

DESAFIO

Desafio final fullstack



Teach
3035

Índice

Desafio final Fullstack.....	03
Mapeamento JPA.....	03
Descritivo Atributos do Usuário.....	04
Descritivo Atributos do Post.....	05
Controllers.....	06
Acesso e entrega.....	11

Desafio final

Fullstack



Utilizando os conhecimentos adquiridos nos treinamentos do curso 3035TEACH, desenvolva um projeto spring boot com o gerenciador de dependências maven e faça o mapeamento 1xN utilizando o JPA, para isso faça uma conexão com o banco de dados postgres adicionando as dependências necessárias no pom do projeto e configurando as credenciais do banco em um arquivo yml ou xml.

Neste desafio você precisará criar uma API de uma rede social onde possa criar usuários, criar posts, deletar, atualizar, excluir e logar.

Mapeamento JPA



O mapeamento da relação entre usuário e post será de 1xN.

Descritivo Atributos do Usuário



Boolean deleted - Esse atributo vai servir para fazer a exclusão apenas lógica do usuário, ou seja, não será necessário excluir fisicamente do banco apenas inativar setando como true o deleted.

LocalDate createdAt - Este atributo vai servir para guardarmos a data em que o usuário foi criado.

LocalDate updatedAt - Este atributo vai servir para guardarmos a data em que o usuário foi atualizado.

String name - Nome completo do Usuário.

String userName - Este atributo vai ser o identificador único do usuário e vai servir para autenticação no login.

String phone - Telefone do usuário não necessitando de regra de validação aqui.

String mail - Email do usuário, este deve ter uma validação para verificar se é um email válido. Dica: verificar se existe um @ na string

String password - Senha do usuário para autenticação no login.

String profileLink - Este atributo vai ser usado para guardar algum link de uma imagem para usar no perfil podendo ser da internet mesmo.

Exemplo: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTKo6EQGkW1RVzpZrawB05Gq0srUxkynU6yP7IOvct1zNQHgu1ilUYIJKhwwq_ECaz2aRo&usqp=CAU

Descritivo Atributos do Post



Boolean deleted - Esse atributo vai servir para fazer a exclusão apenas lógica do post, ou seja, não será necessário excluir fisicamente do banco apenas inativar setando como true o deleted.

LocalDate createdAt - Este atributo vai servir para guardarmos a data em que o post foi criado.

LocalDate updatedAt - Este atributo vai servir para guardarmos a data em que o post foi atualizado.

String title - Título do post pode ser um texto de no máximo 50 caracteres.

String description - Descrição do post pode ser um texto contendo até 200 caracteres.

String photoLink - Este atributo vai ser usado para guardar algum link de uma imagem podendo ser da internet mesmo.

Exemplo: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTKo6EQGkW1RVzpZrawB05Gq0srUxkynU6yP7IOvct1zNQHgu1ilUYIJKhwwq_ECaZ2aRo&usqp=CAU

String videoLink - Este atributo vai ser usado para guardar algum link de um vídeo podendo ser da internet também.

Descritivo Atributos do Post



Boolean private - Este atributo vai ser usado para dizer se o post do usuário pode ser acessado por outros usuários além dele mesmo.

- Exemplo private = true: somente o usuário dono pode ver
- Exemplo private = false: qualquer usuário poderá ver esse post

Controllers



Usuários

- Endpoint para criar um usuário. O usuário vai criar sua própria conta através de um signup.

Na criação deve conter algumas regras de validação como: verificar se o username, email e telefone já são existentes e também se o email é válido.

Obs: Os campos nome, username, email e password não devem ser nulos.

- Endpoint para editar um usuário. O usuário vai poder editar seu perfil através de um update.
- Endpoint para deletar um usuário. O usuário poderá deletar sua conta através de um delete.

Controllers



Usuários

- Endpoint para listar todos os usuários não deletados. O usuário poderá ver os usuários cadastrados para em uma próxima melhoria ele poder adicionar como amigo em sua rede social.
- Criar um login com spring security usando o username e senha que deverá ser criptografada quando salva no banco.

Posts

- Endpoint para criar uma postagem. O usuário poderá criar uma postagem para o seu feed e para isso vai ser preciso verificar junto com a criação, o id do usuário autenticado para o post estar vinculado a esse usuário.

Obs: Os campos title e private não deverão ser nulos.

- Endpoint para atualizar a postagem. O usuário poderá atualizar o post através de um update.
- Endpoint para deletar uma postagem. O usuário poderá deletar um post através do delete
- Endpoint de listagem de uma postagem pelo id do usuário. O usuário poderá ver todas as suas postagens através de um get.
- Endpoint para alterar a privacidade do post. O usuário poderá alterar o post de privado para público setado como false ou true.

Controllers



Utilizando a aplicação backend já executada, criar a interface responsiva de uma rede social.

A aplicação deverá seguir o layout que será disponibilizado no Figma e será composta pelas seguintes telas:

- **Login:** o usuário deve inserir e-mail e senha para acessar a rede social. Necessário ter mensagens informando possíveis erros de acesso (senha errada, e-mail não cadastrado, etc).
- **Extra:** integração para realizar login usando Google, Apple.
- **Página de criação de usuário:** para criar o usuário será necessário informar o nome, e-mail, telefone, descrição, foto e senha. Necessário ter mensagens informando possíveis erros ao realizar o cadastro (campos não preenchidos, usuário já cadastrado, etc).
- **Página para edição/excluir conta:** será possível editar os mesmos campos informados na criação de usuário, além de editar o nome do usuário e incluir biografia. Será possível também excluir a conta. OBS: o nome do usuário será iniciado com o valor cadastrado no campo 'nome', por exemplo:

Nome: Maria da Silva

Nome do usuário: @MariaDaSilva

Controllers



- **Página de perfil:** serão apresentados os dados do usuário (nome, descrição, foto), além das postagens pessoais. A página também deverá conter um botão para adicionar usuário à lista de amigos, botão para fazer a configuração/edição do perfil, botão para acessar a lista de amigos e botão para acessar a home/página do feed.
- **Página de amigos:** listagem dos amigos, contendo paginação. Na lista será apresentado o nome e imagem do perfil, sendo que ao clicar sobre um amigo, usuário será direcionado para a página do perfil desse usuário.
- **Página do feed:** serão apresentadas as postagens dos usuários/amigos em ordem cronológica. Cada postagem irá apresentar o nome do usuário, uma imagem e botão para curtir.

Controllers



- **Implementações no backend:** Além da criação da interface da rede social, será necessário criar novas funcionalidades no backend:
- Envio do atributo da descrição/bio na criação/edição do usuário
- Criar a funcionalidade de curtir a postagem e a contagem do número de curtidas. Não é necessário identificar o usuário que curtiu, por isso, é possível curtir várias vezes (o coração é preenchido apenas para sinalizar a curtida, como se fosse uma animação, após isso, o coração volta a ser vazado).
- Criar endpoint de listagem de usuários pelo id do user logado
- Criar endpoint de post passando os ids dos usuários para fazer a relação de usuário para usuário, podendo assim adicionar amigos.
- Criar relação N-N entre os usuários

Acesso e entrega



- Acesso Figma com design:
<https://www.figma.com/design/kstnnREkgicTbvFxfBzu3C/Teach-3035---Desafio-Front-end-final?node-id=0-1&t=jze2w9JUtb0X97oQ-1>
- A entrega pelo Github e o link postado nos comentários.

BOM DESAFIO E MÃOS À OBRA!! :D