

# Programação Web com Java

## Aula 4

# Padrão MVC

- Model-view-controller (MVC), em português modelo-visão-controlador, é um padrão de arquitetura de software que separa a representação da informação da interação do usuário com ele.

# Padrão MVC

- O MVC prega que uma aplicação pode ser dividida em 3 camadas, isso possibilitará ter uma aplicação mais organizada, uma vez que, o código ficara mais organizado.
- Isso implica em manutenabilidade maior, já que é uma mais fácil dar manutenção de uma forma simples e rápida, deixando o processo menos oneroso.

# Padrão MVC

- Vantagem
  - Estruturar melhor as Classes
  - Isola as camadas da aplicação.

# 3 camadas de código

- View – Camada representada pelas interfaces/Telas.
- Model – Camada de Modelagem.
  - Camada onde ficam as Classes, métodos e atributos das entidades.
- Controller – Camada que Controla a comunicação entre a camada view e a Model.
  - Captura os dados da interface modela esses dados usando a camada model e persiste no BD.

# Camada DAO (data access object)

- Existe ainda um padrão que sugere uma 4ª camada, que seria responsável pela persistência de dados no BD.
- O padrão DAO convencionou que as operações que fazem acesso ao BD (salvar, alterar, excluir, pesquisar), sejam armazenadas numa classe específica chamada DAO



## New Java Package



### Java Package



Package already exists.



Creates folders corresponding to packages.

Source folder:

Name:

☐ Create package-info.java



- Deployment Descriptor: Vendas
- JAX-WS Web Services
- Java Resources
  - src/main/java
    - br.com.main
    - br.com.vendas.dao
    - br.com.vendas.domain
    - br.com.vendas.util
  - src/main/resources
  - src/test/java
    - br.com.vendas.dao
    - br.com.vendas.Util
  - src/test/resources
  - Libraries
- JavaScript Resources
- Deployed Resources
- src
- target
- pom.xml
- VendasSabado
- VendasSegunda

### New Java Class

#### Java Class

Type already exists.

Source folder:

Package:

☐ Enclosing type:

---

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments



```
public class GenericDao<Entidade> {
```

```
public void salvar(Entidade entidade) {
```

```
Session sessao = hibernateUtil.getSessionFactory().openSession();
```

```
Transaction transacao = null;
```

```
try {
```

```
transacao = sessao.beginTransaction();
```

```
sessao.save(entidade);
```

```
transacao.commit();
```

```
}catch(RuntimeException erro) {
```

```
if (transacao != null) {
```

```
transacao.rollback();
```

```
} throw erro;
```

```
} finally {
```

```
sessao.close();
```

```
}
```

```
}
```

# Hibernate.cfg

```
<!-- Mostra as SQLs Geradas -->
<property name="show_sql">true</property>

<!-- Cria as tabelas do banco de dados -->
<property name="hbm2ddl.auto">validate</property>

<!-- Mapeamento das Entidades -->
<mapping class="br.com.vendas.domain.Cidade"/>
<mapping class="br.com.vendas.domain.Cliente"/>
<mapping class="br.com.vendas.domain.Estado" />
<mapping class="br.com.vendas.domain.Fabricante"/>
<mapping class="br.com.vendas.domain.Funcionario"/>
```


- Vendas
  - Deployment Descriptor: Vendas
  - JAX-WS Web Services
  - Java Resources
    - src/main/java
      - br.com.main
      - br.com.vendas.dao
        - EstadoDao.java
        - FabricanteDao.java
        - GenericDao.java
          - GenericDao<Entidade>
        - br.com.vendas.domain
        - br.com.vendas.util
      - src/main/resources
      - src/test/java
        - br.com.vendas.dao
        - br.com.vendas.Util
      - src/test/resources
      - Libraries
    - JavaScript Resources
    - Deployed Resources
    - src
    - target
    - pom.xml
    - VendasSabado
    - VendasSegunda

26

&lt;property name="cache."

## New Java Class

## Java Class

 Type already exists.

Source folder: Vendas/src/main/java

Browse...

Package: br.com.vendas.dao

Browse...

☐ Enclosing type:

Browse...

Name: EstadoDao

Modifiers: ☒ public ☐ package ☐ private ☐ protected☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

Browse...

Interfaces:

Add...

Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)☐ Constructors from superclass☒ Inherited abstract methodsDo you want to add comments? (Configure templates and default values [here](#))

```
package br.com.vendas.dao;
```

```
import br.com.vendas.domain.Fabricante;
```

```
public class FabricanteDao extends  
GenericDao<Fabricante>{
```

```
}
```

- > src/main/resources
- ▼ src/test/java
  - > br.com.vendas.dao
  - > br.com.vendas.Util
- > src/test/resources
- > Libraries
- > JavaScript Resources
- > Deployed Resources
- > src
- > target
- pom.xml
- VendasSabado
- VendasSegunda

5 public class EstadoDao exte

### New Java Package

**Java Package**

Package already exists.


Creates folders corresponding to packages.


Source folder:

Name:

☐ Create package-info.java

- > Deployment Descriptor: Vendas
- > JAX-WS Web Services
- ✓ Java Resources
  - > src/main/java
  - > src/main/resources
  - ✓ src/test/java
    - ✓ br.com.vendas.dao
      - > EstadoDaoTest.java
      - > FabricanteDaoTest.java
    - > br.com.vendas.Util
  - > src/test/resources
  - > Libraries
- > JavaScript Resources
- > Deployed Resources
- > src
- > target
- pom.xml
- VendasSabado
- VendasSegunda

 New Java Class

**Java Class**  
 Type already exists.

Source folder:

Vendas/src/test/java

Browse...

Package:

br.com.vendas.dao

Browse...

☐ Enclosing type:

Browse...

Name:

EstadoDaoTest

Modifiers:

☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass:

java.lang.Object

Browse...

Interfaces:

Add...

Remove

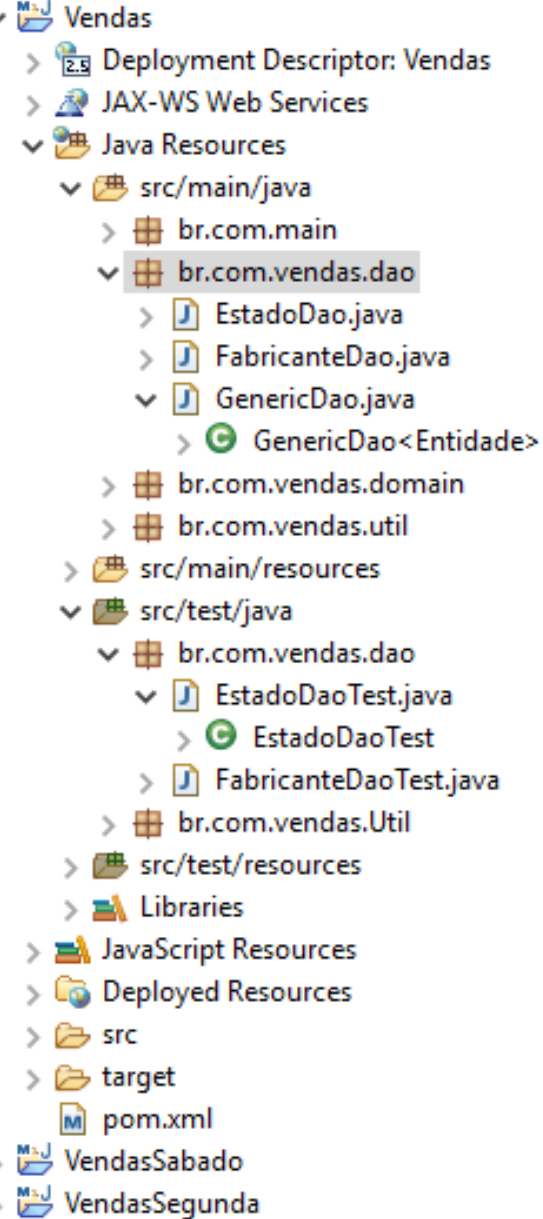
Which method stubs would you like to create?

☐ public static void main(String[] args)  
☐ Constructors from superclass  
☐ Constructors from superclass

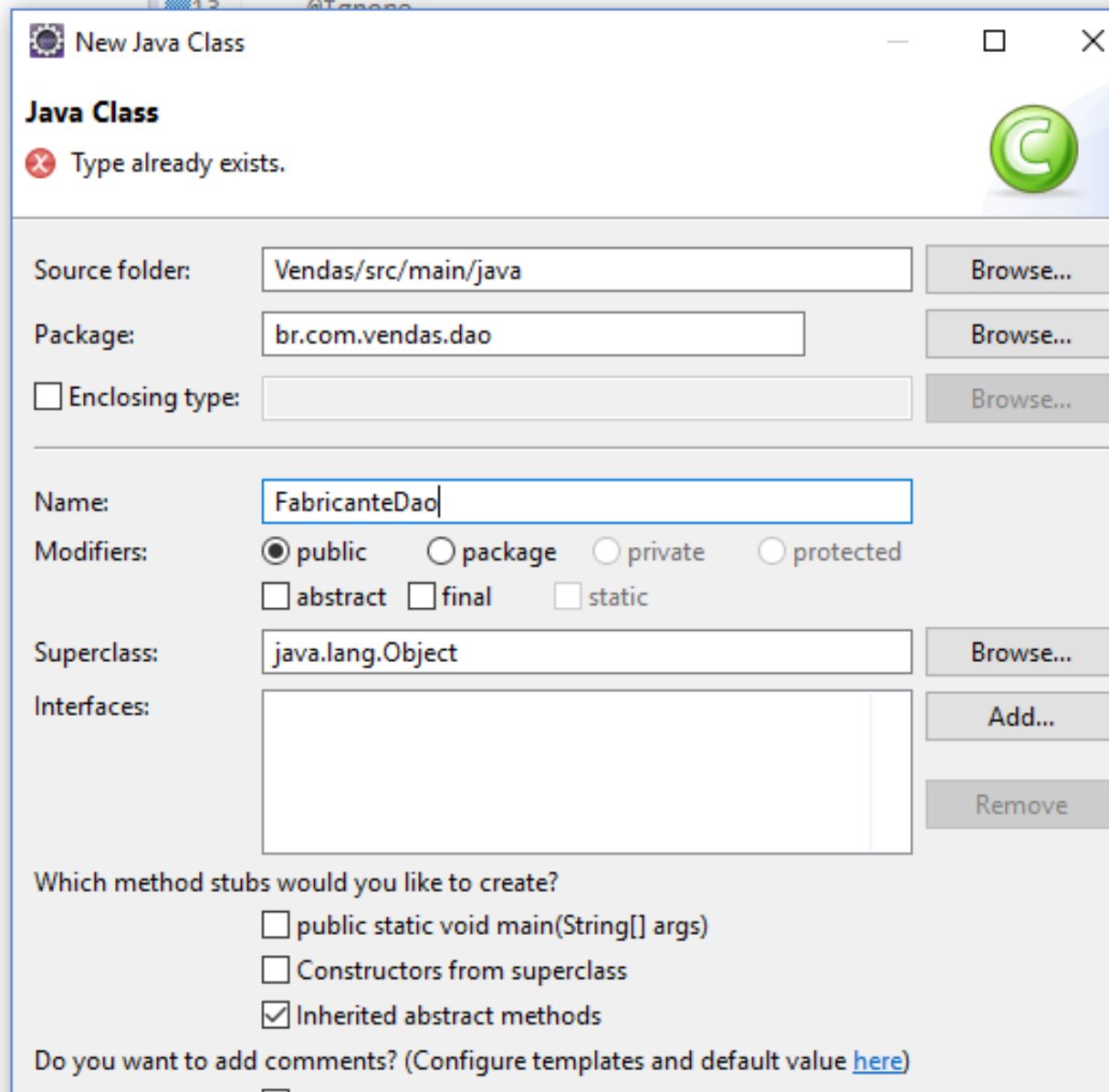
@Test

```
public void salvar() {  
    Estado estado = new Estado();  
    estado.setNome("São Paulo");  
    estado.setSigla("SP");
```

```
        EstadoDao EstadoDao = new  
EstadoDao();  
        EstadoDao.salvar(estado);  
    }
```



```
9  
10 public class EstadoDaoTest {  
11  
12     @Test  
13     @SuppressWarnings("unchecked")
```






```
package br.com.vendas.dao;
```

```
import br.com.vendas.domain.Estado;
```


```
public class EstadoDao extends  
GenericDao<Estado>{
```

```
}
```

- ▼ vendas
  - > Deployment Descriptor: Vendas
  - > JAX-WS Web Services
  - ▼ Java Resources
    - > src/main/java
    - > src/main/resources
    - ▼ src/test/java
      - > br.com.vendas.dao
      - > br.com.vendas.Util
    - > src/test/resources
    - > Libraries
  - > JavaScript Resources
  - > Deployed Resources
  - > src
  - > target
  - pom.xml
  - VendasSabado
  - VendasSegunda

 New Java Class

**Java Class**  
Create a new Java class.



Source folder: Vendas/src/test/java

Browse...

Package: br.com.vendas.dao

Browse...

☐ Enclosing type:

Browse...

Name: FabricanteDaoTes

Modifiers: ☒ public ☐ package ☐ private ☐ protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

Browse...

Interfaces:

Add...

Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

br.com.vendas.dao - Vendas/src/test/java

```
package br.com.vendas.dao;
```

```
import org.junit.Test;
```

```
import br.com.vendas.domain.Fabricante;
```

```
public class FabricanteDaoTest {
```

```
    @Test
```

```
    public void Salvar() {
```

```
        Fabricante fabricante = new Fabricante();
```

```
        fabricante.setDescricao("XYZ Enterprise");
```

```
        FabricanteDao fabricantedao = new FabricanteDao();
```

```
        fabricantedao.salvar(fabricante);
```

```
    }
```

# Atualizar GenericDao

```
private Class<Entidade> classe;
```

```
    @SuppressWarnings("unchecked")
```

```
    public GenericDao() {
```

```
        this.classe = (Class<Entidade>)
```

```
        ((ParameterizedType)getClass().getGenericSuper  
class()).getActualTypeArguments()[0];
```

```
    }
```

```

1 package br.com.vendas.dao;
2
3 import java.lang.reflect.ParameterizedType;
4
5 public class GenericDao<Entidade> {
6
7     private Class<Entidade> classe;
8
9     @SuppressWarnings("unchecked")
10    public GenericDao() {
11        this.classe = (Class<Entidade>) ((ParameterizedType)getClass().getGenericSuperclass()).getActualTypeArguments()[0];
12    }
13
14    public void salvar(Entidade entidade) {
15
16        Session sessao = hibernateUtil.getSessionFactory().openSession();
17        Transaction transacao = null;
18
19        try {
20            transacao = sessao.beginTransaction();
21            sessao.save(entidade);
22            transacao.commit();
23
24        } catch (RuntimeException erro) {
25            if (transacao != null) {
26                transacao.rollback();
27            } throw erro;
28        } finally {
29            sessao.close();
30        }
31    }
32}

```

# Listar GenericDao

```
public List<Entidade> listar(){
    Session sessao =
hibernateUtil.getSessionFactory().openSession();
    try {
        Criteria consulta = sessao.createCriteria(classe);
        List<Entidade> resultado = consulta.list();
        return resultado;

    }catch(RuntimeException erro) {
        throw erro;
    }finally {
        sessao.close();
    }
}
```

```

8         sessao.save(entidade);
9         transacao.commit();
10
11     }catch(RuntimeException erro) {
12         if (transacao != null) {
13             transacao.rollback();
14         } throw erro;
15     } finally {
16         sessao.close();
17     }
18 }
19
20 @SuppressWarnings("unchecked")
21 public List<Entidade> listar(){
22     Session sessao = hibernateUtil.getSessionFactory().openSession();
23     try {
24         Criteria consulta = sessao.createCriteria(classe);
25         List<Entidade> resultado = consulta.list();
26         return resultado;
27
28     }catch(RuntimeException erro) {
29         throw erro;
30     }finally {
31         sessao.close();
32     }
33 }
34
35 }
36
37 }

```

# Atualizar EstadoDaoTest

@Test

```
public void listar() {  
    EstadoDao estadodao = new EstadoDao();  
    List<Estado> resultado = estadodao.listar();  
  
    System.out.println("Total de Registros: " +  
resultado.size());  
  
    for(Estado estado : resultado) {  
        System.out.println(estado.getSigla()+" - "+  
estado.getNome());  
    }  
}
```



```
3+ import java.util.List;
9
10 public class EstadoDaoTest {
11
12-     @Test
13     @Ignore
14     public void salvar() {
15         Estado estado = new Estado();
16         estado.setNome("São Paulo");
17         estado.setSigla("SP");
18
19         EstadoDao EstadoDao = new EstadoDao();
20         EstadoDao.salvar(estado);
21     }
22-     @Test
23     public void listar() {
24         EstadoDao estadodao = new EstadoDao();
25         List<Estado> resultado = estadodao.listar();
26
27         System.out.println("Total de Registros: " + resultado.size());
28
29         for(Estado estado : resultado) {
30             System.out.println(estado.getSigla()+" - "+ estado.getNome());
31         }
32     }
33
34 }
35
36
```