

```
class Banner
```

```
attr_accessible :horiz, :link, :visible, :image, :position  
has_attached_file :image, styles: { vert: '220'
```

```
before_create :assign_position
```

```
class Banner < ActiveRecord::Base
```

```
attr_accessible :horiz, :link, :visible, :image  
has_attached_file :image, styles: { vert: '220'
```

```
before_create :assign_position
```

```
protected
```

```
def assign_position
```

```
max = Banner.maximum(:position)
```

```
self.position = max ? max + 1 : 0
```

FULL STACK DEVELOPMENT
TECH CHALLENGE

FASE 02

Tech Challenge

O Tech Challenge é uma atividade de desenvolvimento de software em grupo que integra os conhecimentos adquiridos durante a fase atual do curso. A entrega deste projeto é obrigatória e corresponde a 90% da nota final em todas as disciplinas desta fase.

O problema

Atualmente, a maioria de professores e professoras da rede pública de educação não têm plataformas onde postar suas aulas e transmitir conhecimento para alunos e alunas de forma prática, centralizada e tecnológica.

Para solucionar esse problema, nós utilizamos os conhecimentos adquiridos na última fase para auxiliar a nossa comunidade com a criação de uma aplicação de blogging dinâmico, utilizando a plataforma OutSystems. A plataforma foi um sucesso e, agora, nossa aplicação escalará para um panorama nacional. Portanto, precisaremos refatorar nosso Back-end, utilizando a plataforma de desenvolvimento node.js, e precisaremos persistir esses dados em um banco de dados, seja ele SQL ou NoSQL, conforme a decisão do grupo.

Requisitos funcionais

Os seguintes endpoints REST serão implementados para a aplicação de blogging:

- **GET /posts - Lista de Posts:**
 - Este endpoint permitirá aos alunos visualizarem uma lista de todos os posts disponíveis na página principal.
- **GET /posts/:id - Leitura de Posts:**
 - Ao acessar este endpoint com um ID específico de post, os alunos poderão ler o conteúdo completo desse post.
- **POST /posts - Criação de Postagens:**

- Permite que docentes criem novas postagens. Este endpoint aceitará dados como título, conteúdo e autor no corpo da requisição.
- **PUT /posts/:id - Edição de Postagens:**
 - Usado para editar uma postagem existente. Professores deverão fornecer o ID do post que desejam editar e os novos dados no corpo da requisição.
- **GET /posts - Listagem de Todas as Postagens:**
 - Este endpoint permitirá que professores vejam todas as postagens criadas, facilitando a gestão do conteúdo.
- **DELETE /posts/:id - Exclusão de Postagens:**
 - Permite que docentes excluam uma postagem específica, usando o ID do post como parâmetro.
- **GET /posts/search - Busca de Posts:**
 - Este endpoint permitirá a busca de posts por palavras-chave. Os usuários poderão passar uma query string com o termo de busca e o sistema retornará uma lista de posts que contêm esse termo no título ou conteúdo.

Requisitos técnicos

- **Back-end em Node.js:**
 - Implementação do servidor usando Node.js.
 - Utilização de frameworks como Express para roteamento e middleware.
- **Persistência de Dados:**
 - Utilização de um sistema de banco de dados (por exemplo, MongoDB, PostgreSQL).
 - Implementação de modelos de dados adequados para as postagens.
- **Containerização com Docker:**

- Desenvolvimento e implantação usando contêineres Docker para garantir consistência entre ambientes de desenvolvimento e produção.
- **Automação com GitHub Actions:**
 - Configuração de workflows de CI/CD para automação de testes e deploy.
- **Documentação:**
 - Documentação técnica detalhada do projeto, incluindo setup inicial, arquitetura da aplicação e guia de uso das APIs.
- **Cobertura de Testes:**
 - O projeto deve garantir que pelo menos 20% do código seja coberto por testes unitários. Essa medida é essencial para assegurar a qualidade e a estabilidade do código, especialmente em funções críticas como criação, edição e exclusão de postagens.

Entrega

- **Código-Fonte:** repositório GitHub com o código do projeto, incluindo Dockerfiles e scripts de CI/CD.
- **Apresentação Gravada:** demonstração em vídeo do funcionamento da aplicação, incluindo detalhes técnicos de implementação.
- **Documentação:** documento descrevendo a arquitetura do sistema, uso da aplicação e relato de experiências e desafios enfrentados pela equipe durante o desenvolvimento, esta documentação deve ser entregue em arquivo de texto da preferência do grupo ou no readme do github.

