





Sistema de Blog Completo







Um sistema completo de blog com painel administrativo e integração com Instagram, desenvolvido com React e Node.js.

Funcionalidades




Blog Principal

-  Visualização pública de posts
-  Sistema de comentários para visitantes
-  Design responsivo e moderno
-  Suporte a imagens, vídeos e áudios

Painel Administrativo

-  Login seguro com autenticação JWT
-  Editor rico para criação/edição de posts
-  Upload de mídia (imagens, vídeos, áudios até 50MB)
-  Sistema de moderação de comentários
-  Gerenciamento de status (rascunho/publicado)
-  Dashboard com estatísticas

Página Instagram

-  Interface preparada para posts do Instagram
-  Integração via API (configurável)
-  Design consistente com o blog

Tecnologias Utilizadas

Frontend

- **React 18** - Interface de usuário
- **React Router** - Navegação
- **Tailwind CSS** - Estilização
- **Shadcn/UI** - Componentes

- **Lucide React** - Ícones
- **Axios** - Requisições HTTP

Backend

- **Node.js** - Runtime
- **Express** - Framework web
- **SQLite** - Banco de dados
- **JWT** - Autenticação
- **Multer** - Upload de arquivos
- **bcrypt** - Hash de senhas
- **CORS** - Cross-origin requests

Instalação e Configuração

1. Instalar Dependências

Frontend:

Bash

```
cd blog-system  
npm install
```

Backend:

Bash

```
cd blog-system/server  
npm install
```

2. Configurar Variáveis de Ambiente

Crie um arquivo `.env` na raiz do projeto:

Plain Text

```
VITE_API_URL=http://localhost:3001/api
```

Para integração com Instagram, configure no servidor:

Plain Text

```
INSTAGRAM_CLIENT_ID=seu_client_id  
INSTAGRAM_ACCESS_TOKEN=seu_access_token
```

3. Executar o Sistema

Backend (Terminal 1):

Bash

```
cd blog-system/server  
npm start
```

Frontend (Terminal 2):

Bash

```
cd blog-system  
npm run dev
```

O sistema estará disponível em:

- Frontend: <http://localhost:5173>
- Backend API: <http://localhost:3001/api>



Credenciais Padrão

Administrador:

- Usuário: admin
- Senha: admin123
- Email: admin@blog.com



Como Usar

Para Administradores

1. Login:

- Clique em "Login Admin" no topo da página
- Use as credenciais padrão

2. Criar Posts:

- No dashboard, clique em "Novo Post"
- Adicione título, conteúdo e mídia
- Defina como rascunho ou publique

3. Gerenciar Comentários:

- Acesse "Comentários" no dashboard
- Aprove, rejeite ou delete comentários

4. Upload de Mídia:

- Use o botão "Fazer Upload" no editor
- Suporte a imagens, vídeos e áudios
- Insira mídia no conteúdo ou defina como destaque

Para Visitantes

1. Visualizar Posts:

- Navegue pela página principal
- Leia posts publicados

2. Comentar:

- Deixe comentários nos posts
- Aguarde moderação do admin

3. Instagram:

- Acesse a aba "Instagram"
- Visualize posts espelhados (quando configurado)



Estrutura do Projeto

Plain Text

```
blog-system/
├── src/                                # Frontend React
│   ├── components/                    # Componentes React
│   ├── contexts/                      # Contextos (Auth)
│   ├── hooks/                         # Hooks customizados
│   └── lib/                           # Utilitários
├── server/                            # Backend Node.js
│   ├── models/                       # Modelos de dados
│   ├── routes/                       # Rotas da API
│   └── middleware/                   # Middlewares
```

```
| └─ uploads/      # Arquivos enviados
└─ dist/           # Build de produção
```

API Endpoints

Autenticação

- `POST /api/auth/login` - Login do admin
- `GET /api/auth/me` - Dados do usuário logado

Posts

- `GET /api/posts` - Listar posts públicos
- `GET /api/posts/admin/all` - Listar todos os posts (admin)
- `POST /api/posts` - Criar post
- `PUT /api/posts/:id` - Atualizar post
- `DELETE /api/posts/:id` - Deletar post
- `POST /api/posts/upload` - Upload de mídia

Comentários

- `GET /api/comments/post/:postId` - Comentários de um post
- `POST /api/comments` - Criar comentário
- `GET /api/comments/admin/all` - Todos os comentários (admin)
- `PUT /api/comments/:id/approve` - Aprovar comentário
- `PUT /api/comments/:id/reject` - Rejeitar comentário
- `DELETE /api/comments/:id` - Deletar comentário

Instagram

- `GET /api/instagram/posts` - Posts do Instagram

Deploy

Build de Produção

```
Bash
```

```
cd blog-system  
npm run build
```

Deploy Frontend

Os arquivos buildados estarão em `dist/` e podem ser servidos por qualquer servidor web.

Deploy Backend

Configure as variáveis de ambiente e execute:

Bash

```
cd server  
npm start
```



Segurança

- Autenticação JWT para admin
- Validação de uploads de arquivo
- Sanitização de dados de entrada
- CORS configurado
- Senhas hasheadas com bcrypt



Licença

Este projeto foi desenvolvido como demonstração de um sistema completo de blog com funcionalidades avançadas.



Suporte

Para dúvidas ou sugestões sobre o sistema, consulte a documentação do código ou entre em contato.

Desenvolvido com ❤️ usando React e Node.js