

Mapeamento Objeto/Relacional com JPA

Evandro César Freiburger

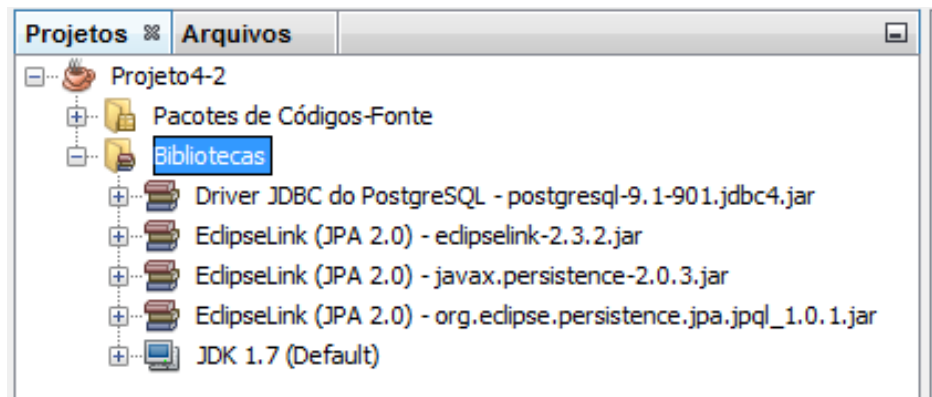
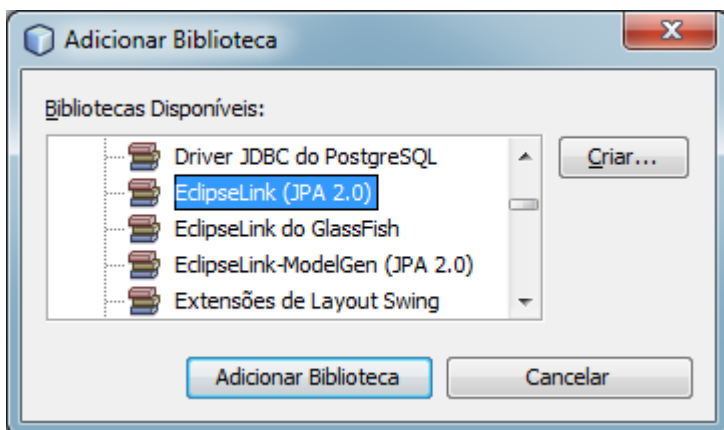
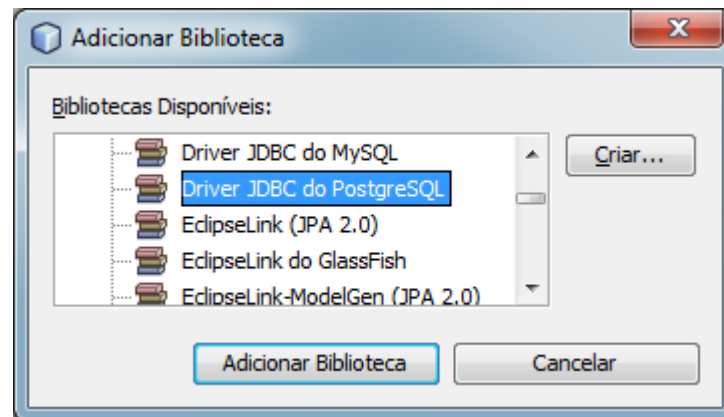
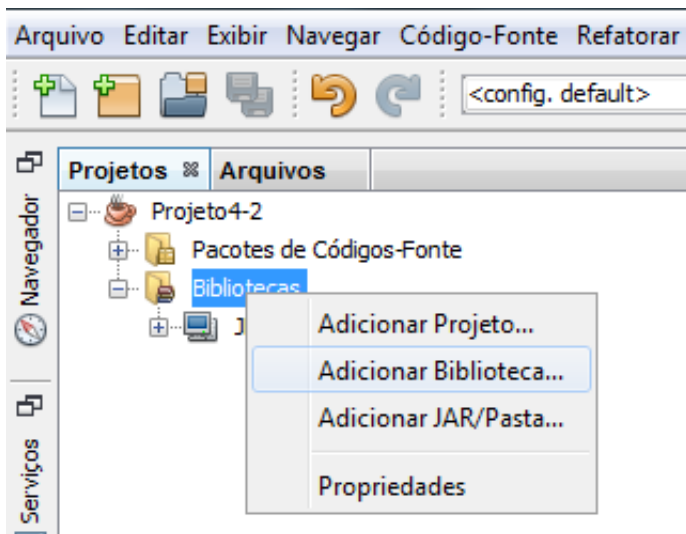
Evandro.freiberger@cba.ifmt.edu.br

JPA 2.0

- ✓ Criar um projeto **Java Application**
- ✓ Adicionar a biblioteca do **Driver** do BD
- ✓ Adicionar a biblioteca da **JPA 2**
- ✓ Criar o arquivo **persistence.xml**
- ✓ Mapear as entidades (**@Entity**)
- ✓ Mapear atributos das entidades
- ✓ Mapear as relações entre entidades
- ✓ Manipular as entidades por meio do **EntityManager**

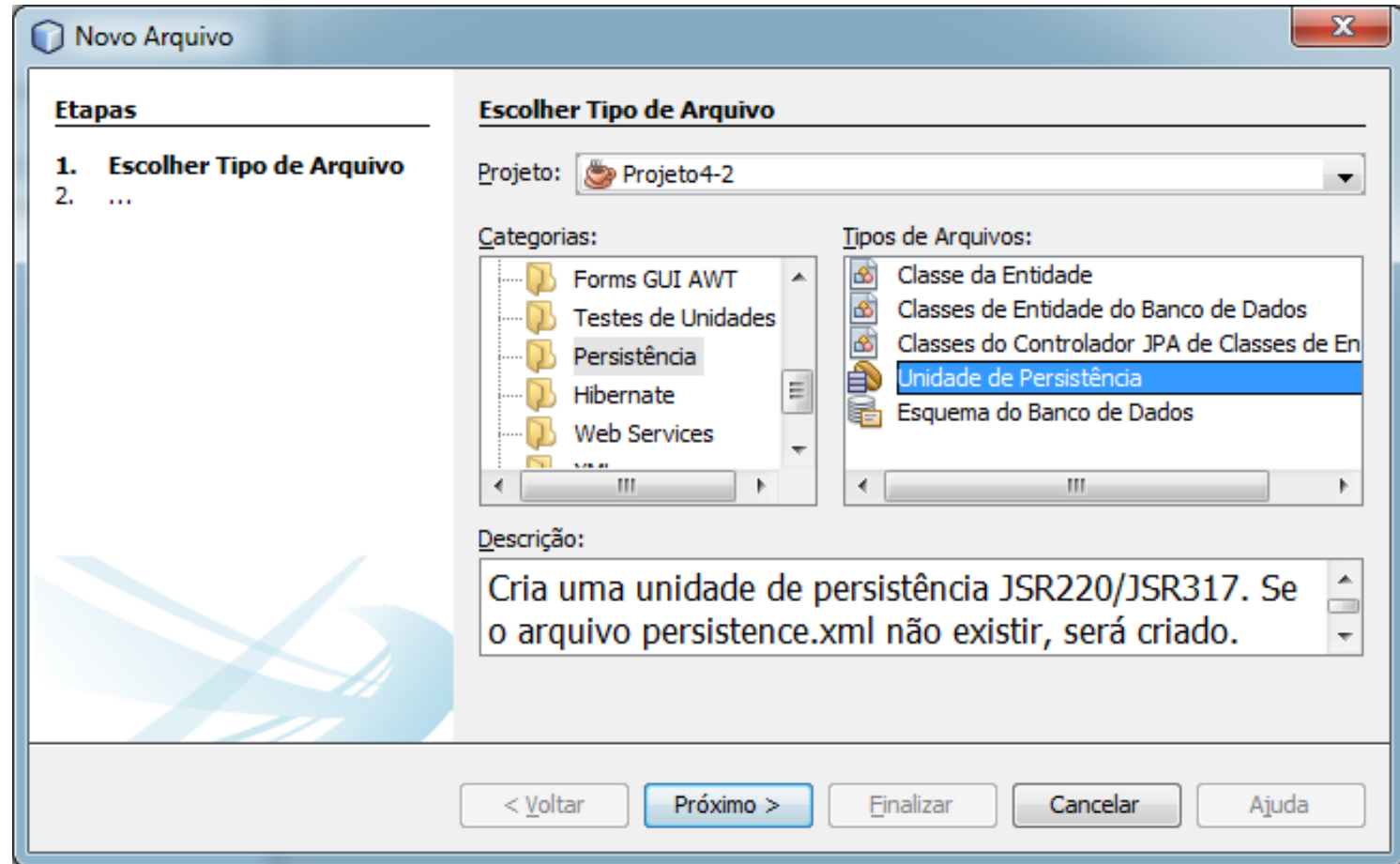
JPA 2.0

Adicionar ao projeto o Driver BD e a JPA 2.0



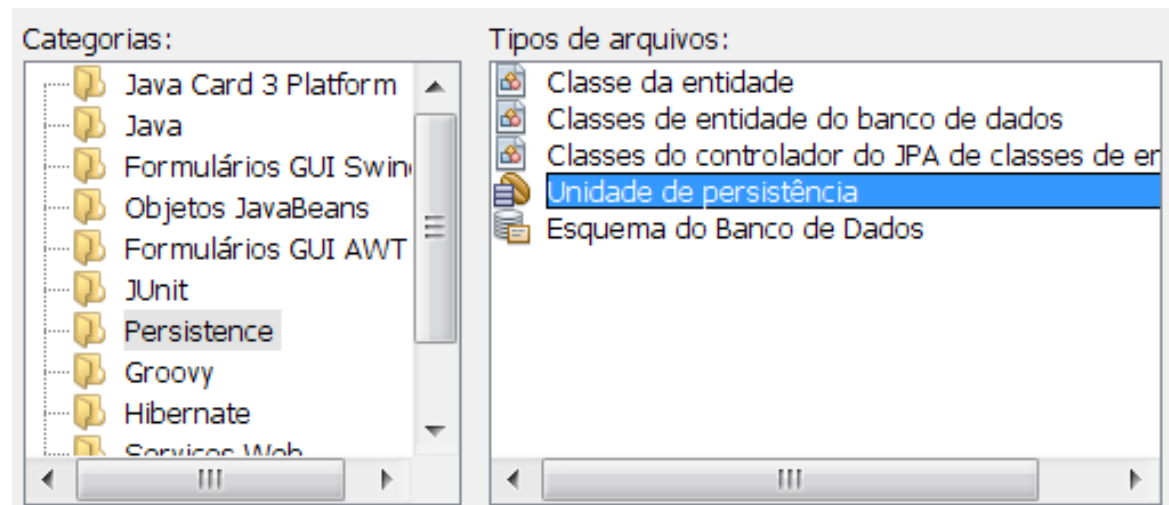
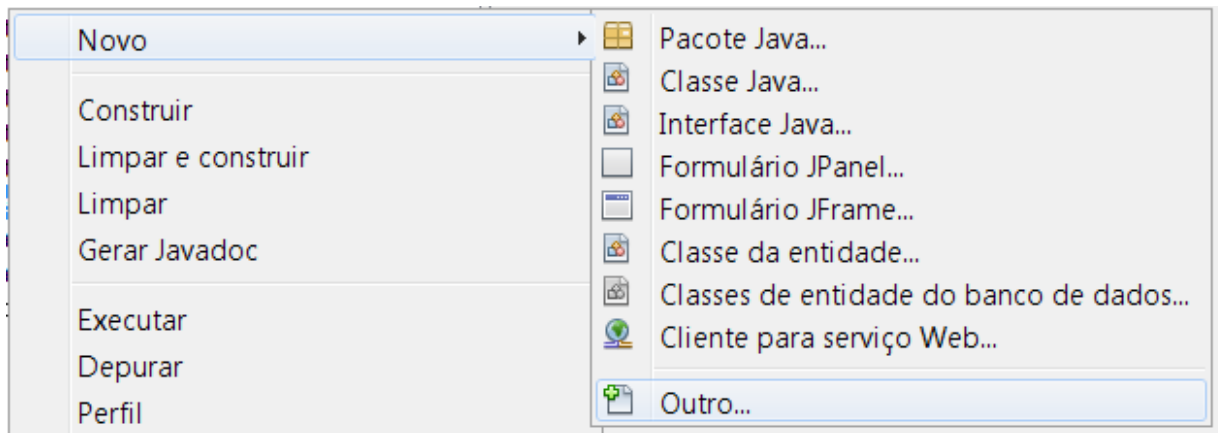
JPA 2.0

- ✓ Criar o arquivo persistence.xml



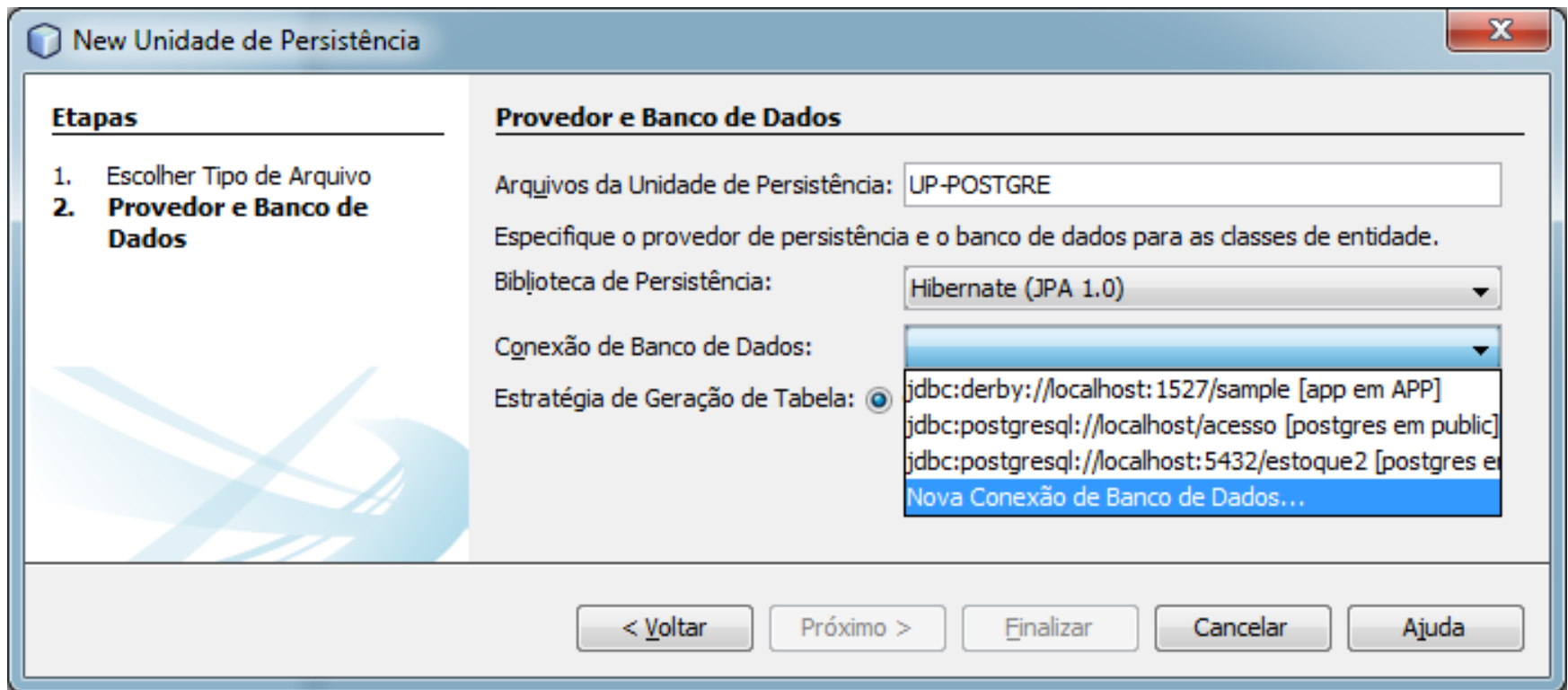
JPA/Hibernate no NetBeans

✓ Adicionar uma Unidade de Persistência ao Projeto



JPA/Hibernate no NetBeans

- ✓ Configurando a Unidade de Persistência
 - Nome da unidade: UP-POSTGRE
 - Biblioteca de Persistência: EclipseLink (JPA 2.1)



JPA/Hibernate no NetBeans

- ✓ Configurando conexão de banco de dados

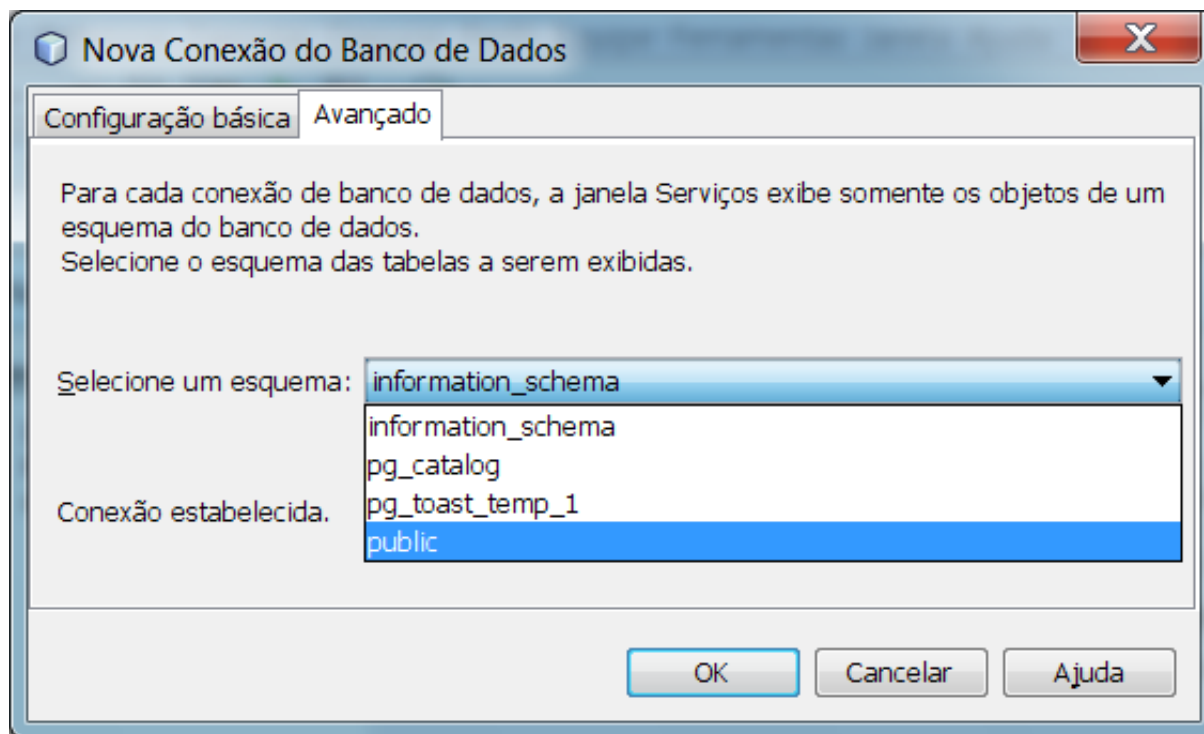
The screenshot shows the 'Assistente de Nova Conexão' (New Connection Wizard) window in NetBeans. The title bar reads 'Assistente de Nova Conexão'. The main tab is 'Personalizar Conexão'. The fields are filled as follows:

- Nome do Driver: PostgreSQL
- Host: localhost
- Porta: 5432
- Banco de dados: estoque2
- Nome do Usuário: postgres
- Senha: (masked with dots)
- ☐ Lembrar senha
- Testar Conexão (button)
- JDBC URL: jdbc:postgresql://localhost:5432/estoque2

At the bottom, there is a status message: 'i Conexão Bem-sucedida.' (Connection Successful). Navigation buttons at the bottom include '< Voltar', 'Próximo >', 'Finalizar', 'Cancelar', and 'Ajuda'.

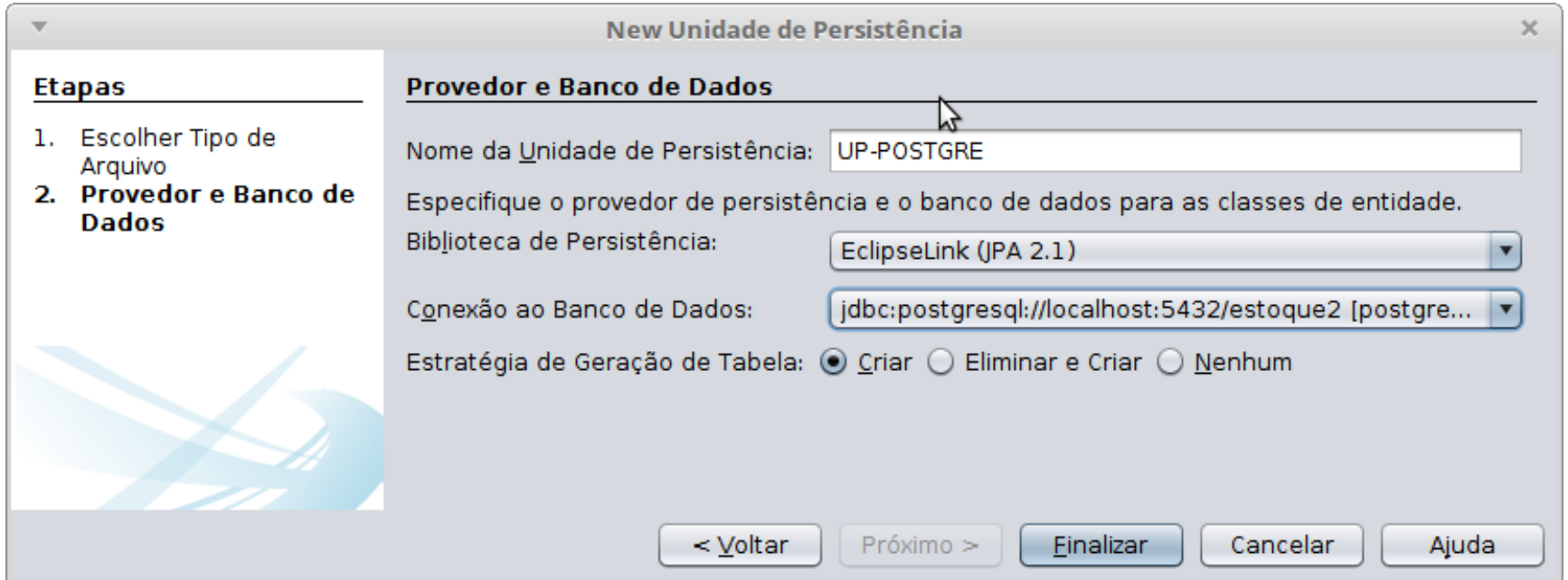
JPA/Hibernate no NetBeans

- ✓ Definir o esquema de banco as ser usado, escolha public



JPA/Hibernate no NetBeans

✓ Estratégia de geração das tabelas



The screenshot shows the 'New Unidade de Persistência' (New Persistence Unit) dialog box in NetBeans. The dialog is titled 'New Unidade de Persistência' and has a close button (X) in the top right corner. On the left side, there is a sidebar titled 'Etapas' (Steps) with two steps: '1. Escolher Tipo de Arquivo' and '2. Provedor e Banco de Dados'. The second step is currently selected. The main area of the dialog is titled 'Provedor e Banco de Dados' and contains the following fields and options:

- Nome da Unidade de Persistência:** A text field containing 'UP-POSTGRE'.
- Especifique o provedor de persistência e o banco de dados para as classes de entidade.**
- Biblioteca de Persistência:** A dropdown menu showing 'EclipseLink (JPA 2.1)'.
- Conexão ao Banco de Dados:** A dropdown menu showing 'jdbc:postgresql://localhost:5432/estoque2 [postgre...'.
- Estratégia de Geração de Tabela:** Three radio buttons: 'Criar' (selected), 'Eliminar e Criar', and 'Nenhum'.

At the bottom of the dialog, there are five buttons: '< Voltar', 'Próximo >', 'Finalizar', 'Cancelar', and 'Ajuda'.

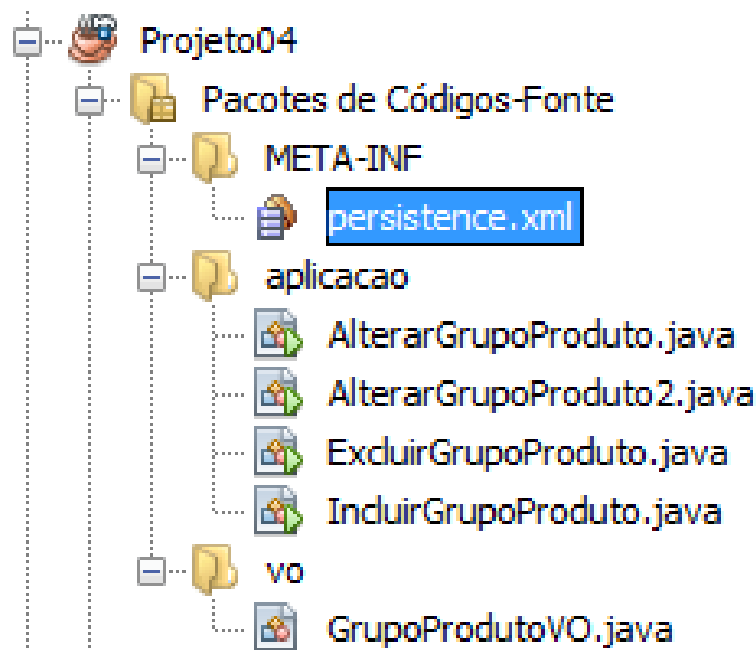
JPA 2.0

✓ Criar o arquivo persistence.xml

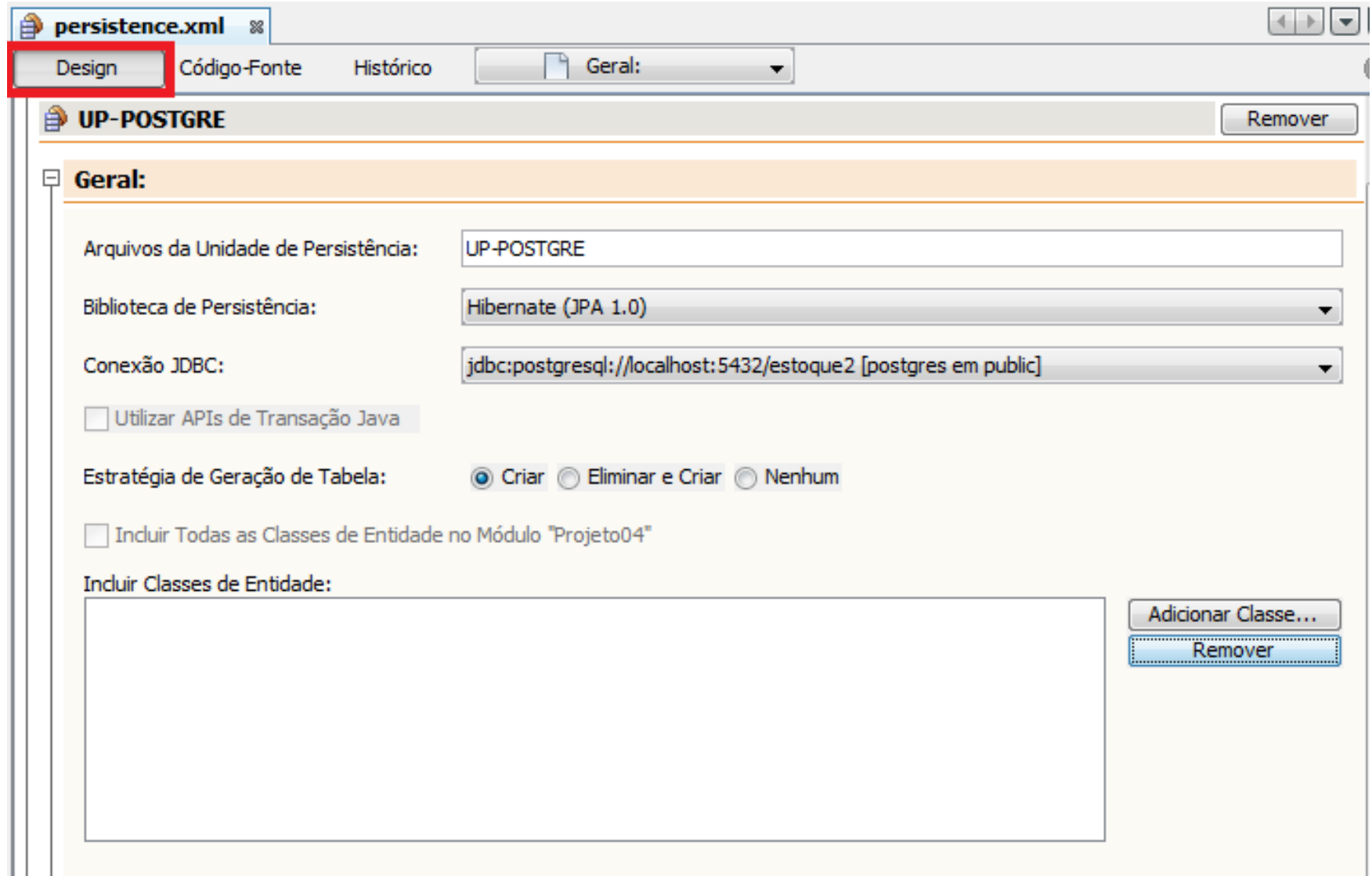
```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
  <persistence-unit name="UP-POSTGRE" transaction-type="RESOURCE_LOCAL">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:postgresql://localhost:5432/estoque2"/>
      <property name="javax.persistence.jdbc.user" value="postgres"/>
      <property name="javax.persistence.jdbc.driver" value="org.postgresql.Driver"/>
      <property name="javax.persistence.jdbc.password" value="postgres"/>
      <property name="javax.persistence.schema-generation.database.action" value="create"/>
    </properties>
  </persistence-unit>
</persistence>
```

JPA/Hibernate no NetBeans

- ✓ As Unidades de Persistência ficam no arquivo persistence.xml na pasta META-INF

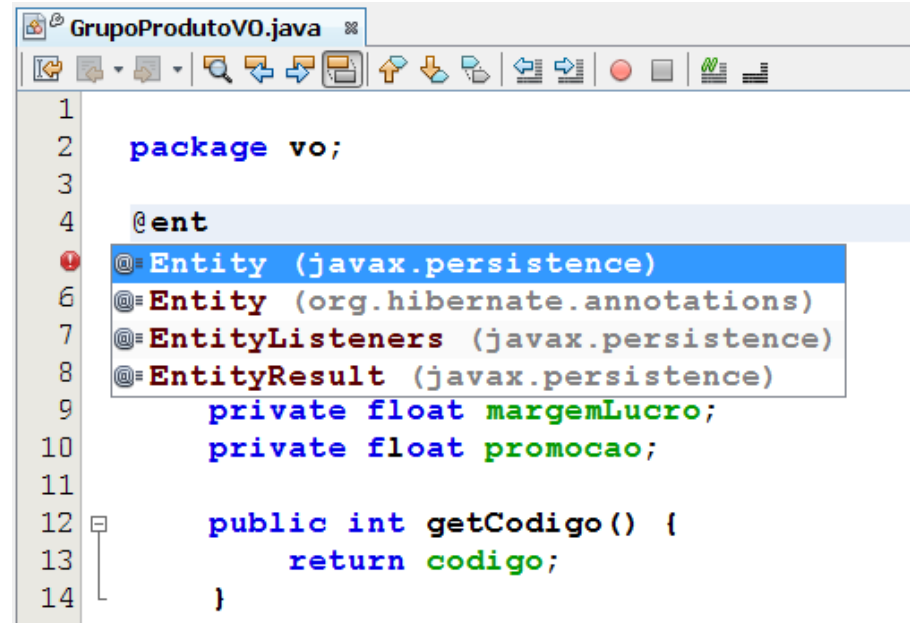


JPA/Hibernate no NetBeans



JPA/Hibernate no NetBeans

- ✓ Mapeamento dos VO's
- ✓ Sempre que possível usar as anotações do pacote da JPA `javax.persistence`
- ✓ Se usar `org.hibernate.annotations` estará vinculando o projeto ao Hibernate



```
1 package vo;
2
3
4 @ent
5 @Entity (javax.persistence)
6 @Entity (org.hibernate.annotations)
7 @EntityListeners (javax.persistence)
8 @EntityResult (javax.persistence)
9 private float margemLucro;
10 private float promocao;
11
12 public int getCodigo() {
13     return codigo;
14 }
```

JPA – Mapeamento das Entidades

```
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "grupoProduto")
public class GrupoProdutoVO {

    @Id
    @GeneratedValue(strategy=GenerationType.SEQUENCE)
    private int codigo;

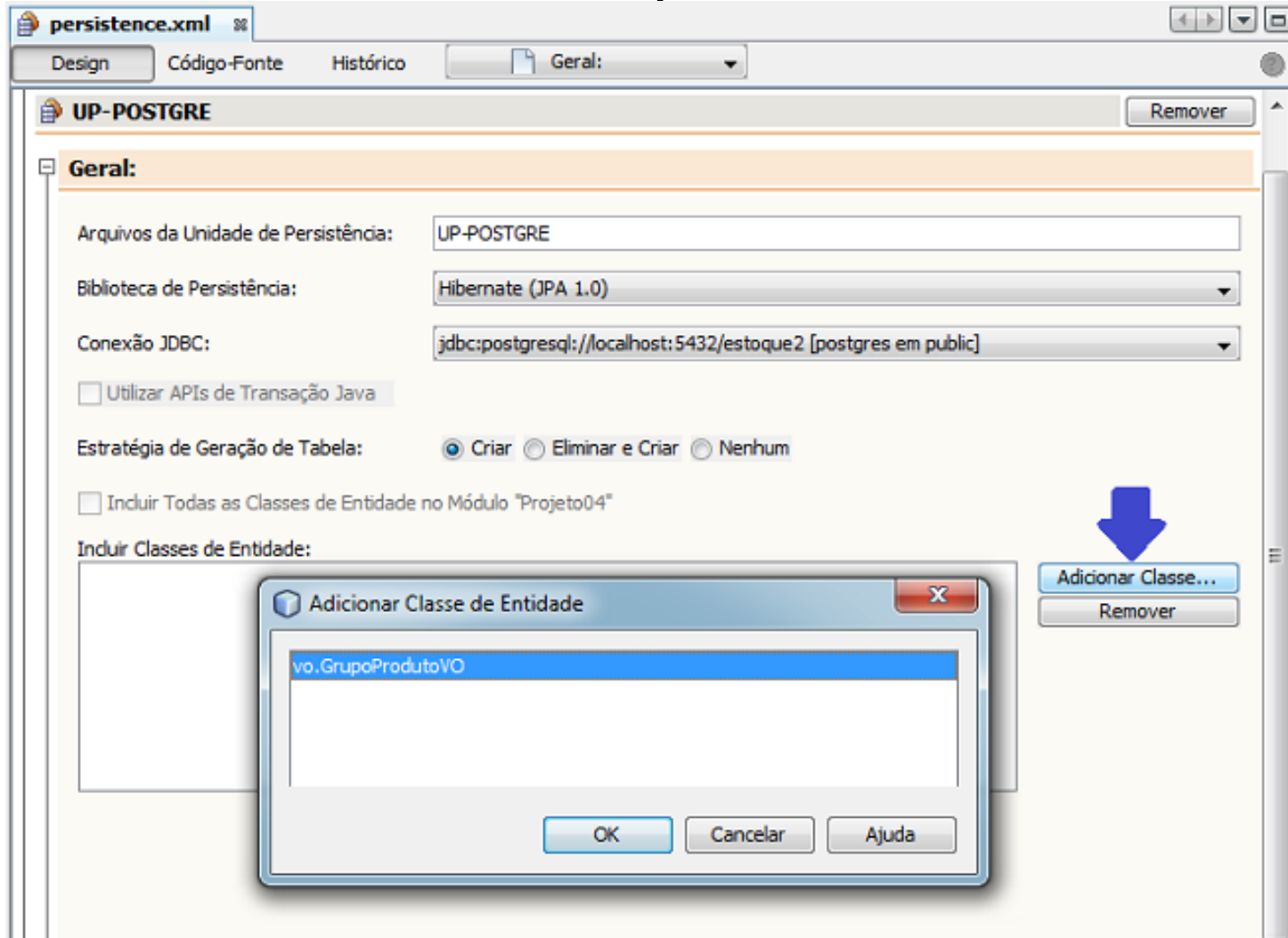
    @Column(length=40, nullable=false)
    private String nome;

    @Column(nullable=false, precision=5, scale=2)
    private float margemLucro;

    @Column(nullable=false, precision=5, scale=2)
    private float promocao;
```

JPA

✓ Adicionar a entidade mapeada



JPA

✓ Classes Importantes da JPA

- **Persistence** – usada para criar fábricas de gerenciadores de entidades e contexto de persistência;
- **EntityManagerFactory** – representa os objetos de fábrica de gerenciadores – usada para criar os gerenciadores de entidades;
- **EntityManager** – representa os objetos gerenciadores de entidades






JPA

```
2 import javax.persistence.EntityManager;
3 import javax.persistence.EntityManagerFactory;
4 import javax.persistence.Persistence;
5 import javax.swing.JOptionPane;
6 import vo.GrupoProdutoVO;
7 public class IncluirGrupoProduto {
8     public static void main(String args[]){
9         EntityManagerFactory fabricaEntityManager=null;
10        EntityManager entityManager=null;
11        try {
12            fabricaEntityManager = Persistence.createEntityManagerFactory("UP-POSTGRE");
13            entityManager = fabricaEntityManager.createEntityManager();
14        } catch (Exception ex) {
15            JOptionPane.showMessageDialog(null, ex.toString());
16        }
17        GrupoProdutoVO grupoVO = new GrupoProdutoVO();
18        grupoVO.setNome("Grupo Teste1");
19        grupoVO.setMargemLucro(40);
20        grupoVO.setPromocao(0);
21        entityManager.getTransaction().begin();
22        entityManager.persist(grupoVO);
23        entityManager.getTransaction().commit();
24    }
25 }
```

JPA

```
8 public class AlterarGrupoProduto {
9     public static void main(String args[]) {
10         EntityManagerFactory fabricaEntityManager = null;
11         EntityManager entityManager = null;
12         try {
13             fabricaEntityManager = Persistence.createEntityManagerFactory("UP-POSTGRE");
14             entityManager = fabricaEntityManager.createEntityManager();
15         } catch (Exception ex) {
16             JOptionPane.showMessageDialog(null, ex.toString());
17         }
18         entityManager.getTransaction().begin();
19         GrupoProdutoVO grupoVO = null;
20         try {
21             Query query = entityManager.createQuery("FROM GrupoProdutoVO WHERE TRIM(NOME) = 'Grupo Teste1'");
22             grupoVO = (GrupoProdutoVO) query.getSingleResult();
23         } catch (Exception ex) {
24             JOptionPane.showMessageDialog(null, ex.toString());
25         }
26         grupoVO.setMargemLucro(50);
27         entityManager.getTransaction().commit();
28     }
29 }
```

JPA

```
9 public static void main(String args[]) {  
10     EntityManagerFactory fabricaEntityManager = null;  
11     EntityManager entityManager = null;  
12     try {  
13         fabricaEntityManager = Persistence.createEntityManagerFactory("UP-POSTGRE");  
14         entityManager = fabricaEntityManager.createEntityManager();  
15     } catch (Exception ex) {  
16         JOptionPane.showMessageDialog(null, ex.toString());  
17     }  
18     entityManager.getTransaction().begin();   
19     GrupoProdutoVO grupoVO = null;  
20     try {  
21         Query query = entityManager.createQuery("FROM GrupoProdutoVO WHERE TRIM(NOME) = 'Grupo Teste1'");  
22         grupoVO = (GrupoProdutoVO) query.getSingleResult();  
23     } catch (Exception ex) {  
24         JOptionPane.showMessageDialog(null, ex.toString());  
25     }  
26     entityManager.getTransaction().commit();   
27     entityManager.getTransaction().begin();   
28     grupoVO.setMargemLucro(60);  
29     entityManager.merge(grupoVO);   
30     entityManager.getTransaction().commit();   
31 }  
32 }
```

JPA

```
9 public static void main(String args[]) {  
10     EntityManagerFactory fabrigaEntityManager = null;  
11     EntityManager entityManager = null;  
12     try {  
13         fabrigaEntityManager = Persistence.createEntityManagerFactory("UP-POSTGRE");  
14         entityManager = fabrigaEntityManager.createEntityManager();  
15     } catch (Exception ex) {  
16         JOptionPane.showMessageDialog(null, ex.toString());  
17     }  
18     entityManager.getTransaction().begin();  
19     GrupoProdutoVO grupoVO = null;  
20     try {  
21         Query query = entityManager.createQuery("FROM GrupoProdutoVO WHERE TRIM(NOME) = 'Grupo Teste1'");  
22         grupoVO = (GrupoProdutoVO) query.getSingleResult();  
23     } catch (Exception ex) {  
24         JOptionPane.showMessageDialog(null, ex.toString());  
25     }  
26     entityManager.getTransaction().commit();  
27     entityManager.getTransaction().begin();  
28     entityManager.remove(grupoVO);  
29     entityManager.getTransaction().commit();  
30 }  
31 }
```