




Instruções: *Todos os programas devem ser resolvidos utilizando os conceitos de Programação Orientada a Objetos, a linguagem Java™ e os conceitos de Java Persistence API.*

12 - RELACIONAMENTO UM PARA MUITOS


1. Banco de Dados

1.1. Acesse o menu **Iniciar** → **Todos os Programas** → **WampServer** e **start WampServer**.

1.2. Aparecerá o ícone  na área de notificação do Windows.

1.3. Acesse o menu **Iniciar** → **Todos os Programas** → **MySQL** e **MySQL Workbench**.

1.4. Clique no usuário **root** na tela do **MySQL Workbench**.

1.5. Na tela de execução de *scripts*, selecione os comandos abaixo e clique no botão .

```
CREATE DATABASE EXERCICIO12;
```

```
USE EXERCICIO12;
```

```
CREATE TABLE TBL_CATEGORIA(  
ID_CATEGORIA BIGINT PRIMARY KEY AUTO_INCREMENT,  
TX_DESCRICAO VARCHAR(25)  
);
```

```
CREATE TABLE TBL_PRODUTO(  
ID_PRODUTO BIGINT PRIMARY KEY AUTO_INCREMENT,  
TX_NOME VARCHAR(35) NOT NULL,  
NR_PRECO DOUBLE NOT NULL,  
ID_CATEGORIA BIGINT NOT NULL,  
FOREIGN KEY (ID_CATEGORIA) REFERENCES TBL_CATEGORIA (ID_CATEGORIA)  
);
```



Instruções: *Todos os programas devem ser resolvidos utilizando os conceitos de Programação Orientada a Objetos, a linguagem Java™ e os conceitos de Java Persistence API.*

2. Criação do projeto

- 2.1. Abra o *Eclipse* e acesse o menu **File** → **New Project**.
- 2.2. Selecione **Web** → **Dynamic Web Project** e clique em **Next**.
- 2.3. Em **Project Name**, coloque o nome de **PC2-Exercicio12**.
- 2.4. Selecione como **Dynamic web module version** a versão 2.5.
- 2.5. Clique no botão **Modify** para acréscimo de suporte ao JSF.
- 2.6. Na tela que abre, marque as opções **JavaScript 1.0** e **JavaServer Faces 2.0**, depois clique em **Ok**.
- 2.7. Clique em **Next** três vezes até chegar à tela de configurações do JSF. Em **Type**, selecione a opção **Disable Library Configuration**.
- 2.8. Ainda nessa tela, em **URL Mapping**, remova o mapeamento **/faces/*** e adicione um novo mapeamento como ***.xhtml**.
- 2.9. Clique em **Finish** para criar o projeto.

3. Importação de Bibliotecas

- 3.1. Abra a pasta **[rede]\\PC1\\lib-final** e copie o conteúdo da pasta.
- 3.2. Acesse o **Eclipse** e navegue através do seguinte caminho: **WebContent/WEB-INF/lib**.
- 3.3. Cole os JARs copiados dentro da pasta **lib**.

4. Pacotes

- 4.1. Crie os pacotes abaixo dentro da pasta **src**:

br.com.etechoracio.jpa.model

br.com.etechoracio.jpa.test



Instruções: *Todos os programas devem ser resolvidos utilizando os conceitos de Programação Orientada a Objetos, a linguagem Java™ e os conceitos de Java Persistence API.*

5. Configuração com MySQL

5.1. Para configurar o Hibernate em uma aplicação, devemos criar a pasta **META-INF**:

5.1.1. Clique com o botão direito em **src** e acesse a opção **New → Other**

5.1.2. Selecione em **General** a opção **Folder**

5.1.3. Digite **META-INF** como nome de pasta e **Finish**

5.2. Copie o arquivo **persistence.xml** localizado em **[rede]/PCII/modelos** para dentro da pasta **META-INF**.

6. Mapeamento

6.1. Criar a classe **Categoria** dentro do pacote **model**.



6.2. Anote a classe com o mapeamento **@Getter** e **@Setter**.

6.3. Mapear a classe com as anotações **@Entity** e **@Table**, conforme **EXEMPLO** abaixo:

```
@Entity
@Table(name="TBL_VEICULO")
public class Veiculo {
```

6.4. Mapear o atributo **id** como chave primária da tabela e auto-incrementável, conforme **EXEMPLO** abaixo:

```
@Id
@GeneratedValue
@Column(name="ID_VEICULO")
private Long id;
```

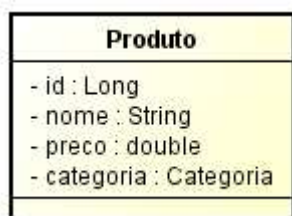
6.5. Mapear o atributo **descricao** com a anotação **@Column**.

6.6. Identificar a classe com **Ctrl + Shift + F** e salvar a classe (**Ctrl + S**).



Instruções: *Todos os programas devem ser resolvidos utilizando os conceitos de Programação Orientada a Objetos, a linguagem Java™ e os conceitos de Java Persistence API.*

6.7. Criar a classe **Produto** abaixo dentro do pacote **model**.



6.8. Mapear a classe com as anotações JPA, inclusive o relacionamento com Categoria (**@ManyToOne**).

6.9. Identificar a classe com **Ctrl + Shift + F** e salvar a classe (**Ctrl + S**).

7. Configuração do Tomcat no Eclipse

7.1. Abra a *View* de *Servers* no Eclipse. Aperte **Ctrl + 3**, digite **Servers** e pressione ENTER.

7.2. Clique com o botão direito dentro da aba *Servers* e vá em **New → Server**.

7.3. Selecione o **Apache Tomcat7.0** e clique em **Next**.

7.4. Na próxima tela, selecione o diretório de instalação do Tomcat. A localização do diretório será: **C:\Arquivos de Programas\Apache Software Foundation\Tomcat 7.0**.

7.5. Clique em **Finish**.

7.6. Clique no botão  para iniciar o **Tomcat**.

8. Testando mapeamento

8.1. Abra a pasta **...\PCII\exemplos** e copie a classe **validacao.xhtml**.

8.2. Acesse o **Eclipse** e navegue até a pasta **WebContent**.

8.3. Cole o arquivo copiado.

8.4. Abra novamente a pasta **...\PCII\exemplos** e copie a classe **ValidadorMB**.

8.5. Acesse o **Eclipse** e navegue até a pasta **src**.

8.6. Cole o arquivo copiado dentro do pacote **test**.



Instruções: *Todos os programas devem ser resolvidos utilizando os conceitos de Programação Orientada a Objetos, a linguagem Java™ e os conceitos de Java Persistence API.*

8.7. Abra a classe ValidadorMB e altere a linha **25** para **Produto.class**.

8.8. Salvar a classe (**Ctrl + S**).

8.9. Acesse o menu **Window → Web Browser → Default system web browser** ou selecione o navegador desejado para executar a aplicação.

8.10. Execute a página validacao.xhtml e clique no botão **Validar** e veja se o resultado é **"Conexão OK"**.

8.11. **Delete a pasta lib** do projeto através do seguinte caminho: **WebContent/WEB-INF/**.

8.12. Acesse a pasta **D:/Projeto-Java-PCII/PC2-Exercicio12** (via **Prompt de Comando**).

8.13. Transforme o diretório **PC2-Exercicio12** em um repositório do Git.

```
git init
```

8.14. Verifique a situação dos arquivos no repositório Git.

```
git status
```

8.15. Faça com que os arquivos sejam rastreados pelo Git.

```
git add .
```

8.16. Verifique a situação dos arquivos no repositório Git novamente.

8.17. Execute o **commit** para gravar as mudanças no repositório com a mensagem **"[Ex.12] – Relacionamento OneToOne"**.

8.18. Acesse o github (<https://github.com/>) e entre na sua conta.

8.19. Crie um repositório chamado **PC2-Exercicio12**.

...or push an existing repository from the command line

```
git remote add origin https://github.com/rogermoraes/exercicio3.git  
git push -u origin master
```

8.20. Execute os comandos listados acima para incluir seu código no repositório remoto.

8.21. Abra a página web do **Github** e verifique se o repositório foi atualizado.