

Curso: Sistemas de Informação	
Disciplina: Mapeamento Objeto Relacional	Ciclo: 8
Professor: Luiz Gustavo Dias	Tipo: Material Complementar
Objetivo: JPA, Maven e Hibernate	

JPA – IMPLEMENTAÇÃO PRÁTICA

ARQUITETURA

Maven:

O Maven é uma ferramenta de código aberto mantida pela Apache. Trata-se de uma ferramenta de gestão de dependências e um task runner. Em outras palavras, o Maven automatiza os processos de obtenção de dependências e de compilação de projetos Java.

Principais Classes:

EntityManager

<https://docs.oracle.com/javaee/7/api/javax/persistence/EntityManager.html>

Um objeto EntityManager encapsula uma conexão com a base de dados e serve para efetuar operações de acesso a dados (inserção, remoção e atualização) em entidades (clientes, produtos, pedidos, etc.) por ele monitoradas em um mesmo contexto de persistência.

Escopo: tipicamente mantém-se uma instância única de EntityManager para cada thread do sistema (no caso de aplicações web, para cada requisição ao sistema).

EntityManagerFactory

<https://docs.oracle.com/javaee/7/api/javax/persistence/EntityManagerFactory.html>

Um objeto EntityManagerFactory é utilizado para instanciar objetos EntityManager.

Escopo: tipicamente mantém-se uma instância única de EntityManagerFactory para toda aplicação.

CRIANDO UMA APLICAÇÃO SIMPLES

Passo 1: Criar uma aplicação Java

- a) A aplicação deve refletir o diagrama abaixo;

Produto
+ cod : int + nome : String + categoria : int
setCod(Integer cod) setNome(String nome) setCategoria(Integer categoria) getCod() : Integer getNome() : String getCategoria : Integer toString() : String

- b) Criar o construtor do produto;
c) Criar métodos get e set;
d) Criar método toString vide exemplo abaixo:

```
@Override  
public String toString() {  
    return "Produto [cod=" + cod + ", nome=" + nome + ", categoria=" + categoria + "];"  
}
```

- e) Criar uma classe contendo o void, instanciar 3 produtos e imprimí-los na tela.

Passo 2: Incluir o JPA para persistir os objetos no banco de dados

- a) Crie um novo schema no MySQL com o nome jpa.
b) Crie um projeto Maven para trabalhar com o MySql e o Java

Maven é o gerenciador de builds e dependências do Java.

- File > New > Other > Maven Project
- Create a Simple Project:
 - Group Id (nome do pacote da empresa): com.univasweb
 - Artifact Id (nome do projeto): ex1jpamaven

- c) Copie as pastas do projeto anterior para este novo projeto:
 - a. Pode copiar o pacote todo.
 - b. Copiar para src/main/java.
- d) Execute o projeto para garantir que esteja funcionando.
- e) Atualize o Maven do projeto para Java 8 ou 11:
 - a. Observe que o compliance do java criado refere-se à versão 5.
 - b. Localize o arquivo pom.xml e inclua a propriedade abaixo:

```
<properties>
  <maven.compiler.source>8</maven.compiler.source>
  <maven.compiler.target>8</maven.compiler.target>
</properties>
```

- c. Clique com o botão direito do mouse no projeto > Maven > Update Project.
- f) Inclua as dependências Maven a serem baixadas.
 - a. Pesquisar a versão da dependência junto aos seus mantenedores.
 - b. Escolher versões que sejam estáveis.

```
<dependencies>
  <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.6.15.Final</version>
    <type>pom</type>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-entitymanager -->
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-entitymanager</artifactId>
    <version>5.6.15.Final</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.28</version>
  </dependency>
</dependencies>
```

- c. quando você salvar o arquivo o Maven já fará o download das dependências indicadas.
- g) Configure o JPA no projeto por meio do arquivo persistence.xml:
 - a. Em resources, crie uma pasta chamada META-INF.
 - b. Dentro dela, crie o arquivo persistence.xml
 - c. Escreva o conteúdo do arquivo.
- h) Incluir os mapeamentos objeto relacional
 - a. Entidade de Domínio

```
@Entity  
public class Produto{
```

- b. Chave Primária

```
@Id  
@GeneratedValue(strategy=GenerationType.IDENTITY)  
private Integer cod;
```

- c. Colunas

```
@Column(name="nomeprod")  
private String nome;  
@Column(name="categprod")  
private Integer categoria;
```

- i) Adequar a classe com o main para persistir os objetos:

- a. Instanciar o EntityManagerFactory e o EntityManager conforme abaixo:

```
EntityManagerFactory emf = Persistence.createEntityManagerFactory("exemplo-jpa");  
EntityManager em = emf.createEntityManager();
```

- b. Utilize o método persist para persistir os objetos conforme abaixo:

```
em.getTransaction().begin();  
  
em.persist(p1);  
em.persist(p2);  
em.persist(p3);  
  
em.getTransaction().commit();
```

getTransaction().begin() inicia a transação com o banco e o .commit() confirma a persistência.

- c. Utilize o método close para encerrar a conexão com o banco de dados.