

Programação Web com Java

Aula 3

- Adicionar dependência do mysql no maven

Classe Util

- Depois de configurar o hibernate é necessário criar uma Classe utilitária para agenciar as conexões da aplicação com o BD.
- O hibernate trabalha com fabrica de sessões e é através das sessões que são feitas as conexões.

Classe Util

- O que essa classe fará?
 1. Lêr as configurações
 2. Registrar o serviço
 3. Criar a Fabrica de sessões.
 1. Uma fabrica de sessões no hibernate é uma classe que prove sessões. É através dessas sessões que é possível conversar com o banco.

package

- Um pacote é um diretório que armazena arquivos java.
- `br.com.vendassabado.util`
 - Nessa classe ficarão todos os arquivos utilitários


- ▼ Vendas
 - > 2.9 Deployment Descriptor: Vendas
 - > JAX-WS Web Se
 - ▼ Java Resources
 - > src/main/jav
 - > src/main/res
 - > src/test/java
 - > src/test/reso
 - > Libraries
 - > JavaScript Reso
 - > Deployed Resou
 - > src
 - > target
 - pom.xml
 - Vendas1
 - > VendasS

- New >
- Go Into
- Open Type Hierarchy F4
- Show In Alt+Shift+W >
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Remove from Context Ctrl+Alt+Shift+Down
- Build Path >
- Source Alt+Shift+S >
- Refactor Alt+Shift+T >
- Import...
- Export...
- Refresh F5
- Coverage As >


- Project...
- Annotation
- Class
- Enum
- Interface
- Package
- Source Folder
- HTML File
- JSP File
- Filter
- Listener
- Servlet
- Example...
- Other... Ctrl+N

```
<property name= show_sql >true</pro  
30  
31  
to de da  
">creat  
es -->  
ria.dom  
ria.dom  
Snippets
```

- ▼ Vendas
 - > Deployment Descriptor: Vendas
 - > JAX-WS Web Services
 - ▼ Java Resources
 - > src/main/java
 - > src/main/resources
 - > src/test/java
 - > src/test/resources
 - > Libraries
 - > JavaScript Resources
 - > Deployed Resources
 - > src
 - > target
 - > pom.xml
- Vendas1
- > VendasS

 New Java Package

Java Package
Create a new Java package.




Creates folders corresponding to packages.

Source folder:

Name:

☐ Create package-info.java



- ▼ Vendas
 - > Deployment Descriptor: Vendas
 - > JAX-WS Web Services
 - ▼ Java Resources
 - ▼ src/main/java
 - br.com.vendas.v
 - > src/main/resources
 - > src/test/java
 - > src/test/resources
 - > Libraries
 - > JavaScript Resources
 - > Deployed Resources
 - > src
 - > target
 - pom.xml
 - Vendas1
 - > VendasS

- New
- Go Into
- Open Type Hierarchy F4
- Show In Alt+Shift+W >
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Remove from Context Ctrl+Alt+Shift+Down
- Build Path >
- Source Alt+Shift+S >
- Refactor Alt+Shift+T >
- Import...
- Export...
- Refresh F5
- Coverage As >
- Run As >
- Debug As >
- Profile As >
- Validate


- Project...
- Annotation
- Class
- Enum
- Interface
- Package
- HTML File
- JSP File
- Filter
- Listener
- Servlet
- Example...
- Other... Ctrl+N

- ▼ Vendas
 - > Deployment Descriptor: Vendas
 - > JAX-WS Web Services
 - ▼ Java Resources
 - ▼ src/main/java
 - br.com.vendas.util
 - > src/main/resources
 - > src/test/java
 - > src/test/resources
 - > Libraries
 - > JavaScript Resources
 - > Deployed Resources
 - > src
 - > target
 - pom.xml
 - Vendas1
 - > VendasS

Mark
No cons

New Java Class

Java Class

 Type name is discouraged. By convention, Java type names usually start with an uppercase letter

Source folder:

Vendas/src/main/java

Browse...

Package:

br.com.vendas.util

Browse...

☐ Enclosing type:

Browse...

Name:

hibernateUtil

Modifiers:

☒ public

☐ package

☐ private

☐ protected

☐ abstract

☐ final

☐ static

Superclass:

java.lang.Object

Browse...

Interfaces:

Add...

Remove

Which method stubs would you like to create?

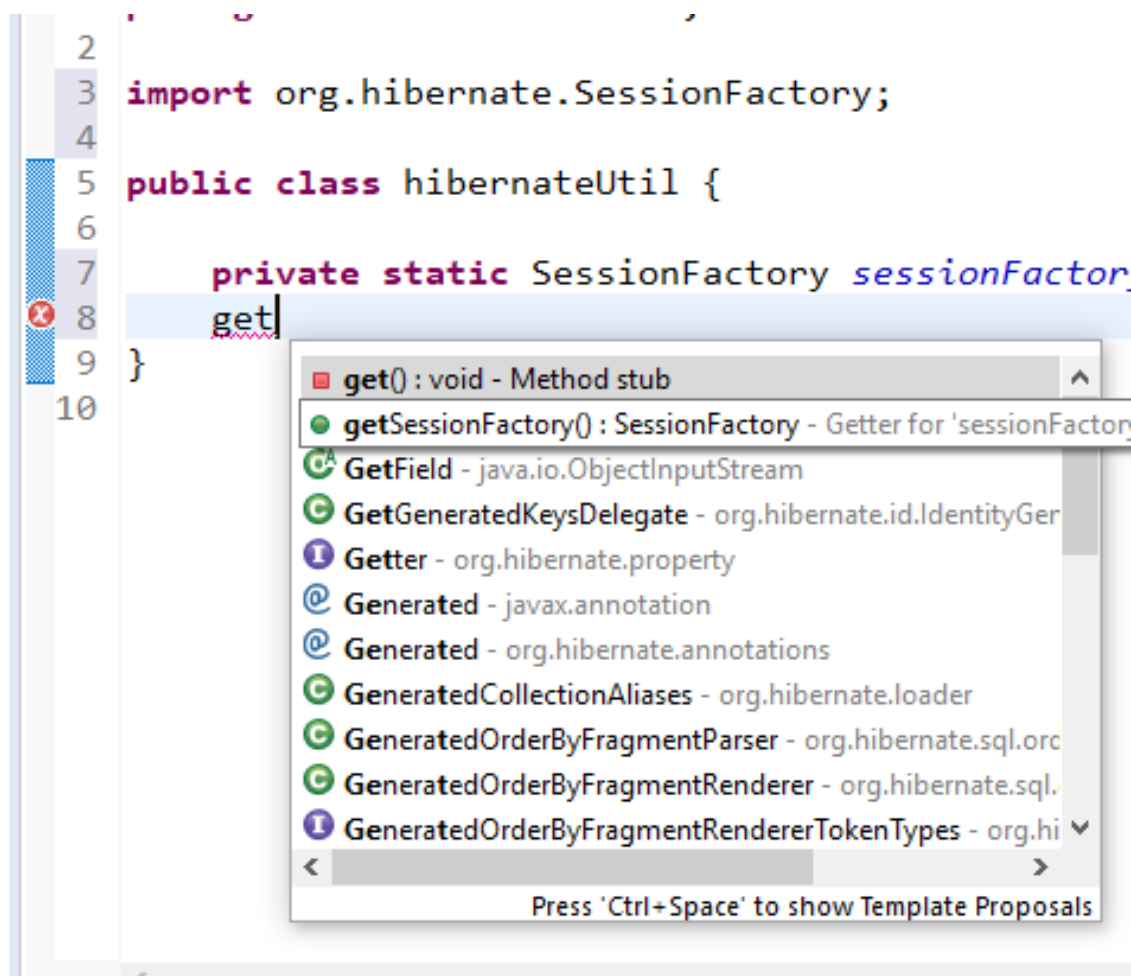
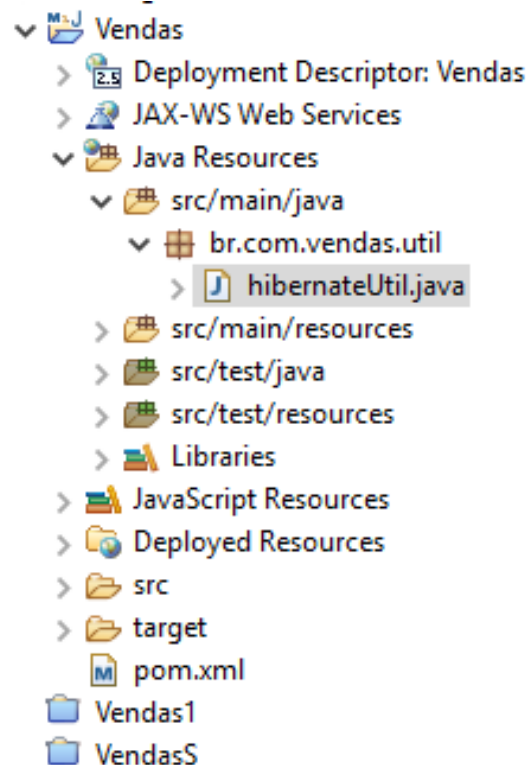
☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

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

☐ Generate comments



```
private static SessionFactory fabricaSessoes = CriarFabricaSessoes();  
public static SessionFactory getFabricaSessoes() {  
    return fabricaSessoes;  
}  
private static SessionFactory CriarFabricaSessoes() {  
    try {  
        Configuration configuracao = new Configuration().configure();  
        ServiceRegistry registro = new  
        StandardServiceRegistryBuilder().applySettings(configuracao.getProperties()).build();  
        SessionFactory fabrica = configuracao.buildSessionFactory(registro);  
        return fabrica;  
  
    }catch(Throwable ex) {  
        System.err.println("A fábrica de sessões não pode ser criada." + ex);  
        throw new ExceptionInInitializerError(ex);  
    }  
}
```

- Configuration configuracao = new Configuration().configure();
 - **Busca o arquivo .cfg, faz a leitura e carrega tudo que foi definido na configuração do hibernate.**

VendasSabado

> 2.5 Deployme

> JAX-WS W

> Java Resou

> src/mai

> br.co

> br.co

> src/mai

> src/test

> br.co

> h

> src/test

> Librarie

> JavaScript

> Deployed I

> src

> target

pom.xml

11

return fabricaSessoes:

New

Go Into

Open Type Hierarchy F4

Show In Alt+Shift+W >

Copy Ctrl+C

Copy Qualified Name

Paste Ctrl+V

Delete Delete

Remove from Context Ctrl+Alt+Shift+Down

Build Path >

Source Alt+Shift+S >

Refactor Alt+Shift+T >

Import...

Export...

Refresh F5

Project...

Annotation

Class

Enum

Interface

Package

Source Folder

HTML File

JSP File

Filter

Listener

Servlet

Example...

Other... Ctrl+N

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

```
10- pub  
11  
12 }  
13- priv  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27 }  
28
```

Java Package

⊗ Package already exists.

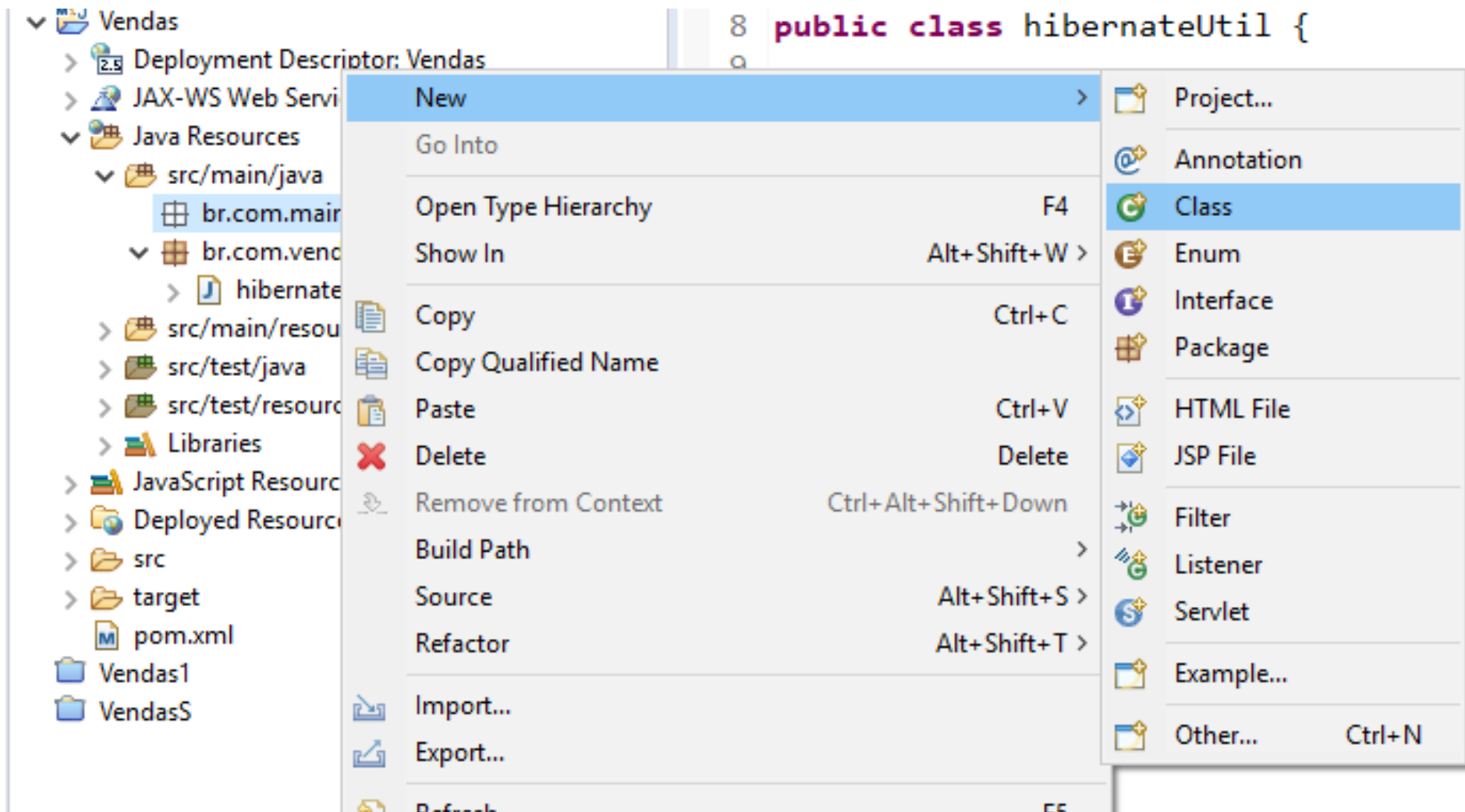
Creates folders corresponding to packages.

Source folder:

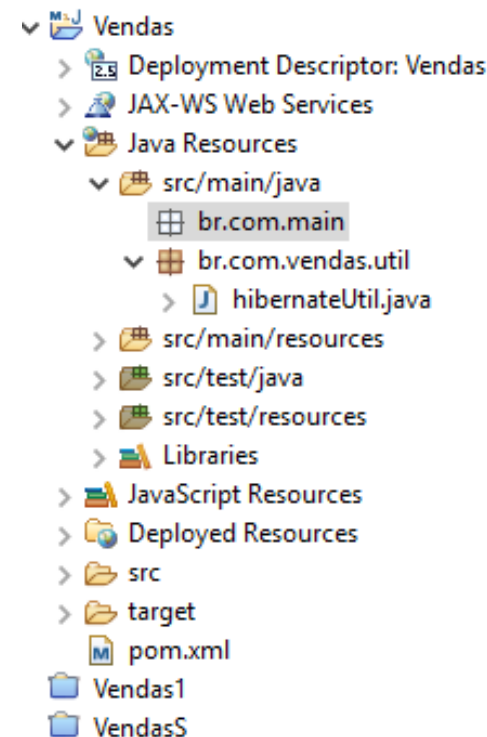
Name:

☐ Create package-info.java


hibernateUtilTest



hibernateUtilTest



Java Class

 Type name is discouraged. By convention, Java type names usually start with an uppercase letter



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


```
Session sessao =  
hibernateUtil.getFabricaSessoes().openSession();  
    sessao.close();
```

```
hibernateUtil.getFabricaSessoes().close();
```

> Deployment Descri
> JAX-WS Web Servic
✓ Java Resources
 ▼ src/main/java
 ▼ br.com.main
 ▼ hibernateUtil
 > hiberna
 > br.com.vend
 > src/main/resour
 ▼ src/test/java
 ▼ br.com.vend
 > hibernateUtil
 > src/test/resourc
 > Libraries
> JavaScript Resource
> Deployed Resource
> src
> target
 pom.xml

	Copy	Ctrl+C
	Copy Qualified Name	
	Paste	Ctrl+V
	Delete	Delete
	Remove from Context	Ctrl+Alt+Shift+Down
	Build Path	
	Source	Alt+Shift+S
	Refactor	Alt+Shift+T
	Import...	
	Export...	
	Refresh	F5
	References	>
	Declarations	>
	Coverage As	>
	Run As	>
	Debug As	>
	Profile As	>
	Validate	

```
assabado.util.hibernateUtil;  
  
nateUtilTest {  
    void main(String[] args) {  
        ssao = hibernateUtil.getFabrica  
        se();  
        til.getFabricaSessoes().close()  
    }  
}
```

	1 Run on Server	Alt+Shift+X, R
	2 Java Application	Alt+Shift+X, J
Run Configurations...		

JUnit

- Permite fazer testes unitários na aplicação.

JUnit

<dependency>

 <groupId>junit</groupId>

 <artifactId>junit</artifactId>

 <version>4.12</version>

 <scope>test</scope>

</dependency>

- ▼ VendasSabado
 - > Deployment D
 - > JAX-WS Web S
 - ▼ Java Resources
 - > src/main/ja
 - > src/main/re
 - > src/test/jav
 - src/test/res
 - > Libraries
 - > JavaScript Reso
 - > Deployed Reso
 - > src
 - > target
 - pom.xml

3e import org.hibernate.Session;

New	>	Project...
Go Into		Annotation
Open Type Hierarchy	F4	Class
Show In	Alt+Shift+W >	Enum
Copy	Ctrl+C	Interface
Copy Qualified Name		Package
Paste	Ctrl+V	Source Folder
Delete	Delete	HTML File
Remove from Context	Ctrl+Alt+Shift+Down	JSP File
Build Path	>	Filter
Source	Alt+Shift+S >	Listener
Refactor	Alt+Shift+T >	Servlet
Import...		Example...
Export...		Other... Ctrl+N
Refresh	F5	

- > JAX-WS Web Services
- ▼ Java Resources
 - > src/main/java
 - > src/main/resources
 - ▼ src/test/java
 - > br.com.vendassabado.Util
 - > src/test/resources
 - > Libraries
- > JavaScript Resources
- > Deployed Resources
- > src
- > target
- pom.xml

37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

<version>4.3.11.Final</version>

New Java Package

Java Package

Create a new Java package.

Creates folders corresponding to packages.

Source folder: VendasSabado/src/test/java

Name: br.com.vendassabado.Util

☐ Create package-info.java

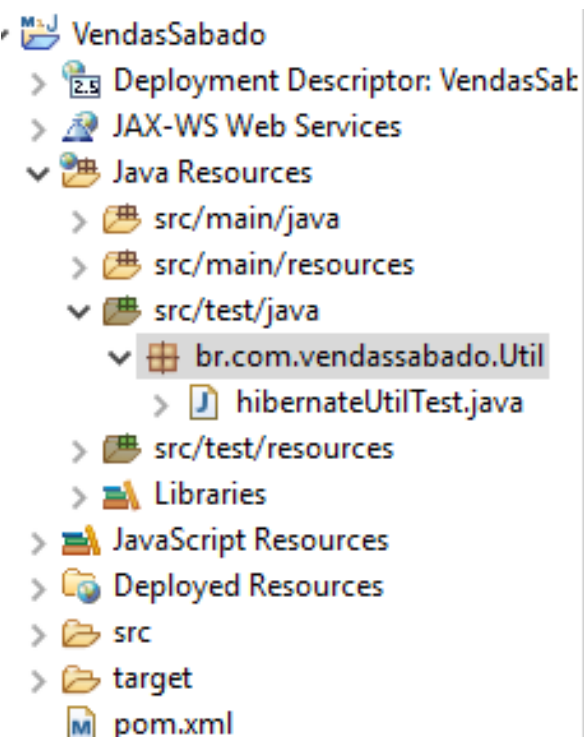


Finish

- VendasSabado
 - > Deployment Descriptor
 - > JAX-WS Web Service
 - Java Resources
 - > src/main/java
 - > src/main/resources
 - src/test/java
 - br.com.vendasSabado
 - hibernate.cfg.xml
 - src/test/resources
 - Libraries
 - > JavaScript Resource
 - > Deployed Resource
 - > src
 - > target
 - pom.xml

Open Type Hierarchy	F4
Show In	Alt+ Shift+W >
Copy	Ctrl+C
Copy Qualified Name	
Paste	Ctrl+V
Delete	Delete
Remove from Context	Ctrl+ Alt+ Shift+ Down
Build Path	>
Source	Alt+ Shift+S >
Refactor	Alt+ Shift+ T >
Import...	
Export...	
Refresh	F5
References	>
Declarations	>

Annotation
Class
Enum
Interface
Package
HTML File
JSP File
Filter
Listener
Servlet
Example...
Other... Ctrl+N



36 Source folder: VendasSabado/src/test/java

37

38 Package: br.com.vendassabado.Util

39

40 ☐ Enclosing type:

41

42 Name: hibernateUtilTest

43

44 Modifiers: ☒ public ☐ package ☐ private ☐ protected

45 ☐ abstract ☐ final ☐ static

46 Superclass: java.lang.Object

47

48 Interfaces:

49

50

51

52 Which method stubs would you like to create?

53 ☐ 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

Over

Finish

Run

@Test

```
public void conectar() {
```

```
    Session sessao =
```

```
hibernateUtil.getFabricaSessoes().openSession();
```

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

```
hibernateUtil.getFabricaSessoes().close();
```

- VendasSabado
 - Deployment Descrip
 - JAX-WS Web Service
 - Java Resources
 - src/main/java
 - src/main/resourc
 - src/test/java
 - br.com.venda
 - hibernateL
 - src/test/resource
 - Libraries
 - JavaScript Resource
 - Deployed Resources
 - src
 - target
 - pom.xml

- Open With >
- Copy Ctrl+C
- Copy Qualified Name
- Paste Ctrl+V
- Delete Delete
- Remove from Context Ctrl+Alt+Shift+Down
- Build Path >
- Source Alt+Shift+S >
- Refactor Alt+Shift+T >
- Import...
- Export...
- Refresh F5
- References >
- Declarations >
- Coverage As >
- Run As >
- Debug As >
- Profile As >
- Validate
- Restore from Local History...

```
hibernate</groupId>  
hibernate-core</artifactId>  
1.Final</version>  
  
</groupId>  
sql-connector-java</artifactId>  
6</version>  
  
</groupId>  
unit</artifactId>  
</version>  
<scope>
```

- 1 Run on Server Alt+Shift+X, R
- 2 JUnit Test Alt+Shift+X, T
- Run Configurations...

Markers Properties Servers Data Source Explorer Snippets Console Progress JUnit Terminal

Finished after 1,345 seconds

Close

Runs: 1/1 Errors: 0 Failures: 0

> br.com.vendassabado.Util.hibernateUtilTest [Runner: JUnit 4] (1,309 s)

Failure Trace

Design Source

Markers Properties Servers Data Source Explorer Snippets Console Progress JUnit Terminal

Finished after 0,992 seconds

Runs: 1/1 Errors: 1 Failures: 0

▼ br.com.vendassabado.Util.hibernateUtilTest [Runner: JUnit 4] (0,964 s)

conectar (0,964 s)

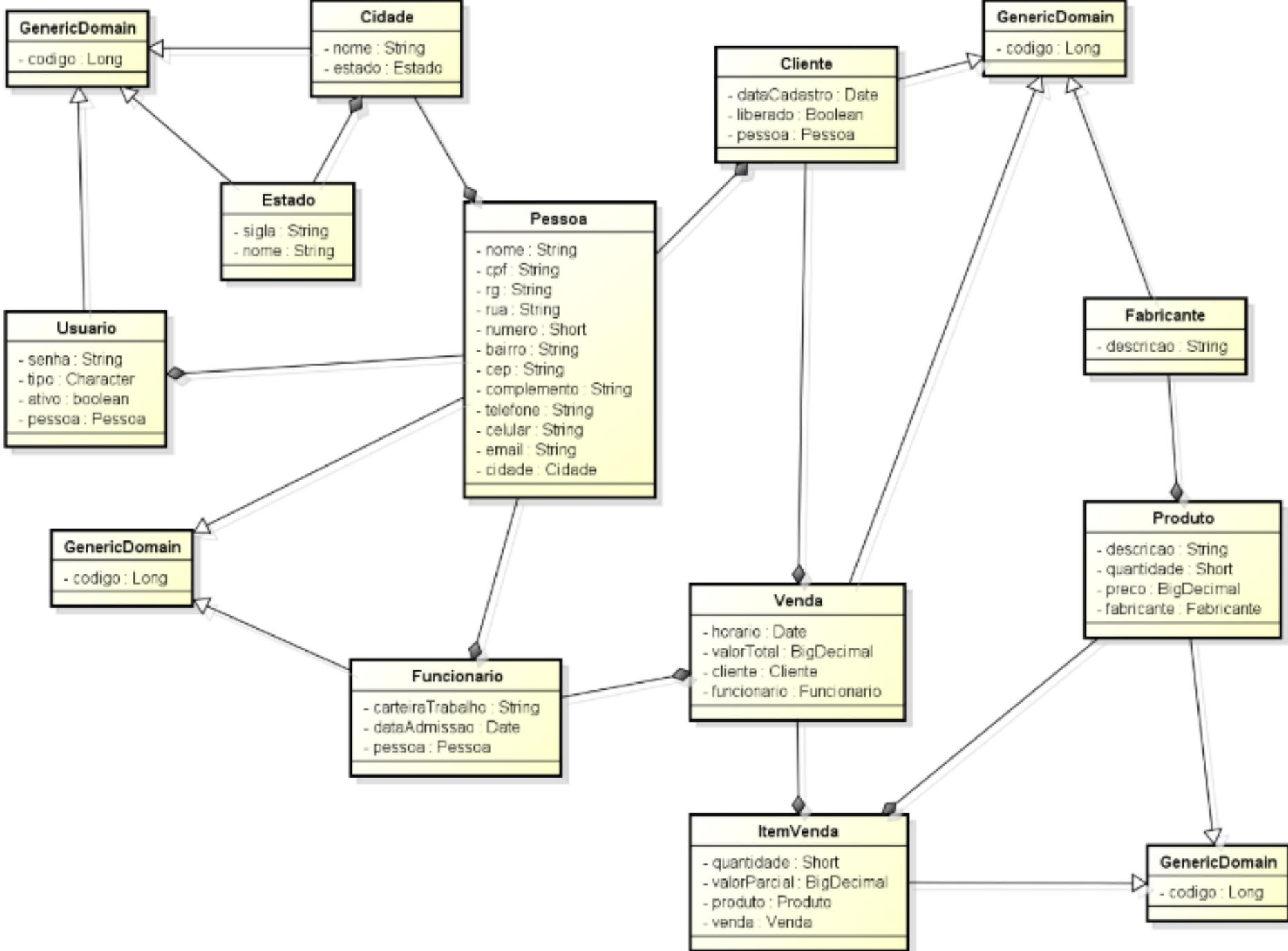
Failure Trace

java.lang.ExceptionInInitializerError

at br.com.vendassabado.util.hibernateUtil.CriarFabricaSes

<

Modelo de dados



Camada Domain

- A forma mais fácil de trabalhar com hibernate é aquela em q toda tabela do banco possui uma chave primaria do tipo auto incremento.
- Para evitar o retrabalho de ter q criar um código em todas as tabelas usa-se o Generic Domain, que é uma classe onde estará a chave primária que poderá ser compartilhada por todas as tabelas.

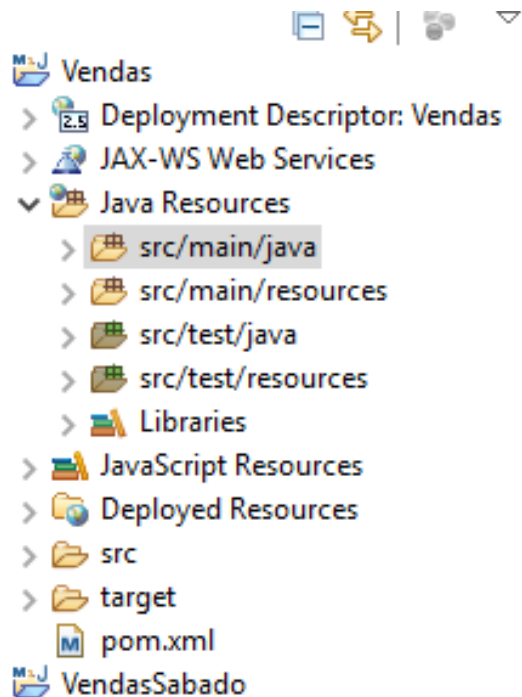
Camada Domain

- A camada domain corresponde as nossas tabelas no banco de dados.
- O Generic domain servirá como gerador de chave primária para todas as tabelas do BD. Entretanto não será uma tabela, será uma classe java que injetara a chave primária no banco

veritas
> 2.5 Deployment Descriptor Vendor
> JAX-WS Vendor
▼ Java Resources
 > src/main
 > src/main
 > src/test
 > src/test
 > Library
> JavaScript
> Deployed
> src
> target
 pom.xml
VendasSaba

New	>
Go Into	
Open Type Hierarchy	F4
Show In	Alt+Shift+W >
Copy	Ctrl+C
Copy Qualified Name	
Paste	Ctrl+V
Delete	Delete
Remove from Context	Ctrl+Alt+Shift+Down
Build Path	>
Source	Alt+Shift+S >
Refactor	Alt+Shift+T >
Import...	
Export...	
Refresh	F5

Project...
Annotation
Class
Enum
Interface
Package
Source Folder
HTML File
JSP File
Filter
Listener
Servlet
Example...
Other... Ctrl+N



```
1 <?xml version="
2 <!DOCTYPE hiber
3 "-//Hibernate/H
4 "http://www.hib
5
6 <hibernate-conf
7
8 <session-fa
9 <!-- Da
10 <propert
11 <propert
12 <propert
13 <propert
14
15 <!-- Po
16 <propert
17
18 <!-- SQ
19 <propert
20
21 <!-- Ge
```

Java Package

Create a new Java package.

Creates folders corresponding to packages.


Source folder: Vendas/src/main/java

Name: br.com.vendas.domain


☐ Create package-info.java





Fin


✓  Vendas


>  Deployment Descriptor Vendas


>  JAX-WS Web Service Vendas


✓  Java Resources


 >  src/main/java


 >  br.com.ma

 br.com.ver


 >  br.com.ver


 >  src/main/resources

 >  src/test/java


 >  src/test/resources


 >  Libraries


>  JavaScript Resources

>  Deployed Resources

>  src

>  target

 pom.xml

>  VendasSabado

New

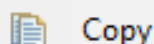
Go Into

Open Type Hierarchy

F4

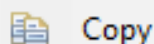
Show In

Alt+Shift+W >

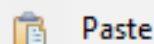


Copy

Ctrl+C

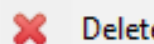


Copy Qualified Name



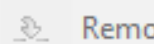
Paste

Ctrl+V



Delete

Delete



Remove from Context

Ctrl+Alt+Shift+Down

Build Path

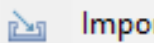
>

Source

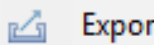
Alt+Shift+S >

Refactor

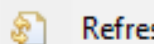
Alt+Shift+T >



Import...




Export...





Refresh


F5


 Project...


 Annotation


 Class


 Enum


 Interface


 Package


 HTML File


 JSP File

 Filter

 Listener

 Servlet

 Example...

 Other... Ctrl+N

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

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

Implements Serializable

- Em aplicações web o fluxo de rede é constante. Por conta disso é importante que os objetos sejam serializados. Serializar um objeto significa transforma-los em bytes que podem ser transmitidos ou gravados. Usa-se serialização quando se quer gravar objetos em arquivos e transmiti-los pela rede.

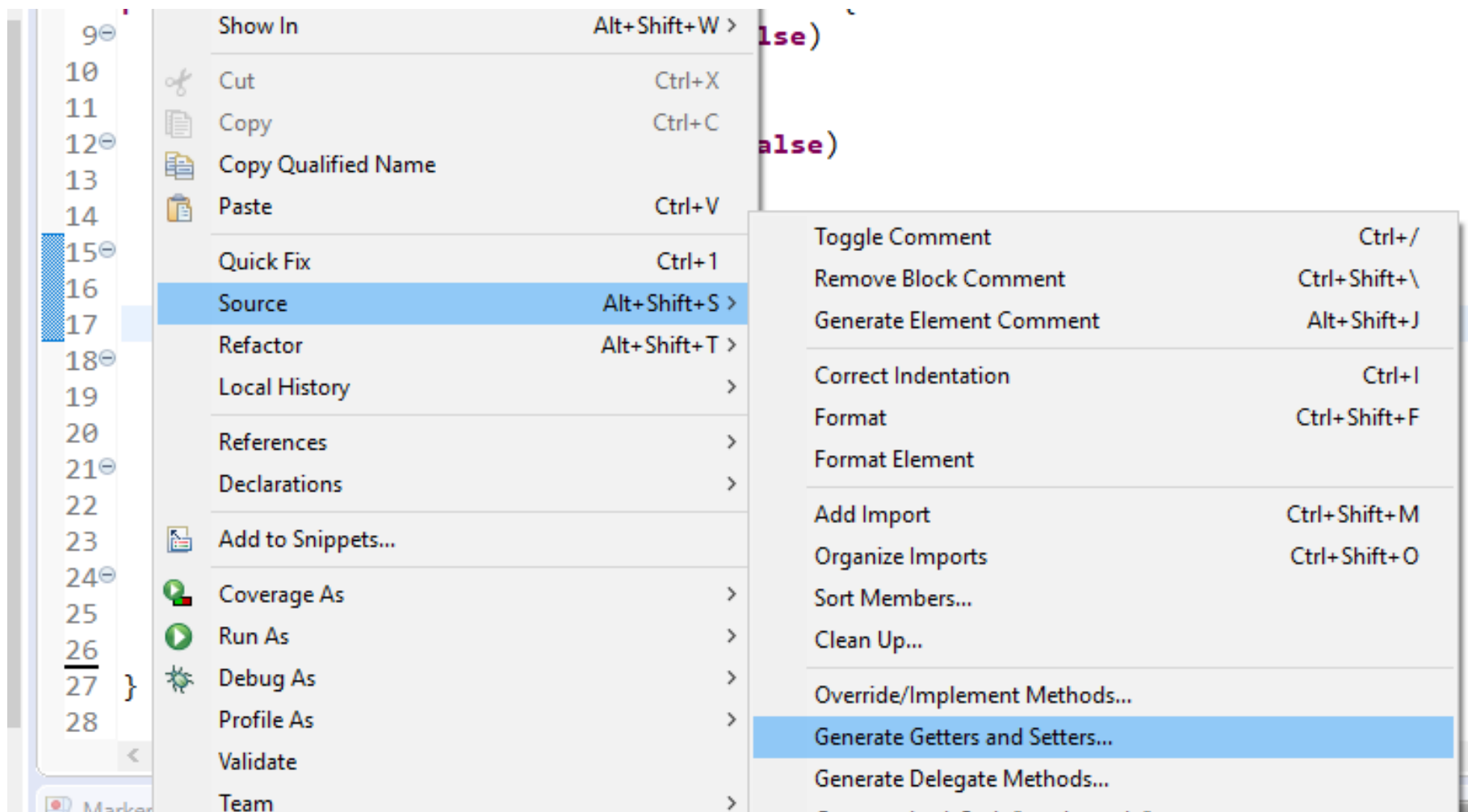
```
@MappedSuperclass
public class GenericDomain implements Serializable{
    @Id
    @GeneratedValue(strategy =
GenerationType.AUTO)
    private long codigo;

    public long getCodigo() {
        return codigo;
    }
    public void setCodigo(long codigo) {
        this.codigo = codigo;
    }
}
```

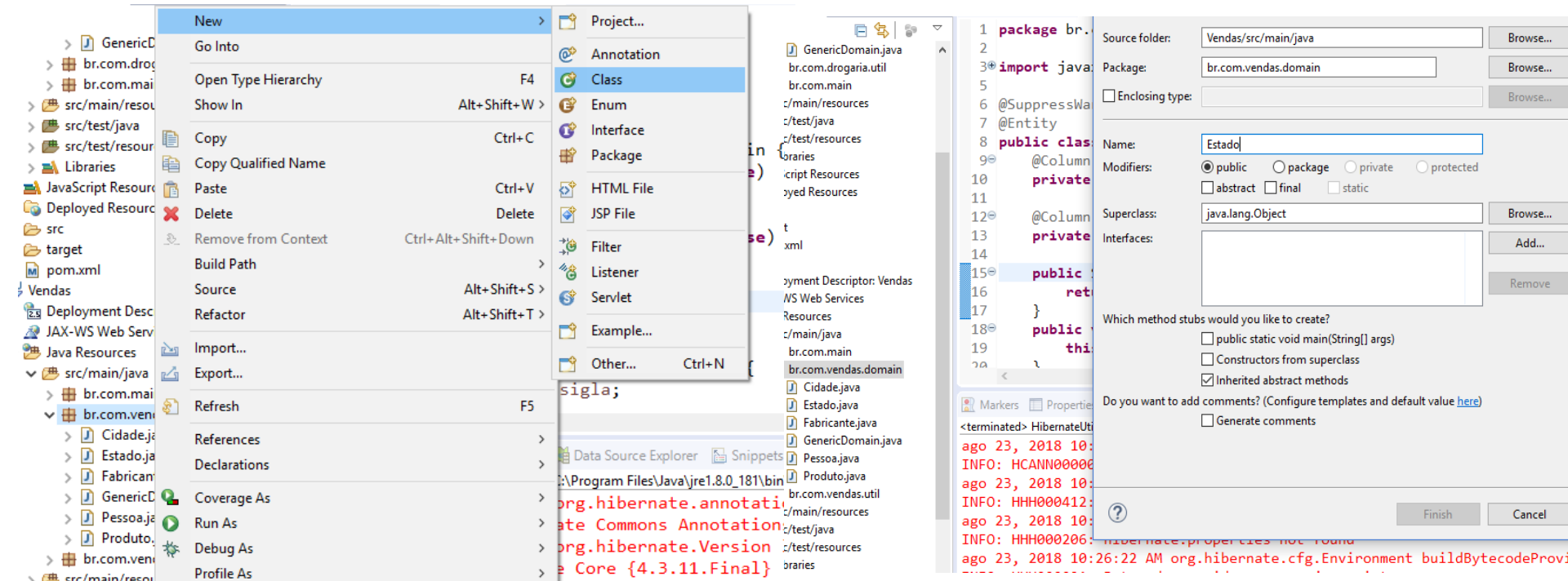
```

1  package br.com.drogaria.domain;
2
3+ import java.io.Serializable;
9
10 @SuppressWarnings("serial")
11 @MappedSuperclass
12 public class GenericDomain implements Serializable {
13-     @Id
14     @GeneratedValue(strategy = GenerationType.AUTO)
15     private Long codigo;
16
17-     public Long getCodigo() {
18         return codigo;
19     }
20
21-     public void setCodigo(Long codigo) {
22         this.codigo = codigo;
23     }
24

```



Criando as Entidades




```
1 package br.com.vendas.domain;
2
3 import javax.persistence.Column;
4
5
6 @SuppressWarnings("serial")
7 @Entity
8 public class Estado extends GenericDomain {
9     @Column(length = 2, nullable = false)
10     private String sigla;
11
12     @Column(length = 50, nullable = false)
13     private String nome;
14
15     public String getSigla() {
16         return sigla;
17     }
18     public void setSigla(String sigla) {
19         this.sigla = sigla;
20     }
21     public String getNome() {
22         return nome;
23     }
24     public void setNome(String nome) {
25         this.nome = nome;
26     }
27 }
```

@SuppressWarnings("serial")

@Entity

public class Estado extends GenericDomain {

@Column(length = 2, nullable = false)

private String sigla;

@Column(length = 50, nullable = false)

private String nome;

public String getSigla() {

return sigla;

}

public void setSigla(String sigla) {

this.sigla = sigla;

}

public String getNome() {

return nome;

}

public void setNome(String nome) {

this.nome = nome;

}

}

Mapeando as entidades

```
13 <property name="connection.username">root</property>
14 <property name="connection.password">q1w2e3r4</property>
15
16 <!-- Pool de Conexões /Quantas conexões ao msm tempo -->
17 <property name="connection.pool_size">1</property>
18
19 <!-- SQL dialect -->
20 <property name="dialect">org.hibernate.dialect.MySQL5In
21
22 <!-- Gerenciamento do Contexto das Sessões -->
23 <property name="current_session_context_class">thread</
24
25 <!-- Cache de Segundo Nível -->
26 <property name="cache.provider_class">org.hibernate.cac
27
28 <!-- Mostra as SQLs Geradas -->
29 <property name="show_sql">>true</property>
30
31 <!-- Cria as tabelas do banco de dados -->
32 <property name="hbm2ddl.auto">create</property>
33
34 <!-- Mapeamento das Entidades -->
35 <mapping class="br.com.vendas.domain.Estado" />
36
```

Mapeando as entidades cfg.xml

- <!-- Mapeamento das Entidades -->
- <mapping
class="*br.com.vendas.domain.Estado*" />

```
1 package br.com.vendas.domain;
2
3 import javax.persistence.Column;
4
5
6 @SuppressWarnings("serial")
7 @Entity
8 public class Fabricante extends GenericDomain {
9     @Column(length = 50, nullable = false)
10     private String descricao;
11
12     public String getDescricao() {
13         return descricao;
14     }
15
16     public void setDescricao(String descricao) {
17         this.descricao = descricao;
18     }
19
20
21
22 }
23
```

@SuppressWarnings("serial")

@Entity

public class Fabricante extends GenericDomain {

@Column(length = 50, nullable = false)

private String descricao;

public String getDescricao() {

return descricao;

}

public void setDescricao(String descricao) {

this.descricao = descricao;

}

}

```
/
8 @SuppressWarnings("serial")
9 @Entity
0 public class Cidade extends GenericDomain {
1     @Column(length = 50, nullable = false)
2     private String nome;
3
4     @ManyToOne
5     @JoinColumn(nullable = false)
6     private Estado estado;
7
8     public String getNome() {
9         return nome;
0     }
1
2     public void setNome(String nome) {
3         this.nome = nome;
4     }
5
6     public Estado getEstado() {
7         return estado;
8     }
9
0     public void setEstado(Estado estado) {
1         this.estado = estado;
2     }
3 }
```

```
@SuppressWarnings("serial")
@Entity
public class Cidade extends GenericDomain {
    @Column(length = 50, nullable = false)
    private String nome;

    @ManyToOne
    @JoinColumn(nullable = false)
    private Estado estado;

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public Estado getEstado() {
        return estado;
    }

    public void setEstado(Estado estado) {
        this.estado = estado;
    }
}
```


Mapeando as entidades cfg

```
<!-- Mapeamento das Entidades -->
```

```
<mapping class=
```

```
"br.com.vendas.domain.Cidade"/>
```

```
3 @SuppressWarnings("serial")
1 @Entity
2 public class Produto extends GenericDomain {
3     @Column(length = 80, nullable = false)
4     private String descricao;
5
6     @Column(nullable = false)
7     private Short qtde;
8
9     @Column(nullable = false, precision = 6, scale = 2)
10    private BigDecimal preco;
11
12    @ManyToOne
13    @JoinColumn(nullable = false)
14    private Fabricante fabricante;
15
16    public String getDescricao() {
17        return descricao;
18    }
19
20    public void setDescricao(String descricao) {
21        this.descricao = descricao;
22    }
23
24    public Short getQtde() {
25        return qtde;
26    }
27 }
```

```
32     }
33
34     public Short getQtde() {
35         return qtde;
36     }
37
38     public void setQtde(Short qtde) {
39         this.qtde = qtde;
40     }
41
42     public BigDecimal getPreco() {
43         return preco;
44     }
45
46     public void setPreco(BigDecimal preco) {
47         this.preco = preco;
48     }
49
50     public Fabricante getFabricante() {
51         return fabricante;
52     }
53
54     public void setFabricante(Fabricante fabricante) {
55         this.fabricante = fabricante;
56     }
57 }
```

```
@SuppressWarnings("serial")
@Entity
public class Produto extends GenericDomain {
    @Column(length = 80, nullable = false)
    private String descricao;

    @Column(nullable = false)
    private Short qtde;

    @Column(nullable = false, precision = 6, scale = 2)
    private BigDecimal preco;

    @ManyToOne
    @JoinColumn(nullable = false)
    private Fabricante fabricante;

    public String getDescricao() {
        return descricao;
    }

    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }

    public Short getQtde() {
        return qtde;
    }

    public void setQtde(Short qtde) {
        this.qtde = qtde;
    }

    public BigDecimal getPreco() {
        return preco;
    }

    public void setPreco(BigDecimal preco) {
        this.preco = preco;
    }

    public Fabricante getFabricante() {
        return fabricante;
    }

    public void setFabricante(Fabricante fabricante) {
        this.fabricante = fabricante;
    }
}
```

Mapeando as entidades cfg

```
<mapping  
class="br.com.vendas.domain.Produto"/>
```

```
7
8 @SuppressWarnings("serial")
9 @Entity
10 public class Pessoa extends GenericDomain {
11     @Column(length = 50, nullable = false)
12     private String nome;
13
14     @Column(length = 12, nullable = false)
15     private String cpf;
16
17     @Column(length = 9, nullable = false)
18     private String rg;
19
20     @Column(length = 50, nullable = false)
21     private String rua;
22
23     @Column(nullable = false)
24     private Short numero;
25
26     @Column(length = 50, nullable = false)
27     private String bairro;
28
29     @Column(length = 9, nullable = false)
30     private String cp;
31
32     @Column(length = 50)
33     private String complemento;
```

```
35 @Column(length =10, nullable =false)
36 private String telefone;
37
38 @Column(length = 10)
39 private String celular;
40
41 @Column(length = 50)
42 private String email;
43
44 @ManyToOne
45 @JoinColumn(nullable = false)
46 private Cidade cidade;
47
48 public String getNome() {
49     return nome;
50 }
51
52 public void setNome(String nome) {
53     this.nome = nome;
54 }
55
56 public String getCpf() {
57     return cpf;
58 }
59
60 public void setCpf(String cpf) {
```

```
64 public String getRg() {  
65     return rg;  
66 }  
67  
68 public void setRg(String rg) {  
69     this.rg = rg;  
70 }  
71  
72 public String getRua() {  
73     return rua;  
74 }  
75  
76 public void setRua(String rua) {  
77     this.rua = rua;  
78 }  
79  
80 public Short getNumero() {  
81     return numero;  
82 }  
83  
84 public void setNumero(Short numero) {  
85     this.numero = numero;  
86 }  
87  
88 public String getBairro() {  
89     return bairro;  
90 }
```



```
91
92 public void setBairro(String bairro) {
93     this.bairro = bairro;
94 }
95
96 public String getCp() {
97     return cp;
98 }
99
100 public void setCp(String cp) {
101     this.cp = cp;
102 }
103
104 public String getComplemento() {
105     return complemento;
106 }
107
108 public void setComplemento(String complemento) {
109     this.complemento = complemento;
110 }
111
112 public String getTelefone() {
113     return telefone;
114 }
115
116 public void setTelefone(String telefone) {
117     this.telefone = telefone;
```

```
118     }
119
120     public String getCelular() {
121         return celular;
122     }
123
124     public void setCelular(String celular) {
125         this.celular = celular;
126     }
127
128     public String getEmail() {
129         return email;
130     }
131
132     public void setEmail(String email) {
133         this.email = email;
134     }
135
136     public Cidade getCidade() {
137         return cidade;
138     }
139
140     public void setCidade(Cidade cidade) {
141         this.cidade = cidade;
142     }
143 }
144
```

```

• @SuppressWarnings("serial")
• @Entity
• public class Pessoa extends GenericDomain {
•     @Column(length = 50, nullable = false)
•         private String nome;
•
•
•     @Column(length = 12, nullable = false)
•         private String cpf;
•
•
•     @Column(length = 9, nullable = false)
•         private String rg;
•
•
•     @Column(length = 50, nullable = false)
•         private String rua;
•
•
•     @Column(nullable = false)
•         private Short numero;
•
•
•     @Column(length = 50, nullable = false)
•         private String bairro;
•
•
•     @Column(length = 9, nullable = false)
•         private String cp;
•
•
•     @Column(length = 50)
•         private String complemento;
•
•
•     @Column(length = 10, nullable = false)
•         private String telefone;
•
•
•     @Column(length = 10)
•         private String celular;
•
•
•     @Column(length = 50)
•         private String email;
•
•
•     @ManyToOne
•     @JoinColumn(nullable = false)
•     private Cidade cidade;
•
•
•     public String getNome() {
•         return nome;
•     }
•
•
•     public void setNome(String nome) {
•         this.nome = nome;
•     }

```

```

•     public String getCpf() {
•         return cpf;
•     }
•
•     public void setCpf(String cpf) {
•         this.cpf = cpf;
•     }
•
•     public String getRg() {
•         return rg;
•     }
•
•     public void setRg(String rg) {
•         this.rg = rg;
•     }
•
•     public String getRua() {
•         return rua;
•     }
•
•     public void setRua(String rua) {
•         this.rua = rua;
•     }
•
•     public Short getNumero() {
•         return numero;
•     }
•
•     public void setNumero(Short numero) {
•         this.numero = numero;
•     }
•
•     public String getBairro() {
•         return bairro;
•     }
•
•     public void setBairro(String bairro) {
•         this.bairro = bairro;
•     }
•
•     public String getCp() {
•         return cp;
•     }
•
•     public void setCp(String cp) {
•         this.cp = cp;
•     }
•
•     public String getComplemento() {
•         return complemento;
•     }
•
•     public void setComplemento(String complemento) {
•         this.complemento = complemento;
•     }
•
•     public String getTelefone() {
•         return telefone;
•     }
•
•     public void setTelefone(String telefone) {
•         this.telefone = telefone;
•     }
•
•     public String getCelular() {
•         return celular;
•     }
•
•     public void setCelular(String celular) {
•         this.celular = celular;
•     }

```

Mapeando as entidades cfg

```
<!-- Mapeamento das Entidades -->
```

```
<mapping class=
```

```
"br.com.vendas.domain.Pessoa"/>
```