

Aula 10: Criando a Entidade e Tabela de Usuários

September 18, 2025

1 Introdução

Na Aula 9, adicionamos o Spring Security ao projeto e observamos seu comportamento padrão, que bloqueia todas as requisições com o código 401 (Unauthorized) e exibe um formulário de login para aplicações web. Para implementar autenticação stateless com JWT, precisamos criar uma entidade para representar usuários e uma tabela no banco de dados. Nesta aula, criaremos a entidade JPA Usuario e uma migration Flyway para a tabela usuarios. As alterações serão feitas no Visual Studio Code ou sua IDE preferida.

2 Criando a Entidade Usuario

Para suportar autenticação, criaremos uma entidade JPA no pacote `med.voll.api.domain.usuario`, que representará os usuários da API. No Visual Studio Code ou sua IDE preferida, criamos o pacote `usuario` dentro de `med.voll.api.domain` e adicionamos a classe `Usuario`.

O código da entidade é:

```
1 package med.voll.api.domain.usuario;
2
3 import jakarta.persistence.*;
4 import lombok.AllArgsConstructor;
5 import lombok.EqualsAndHashCode;
6 import lombok.Getter;
7 import lombok.NoArgsConstructor;
8
9 @Table(name = "usuarios")
10 @Entity(name = "Usuario")
11 @Getter
12 @NoArgsConstructor
13 @AllArgsConstructor
14 @EqualsAndHashCode(of = "id")
15 public class Usuario {
16
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     private Long id;
20     private String login;
21     private String senha;
22 }
```

2.1 Explicação do Código

- `@Table(name = "usuarios")`: Mapeia a entidade para a tabela `usuarios` no banco de dados.
- `@Entity(name = "Usuario")`: Define a classe como uma entidade JPA.
- `@Id` e `@GeneratedValue(strategy = GenerationType.IDENTITY)`: Configura o campo `id` como chave primária com auto-incremento.

- login e senha: Representam o e-mail do usuário e a senha (que será armazenada com hash, como veremos em aulas futuras).
- @Getter, @NoArgsConstructor, @AllArgsConstructor, @EqualsAndHashCode(of = "id"): Anotações do Lombok para gerar getters, construtores e métodos de igualdade baseados no id.

3 Criando a Migration Flyway

Para criar a tabela usuarios no banco de dados, usaremos o Flyway para gerenciar a evolução do schema. Antes de criar a migration, interrompemos o projeto (atalho: Ctrl + F12) para evitar conflitos.

No diretório src/main/resources/db/migration, criamos um novo arquivo de migration chamado V5__create-table-usuarios.sql, copiando a estrutura de uma migration existente (ex.: V1__create-table-medicos.sql) e adaptando-a:

```
1 create table usuarios (
2     id bigint not null auto_increment,
3     login varchar(100) not null,
4     senha varchar(255) not null,
5     primary key(id)
6 );
```

3.1 Explicação da Migration

- id bigint not null auto_increment: Chave primária com auto-incremento.
- login varchar(100) not null: Armazena o e-mail do usuário, com até 100 caracteres.
- senha varchar(255) not null: Armazena a senha (com hash, configurado em aulas futuras), com até 255 caracteres para suportar algoritmos de hash.
- primary key(id): Define o id como chave primária.

Após salvar a migration, reiniciamos o projeto. O Flyway detecta e executa a nova migration, criando a tabela usuarios. No console, verificamos os logs:

```
Current version of schema 'vollmedapi': 4
Migrating schema 'vollmedapi' to version "5 - create-table-usuarios"
Successfully applied 1 migration to schema 'vollmedapi', now at version v5
```

4 Próximos Passos

Com a entidade Usuario e a tabela usuarios criadas, estamos prontos para implementar o processo de autenticação com JWT, incluindo um endpoint de login e lógica para geração de tokens. Na próxima aula, configuraremos o Spring Security para suportar autenticação stateless. Continue praticando no Visual Studio Code ou sua IDE preferida!

5 Dica do Professor

- **Aprofunde-se em JPA e Flyway:** Consulte a documentação do Spring Data JPA (<https://spring.io/projects/spring-data-jpa>) e do Flyway (<https://flywaydb.org/documentation/>) para entender entidades e migrations.
- **Comunidade no X:** Siga perfis como @SpringDataPro e @DBMigration no X para dicas sobre JPA e gerenciamento de schemas.
- **Pratique:** Verifique no banco de dados (ex.: MySQL Workbench) se a tabela usuarios foi criada e analise os logs do Flyway para confirmar a execução da migration.