

# **Mapeamento Objeto/Relacional com JPA**

**Evandro César Freiburger**

**[Evandro.freiberger@cba.ifmt.edu.br](mailto:Evandro.freiberger@cba.ifmt.edu.br)**

# JPA/Hibernate

## ✓ Uso da API JPA

- Inclusão do *driver* do banco de dados
- Inclusão da API JPA/Hibernate
- Configuração do arquivo *persistence.xml*
- Mapear as entidades a serem persistidas
- Criar uma instância de *EntityManagerFactory*
  - `Persistence.createEntityManagerFactory("unidade")`
- Criar uma instância de *EntityManager*
  - `fabrigaEntityManager.createEntityManager()`
- Gerenciar as entidades com o objeto *entityManager*

# JPA/Hibernate

```
1 package aplicacao;
2 import ...
7 public class IncluirGrupoProduto {
8     public static void main(String args[]){
9         EntityManagerFactory fabrigaEntityManager;
10        EntityManager entityManager=null;
11        try {
12            fabrigaEntityManager = Persistence.createEntityManagerFactory("UP-POSTGRE");
13            entityManager = fabrigaEntityManager.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 }
```

Código de  
Inicialização da  
Unidade de  
Persistência

# JPA/Hibernate

- A Fábrica mantém uma instância global (static) da EntityManagerFactory.

```
1 package persistencia;
2 import ...
5 public class FabricaEntityManager {
6
7     private static EntityManagerFactory entityManagerFactory = null;
8
9     private FabricaEntityManager() {} // para impedir instanciamento
10
11     static {
12         entityManagerFactory = Persistence.createEntityManagerFactory("UP-POSTGRE");
13     }
14
15     public static EntityManager getEntityManager() throws PersistenciaException {
16         if (entityManagerFactory == null) {
17             throw new PersistenciaException("Unidade de Persistência não iniciada");
18         }
19         return entityManagerFactory.createEntityManager();
20     }
21 }
```

# JPA/Hibernate

- ✓ DAO – Genérico (super classe de todas os DAO's)

```
1 package persistencia;
2 import javax.persistence.EntityManager;
3
4 public class DAO<VO> {
5     protected EntityManager entityManager;
6
7     public DAO(EntityManager entityManager) {
8         this.entityManager = entityManager;
9     }
10
11     public void incluir(VO vo) throws PersistenciaException {
12         this.entityManager.persist(vo);
13     }
14
15     public void alterar(VO vo) throws PersistenciaException{
16         this.entityManager.merge(vo);
17     }
18 }
```

# JPA/Hibernate

- ✓ DAO – Genérico (super classe de todas os DAO's)

```
18
19 public void excluir(VO vo) throws PersistenciaException{
20     this.entityManager.remove(vo);
21 }
22
23 public void iniciarTransacao() {
24     this.entityManager.getTransaction().begin();
25 }
26
27 public void confirmarTransacao() {
28     this.entityManager.getTransaction().commit();
29 }
30
31 public void cancelarTransacao() {
32     this.entityManager.getTransaction().rollback();
33 }
34 }
```

# JPA/Hibernate

## ✓ DAO – Específico para GrupoProduto

```
1 package persistencia;
2 import ...
7 public class GrupoProdutoDAO extends DAO<GrupoProdutoVO> {
8
9     public GrupoProdutoDAO(EntityManager entityManager) throws PersistenciaException {
10         super(entityManager);
11     }
12
13     public GrupoProdutoVO buscarPorCodigo(int codigo) throws PersistenciaException {
14         return this.entityManager.find(GupoProdutoVO.class, codigo);
15     }
16
17     public List<GrupoProdutoVO> buscarTodos() throws PersistenciaException {
18         try {
19             Query query = this.entityManager.createQuery("FROM GupoProdutoVO ORDER BY NOME");
20             return query.getResultList();
21         } catch (PersistenceException ex) {
22             throw new PersistenciaException("Não foi possível obter a lista de Grupos de Produtos");
23         }
24     }
25 }
```

# JPA/Hibernate

## Usando a Classe DAO

```
1 package aplicacao;
2 import ...
7 public class IncluirGrupoProduto2 {
8     public static void main(String args[]) {
9         GrupoProdutoDAO grupoDAO = null;
10
11         GrupoProdutoVO grupoVO = new GrupoProdutoVO();
12         grupoVO.setNome(JOptionPane.showInputDialog("Forneça o nome do Grupo de Produto"));
13         grupoVO.setMargemLucro(Float.parseFloat(JOptionPane.showInputDialog("Forneça o % da margem de lucro")));
14         grupoVO.setPromocao(Float.parseFloat(JOptionPane.showInputDialog("Forneça o % de promoção")));
15         try {
16             grupoDAO = new GrupoProdutoDAO(FabricaEntityManager.getEntityManager());
17             grupoDAO.iniciarTransacao();
18             grupoDAO.incluir(grupoVO);
19             grupoDAO.confirmarTransacao();
20         } catch (PersistenciaException ex) {
21             JOptionPane.showMessageDialog(null, ex.toString());
22             grupoDAO.cancelarTransacao();
23         }
24     }
25 }
```



# JPA/Hibernate

- ✓ Classe de Negócio – usa a fábrica para gerar objetos EntityManager;
- ✓ Passa a instância de EntityManager para o DAO

```
1  package negocio;
2  import ...
7  public class GrupoProduto {
8
9      private GrupoProdutoDAO grupoDAO;
10
11     public GrupoProduto() throws NegocioException {
12         try {
13             grupoDAO = new GrupoProdutoDAO(FabricaEntityManager.getEntityManager());
14         } catch (PersistenciaException ex) {
15             throw new NegocioException("Erro ao iniciar Persistência " + ex.getMessage());
16         }
17     }
}
```

# JPA/Hibernate

- ✓ Classe de Negócio – reúne as operações e validações de negócio;

```
19 public void inserir(GrupoProdutoVO grupoVO) throws NegocioException {  
20  
21     String mensagemErros = this.validarDados(grupoVO);  
22  
23     if (!mensagemErros.isEmpty()) {  
24         throw new NegocioException(mensagemErros);  
25     }  
26  
27     try {  
28         grupoDAO.iniciarTransacao();  
29         grupoDAO.incluir(grupoVO);  
30         grupoDAO.confirmarTransacao();  
31     } catch (PersistenciaException ex) {  
32         grupoDAO.cancelarTransacao();  
33         throw new NegocioException("Erro ao incluir novo grupo de produto - " + ex.getMessage());  
34     }  
35 }
```

# JPA/Hibernate

## ✓ Classe de Negócio – validações de negócio;

```
90 private String validarDados(GrupoProdutoVO grupoVO) {  
91  
92     String mensagemErros = "";  
93  
94     if (grupoVO.getNome().length() == 0) {  
95         mensagemErros += "Nome não pode ser vazio";  
96     }  
97  
98     if (grupoVO.getMargemLucro() < 0) {  
99         mensagemErros += "\nA margem de Lucro não pode ser negativa";  
100    }  
101  
102    if (grupoVO.getPromocao() < 0 || grupoVO.getPromocao() > 100) {  
103        mensagemErros += "\nO Percentual de Promoção deve ser entre 0 e 100";  
104    }  
105  
106    return mensagemErros;  
107 }
```

# JPA/Hibernate

## ✓ Usando a classe de Negócio/Domínio

```
1 package aplicacao;
2 import ...
9 public class IncluirGrupoProduto3 {
10     public static void main(String args[]) {
11         GrupoProdutoVO grupoVO = new GrupoProdutoVO();
12         grupoVO.setNome(JOptionPane.showInputDialog("Forneça o nome do Grupo de Produto"));
13         grupoVO.setMargemLucro(Float.parseFloat(JOptionPane.showInputDialog("Forneça o % da margem de lucro")));
14         grupoVO.setPromocao(Float.parseFloat(JOptionPane.showInputDialog("Forneça o % de promoção")));
15         try {
16             GrupoProduto grupoProduto = new GrupoProduto();
17             grupoProduto.inserir(grupoVO);
18         } catch (NegocioException ex) {
19             JOptionPane.showMessageDialog(null, ex.toString());
20         }
21     }
22 }
```

# JPA/Hibernate

## ✓ Classe ProdutoVO

```
@Entity
@Table(name = "produto")
public class ProdutoVO implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private int codigo;
    @Column(length = 50, nullable = false)
    private String Nome;
    @Column(nullable = false)
    private int estoque;
    @Column(nullable = false, precision = 8, scale = 2)
    private float valorCompra;
    @Column(nullable = false, precision = 5, scale = 2)
    private float margemLucro;
    @Column(nullable = false, precision = 5, scale = 2)
    private float promocao;
    @Transient
    private float venda;
    @ManyToOne(fetch=FetchType.EAGER, cascade={CascadeType.REFRESH})
    private GrupoProdutoVO grupo;
```

# JPA/Hibernate

## ✓ Classe BaixaVO

```
@Entity
@Table(name = "registrobaixa")
public class BaixaVO implements Serializable {

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

    @Column(nullable=false)
    private int motivo;

    @Column(nullable=false)
    @Temporal(TemporalType.TIMESTAMP)
    private Date dataRegistro;

    @Column(nullable=false)
    private int quantidade;

    @ManyToOne(fetch=FetchType.EAGER, cascade={CascadeType.MERGE, CascadeType.REFRESH})
    private ProdutoVO produto;
```

# JPA/Hibernate

## ✓ Persistence.xml

```
<persistence-unit name="UP-POSTGRE" transaction-type="RESOURCE_LOCAL">
  <provider>org.hibernate.ejb.HibernatePersistence</provider>
  <class>vo.GrupoProdutoVO</class>
  <class>vo.ProdutoVO</class>
  <class>vo.BaixaVO</class>
  <properties>
    <property name="hibernate.connection.driver_class" value="org.postgresql.Driver"/>
    <property name="hibernate.connection.username" value="postgres"/>
    <property name="hibernate.connection.password" value="postgres"/>
    <property name="hibernate.connection.url" value="jdbc:postgresql://localhost:5432/estoque"/>
    <property name="hibernate.cache.provider_class" value="org.hibernate.cache.NoCacheProvider"/>
    <property name="hibernate.dialect" value="org.hibernate.dialect.PostgreSQLDialect"/>
    <property name="hibernate.archive.autodetection" value="none"/>
    <property name="hibernate.show_sql" value="true"/>
    <property name="hibernate.format_sql" value="true"/>
    <property name="hibernate.hbm2ddl.auto" value="update"/>
  </properties>
</persistence-unit>
```

# JPA/Hibernate

```
package persistencia;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;
import javax.persistence.Query;
import vo.ProdutoVO;
public class ProdutoDAO extends DAO<ProdutoVO> {

    public ProdutoDAO(EntityManager entityManager) throws PersistenciaException {
        super(entityManager);
    }

    public ProdutoVO buscarPorCodigo(int codigo) throws PersistenciaException {
        try {
            return this.entityManager.find(ProdutoVO.class, codigo);
        } catch (PersistenceException ex) {
            throw new PersistenciaException("Não foi possível obter a lista de Grupos de Produtos");
        }
    }
}
```

Continua...



# JPA/Hibernate

## Continuação...

```
public List<ProdutoVO> buscarTodos() throws PersistenciaException {  
    try {  
        Query query = this.entityManager.createQuery("FROM ProdutoVO ORDER BY NOME");  
        return query.getResultList();  
    } catch (PersistenceException ex) {  
        throw new PersistenciaException("Não foi possível obter a lista de Grupos de Produtos");  
    }  
}
```

```
public List buscarPorNome(String nome) throws PersistenciaException {  
    try {  
        Query q = this.entityManager.createQuery(  
            "FROM ProdutoVO WHERE UPPER(NOME) LIKE '"  
            + nome.trim().toUpperCase() + "%' ORDER BY NOME");  
        return q.getResultList();  
    } catch (PersistenceException ex) {  
        throw new PersistenciaException("Não foi possível obter a lista de Grupos de Produtos");  
    }  
}
```

# JPA/Hibernate

```
package persistencia;
import javax.persistence.EntityManager;
import vo.BaixaVO;
public class RegistroBaixaDAO extends DAO<BaixaVO> {

    public RegistroBaixaDAO(EntityManager entityManager) throws PersistenciaException {
        super(entityManager);
    }
}
```