

Mãos à obra: JPA - A classe Venda como Entidade

Vamos colocar mais uma vez a mão na massa. Com base no que você aprendeu, seguem linhas gerais do que deve ser feito.

1 - Nossa classe `Venda` não é uma Entidade, ou seja, ela não é persistida no banco de dados. Para deixarmos nossa aplicação ainda mais completa, vamos persistir as informações de vendas. Ao longo desse exercício, você perceberá que você trabalhará muito mais com JPA/Hibernate do que o JSF. Isso porque, assim como CDI casa tão bem com JSF, JPA também faz parte desse trio. Não esqueça que os três são especificações Java EE. Se por acaso você tiver alguma dúvida com JPA, o mais recomendado é que você assista os treinamentos relacionados com JPA. No fundo, o que queremos aqui, é construir uma aplicação ainda mais completa.

2 - Transforme a classe `Venda` em uma entidade. Lembre-se que entrará em jogo as anotações `@Entity`, `@Id` e `@GeneratedValue`, inclusive a adição da classe `Venda` no arquivo `persistence.xml` e também `@ManyToOne` no livro. **Não esqueça de adicionar o construtor padrão.**

3 - Reinicie a aplicação e faça um login. JPA detectará que a tabela `Venda` não existe e a criará para nós.

4 - Como apresentado no vídeo, acesse seu banco de dados pelo terminal e inclua algumas vendas.

5 - Em `VendasBean`, injete um `EntityManager` e através dele execute uma JPQL que retorne todas as vendas que você acabou de inserir no banco. Não domina JPQL? Não se preocupe, você pode recorrer ao vídeo e mais tarde estudar conosco aqui JPA/Hibernate em outros treinamentos para ficar ainda melhor.

6 - Teste sua aplicação e verifique o gráfico, que deve exibir os dados com base nas vendas cadastradas.