



SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PARÁ



## PROGRAMAÇÃO WEB II

**Eudo Tabosa de oliveira**

Trabalho apresentado como avaliação da disciplina PROGRAMAÇÃO WEB II do curso Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Pará. **Prof: Obedio de Sousa Albuquerque**

**Altamira-PA  
2024**

# **Sumário**

## **Backend**

- 1. Banco de Dados: possui três tabelas principais:**
- 2. Models e Relacionamentos**
- 3. Controladores e Rotas**
- 4. Segurança e Autenticação**
- 5. Validação e Tratamento de Erros**

## **Frontend**

- 1. Design e Estrutura HTML/CSS**
- 2. Interatividade e Navegação**
- 3. Mensagens e Alertas**
- 4. Fluxo de Usuário**

## BlogTads: Visão Geral

O sistema desenvolvido é um blog com funcionalidades de cadastro, login, criação e gestão de publicações e comentários. Cada usuário pode fazer CRUD nas próprias publicações, enquanto todos os usuários podem ver, comentar e interagir nas publicações de outros. A interface foi construída para ser intuitiva e amigável.

### Backend

#### 1. Banco de Dados: possui três tabelas principais:

- **Users:** Armazena os dados dos usuários, incluindo email e senha criptografada.
- **Articles:** Armazena as publicações criadas pelos usuários, com referências aos usuários que as criaram.
- **Comments:** Armazena os comentários feitos pelos usuários nas publicações, incluindo o ID do artigo e do usuário.

#### 2. Models e Relacionamentos

- **User:** O modelo User tem um relacionamento has\_many com articles e comments. O has\_secure\_password é usado para autenticação, garantindo que as senhas sejam armazenadas de forma segura.
- **Article:** Cada artigo pertence a um usuário (belongs\_to :user) e possui vários comentários (has\_many :comments). Isso permite que apenas o autor do artigo possa editar ou excluir o conteúdo.
- **Comment:** Cada comentário pertence a um artigo (belongs\_to :article) e a um usuário (belongs\_to :user). Esse relacionamento permite identificar o autor de cada comentário e o artigo correspondente.

#### 3. Controladores e Rotas

- **SessionsController:** Gerencia as sessões de usuário, incluindo login e logout.
- **UsersController:** Controla o cadastro de novos usuários. Após o cadastro, o usuário é redirecionado para a tela de login.
- **ArticlesController:** Controla o CRUD das publicações. O autor pode criar, editar e excluir apenas as próprias publicações.

- **CommentsController:** Controla a criação e exclusão de comentários. Cada usuário pode comentar nas publicações de outros, mas apenas o autor do artigo e do comentário têm permissão para excluir um comentário.

#### 4. Segurança e Autenticação

- **Autenticação:** O sistema utiliza `has_secure_password` no modelo `User` para autenticar as credenciais dos usuários.
- **Autorização:** Há verificações em cada controlador para garantir que apenas os usuários autorizados possam editar ou excluir conteúdo. Por exemplo, apenas o autor de um comentário ou do artigo pode excluir um comentário específico.

#### 5. Validação e Tratamento de Erros

- O backend inclui validações no modelo `User` para garantir que o email seja único e que a senha tenha pelo menos 4 caracteres. Em caso de erro, o sistema retorna mensagens de erro específicas para ajudar o usuário a corrigir a entrada.

### Frontend: Interface do Usuário e Interatividade

#### 1. Design e Estrutura HTML/CSS

- **Layout Responsivo:** O frontend foi estilizado para ser amigável, utilizando CSS para definir layouts e estilos que facilitam a navegação.
- **Formulários de Cadastro e Login:** Formulários construídos com `form_with` do Rails, onde mensagens de erro são exibidas junto aos campos.
- **CRUD da Publicação e Comentários:** Formulários e botões de ação para criação, edição e exclusão estão organizados em layouts simples. Cada comentário exibe o email do usuário que o fez, o conteúdo e a data de criação.

#### 2. Interatividade e Navegação

- **JavaScript Modularizado:** Para carregar funcionalidades específicas como confirmação de exclusão e algumas validações do lado do cliente.

- **Condições de Exibição:** Elementos como botões de edição e exclusão são exibidos apenas para usuários autorizados por exemplo, o autor de uma publicação ou comentário.
- **Validação do Lado do Cliente:** Antes de enviar o formulário, é necessário validar se a senha possui o número mínimo de caracteres exigidos e exibir mensagens para orientar o usuário

### 3. Mensagens e Alertas

- Mensagens de sucesso ou erro são exibidas para o usuário, informando sobre o resultado das ações, como a adição ou remoção de um comentário.
- As mensagens foram adicionadas de forma a não interromper a experiência do usuário, mantendo a usabilidade da interface.

### 4. Fluxo de Usuário

1. **Cadastro e Login:** Um novo usuário pode se cadastrar fornecendo email e senha. Após o cadastro, é redirecionado para a tela de login.
2. **Publicações:** Após o login, o usuário pode ver todas as publicações, criar novas, editar ou excluir as próprias.
3. **Comentários:** O usuário pode comentar em publicações de outros.
4. Método não implementado (excluir comentários próprios e, caso seja o autor de uma publicação, excluir qualquer comentário feito nela.)