licença

O projeto terá **Licença MIT**.

A Licença MIT é uma licença de software livre que permite a qualquer pessoa utilizar, copiar, modificar, fundir, publicar, distribuir, sublicenciar e/ou vender cópias do software. Ela também permite que qualquer pessoa a quem o software seja fornecido tenha os mesmos direitos, desde que a licença original seja incluída em todas as cópias ou partes substanciais do software.

Para os estudantes do IFSC, a Licença MIT permite que eles utilizem e modifiquem o projeto para seus próprios fins acadêmicos ou pessoais, promovendo a colaboração e o compartilhamento de conhecimento. Isso também significa que outros estudantes podem aprender e contribuir para o projeto, ajudando a melhorar suas habilidades e a criar uma comunidade de desenvolvedores dentro do instituto.

4. Casos de Uso

4.1. UC001 – Autenticação de Usuários

4.1.1. **Ator Principal:** Usuário

4.1.2. **Descrição:** Permitir que os usuários realizem o login no sistema utilizando o Laravel Breeze.

4.1.3. **Pré-condição:** O usuário deve estar registrado no sistema.

4.1.4. **Fluxo Principal:**

4.1.5. O usuário acessa a página de login.

4.1.6. O usuário insere suas credenciais.

4.1.7. O sistema valida as credenciais e, se corretas, concede acesso ao usuário.

4.1.8. O usuário seja redirecionado para o painel principal.

4.1.9. **Pós-condição:** O usuário está autenticado no sistema

4.2. UC002 – Gestão de Postagens

4.2.1. **Ator Principal:** Autor, Administrador

4.2.2. **Descrição:** Permitir que os autores e administradores criem, editem, listem e excluam postagens no blog.

4.2.3. **Fluxo Principal:**

4.2.4. O autor ou administrador acessa a página de gestão de postagens.

4.2.5. O autor ou administrador seleciona a opção de criar postagem.

4.2.6. O autor ou administrador insere as informações da postagem e salva.

4.2.7. **Fluxos Alternativos:**

4.2.8. **Edição de Postagem:** O autor ou administrador seleciona uma postagem existente e a edita.

4.2.9. **Exclusão de Postagem:** O autor ou administrador seleciona uma ou várias postagens existentes e exclui.

4.2.10. **Pós-condição:** A postagem é criada, editada ou excluída no sistema.

4.3. UC003 – Gestão de Categorias de Postagens

4.3.1. **Ator Principal:** Administrador

4.3.2. **Descrição:** Permitir que os administradores criem, editem, listem e excluam categorias de postagens.

4.3.3. **Fluxo Principal:**

4.3.4. O administrador acessa a página de gestão de categorias.

4.3.5. O administrador seleciona a opção de criar categoria.

4.3.6. O administrador insere as informações da categoria e salva.

4.3.7. **Fluxos Alternativos:**

4.3.8. **Edição de Categoria:** O administrador seleciona uma categoria existente e a edita.

4.3.9. **Exclusão de Categoria:** O administrador seleciona uma ou várias categorias existentes e exclui.

4.3.10. **Pós-condição:** A categoria é criada, editada ou excluída no sistema.

4.4. UC004 – Gestão de Banners

4.4.1. **Ator Principal:** Administrador

4.4.2. **Descrição:** Permitir que os administradores criem, editem, listem e excluam banners.

4.4.3. **Fluxo Principal:**

4.4.4. O administrador acessa a página de gestão de banners.

4.4.5. O administrador seleciona a opção de criar banner.

4.4.6. O administrador insere as informações do banner e salva.

4.4.7. **Fluxos Alternativos:**

4.4.8. **Edição de Banner:** O administrador seleciona um banner existente e o edita.

4.4.9. **Exclusão de Banner:** O administrador seleciona um ou vários banners existentes e exclui.

4.4.10. **Pós-condição:** O banner é criado, editado ou excluído no sistema.

4.5. UC005 – Gestão de Categorias de Banners

4.5.1. **Ator Principal:** Administrador

4.5.2. **Descrição:** Permitir que os administradores criem, editem, listem e excluam categorias de banners.

4.5.3. **Fluxo Principal:**

4.5.4. O administrador acessa a página de gestão de categorias de banners.

4.5.5. O administrador seleciona a opção de criar categoria de banner.

4.5.6. O administrador insere as informações da categoria de banner e salva.

4.5.7. **Fluxos Alternativos:**

4.5.8. **Edição de Categoria de Banner:** O administrador seleciona uma categoria de banner existente e a edita.

4.5.9. **Exclusão de Categoria de Banner:** O administrador seleciona uma ou várias categorias de banners existentes e exclui.

4.5.10. **Pós-condição:** A categoria de banner é criada, editada ou excluída no sistema.

4.6. UC006 – Configuração Geral do Site

4.6.1. **Ator Principal:** Administrador, Super-Administrador

4.6.2. **Descrição:** Permitir que os administradores configurem as informações gerais do site.

4.6.3. **Fluxo Principal:**

4.6.4. O administrador acessa a página de configurações do site.

4.6.5. O administrador atualiza as informações gerais do site (nome, status, logotipo, favicon) e salva.

4.6.6. **Pós-condição:** As configurações do site são atualizadas no sistema.

4.7. UC007 – Configuração de Email

4.7.1. **Ator Principal:** Administrador, Super-Administrador

4.7.2. **Descrição:** Permitir que os administradores configurem as informações de email do site.

4.7.3. **Fluxo Principal:**

4.7.4. O administrador acessa a página de configurações de email.

4.7.5. O administrador atualiza as informações de email (driver, host, porta, criptografia, usuário, senha, remetente, nome do remetente) e salva.

4.7.6. **Pós-condição:** As configurações de email são atualizadas no sistema.

4.8. UC008 – Gerenciamento de Exceções e Atividades

4.8.1. **Ator Principal:** Administrador, Super-Administrador

4.8.2. **Descrição:** Permitir que os administradores visualizem e gerenciem exceções e atividades do sistema.

4.8.3. **Fluxo Principal:**

4.8.4. O administrador acessa a página de gerenciamento de exceções.

4.8.5. O administrador visualiza e gerencia as exceções registradas.

4.8.6. O administrador acessa a página de logs de atividades.

4.8.7. O administrador visualiza e gerencia os logs de atividades.

4.8.8. **Pós-condição:** As exceções e atividades são gerenciadas no sistema.

5. Requisitos Funcionais

5.1. RF01 – Gestão de Usuários

5.1.1. **Descrição:** Utilizando o kit Laravel Breeze, os usuários podem se autenticar no sistema. Este kit fornece uma implementação pronta e robusta para autenticação, incluindo registro, login, reset de senha e verificação de email.

5.1.2. **Níveis de Permissão:** O sistema suporta múltiplos níveis de permissão, permitindo a gestão de super-administradores, administradores comuns, autores e qualquer outra função que se deseje cadastrar.

5.1.3. Funcionalidades Incluídas:

- *Login:* Permitir que os usuários façam login utilizando as capacidades de autenticação disponíveis.
- *Logout:* Permitir que os usuários encerrem suas sessões com as funcionalidades de logout disponíveis.
- *Recuperação de Senha:* Fornecer a opção para os usuários redefinirem suas senhas utilizando os recursos de recuperação de senha.
- *Controle de Sessão:* Gerenciar as sessões de usuários com base nas configurações padrão do framework.

5.1.4. **Dependências:** A implementação do requisito funcional RF01 depende da integração correta com o framework e das configurações de segurança e privacidade do sistema. Quaisquer customizações adicionais devem seguir os padrões do framework para garantir a consistência e a segurança.

5.2. RF02 – Gestão de Postagens

5.2.1. **Descrição:** Permitir aos usuários autenticados criar, editar, listar e excluir postagens em blog.

5.2.2. **Funcionalidades:**

- Criação de nova postagem com título, conteúdo e imagem destacada.
- Edição de postagens existentes.
- Listagem de todas as postagens com paginação.
- Exclusão de postagens.

5.3. RF03 – Dashboard de Administração

5.3.1. **Descrição:** O sistema deve fornecer um painel de administração onde os administradores podem visualizar e gerenciar postagens, usuários, categorias, permissões (se for o caso de se tratar do super-administrador). O painel de administração deve oferecer uma visão geral das atividades do site, incluindo estatísticas de postagens, comentários e usuários, além de permitir a gestão completa dos recursos, configurações do site, logotipo, endereço de email.

5.3.2. **Funcionalidades:**

- Listagem e gerenciamento de usuários.
- Listagem e gerenciamento de todas as postagens.
- Listagem e gerenciamento de categorias.

5.4. RF04 – Gestão de Categorias de Postagens

5.4.1. **Descrição:** Permitir aos administradores criar, editar, listar e excluir categorias de postagens.

5.4.2. **Funcionalidades:**

- Criação de nova categoria de postagens.
- Edição de categorias de postagens existentes.
- Listagem de todas as categorias de postagens.
- Exclusão de categorias de postagens.

5.5. RF05 – Gestão de Banners

5.5.1. **Descrição:** Permitir aos administradores criar, editar, listar e excluir banners.

5.5.2. **Funcionalidades:**

- Criação de novo banner com título, descrição, imagem, agendamento e configurações adicionais.
- Edição de banners existentes.
- Listagem de todos os banners com paginação.
- Exclusão de banners.

5.6. RF06 – Gestão de Categorias de Banners

5.6.1. **Descrição:** Permitir aos administradores criar, editar, listar e excluir categorias de banners.

5.6.2. **Funcionalidades:**

- Criação de nova categoria de banners.
- Edição de categorias de banners existentes.
- Listagem de todas as categorias de banners.
- Exclusão de categorias de banners.

5.7. RF07 – Configuração Geral do Site

5.7.1. **Descrição:** Permitir aos administradores configurar as informações gerais do site.

5.7.2. **Funcionalidades:** Atualização do nome do site, status, logotipo e favicon.

5.7.3. **Dependências:** As configurações de segurança e privacidade devem ser configuradas para controlar o acesso dos usuários não autenticados a funcionalidades restritas.

5.8. RF08 – Configuração de Email

5.8.1. **Descrição:** Permitir aos administradores configurar as informações de email do site.

5.8.2. **Funcionalidades:**

- Atualização do driver de email, host, porta, criptografia, usuário, senha, remetente e nome do remetente.
- Dependências: As configurações de segurança e privacidade devem ser configuradas para controlar o acesso dos usuários não autenticados a funcionalidades restritas.

6. Requisitos Não-Funcionais

6.1. Desempenho

6.1.1. Tempo de resposta das páginas inferior a 30 segundos.

6.2. Segurança

6.2.1. Utilizar criptografia SSL/TLS para todas as comunicações.

6.2.2. Armazenar senhas utilizando hashing seguro.

6.2.3. Incluir mecanismos de prevenção contra ataques de força bruta e tentativas de login maliciosas.

6.3. Escalabilidade

6.3.1. Projetado para ser escalável horizontalmente, permitindo a adição de novos servidores conforme necessário.

6.4. Manutenibilidade

6.4.1. Seguir as melhores práticas de desenvolvimento (padrões de design, princípios de código limpo).

6.5. Usabilidade

6.5.1. Interface intuitiva e fácil de usar, seguindo princípios de design centrado no usuário.

6.5.2. Acessível conforme as diretrizes do WCAG 2.1.

6.6. Tradução O sistema deve suportar múltiplos idiomas para garantir que os usuários possam interagir com a aplicação na sua língua preferida. A tradução deve abranger todas as partes da interface do usuário, incluindo formulários, mensagens de erro, e outros textos visíveis.

6.6.1. Acessibilidade: Facilita o acesso ao sistema para usuários que não falam o idioma principal do sistema, promovendo uma experiência inclusiva.

6.6.2. Expansão Global: Permite a expansão para mercados internacionais, atendendo a uma audiência mais ampla e diversificada.

6.6.3. Melhoria da Experiência do Usuário: Oferece uma experiência mais amigável e personalizada, o que pode aumentar a satisfação e a retenção dos usuários.

6.6.4. Compliance: Em alguns casos, pode ser um requisito legal ou regulatório para garantir que a aplicação esteja acessível em diversos idiomas.

6.7. Justificativas

- **Segurança:** A plataforma utiliza hashing seguro para senhas e criptografia para comunicações, além de fornecer proteção contra ataques de força bruta.
- **Manutenibilidade:** O código é desenvolvido seguindo padrões de design e práticas recomendadas, o que facilita a manutenção e evolução do sistema.
- **Desempenho e Escalabilidade:** A arquitetura permite a escalabilidade horizontal e a manutenção de desempenho mesmo sob alta carga.
- **Usabilidade:** As interfaces são projetadas para serem intuitivas e acessíveis, garantindo uma experiência de usuário consistente.

7. Plugins Recomendados

Algumas recomendações de plugins que podem complementar o desenvolvimento e a administração do seu CMS. Estes plugins são altamente recomendados para melhorar a funcionalidade e a experiência do sistema:

- 7.1. **Rupadana - API Resources:** Gere APIs para seus recursos de maneira simples e eficaz.
- 7.2. **Bezhan Salleh - Language Switch:** Um plugin para troca de idiomas com configuração mínima para painéis do Filament.
- 7.3. **Kenepa - Resource Lock:** Bloqueio de recursos quando outro usuário começa a editar um recurso, evitando conflitos de edição.
- 7.4. **Baptiste Bouillot - Jobs Monitor:** Monitore trabalhos em segundo plano para acompanhar o desempenho e o status das tarefas.
- 7.5. **Ralph J. Smit - Components:** Uma coleção de componentes úteis para aprimorar a interface e a funcionalidade do seu sistema.
- 7.6. **Tapp Network - Laravel Auditing:** Plugin para auditoria de recursos, ideal para monitorar e registrar alterações.
- 7.7. **Shuvro Roy - Spatie Laravel Health:** Monitoramento de saúde para o Filament, garantindo que o sistema esteja operando de maneira saudável.

Estes plugins podem proporcionar funcionalidades adicionais e facilitar a gestão e manutenção do seu sistema, contribuindo para um desenvolvimento mais ágil e eficiente.

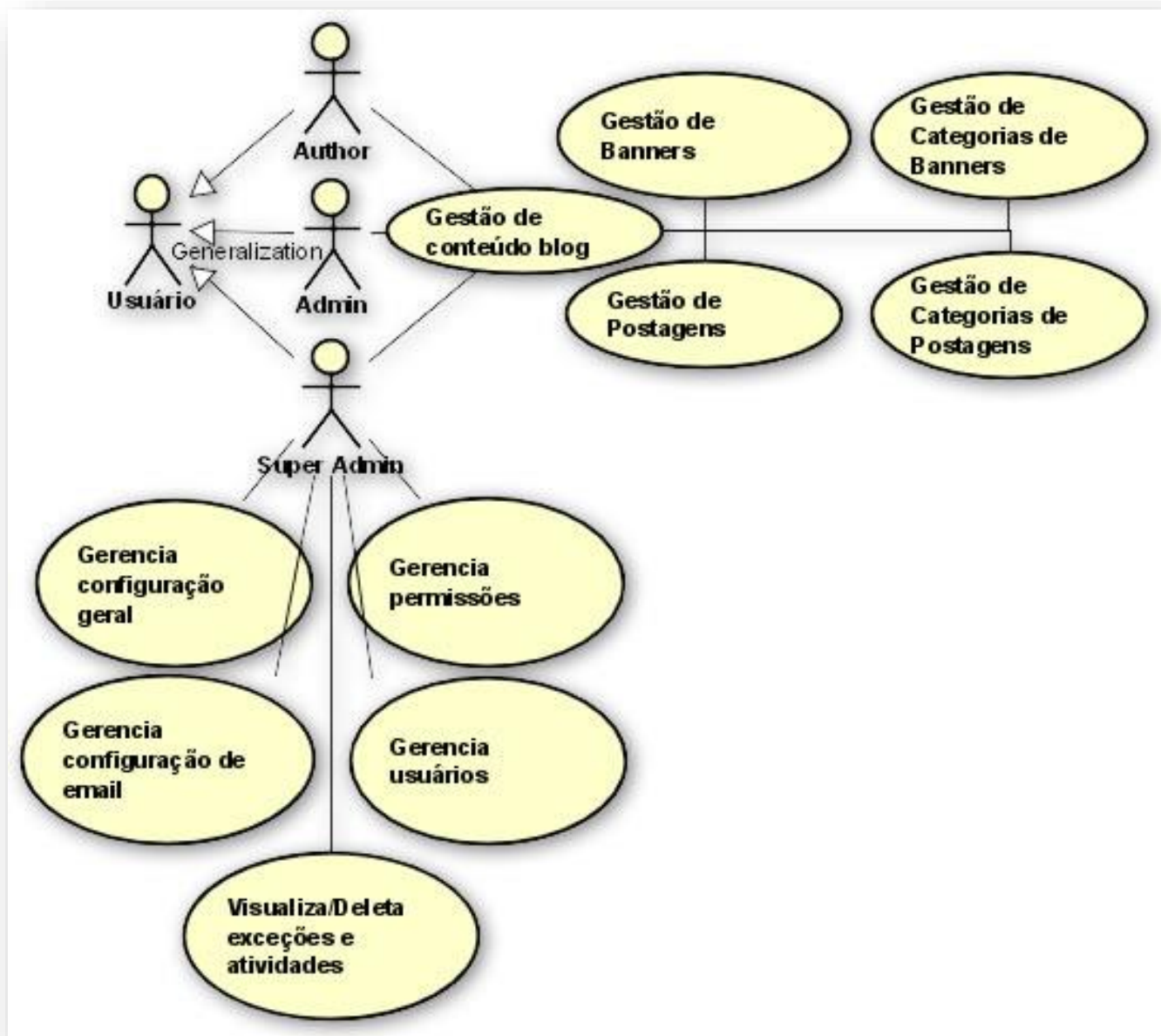


Figura 1: Diagrama de Casos de Uso

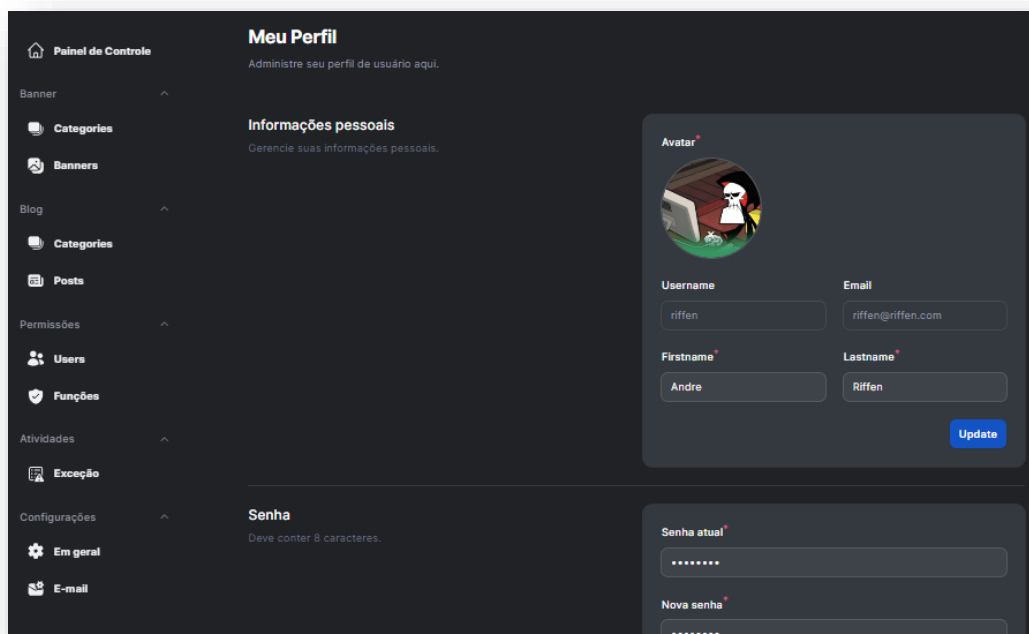


Figura 2: Captura de tela do painel de edição de informações do usuário.

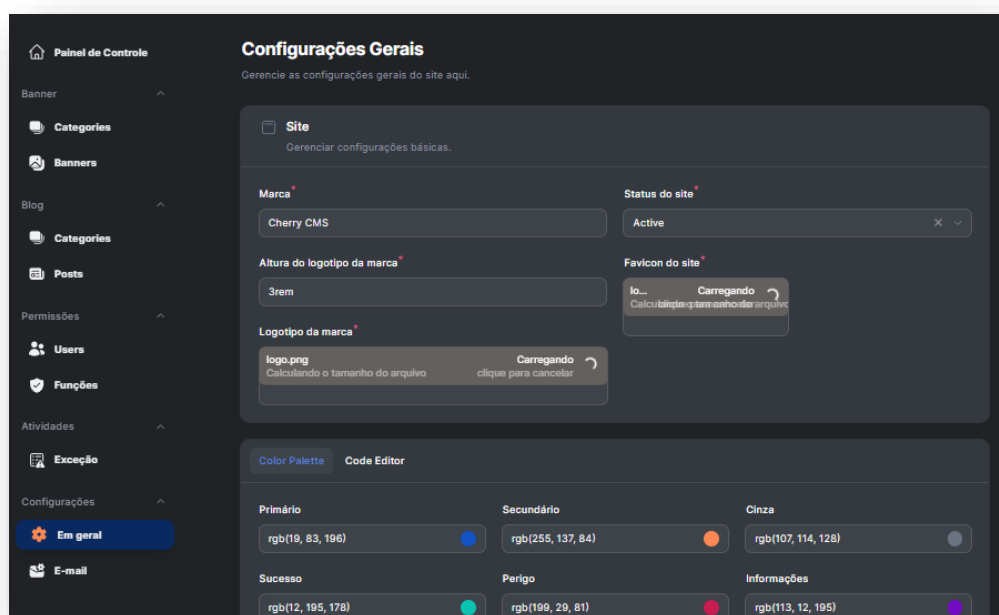


Figura 3: Captura de tela da página de edição de configurações da aplicação.



Figura 4: Estrutura das tabelas do banco de dados.

8. Tecnologias e Recursos Utilizados

Neste tópico, descreveremos os recursos tecnológicos necessários para o desenvolvimento e execução do sistema. Isso incluirá linguagens de programação, frameworks, ferramentas de desenvolvimento e outras tecnologias relevantes.

8.1. Linguagem de Programação:

PHP: Utilizaremos PHP como a linguagem principal para o desenvolvimento do backend do sistema devido à sua ampla adoção e suporte para desenvolvimento web.

Link de referência

<https://www.php.net/docs.php> [↗]

8.2. Banco de Dados:

MySQL: Faremos uso do MySQL como o sistema de gerenciamento de banco de dados (SGBD) para armazenar e gerenciar os dados do sistema.

Link de referência

<https://downloads.mysql.com/docs/refman-4.1-pt.a4.pdf> [↗]

8.3. Framework:

Laravel: Pra garantir a escalabilidade de maneira eficaz, utilizando diversas bibliotecas e dependências incluídas na stack de desenvolvimento.

Link de referência

<https://laravel.com/> [↗]

8.4. Ferramenta de Desenvolvimento:

WampServer: Para simular um ambiente de servidor web local, faremos uso do WampServer, que inclui o Apache, MySQL e PHP, permitindo-nos desenvolver e testar o sistema em nossa máquina local.

Link de referência

<https://www.wampserver.com/en/> [↗]

8.5. Ambiente de Desenvolvimento Integrado (IDE):

PHP Storm: Utilizaremos uma IDE PHP Storm para desenvolver e gerenciar o código-fonte do projeto, proporcionando recursos avançados de edição e depuração.

Link de referência

<https://www.jetbrains.com/phpstorm/> [↗]

9. Glossário

CRUD: Sigla que representa as operações básicas de um sistema de gerenciamento de banco de dados - Create, Read, Update e Delete - usadas para manipular dados.

<https://pt.wikipedia.org/wiki/CRUD>

PHP: Uma linguagem de programação amplamente utilizada para o desenvolvimento de aplicativos web.

<https://pt.wikipedia.org/wiki/PHP>

MySQL: Um sistema de gerenciamento de banco de dados relacional usado para armazenar e gerenciar dados.

<https://pt.wikipedia.org/wiki/MySQL>

IDE: Sigla para Ambiente de Desenvolvimento Integrado, uma ferramenta de software que oferece recursos avançados para desenvolver, depurar e gerenciar código-fonte.

https://pt.wikipedia.org/wiki/Ambiente_de_desenvolvimento_integrado

Framework: Uma estrutura de software que fornece funcionalidades comuns e reutilizáveis para facilitar o desenvolvimento de aplicativos.

<https://pt.wikipedia.org/wiki/Framework>

Front-end: A parte de um aplicativo ou sistema que interage diretamente com os usuários finais, incluindo a interface do usuário e a lógica que controla a apresentação dos dados.

https://en.wikipedia.org/wiki/Front-end_web_development