

06

## Criando a entidade de Ordem

Na aula anterior, ficamos com o desafio de criar a entidade de ordens. Para quem ficou com alguma dúvida, nesta seção nós iremos detalhar o passo a passo da atividade.

1º: Criar uma classe no pacote `br.com.alura.model` e nomeá-la como **Ordem**. Como nossa classe é uma entidade, precisamos anotá-la como `@Entity`:

```
@Entity  
public class Ordem {  
  
}
```

2º: Com a classe criada, precisamos inserir os atributos da classe. A ordem possui um id, preço, tipo, data, status e id do usuário (userId). Vamos criá-los:

```
private Long id;  
  
private Double preco;  
  
private String tipo;  
  
private LocalDate data;  
  
private String status;  
  
private Long userId;
```

3º: Precisamos informar qual atributo é nossa PK (Primary Key) e no nosso caso será o id. A anotação responsável por esta configuração é a `@Id`.

```
@Id  
private Long id;
```

4º: Como definimos na configuração do banco de dados que o id será auto gerado, não é necessário informá-lo pela nossa aplicação. Precisamos realizar uma configuração para informar isso também e conseguimos fazê-lo com a anotação `@GeneratedValue`

```
@Id  
@GeneratedValue(strategy = GenerationType.IDENTITY)  
private Long id;
```

*Identity indica que o provedor de persistência deve atribuir chaves primárias para a entidade usando uma coluna de identidade do banco de dados*

5º: Quando estamos utilizando o JPA, os atributos das nossas entidades devem ter os mesmos nomes dos atributos da tabela do banco de dados, caso contrário não ocorrerá o binding. O padrão de nomenclatura de atributos no Java, segue o padrão camelCase (todas as letras da primeira palavra minúsculas, se houver uma segunda palavra, deve-se usar a primeira letra maiúscula e não deve haver espaço. Ex: primeiraPalavra) e o nosso banco de dados usa o *underline* para separação das palavras. Ex: primeira\_palavra. O nosso atributo userId do Java é diferente do user\_id da tabela e dessa forma nossa informação não seria persistida no banco de dados. Para que a operação seja possível e ainda mantermos as boas práticas de nomenclatura dos nossos atributos, podemos usar a anotação `@Column` do JPA e informar qual é o nome do atributo da tabela do banco de dados que o nosso objeto fará o binding.

```
@Column(name = "user_id")
private Long userId;
```

6º: Agora um último passo é criar os getters e setters necessários:

```
public void setPreco(Double preco) {
    this.preco = preco;
}

public void setTipo(String tipo) {
    this.tipo = tipo;
}

public void setData(LocalDate data) {
    this.data = data;
}

public void setStatus(String status) {
    this.status = status;
}

public void setUserId(Long userId) {
    this.userId = userId;
}
```

Pronto!! Nossa entidade está pronta para uso =)