ComTec Hub

Descrição do projeto - ComTec Hub:

Objetivo:

Desenvolver uma **plataforma digital voltada para a comunidade tecnológica**, onde usuários possam se reunir em comunidades públicas ou privadas para compartilhar conhecimentos, tirar dúvidas e publicar artigos sobre tecnologia.

Escopo do projeto:

A plataforma permitirá:

- **Cadastro e autenticação de usuários.**
- **Criação de comunidades** (públicas ou privadas) por qualquer usuário.
- **Comunicação dentro das comunidades**:
  - Apenas membros podem comentar nas comunidades.
  - Para acessar uma comunidade privada, o usuário deve informar um **código de acesso** gerado pelo administrador da comunidade.
- **Visualização pública** de mensagens e postagens em comunidades públicas.
- **Publicação de artigos técnicos** (visíveis a todos os usuários, inclusive não logados).

Utilizaremos o modelo de processo ágil **Scrum** durante o desenvolvimento deste projeto.

Utilizaremos a ferramenta **Trello** para auxiliar em nosso quadro **Scrum.**

Tecnologias usadas:

- **Linguagem de programação**: Java.
- **Banco de Dados**: PostgreSQL.
- **Frameworks**:
  - Spring Boot (estrutura principal da aplicação)
  - Spring Security (autenticação e autorização)
  - Spring Data JPA (persistência de dados)
  - MapStruct (mapeamento entre entidades e DTOs)
  - JUnit(usado para teste unitários)

- Postman(para teste manual da aplicação Rest)

- **Controle de Versão**:
  - Git + GitHub

Estrutura do projeto

- **Entidades**:
  - **Usuario**: UUID id, String email, String login, String senha, List < UsuarioComunidade > usuarioComunidade, List<Mensagem> mensagens.
  - **Comunidade**: Integer id, String nome, String descricao, Integer codigoAcesso, TipoComunidade tipoComunidade, Chat chat, List < Usuario > usuarios, Feed feed.
  - **Chat**: UUID id ,List< Mensagem > mensagens, Comunidade comunidade.
  - **Mensagem**: UUID id, String texto, Usuario usuario, LocalDateTime dataHoraMensagem, Chat chat.
  - **Feed**: List< Comunidade > comunidades;
  - **UsuarioComunidade**: Usuario usuario, Comunidade comunidade, RoleNaComunidade role.
- **Enums**:
  - **TipoComunidade**: PUBLICA, PRIVADA.
  - **RolesNaComunidade**: ADMIN, SUB_ADMIN, MEMBRO.
- **Repositórios**:

  Obs: Todos os repositórios devem ter métodos save, update, findById, findAll, delete.
  - **UsuarioRepository**: findByLogin(String login) : Usuario, findByEmail(String email) : Usuario.
  - **ComunidadeRepository**: findByNome(String nome) : Comunidade, findByTipoComunidade(TipoComunidade tipoComunidade) : List, findByDescricaoContaining(String descricao) : List, findAllUsuario() : List, findByUsuarioContaining(String loginUsuario) : List<>.
  - **ChatRepository**: findByMensagemTextoContaining(String texto) : List, findByComunidade(Comunidade comunidade) : Chat.
  - **MensagemRepository**: findByDataHoraMensagem(LocalDateTime data) : List, findByUsuario(Usuario usuario) : List.
  - **FeedRepository**: Crud padrão
  - **UsuarioComunidadeRepository**: Crud padrão, findByUsuarioLogin(String loginUsuario) : Usuario, findByComunidadeNomeContaining(String nome) : List < Comunidade >
- **Serviços**:

  Obs: todos os serviços devem ter métodos save, update, findById, findAll, delete.

- **UsuarioService**: findAllComunidade : List< Comunidade >.
- **ComunidadeService**: findByNome(String nome) : Comunidade, findByTipoComunidade(TipoComunidade tipoComunidade) : List < Comunidade >, findByDescricaoContaining(String descricao) : List< Comunidade >, findByUsuarioContaining(Usuario usuario) : List< Usuario >, findAllUsuario() List< Usuario >.
- **ChatService**: findByMensagemTextoContaining(String texto) : List < Mensagem >, findByComunidade(Comunidade comunidade) : Chat.
- **MensagemService**: findByMensagemTextoContaining(String texto) : Mensagem, findByUsuario(Usuario usuario) : List < Mensagem >.
- **Feed**: findByTipoComunidade(TipoComunidade) : List < Comunidade >, findByComunidadeNomeContaining(String nome) : List < Comunidade >, findByComunidadeDescricao(String descricao) : List < Comunidade >.

- **DTOs:**

  As nomeações 'Request' e 'Response' devem ser colocadas no final do nome e antes de DTOs. Cada DTO deve ter um mapper.

  - **Request**:
    - UsuarioDTO: String email, String login, String senha.
    - MensagemDTO: String texto.
    - ComunidadeDTO: String nome, String descricao, Integer codigoAcesso, String tipoComunidade.
  - **Response**:
    - UsuarioDTO: UUID id, String email, String login, String senha.
    - MensagemDTO: String usuario, LocalDateTime dataHoraMensagem, String texto.
    - ComunidadeDTO: Integer id, String nome, String descricao, String tipoComunidade.
    - ChatDTO: List < ResponseMensagemDTO > mensagens
    - FeedDTO: List < ComunidadeDTO >