



UNIVERSIDADE ESTADUAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA – CAMPUS I
CURSO DE COMPUTAÇÃO (BACHARELADO)
COMPONENTE CURRICULAR: PROGRAMAÇÃO WEB
PROF. ANA ISABELLA MUNIZ LEITE

LUCAS FERREIRA MAIA
TÁSSIO SALES DOS SANTOS
ANTÔNIO NERI PEREIRA MONTEIRO VIEIRA DE MELO
DOCUMENTO DE VISÃO - BLOG

CAMPINA GRANDE – PB

16 de maio de 2025

Sumário

Sumário.....	2
1. Introdução.....	3
1.1 Objetivo.....	3
1.2 Necessidade de Negócio.....	3
1.3 Público Alvo.....	3
2. Fluxograma da Aplicação.....	4
3. Requisitos Funcionais.....	4
4. Requisitos Não Funcionais.....	6
5. Perfil do Usuário.....	6
5.1. Usuário Comum.....	6
5.2. Administrador.....	7
6. Tecnologias.....	8
7. Modelagem de Dados.....	9
7.1 Diagrama de Casos de Usos.....	9
Atores.....	10
Casos de Uso do Admin.....	10
Casos de Uso do User.....	10
Relacionamentos entre Casos de Uso.....	10
7.2 Diagrama de Classes.....	11
Classes Principais.....	11
Serviços.....	12
Relacionamentos.....	12
Anexos.....	12

1. Introdução

O sistema é uma plataforma de blog pessoal desenvolvida para permitir a criação, publicação e interação de conteúdos de forma simples e organizada. Usuários poderão se registrar, fazer login e acessar suas contas para publicar posts, ler artigos, curtir e comentar em publicações de outros usuários, além de editar suas próprias postagens e gerenciar seus perfis pessoais.

O sistema contará com um painel administrativo exclusivo para usuários com perfil de administrador, oferecendo funcionalidades avançadas, como gerenciamento de usuários (visualização, edição, exclusão e alteração de permissões) e monitoramento de postagens (moderação de conteúdo, exclusão de posts inadequados e acompanhamento de interações).

1.1 Objetivo

O documento tem por objetivo descrever a situação atual do negócio e fornecer uma visão geral do escopo do produto de software a ser desenvolvido.

1.2 Necessidade de Negócio

O blog pessoal será uma plataforma dedicada à publicação de posts relacionados principalmente à programação, desenvolvimento de software e tecnologia em geral. Seu principal objetivo é compartilhar conhecimentos técnicos, tutoriais, boas práticas, dicas e experiências adquiridas ao longo do tempo, visando contribuir para a comunidade de desenvolvedores e interessados na área. A plataforma deve proporcionar uma experiência intuitiva e organizada tanto para leitores quanto para administradores do blog, garantindo acessibilidade, clareza e segurança dos conteúdos publicados.

1.3 Público Alvo

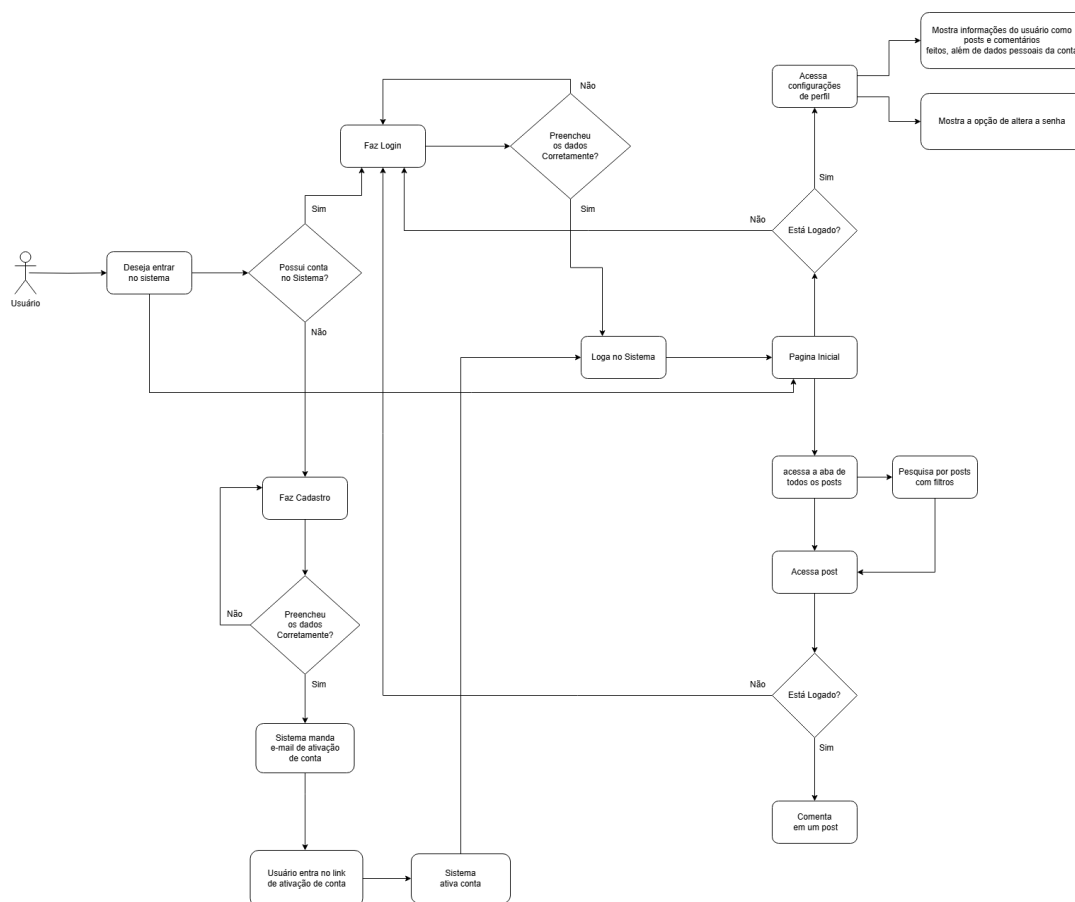
O público-alvo da aplicação inclui:

- **Desenvolvedores iniciantes:** Pessoas que estão aprendendo programação e buscam tutoriais, dicas e explicações claras sobre conceitos técnicos.
- **Desenvolvedores avançados:** Profissionais que desejam compartilhar seu conhecimento, contribuir com a comunidade ou explorar tópicos avançados para ampliar suas habilidades.

- **Entusiastas de tecnologia:** Pessoas interessadas em programação e temas relacionados, mesmo que não atuem profissionalmente na área.
- **Estudantes:** Alunos de cursos técnicos, universitários ou autodidatas que utilizam a plataforma como fonte de aprendizado e referência.

2. Fluxograma da Aplicação

Segue o fluxograma da aplicação mostrando as possíveis interações do usuário



3. Requisitos Funcionais

1. Login:

Permitir que usuários registrados autentiquem suas credenciais (e-mail e senha) para acessar o sistema de forma segura. A funcionalidade deve incluir tratamento adequado para erros de autenticação, como credenciais inválidas, e oferecer a opção de recuperação de senha.

2. **Registro:**

Permitir que novos usuários se cadastrem fornecendo nome, e-mail e senha. O sistema deve validar as informações fornecidas, garantindo que o e-mail seja único e válido. Após o registro, o usuário poderá realizar login imediatamente ou mediante confirmação por e-mail (se aplicável).

3. **Postagens:**

Fornecer um sistema de gerenciamento de postagens onde usuários administradores podem criar, editar, deletar e visualizar posts. Cada postagem deve conter título, conteúdo, autor, data de publicação e categoria. Usuários comuns podem visualizar os posts publicados e interagir com eles por meio de comentários e curtidas, mas não podem modificá-los.

4. **Comentários em Posts:**

Permitir que usuários autenticados publiquem comentários em postagens existentes, promovendo discussões e interações. Os comentários devem incluir autor, data e conteúdo, e ser exibidos em ordem cronológica. Administradores devem ter a capacidade de moderar ou excluir comentários inadequados.

5. **Dashboard de Administração:**

Oferecer uma interface exclusiva para administradores, onde podem monitorar e gerenciar postagens e usuários. As funcionalidades incluem criação, edição e exclusão de posts, além de gerenciamento de usuários (visualização, edição, exclusão e ajuste de permissões). O painel deve fornecer uma visão geral das interações, como número de postagens, comentários e usuários registrados.

6. **Área de Perfil do Usuário:**

Permitir que os usuários autenticados visualizem e editem suas informações pessoais, como nome, e-mail e senha. Deve haver suporte para atualização de dados com validação adequada e feedback ao usuário em caso de sucesso ou falha.

7. **Página de Pesquisa de Posts:**

Fornecer uma página dedicada à busca de postagens, onde os usuários podem encontrar conteúdos específicos por meio de uma barra de pesquisa e aplicar filtros avançados, como data de publicação, categoria e palavras-chave. Os resultados devem ser exibidos de forma organizada e relevante.

4. Requisitos Não Funcionais

1. Segurança

Criptografia de Dados: Todos os dados sensíveis dos usuários, como senhas e informações pessoais, devem ser criptografados antes de serem armazenados no banco de dados. A criptografia deve utilizar algoritmos robustos, como bcrypt, para garantir a integridade e confidencialidade das informações.

Proteção contra Ataques Comuns: Implementar mecanismos de defesa contra vulnerabilidades conhecidas, incluindo:

- **SQL Injection:** Uso de consultas parametrizadas e ORM para evitar a manipulação maliciosa de comandos SQL.
- **Cross-Site Scripting (XSS):** Sanitização de dados inseridos por usuários e uso de cabeçalhos de segurança apropriados.

2. Usabilidade

- O sistema deve ser totalmente responsivo, adaptando-se adequadamente às principais larguras de tela (Mobile, Tablet e Desktop). Deve-se garantir uma experiência de navegação consistente e agradável em diferentes dispositivos.
- Fornecer uma interface limpa e organizada, utilizando componentes visuais bem estruturados e acessíveis. A navegação deve ser fácil de entender, com feedback visual adequado para ações do usuário (como notificações de erro ou sucesso).

5. Perfil do Usuário

5.1. Usuário Comum

Descrição: O Usuário Comum é o tipo de usuário que interage principalmente como leitor e comentador do conteúdo postado no blog. Ele possui acesso limitado às funcionalidades do sistema, focado na navegação e interação com o conteúdo existente.

Tipo: Leitor / Comentador

Responsabilidades e Permissões:

- **Ler Posts:** Pode visualizar e navegar por todos os posts publicados no blog, acessando o conteúdo completo e interagindo com ele através de curtidas e comentários.
- **Comentar em Posts:** Pode publicar comentários em postagens, respondendo a outros usuários ou contribuindo com suas opiniões e discussões.
- **Editar Perfil:** Pode visualizar e atualizar suas próprias informações pessoais, como nome, e-mail e senha, garantindo que seus dados estejam sempre atualizados.
- **Pesquisar Conteúdo:** Pode utilizar a ferramenta de busca para encontrar postagens específicas aplicando filtros como data, categoria e palavras-chave.

5.2. Administrador

Descrição: O Administrador é um usuário com permissões avançadas que permitem o gerenciamento completo do sistema. Além de poder interagir com o conteúdo como um usuário comum, ele também é responsável por supervisionar o funcionamento do blog, criando e moderando postagens e comentários, bem como gerenciando outros usuários.

Tipo: Gestor do Sistema / Leitor / Comentador

Responsabilidades e Permissões:

- **Gerenciamento de postagens:** Pode criar, editar e excluir posts do blog, garantindo que o conteúdo seja atualizado, relevante e organizado.
- **Moderação de Comentários:** Pode monitorar e excluir comentários considerados inadequados ou que violam as diretrizes do sistema.
- **Gerenciamento de Usuários:** Pode visualizar, editar, excluir ou alterar as permissões de outros usuários (como promover ou despromover usuários para administradores).
- **Acesso ao Painel Administrativo:** Possui acesso exclusivo a um painel de administração onde é possível monitorar estatísticas, visualizar relatórios e gerenciar todo o conteúdo do sistema.

- **Interação com o Conteúdo:** Além das permissões administrativas, o administrador pode ler posts, comentar e interagir normalmente com o conteúdo como um usuário comum.

6. Tecnologias

1. HTML, CSS, Bootstrap

Descrição de Uso: Essas tecnologias são usadas para criar a interface do usuário do e-commerce. O HTML fornece a estrutura básica das páginas, o CSS é responsável pela estilização e personalização da aparência, enquanto o Bootstrap é utilizado para garantir responsividade e agilidade no desenvolvimento, oferecendo componentes pré-definidos e estilos prontos para uso.

2. Next.js com TypeScript

Descrição de Uso: Next.js é o framework React utilizado para o desenvolvimento frontend e backend da aplicação. Ele oferece renderização no lado do servidor (SSR) e geração estática (SSG), melhorando a performance e a experiência do usuário. O uso de TypeScript adiciona tipagem estática ao código, ajudando a prevenir erros e facilitando a manutenção do projeto.

3. PostgreSQL com Prisma

Descrição de Uso: O PostgreSQL é o banco de dados relacional escolhido para armazenar informações estruturadas de forma segura e eficiente, incluindo dados de usuários, produtos, pedidos e interações do e-commerce. O Prisma é utilizado como **ORM** (Object-Relational Mapping) para facilitar a integração entre o backend e o banco de dados, oferecendo uma interface intuitiva e tipada para manipulação dos dados com TypeScript.

4. Docker

Descrição de Uso: Docker é utilizado para containerização a aplicação web e o banco de dados PostgreSQL, garantindo consistência no ambiente de desenvolvimento e produção. Com o uso de containers, é possível criar um ambiente padronizado e isolado, simplificando o deploy e a escalabilidade da aplicação.

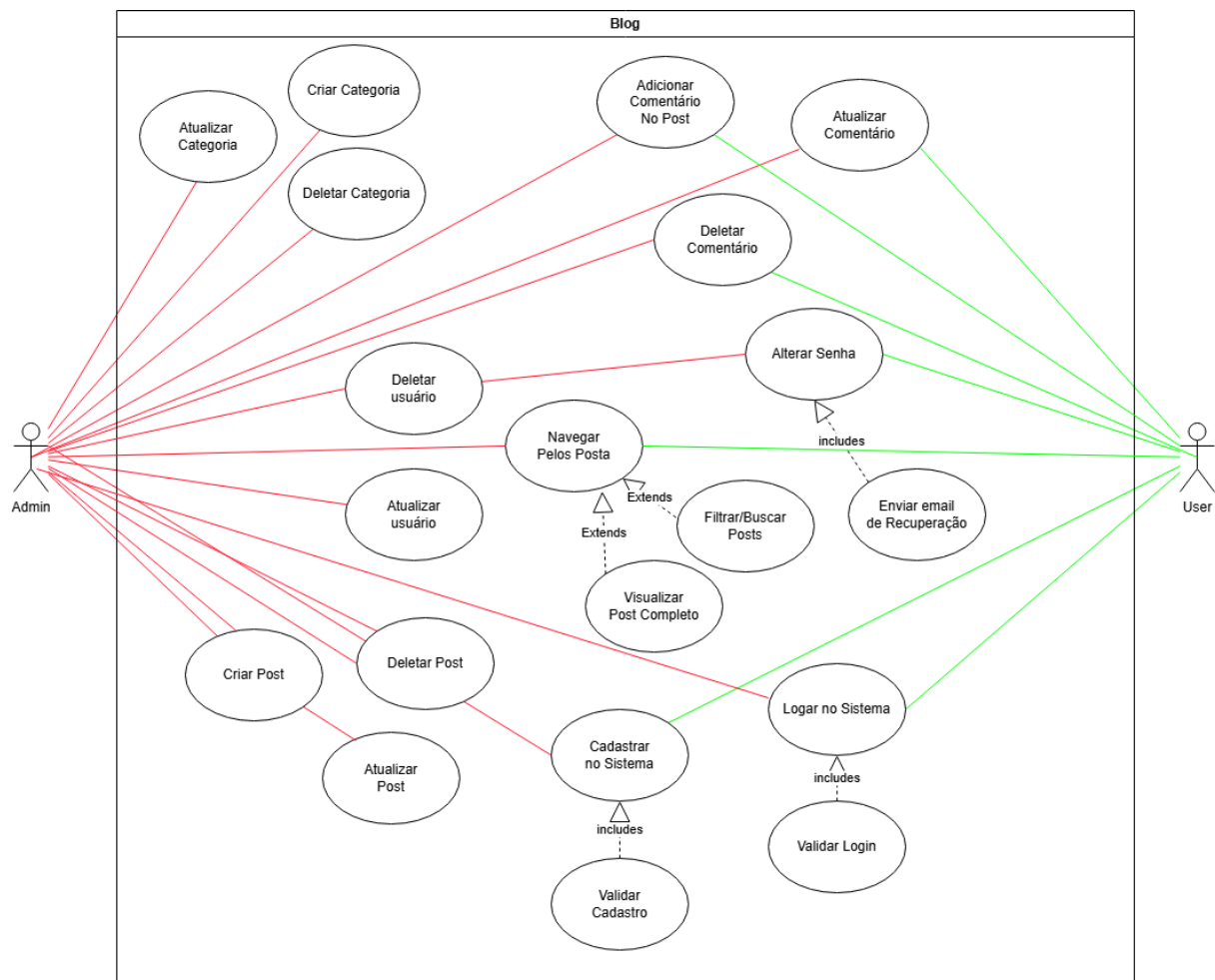
5. Git e GitHub

Descrição de Uso: GitHub é a plataforma utilizada para controle de versão e gerenciamento do código-fonte. Permite colaboração eficiente entre desenvolvedores,

além de possibilitar integração contínua (CI/CD) e fácil rastreamento de alterações no projeto, garantindo organização e segurança durante o desenvolvimento.

7. Modelagem de Dados

7.1 Diagrama de Casos de Usos



O diagrama de casos de uso apresentado representa as funcionalidades principais do sistema de Blog, diferenciando os papéis de **Admin** e **User**, e demonstrando as interações de cada ator com o sistema. O objetivo deste diagrama é oferecer uma visão geral das funcionalidades disponíveis e como os diferentes tipos de usuários podem interagir com o sistema

Atores

- **Admin:** Representa o administrador do sistema, com permissões amplas de gerenciamento de conteúdo e usuários.
- **User:** Representa o usuário comum do blog, com permissões limitadas a navegação, comentários e autenticação.

Casos de Uso do Admin

O administrador possui acesso a funcionalidades avançadas, incluindo:

- **Gerenciamento de Categorias:** Criar, atualizar e deletar categorias de posts.
- **Gerenciamento de Usuários:** Atualizar e deletar usuários cadastrados.
- **Gerenciamento de Posts:** Criar, atualizar e deletar posts.
- **Gerenciamento de Comentários:** Adicionar, atualizar e deletar comentários em posts.
- **Visualização e Navegação:** Navegar pelos posts, visualizar post completo e utilizar filtros de busca.
- **Controle de Acesso:** Cadastrar-se e fazer login no sistema.

Casos de Uso do User

O usuário comum pode realizar as seguintes ações:

- **Interação com Conteúdo:** Navegar pelos posts, visualizar post completo, adicionar e atualizar comentários.
- **Autenticação e Cadastro:** Cadastrar-se no sistema, validar cadastro, fazer login, validar login e alterar senha.
- **Recuperação de Senha:** Enviar e-mail de recuperação de senha.

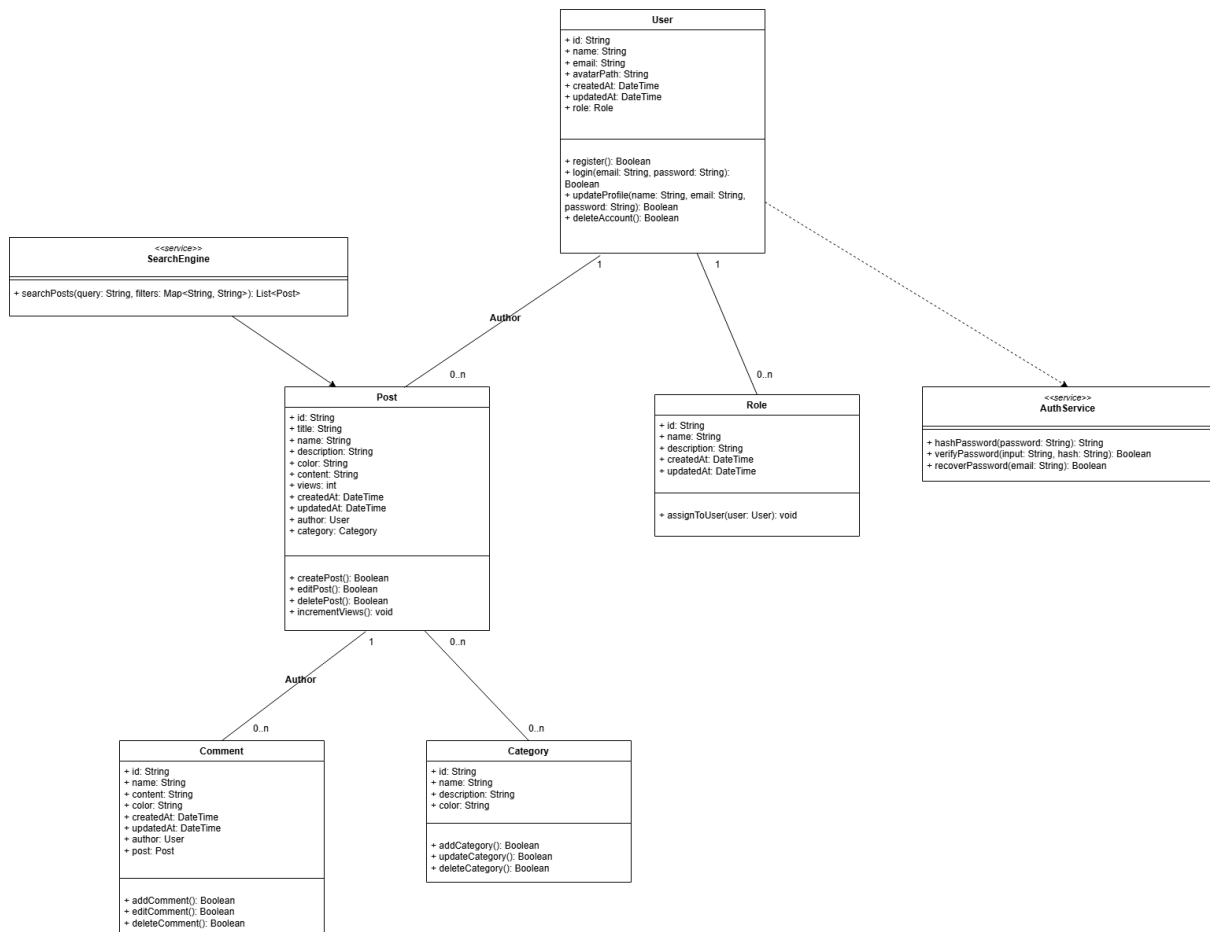
Relacionamentos entre Casos de Uso

- O caso de uso "**Navegar Pelos Posts**" estende o uso de "**Visualizar Post Completo**" e "**Filtrar/Buscar Posts**", indicando que estas ações são opcionais e dependem do contexto de navegação.
- O caso de uso "**Cadastrar no Sistema**" inclui a ação de "**Validar Cadastro**", e "**Logar no Sistema**" inclui "**Validar Login**", caracterizando etapas obrigatórias no

processo de autenticação.

- O caso de uso **"Alterar Senha"** inclui **"Enviar e-mail de Recuperação"**, representando um fluxo complementar para redefinição de senha.

7.2 Diagrama de Classes



O diagrama de classes apresentado modela a estrutura lógica do sistema de Blog, representando as entidades principais, seus atributos, comportamentos (métodos) e os relacionamentos entre elas. Ele serve como base para a definição da estrutura de dados do sistema, orientação da arquitetura e implementação dos componentes de software.

Classes Principais

- **User** Representa o usuário do sistema. Possui atributos como `id`, `name`, `email`, `avatarPath`, datas de criação e atualização, e o papel associado (`role`).
- **Role** Define os papéis dos usuários no sistema, como "Admin" ou "User", com atributos descritivos.

- **Post** Representa uma postagem no blog. Contém título, nome, descrição, cor, conteúdo, número de visualizações e informações de autoria e categoria.
- **Comment** Representa um comentário feito em um post. Possui atributos básicos de identificação, conteúdo e referência ao autor e ao post.
- **Category** Representa a categoria de um post. Possui atributos como nome, descrição e cor.

Serviços

- **AuthService (serviço de autenticação)** Fornece funcionalidades de segurança:
- **SearchEngine (serviço de busca)** Serviço utilitário que oferece suporte à busca de postagens com filtros:
 - `searchPosts(query, filters)`: retorna uma lista de posts de acordo com a consulta e os filtros aplicados.

Relacionamentos

- Um **User** pode ter muitos **Posts** (relação 1 para 0..n).
- Um **User** também pode escrever muitos **Comments** (relação 1 para 0..n).
- Um **Post** pode ter muitos **Comments** e pertence a exatamente uma **Category**.
- Um **User** possui exatamente um **Role**, e um **Role** pode estar associado a vários usuários (relação 1 para 0..n).
- As classes **Post**, **Comment**, **Category** e **User** estão associadas de forma coerente para representar o fluxo natural de criação e interação com o conteúdo.

Anexos

Diagrama de fluxograma da aplicação. Disponível em:
<https://drive.google.com/file/d/1I72bBW8uBSl4v-GTglQwpVFh0HuPS7Xe/view?usp=sharing>

Acesso: 16/05/2025 | (Apenas com e-mail institucional)