

Vídeo 5.1: Configurando o Hibernate e JPA

Olá! Na videoaula aula 4, você conheceu a ferramenta JSP e sua aplicação para o desenvolvimento de páginas dinâmicas, utilizando a linguagem Java. Agora, você dará continuidade ao seu projeto, só que irá conhecer o outro lado da sua aplicação, ou seja, aquele que realiza a comunicação com o banco de dados. Para isso, aprenderá a utilizar uma ferramenta de ORM, Object relational mapping, para tratar a persistência dos dados.

Object Relational Mapper ou Mapeamento Objeto Relacional, traduzindo para o português, foi a alternativa desenvolvida pela comunidade de desenvolvedores Java Web, quando perceberam que os profissionais da programação perdiam bastante tempo para implementares queries SQL, Structured Query Language, em suas aplicações. Outro problema bem comum era a diferença de escrita do SQL, pois há variações de escrita quando se trata de bancos de dados distintos.

Diante dessas problemáticas e da facilidade do desenvolvedor em trabalhar com o paradigma orientado a objeto, os frameworks de ORM foram projetados no mercado e ganharam bastante apoio da comunidade. Esses frameworks realizam o mapeamento das suas classes com as entidades no banco, através de anotações na sua classe. Com um mapeamento realizado corretamente, esses frameworks conseguem realizar operações no banco, como insert, select, delete e update dos dados da sua base sem a necessidade da escrita de um código SQL, fazendo-se uso apenas da programação em Java.

Agora que você conhece mais acerca do ORM, você irá utilizá-los para avançar no desenvolvimento do seu projeto. Nesta videoaula, você trabalhará com os frameworks Hibernate e JPA. O JPA, Java Persistence API, é uma especificação Java que define como deve ser o comportamento dos frameworks que lidam com essa parte de persistência de dados. Já o Hibernate é uma implementação dessa especificação, que, atualmente, é uma das mais populares. Para que possa utilizá-las, você precisa importar as bibliotecas dessas ferramentas, já que elas não vêm no pacote padrão do Java.

Inicialmente, você irá adicionar as dependências dessas ferramentas ao seu projeto. Para isso, você utilizará o Maven. No arquivo do Maven pom.xml, crie um bloco com as tags `<dependencies>` `(dependencies)` e `</dependencies>` `(barra dependencies)`, onde você irá definir todas as dependências que irá necessitar, que, no caso, você precisará das dependências hibernate-core e hibernate-entitymanager. Cada dependência deverá estar entre as tags `<dependency>` `(dependency)` e `</dependency>` `(barra dependency)`, e, dentro dessas tags, deve ser informado o groupId, o artifactId e a version.

#Audiodescrição: Segue código.

```
<dependencies>
  <!-- HIBERNATE -->
  <dependency>
```

```
<groupId>org.hibernate</groupId>
<artifactId>hibernate-core</artifactId>
<version>5.2.6.Final</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-entitymanager</artifactId>
  <version>5.2.6.Final</version>
</dependency>
</dependencies>
```

Feito isso, o Maven irá tratar de capturar essas dependências.

Outra dependência necessária para o projeto é a dependência do driver do banco de dados, isso se faz necessário para que o Hibernate possa se comunicar com um banco de dados a fim de realizar a persistência das informações. Nesse sentido, você deve adotar o MySQL por se tratar do banco de dados em que a instalação já foi realizada na sua máquina. Para isso, você utilizará o MySQL versão 8.0.20 que é a versão mais atualizada do software. Existem diversos bancos de dados que você pode utilizar, mas deve se atentar a utilizar o driver do banco de dados escolhido. Lembrando que você também precisará adicionar, como uma dependência do projeto, esse driver do MySQL ao seu arquivo pom.

#Audiodescrição: Segue código.

```
<dependencies>
<!-- HIBERNATE -->
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.2.6.Final</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-entitymanager</artifactId>
  <version>5.2.6.Final</version>
</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
```

```
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.13</version>
</dependency>
</dependencies>
```

Pronto, após adicionar essas dependências, agora você precisará solicitar ao Maven que realize o update do seu projeto para que ele possa realizar o download desses artefatos. Em seguida, na ferramenta Eclipse, navegue até o seu projeto para acessar as opções do menu, que pode ser acessado pelo seu teclado, utilizando a tecla Aplicação. Depois, navegue até a opção Maven, e, por fim, navegue até a opção Update Project.

#A imagem mostra uma captura de tela do Eclipse. Nela, é exibida uma janela com várias opções, entre elas “Maven”, que está selecionada. Dessa opção, sai uma nova janela, na qual está selecionada a opção “Update Project”.

O Update Project pode ser realizado também utilizando o atalho do teclado Alt+f5. Quando você realiza esse comando, será aberta uma janela de confirmação, onde você irá apenas marcar o nome do seu projeto, e, em seguida, acionar o botão Ok.

#A imagem mostra a tela da janela de confirmação do “Update Maven Project”. Nela, o projeto “CatalogoDeFilmes” está ticado. Logo abaixo, há as seguintes caixas de seleção ticadas: “Update Project configuration from pom.xml”, “Refresh workspace resources from local filesystem” e “Clean projects”. Na parte inferior, há os botões “OK”, que está selecionado, e “Cancel”.

A partir dessa ação, o Maven irá se encarregar de baixar os artefatos, e, assim, será possível utilizar essas ferramentas em seu projeto. Outro ponto importante para o prosseguimento da aula é que você tenha um banco de dados previamente configurado em sua máquina. Como mencionado anteriormente, o MySQL é um dos bancos de dados bastante utilizado pela comunidade de desenvolvimento; possui diversos artigos a respeito do seu uso; e será o banco de dados adotado para a configuração do seu projeto. Com as dependências já configuradas no Maven e o MySQL instalado e configurado em sua máquina Windows, você pode prosseguir com o curso.

Para o JPA e o Hibernate conseguirem trabalhar corretamente, eles necessitam de um arquivo de configuração chamado persistence.xml. É esse arquivo em que você irá definir e configurar as informações necessárias para o seu projeto se comunicar corretamente com o banco de dados.

São exemplos dessas informações: dados como login e senha do banco de dados; dialeto, que é a linguagem que o banco de dados entende; e o driver do banco de dados.

O Hibernate consegue se comunicar com diversas bases de dados, sendo essa uma das vantagens de se utilizar frameworks ORM. Caso sua aplicação, por algum motivo, necessite mudar o sistema de banco de dados, você não terá tantos problemas com essa alteração, pois, basicamente, a maneira de trabalhar, orientado a objeto que é utilizado, não irá mudar.

Para criar esse arquivo persistence.xml, deve-se utilizar o Eclipse. Para isso, em primeiro lugar, crie uma pasta em que o arquivo persistence.xml irá se localizar. Logo em seguida, crie o arquivo persistence.xml. Com isso, navegue até seu projeto, no Eclipse, e acesse as subpastas do projeto. Depois, navegue até a pasta src, que fica dentro de Java Resources. Acione as opções através do seu teclado, utilizando a tecla aplicação. Em seguida, escolha a opção New e, por fim, Other.

#A imagem mostra a tela do Eclipse-Workspace no Eclipse IDE. Nela, a pasta "Project Explorer" está aberta, mostrando uma lista de arquivos. A pasta src está selecionada, e uma janela aparece sobreposta, com a opção "New" selecionada. Dela, sai uma outra janela, onde a opção "Other" está marcada.

Na janela apresentada, digite folder. Dentre as opções apresentadas, selecione a opção Folder para poder criar uma pasta. Em seguida, acione a opção Next.

#A imagem mostra a janela "New". Nela está escrito: "Select a wizard. Create a new folder resource". Logo abaixo, há o campo "Wizards", onde está escrito "folder" e, logo depois, a pasta aparece selecionada. Na parte inferior da janela, há os botões "Next" e "Cancel".

Na tela seguinte, insira o nome da pasta que será criada, ou seja, digite META-INF. Depois, acione a opção Finish localizada no canto inferior direito.

#A imagem mostra a janela "New Folder". Nela está escrito: "Create a new folder resource". Logo abaixo, há o campo "Enter or select the parent folder", onde está escrito "CatalogoDeFilmes/src". A pasta "src" aparece selecionada em seguida. Depois, há o campo "Folder name" preenchido com o texto "META-INF" e o botão "Advanced". Na parte inferior da janela, há os botões "Back", "Finish" e "Cancel".

Com isso, a pasta META-INF será criada no seu projeto. Uma vez que a pasta foi criada, o próximo passo é a criação do arquivo persistence.xml. Assim, você deve navegar até a pasta criada e acionar as opções, novamente, através do teclado utilizando a tecla aplicação. Depois, navegue até New e, em seguida, Other, mais uma vez.

#A imagem mostra a tela do Eclipse-Workspace no Eclipse IDE. Nela, a pasta “Project Explorer” está aberta, mostrando uma lista de arquivos. A pasta src está selecionada, e uma janela aparece sobreposta, com a opção “New” selecionada. Dela, sai uma outra janela, onde a opção “Other” está marcada.

Na janela seguinte, digite xml e, dentre as opções apresentadas, escolha a opção XML File. Em seguida, acione a opção Next.

#A imagem mostra a janela ‘New’. Nela está escrito: “Select a wizard. Create a new XML File”. Logo abaixo, há o campo “Wizards”, onde está escrito “xml” e, depois, a pasta “XML File” aparece selecionada. Na parte inferior da janela, há os botões “Next” e “Cancel”.

Na tela seguinte, insira o nome do arquivo. Digite persistence.xml e acione a opção Finish.

#A imagem mostra a janela “New XML File”. Nela está escrito: “Create a new XML File”. Logo abaixo, há o campo “Enter or select the parent folder”, onde está escrito “CatalogoDeFilmes/src/META-INF” e, depois, a pasta “META-INF” aparece selecionada. Depois, há o campo “File name” preenchido com o texto “persistence.xml” e o botão “Advanced”. Na parte inferior da janela, há os botões “Back”, “Next”, “Finish” e “Cancel”.

Pronto! O arquivo foi criado.

O próximo passo que você deve executar é mapear quais entidades serão persistidas, que, por enquanto, será apenas a entidade Filme. Esses dados de configurações citados anteriormente ficam todos dentro de uma tag chamada <persistence-unit> (persistence unit). Essa tag fica localizada na parte superior do arquivo persistence.xml. Nessa tag, você deve informar o nome da unidade de persistência e esse também deve ser o mesmo nome do seu schema no banco de dados. Não se preocupe, pois mais à frente será apresentado como você criará um schema. Uma informação importante acerca do arquivo persistence.xml é que ele deve ser criado dentro da pasta META-INF, que se localiza dentro do pacote de código fonte do projeto.

Para o seu projeto, o arquivo com todos os dados preenchidos deverá ficar da seguinte maneira:

#Audiodescrição: segue código

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
        http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd"
    version="2.0">

    <!-- unidade de persistencia com o nome catalogofilmes -->

    <persistence-unit name="catalogofilmes">

        <!-- Implementação do JPA, no nosso caso Hibernate -->

        <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>

        <!-- Aqui são listadas todas as entidades -->

        <class>br.com.lead.modelo.Filme</class>

        <properties>

            <!-- Propriedades JDBC -->

            <property name="javax.persistence.jdbc.driver"
                value="com.mysql.jdbc.Driver" />

                <property name="javax.persistence.jdbc.url"
value="jdbc:mysql://localhost:3306/catalogofilmes?useTimezone=true&serverTimezone=UTC"
/>

            <property name="javax.persistence.jdbc.user" value="root" />

            <property name="javax.persistence.jdbc.password" value="admin" />

            <!-- Configurações específicas do Hibernate -->

            <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect" />

            <property name="hibernate.hbm2ddl.auto" value="update" />

            <property name="hibernate.show_sql" value="true" />

            <property name="hibernate.format_sql" value="true" />

        </properties>

    </persistence-unit>
```

</persistence>

Agora, você deve acessar o seu banco de dados para criar o schema que foi definido no arquivo persistence.xml. Você irá utilizar o Comand Line Cliente do MySQL que já vem na instalação desse banco de dados. Para isso, abra MySQL, e o programa irá solicitar a inserção de uma senha. Esta deve ser a mesma senha informada para o acesso ao banco de dados, que foi configurada no momento da instalação do MySQL. Após informar a senha, você irá obter acesso ao banco de dados.

#A imagem mostra a tela de início do “MySQL 8.0 Command Line Client”. Sobre fundo preto, há uma saudação e informações do banco de dados.

Feito isso, você deve agora inserir o comando create database catalogodefيلمes; (create database catalogodefيلمes ponto e virgula) para criar o schema catalogodefيلمes. Em seguida, você irá listar os schemas criados, para isso, use o comando show schemas. Concluída essa etapa, o schema catalogodefيلمes deve ser informado.

#A imagem mostra a tela de início do “MySQL 8.0 Command Line Client”. Sobre fundo preto, há uma saudação e informações do banco de dados. Entre as informações do banco de dados, há o comando “mysql>show schemas;” e o código:

```
“Database
catalogodefيلمes
information_schema
mysql
performance_schema
sakila
sys
world
```

Em seguida, está escrito: “7 rows in set (0.00 sec)
mysql>”

Pronto! Até aqui, você conheceu como configurar o seu projeto para trabalhar com JPA e Hibernate, criou o arquivo de persistence.xml e criou o schema do banco de dados através do MySQL. Na próxima videoaula você irá realizar o mapeamento das suas entidades para poder então persistir informação no seu banco de dados.

Até lá!