**Documentação - Models**

Visando a entrega do produto com funcionalidades além das solicitadas na task, não só criamos o Model do tema da nossa rede social mas também o Model de usuário e de Postagem, com a devida inter-relação entre as tabelas criadas.

**Model Tema**

**Atributos de Tema:**

**1- ID-PK:** Primary key referente a cada novo tema inserido, que aponta para todas as informações de cada tema. Colocamos a anotação @Id para definir esse atributo como a PK da tabela e também a anotação @GeneratedValue com a estratégia de autoincremento.(GenerationType.Identity).

**2 - Descrição:** Nome do tema inserido no banco de dados. Utilizamos a anotação @NotNull para que o campo seja obrigatoriamente preenchido por algo; Colocamento o @Size para que o campo seja composto entre 3 e 20 caracteres.

**3 - Postagem-FK:** Atributo referente a foreign key de postagem, para que cada tema esteja ligado a N postagens. Para sua correlação com a tabela de postagem, utilizamos a anotação @OneToMany e colocamos a anotação @JsonIgnoreProperties(“Tema”) para evitar problemas de recursividade.

Obs: Conversamos com o Instrutor Marcelo sobre a troca dos atributos da tabela de tema, já que na task passada foi solicitado o uso de 3 atributos diferentes além as ID e da FK e, com a nossa proposta de rede social, 1 atributo só já seria o suficiente, pois poderíamos gerar vários temas apenas mudando a descrição do tema.

**Model Postagem**

**Atributos de Postagem:**

**1 - ID:** Assim como no model anterior, campo destinado a Primary Key. Utilizadas as anotações de @Id e @GeneratedValue para que o valor seja autogerado.

**2 - Titulo:** Titulo da postagem Anotações utilizadas: @NotNull e @Size (min 5 e máx 100);

**3 - Texto:** Texto referente ao título indicado. Anotações utilizadas: NotNull e Size (min 1 e max 1000);

**4 - Data:** recebe a data exata da publicação e a hora. Anotações utilizadas @Temporal(TemporalType.TIMESTAMP) para que a data da postagem seja printada. A data definida = new java.sql.Date(System.currentTimeMillis().

**5 - Usuário - FK:** Usuário refere-se ao usuário que faz uso da plataforma. Anotações utilizadas: @OneToMany (mappedBy = "usuario", cascade = CascadeType.ALL) - faz a correlação das tabelas de usuário e de postagem. @JsonIgnoreProperties("usuario") - É uma anotação para evitar problemas com a recursividade.

**6 - Tema - FK:**

**Model Usuário**

**Atributos de Usuário:**

**1 - ID-PK:** Primary key referente a cada novo tema inserido, que aponta para todas as informações de cada usuário. Colocamos a anotação @Id para definir esse atributo como a PK da tabela e também a anotação @GeneratedValue com a estratégia de auto incremento.(GenerationType.Identity).

**2 - Nome:** Nome do usuário.A anotação @NotNull e @Size (min =3, max = 50) pois todo usuário tem que inserir nome para se cadastrar.

**3 - Senha:** Senha do usuário. A anotação @NotNull e @Size (min = 6, max = 100) pois todo usuário tem que inserir senha para se cadastrar.

**4 - Email:** Email do usuário. A anotação @NotNull pois todo usuário tem que inserir email para se cadastrar.

**5 - Postagem - FK:** Atributo referente a foreign key de postagem, para que cada usuário esteja ligado a N postagens. Anotações utilizadas: @OneToMany(mappedBy = "usuario", cascade = CascadeType.ALL); @JsonIgnoreProperties("usuario")