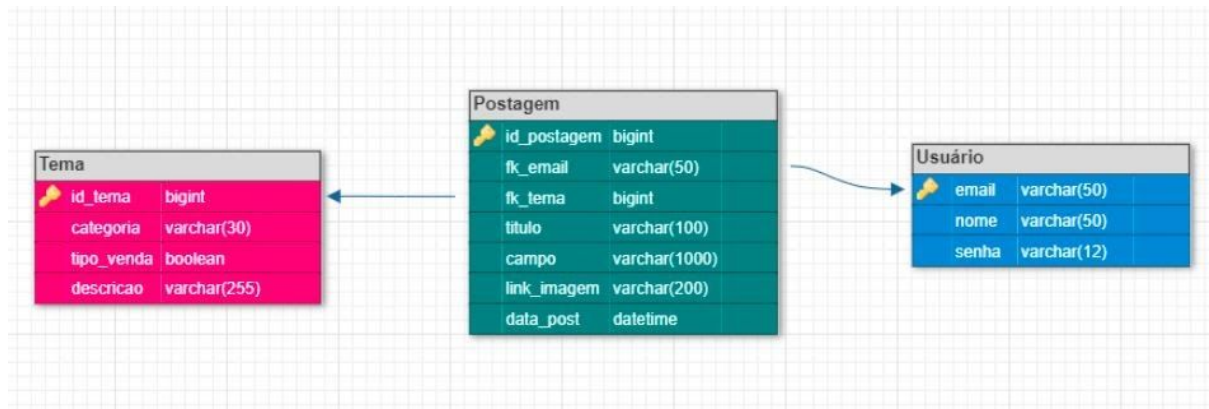


Grupo 1 I - Mei

Integrantes: Cinthia Tengan, Lucas Corrêa, João Paulo Oliveira, Sthefany Silva, Sérgio Ruiz.

Descrição das tabelas e atributos do modelo DER.



De acordo com as instruções propostas pelo instrutor, foram criadas 3 tabelas com seus respectivos atributos.

- **Tema**

id_tema: chave primária para poder identificar o tema escolhido;
categoria: classificação do tema;

categoria: classificar o tema;

tipo_venda: de acordo com a regra de negócio, escolhemos que devem ter dois tipos produtos e serviços, e neste caso usamos um valor booleano no qual 0 representa produto e 1 o serviço;

descricao: descrever o tema selecionado com mais detalhe;

- **Usuario**

email: chave primária para identificar o usuário e consideramos que cada email é único.

nome: nome do usuario completo;

senha: senha de acesso do usuário;

- **Postagem**

id_postagem: chave primária para identificar a postagem;

fk_email: chave estrangeira da tabela usuario para poder identificar o usuário que fez a postagem;

fk_tema: chave estrangeira da tabela tema para poder identificar o tema escolhido pelo o usuário na postagem;

titulo: titulo da postagem;

campo: texto principal da postagem do usuário;

link_imagem: armazena a 'url' da imagem e 'linka' com a postagem;

data_post: data da postagem feita do usuário.

Criado o arquivo Application Properties:

```
spring.jpa.hibernate.ddl-auto=update
```

```
spring.datasource.url=jdbc:mysql://localhost:3306/db_l_mei?createDatabaseIfNotExist=true&serverTimezone=UTC&useSSL=false
```

```
spring.datasource.username=root
```

```
spring.datasource.password=root
```

```
spring.jpa.show-sql=true
```

// Com esses comandos spring, o banco de dados atualiza automatico, e se o banco não existe ele é criado.

Criação do pacote Model , com a classe tema

@Entity // essa classe será definida como uma entidade do jpa hibernate.

@Table(name = "tema") // dentro do banco de dados uma tabela será criada com o nome de tema.

//criação da classe tema:

```
public class Tema {
```

//Dividindo os atributos para explicar as suas anotações:

@Id // Identifica o atributo id como uma primary key.

@Column(name = "id_tema") // renomeia a coluna como id_tema no banco de dados ao invés do atributo id no java.

@GeneratedValue(strategy = GenerationType.IDENTITY) // diz que esse atributo será auto incrementado.

```
private long id;
```

@NotNull // anotação para dizer que o valor não pode ser nulo

@Size(max = 30) // tamanho máximo do atributo

```
private String categoria;
```

@Column(name = "tipo_venda") //renomeia a coluna como tipo_venda no banco de dados ao invés do atributo tipoVenda no java.

@NotNull // anotação para dizer que o valor não pode ser nulo

```
private boolean tipoVenda;
```

@NotNull // anotação para dizer que o valor não pode ser nulo

```
private String descricao;
```
