

```
STATUS
ACCOUNT
STATUS
class Banner
  attr_accessible :horiz, :link, :visible, :image, :position
  has_attached_file :image, styles: { vert: '220'
  before_create :assign_position
  ANALYSIS
  def assign_position
    max = Banner.maximum(:position)
    self.position = max ? max + 1 : 0
  end
end
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
  end
end
```

FULL STACK DEVELOPMENT

WELCOME TO MOBILE

# FASE 04

## **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 compreende 90% da nota final em todas as disciplinas desta fase.

## **O problema**

Após o sucesso do desenvolvimento da aplicação de blogging dinâmico com a implementação do back-end em Node.js e o front-end utilizando React, chegou a hora de criarmos uma interface gráfica mobile robusta, intuitiva e eficiente para esta aplicação. Esta atividade focará em desenvolver o front-end mobile utilizando React Native, proporcionando uma experiência de usuário excelente tanto para docentes quanto para estudantes.

## **Objetivo**

Desenvolver uma interface gráfica para a aplicação de blogging utilizando React Native. A aplicação deve ser acessível e fácil de usar, permitindo que os docentes e alunos(as) possam interagir com os diversos endpoints REST já implementados no back-end.

## **Requisitos funcionais**

A interface gráfica deve incluir as seguintes páginas e funcionalidades:

### **1. Página principal (lista de posts)**

- Exibir uma lista de todos os posts disponíveis.
- Cada item da lista deve mostrar o título, autor e uma breve descrição do post.
- Incluir um campo de busca para filtrar posts por palavras-chave.

### **2. Página de leitura de post**

- Exibir o conteúdo completo de um post selecionado.
- Permitir comentários nos posts (opcional).

### **3. Página de criação de postagens**

- Formulário para que professores possam criar postagens.
- Campos para título, conteúdo e autor.
- Botão para enviar o post ao servidor.

### **4. Página de edição de postagens**

- Formulário para que docentes possam editar postagens existentes.
- Carregar os dados atuais do post para edição.
- Botão para salvar as alterações.

### **5. Página de criação de professores**

- Formulário para que professores possam cadastrar outros professores.
- Botão para enviar o post ao servidor.

### **6. Página de edição de professores**

- Formulário para que professores possam editar docentes já cadastrados.
- Botão para salvar as alterações.

### **7. Página de listagem de professores**

- Página para listagem paginada dos professores e, nas tabelas para cada professor, teremos um botão de editar que leva para a página de edição e um botão de excluir que vai deletar o docente do sistema.

### **8. Replique os requisitos 5, 6 e 7 para estudantes**

- Seguindo o padrão de páginas administrativas feitas para professores, faça o mesmo para alunos.

### **9. Página administrativa**

- Exibir uma lista de todas as postagens, com opções para editar e excluir cada post.
- Botões para editar e excluir postagens específicas.

## 10. Autenticação e autorização

- Implementar login para professores.
- Garantir que apenas usuários autenticados possam acessar as páginas de criação, edição e administração de postagens.

## Requisitos técnicos

### 1. Desenvolvimento em React Native

- Utilizar React Native para desenvolver a interface gráfica do aplicativo.
- Utilização de hooks e componentes funcionais.

### 2. Estilização

- Estilizar o projeto de acordo com layout definido pelo grupo.

### 3. Integração com Back-End

- Realizar chamadas aos endpoints REST para obter, criar, editar e excluir **posts**.
- Realizar chamadas aos endpoints REST para obter, criar, editar e excluir **alunos**.
- Realizar chamadas aos endpoints REST para obter, criar, editar e excluir **professores**.
- Realizar chamadas aos endpoints REST para autenticação.
- Validar permissão para professores e alunos, onde professores podem modificar/criar um post e os alunos podem apenas visualizar.
- Gerenciar o estado da aplicação com ferramentas como Context API ou Redux (opcional).

### 4. Documentação

- Documentação técnica detalhada do mobile no README do repositório, incluindo setup inicial, arquitetura da aplicação e guia de uso.

## Entrega

1. **Código-Fonte:** repositório GitHub com o código do projeto.
2. **Apresentação gravada:** demonstração em vídeo do funcionamento da aplicação, incluindo detalhes técnicos de implementação. O tempo máximo do vídeo é 15min.
3. **Documentação:** documento descrevendo a arquitetura do sistema, uso da aplicação e relato de experiências e desafios enfrentados pela equipe durante o desenvolvimento.

# POSTECH

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