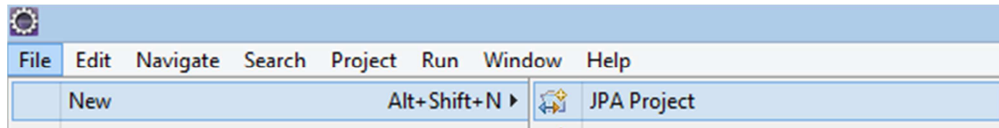


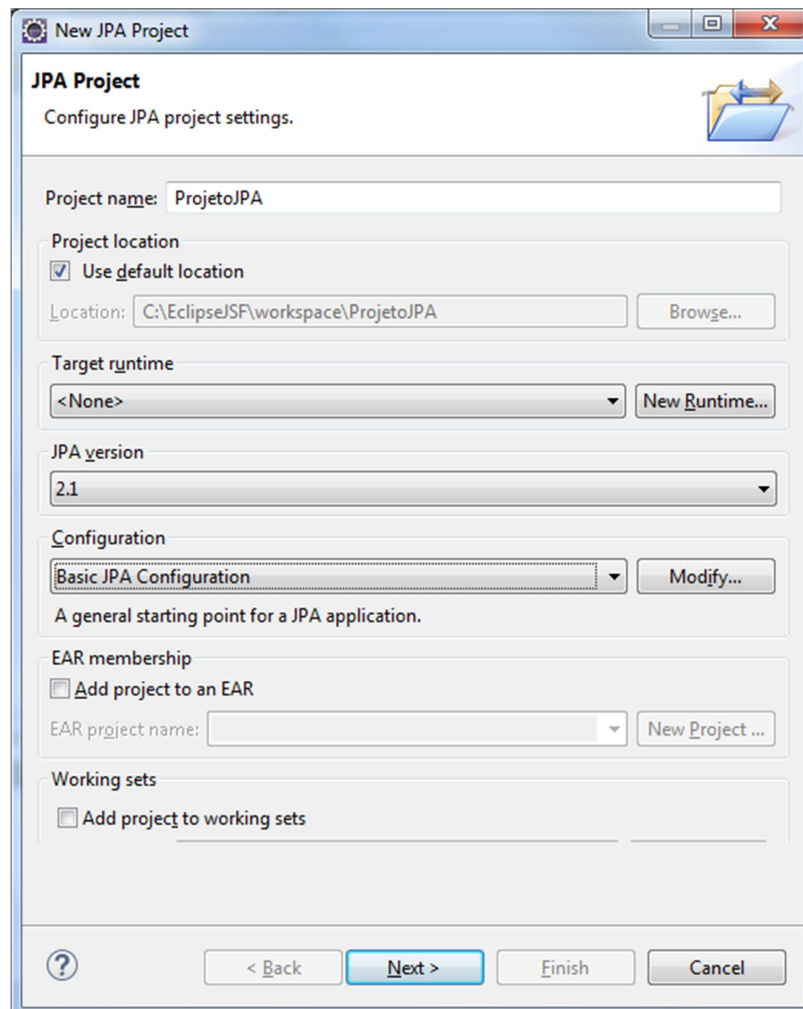
Roteiro com alternativa para implementar um projeto com JPA no Eclipse

Criar o banco de dados usando o cliente do SGBD (pgAdmin)

Criar um projeto com JPA

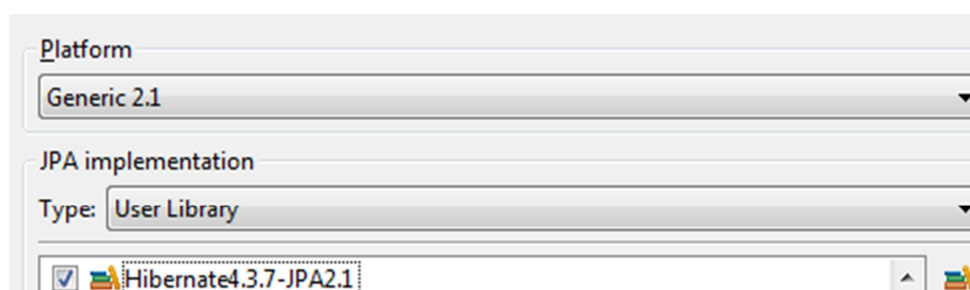


Informar dados do projeto



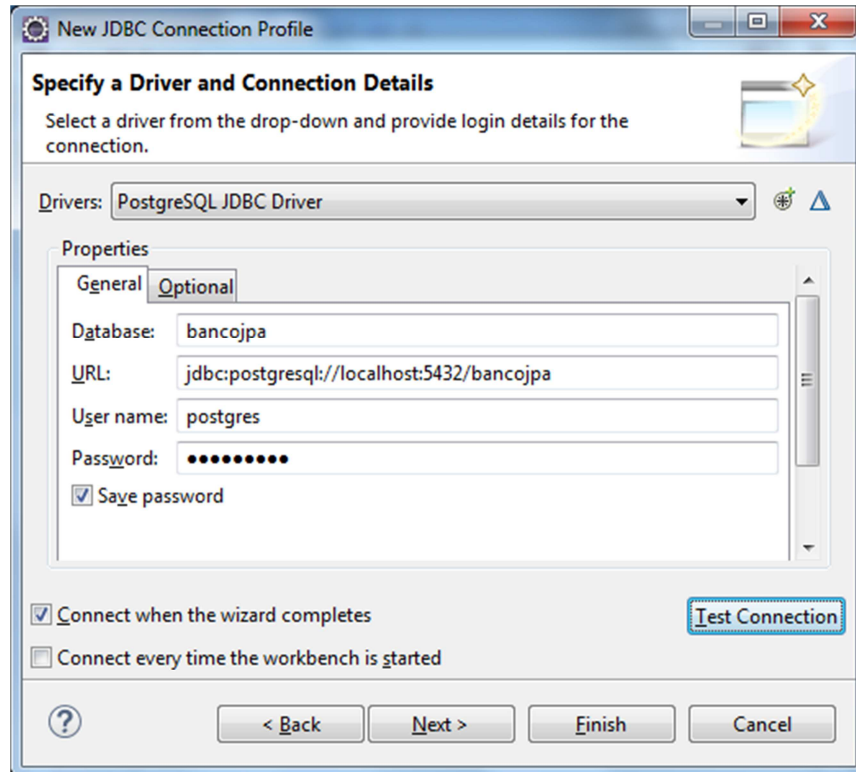
Clicar next e novamente next para as configurações do JPA

- Informar Generic 2.1
- Informar biblioteca do Hibernate. Se não existir, clicar em Manage Libraries e adicionar bibliotecas do Hibernate.



Clicar em Add Connection

- Selecionar PostgreSQL, informar nome da conexão e next
- Em drivers, clicar em New Driver Definition
 - Em JAR List, limpar e adicionar driver desejado > OK
- Informar as propriedades para a conexão JDBC > Test Connection

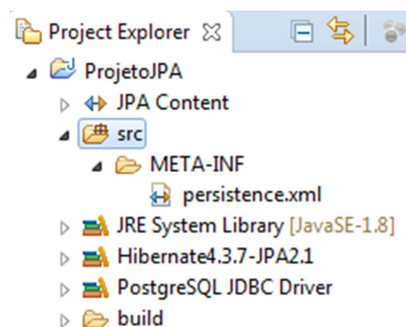


- Finish

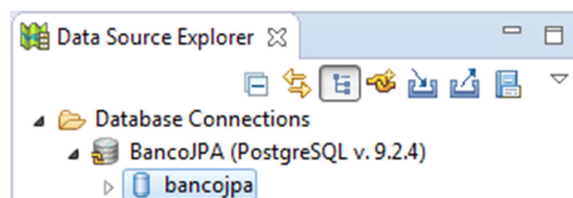
Marcar Add driver library to build path

Finish > Se solicitar para mudar para perspectiva JPA, aceitar.

O projeto JPA será criado com as configurações necessárias.



OBS: Os dados da conexão criada podem ser modificados na área "Data Source Explorer". (Window > Show View > Data Management > Data Source Explorer)





PROGRAMAÇÃO WEB COM JAVA

Prof. Jaqson Dalbosco – jaqson@upf.br
Universidade de Passo Fundo

Configurar o arquivo persistence.xml (Unidade de persistência)

- Abrir o arquivo persistence.xml. Na guia Connection, configurar:

Transaction type: Resource Local

JDBC Connections properties: Clicar em Populate from Connection > escolher banco criado > Ok

Clicar na guia Source:

- Adicionar elemento XML property para geração automática do BD

<property name="hibernate.hbm2ddl.auto" value="create-drop"/>

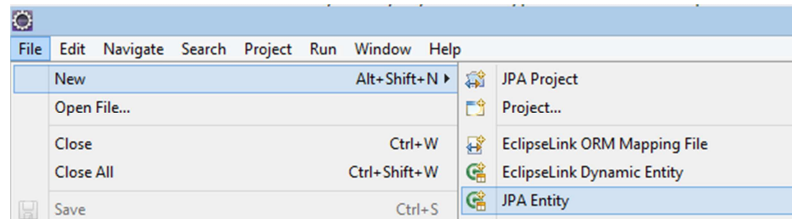
```
*persistence.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/200
3   <persistence-unit name="ProjetoJPA" transaction-type="RESOURCE_LOCAL">
4     <properties>
5       <property name="javax.persistence.jdbc.url" value="jdbc:postgresql://localhost:5432/bancojpa"/>
6       <property name="javax.persistence.jdbc.user" value="postgres"/>
7       <property name="javax.persistence.jdbc.password" value="masterkey"/>
8       <property name="javax.persistence.jdbc.driver" value="org.postgresql.Driver"/>
9       <property name="hibernate.hbm2ddl.auto" value="create-drop"/>
10    </properties>
11  </persistence-unit>
12 </persistence>
```



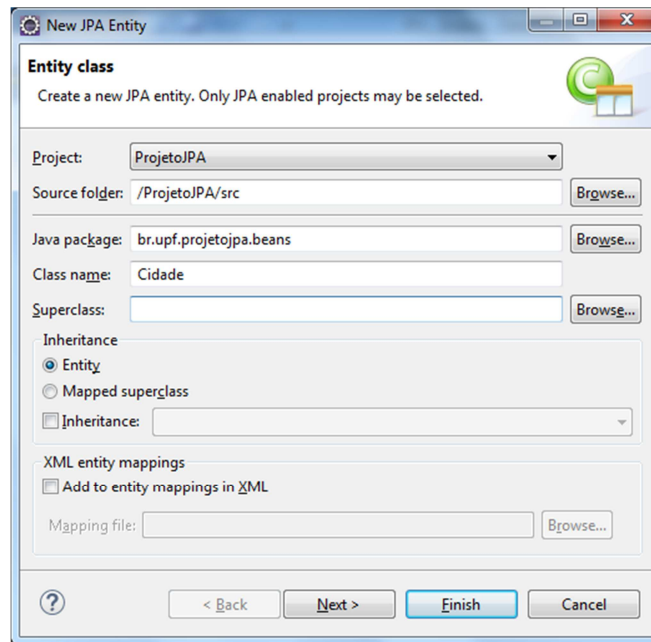
PROGRAMAÇÃO WEB COM JAVA

Prof. Jaqson Dalbosco – jaqson@upf.br
Universidade de Passo Fundo

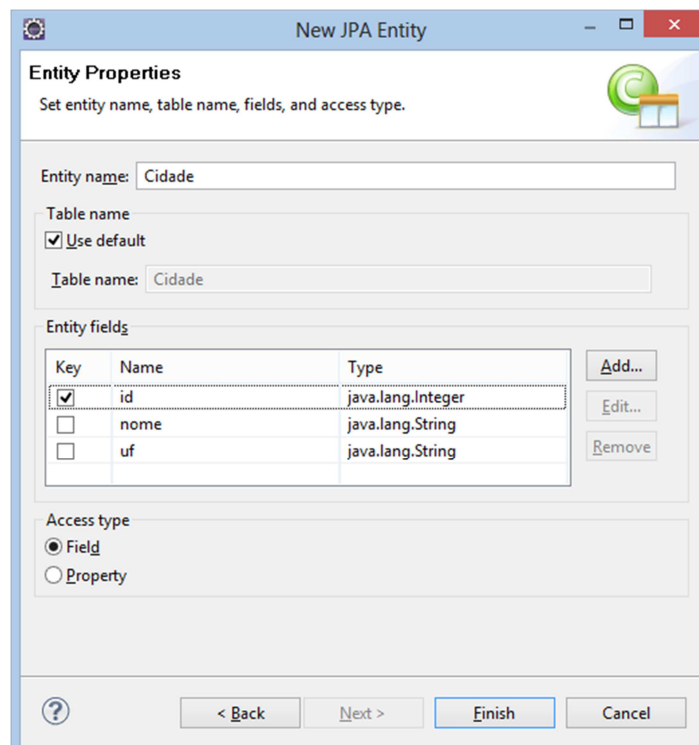
Criar uma classe de entidade persistente com JPA



Informar dados para a classe a ser criada



Clicar next > Informar os atributos para a classe. Marcar atributo identificador (key).



Clicar Finish. OBS: A classe será gerada com os atributos, métodos e anotações básicas JPA para ter uma entidade no banco de dados. A classe também é adicionada automaticamente na unidade de persistência.



PROGRAMAÇÃO WEB COM JAVA

Prof. Jaqson Dalbosco – jaqson@upf.br
Universidade de Passo Fundo

A IDE pode acusar erro na anotação @Entity. Para resolver pode desativar na IDE a validação JPA para build, em Window > Preferences > Validation.

```
*Cidade.java
1 package br.upf.projetojpa.beans;
2
3 import java.io.Serializable;
4
5
6
7
8 @Entity
9 public class Cidade implements Serializable {
10     @Id
11     private Integer id;
12     private String nome;
13     private String uf;
```

Criando uma classe para testar a geração da factory e geração de tabelas no BD

- File > new > class (marcar para gerar o método main)

Criar uma factory para persistência com EntityManager

```
persistence.xml  Cidade.java  TesteGeracaoBD.java
package br.upf.projetojpa.testes;

import javax.persistence.Persistence;

public class TesteGeracaoBD {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Persistence.createEntityManagerFactory("ProjetoJPA");
    }
}
```



PROGRAMAÇÃO WEB COM JAVA

Prof. Jaqson Dalbosco – jaqson@upf.br
Universidade de Passo Fundo

Tabela gerada ao criar a factory.

