Hibernate

- Mapeamento Objeto-Relacional
 - É o nome dado para a persistência automática e transparente de classes de um sistema em tabelas de um SGBD
- Framework ORM
 - Framework de infraestrutura que oferece serviços de ORM
- Mak Fussel
 - Define quatro níveis de maturidade para soluções
 ORM

Níveis ORM

- Relacional puro
 - JDBC + SQL
- Mapeamento leve
 - DAO
- Mapeamento médio
 - EJB
- Mapeamento completo
 - Hibernate



Hibernate

• Classe Aluno:

Aluno

- id : int

- matricula : int

- nome : String

- cpf:long





```
public class Aluno {
   private int id;
   private int matricula;
   private String nome;
   private int cpf;
   public int getId() { return id; }
   public void setId(int id) { this.id = id; }
   public int getMatricula() { return matricula; }
   public void setMatricula(int matricula) { this.matricula = matricula; }
   public String getNome() { return nome; }
   public void setNome(String nome) { this.nome = nome; }
   public int getCpf() { return cpf; }
   public void setCpf(int cpf) { this.cpf = cpf; }
   public Aluno(){}
```





Hibernate

Tabela Aluno:

```
CREATE TABLE ALUNO (
id_aluno integer NOT NULL GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),
matricula integer NOT NULL, -- Matrícula do aluno
nome character(40) NOT NULL, -- Nome do aluno
cpf bigint NOT NULL, -- CPF do aluno
CONSTRAINT pk_aluno PRIMARY KEY (id_aluno),
CONSTRAINT un_cpf_aluno UNIQUE (cpf),
CONSTRAINT un_matricula_aluno UNIQUE (matricula)

Página Inicial 
CONEXÃO: jdbc:derby://localhost:1527/teste [romualdo em ROMU... $

CONEXÃO: jdbc:derby://localhost:1527/teste [romualdo em ROMU... $

CONEXÃO: JCREATE SEQUENCE aluno_seq START WITH 1 INCREMENT BY 1 NO CYCLE;
```

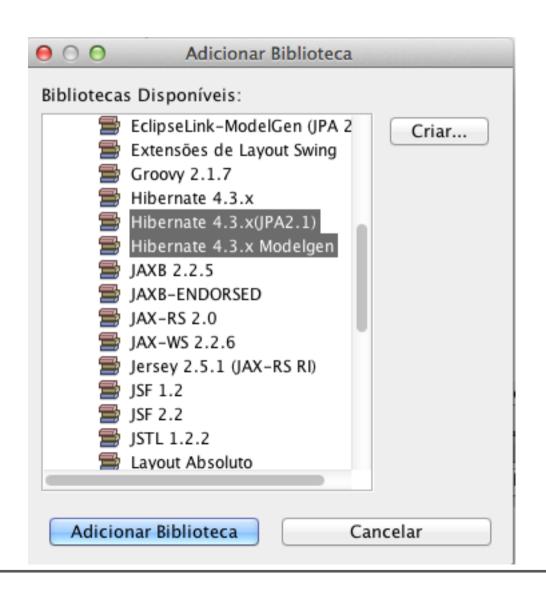
Anotações

- @Entity
 - Todas as classes persistentes mapeadas com anotações são declaradas com essa anotação
- @Table
 - Usada quando o nome da tabela mapeada é diferente do nome da classe
- @ld
 - Usado para mapear um atributo da classe como chave primária da tabela

Anotações

- @GeneratedValue
 - Permite a geração automática para o valor do identificador
 - @GeneratedValue(strategy = GenerationType.SEQUENCE)
- @Column
 - Utilizado quando o nome do atributo é diferente da coluna mapeada
 - @Column(name="id_aluno")

Bibliotecas



Anotações

Importar: javax.persistence.*;

```
* @author romualdocosta
*/
//Anotação que informa que a classe mapeada é persistente
@Entity
//Informando nome e esquema da tabela mapeada
@SequenceGenerator(name = "seq", sequenceName = "aluno_seq")
@Table(name="aluno")
public class Aluno {
//Definição da chave primária
@Id
//Definição do mecanismo de definição da chave primária
@GeneratedValue(generator = "seq")
//Informa o nome da coluna mapeada para o atributo
@Column(name="id_aluno")
    private int id;
```

Conexão com Banco de Dados

- Instanciar um objeto de configuração
 - org.hibernate.cfg.Configuration
 - Inserir as propriedades no programa
- Usar o arquivo .properties
 - Indicar as classes mapeadas no programa
- Usar um arquivo de configuração
 - hibernate.cfg.xml
 - Especificar as propriedades de inicialização e caminhos das classes mapeadas

hibernate.cfg.xml

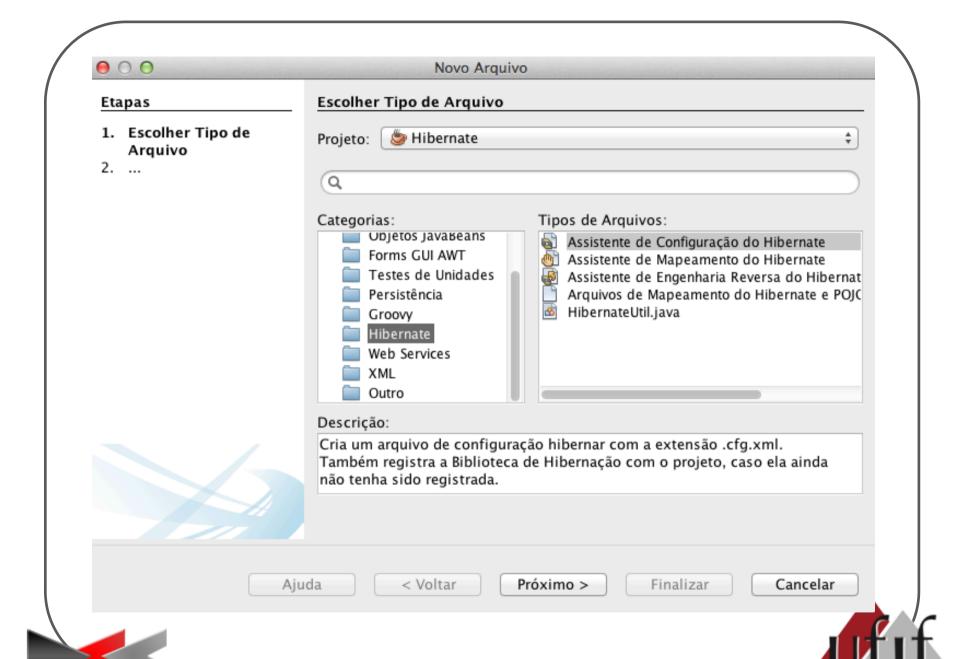
- Url de conexão com o banco de dados
 - hibernate.connection.driver_class
 - hibernate.connection.url
- Usuário e senha do banco de dados
 - hibernate.connection.username
 - hibernate.connection.password
- Número máximo e mínimo de conexões:
 - hibernate.connection.pool_size
- Dialeto SQL
 - hibernate.dialect

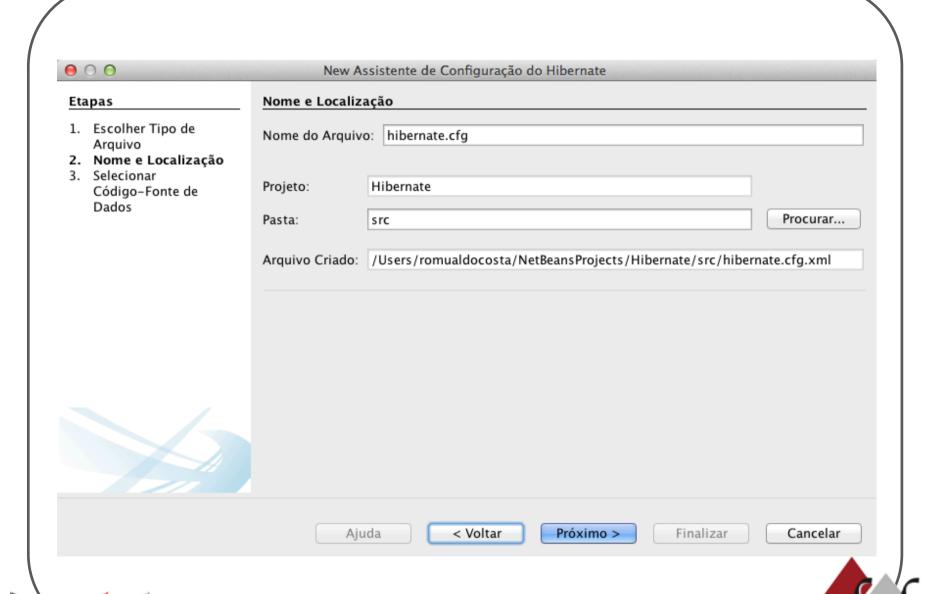


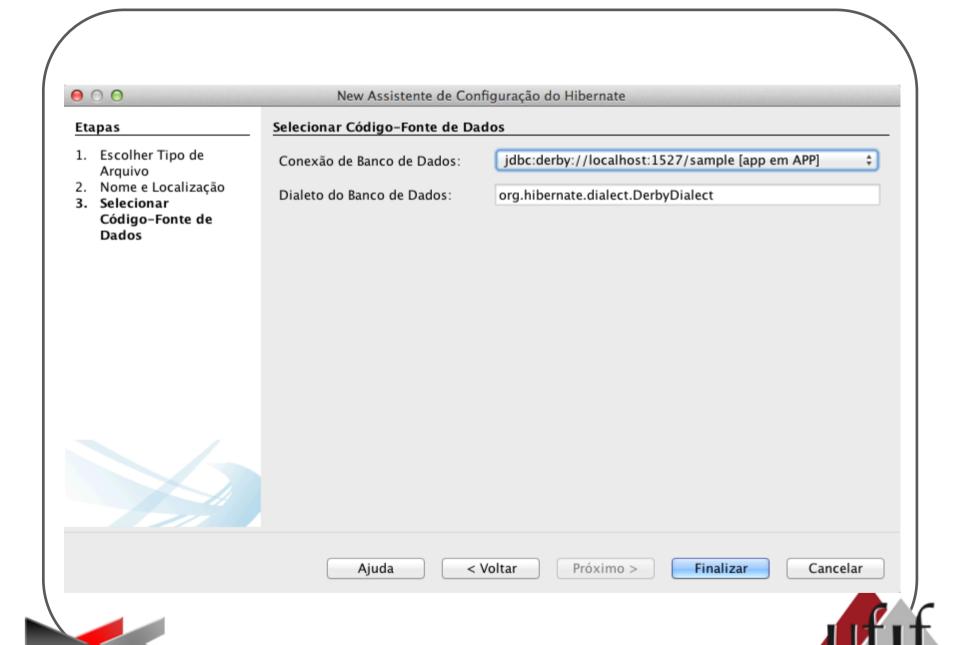


Dialetos

```
DB2 - org.hibernate.dialect.DB2Dialect
HypersonicSQL - org.hibernate.dialect.HSQLDialect
Informix - org.hibernate.dialect.InformixDialect
Ingres - org.hibernate.dialect.IngresDialect
Interbase - org.hibernate.dialect.InterbaseDialect
Pointbase - org.hibernate.dialect.PointbaseDialect
PostgreSQL - org.hibernate.dialect.PostgreSQLDialect
Mckoi SQL - org.hibernate.dialect.MckoiDialect
Microsoft SQL Server - org.hibernate.dialect.SQLServerDialect
MySQL - org.hibernate.dialect.MySQLDialect
Oracle (any version) - org.hibernate.dialect.OracleDialect
Oracle 9 - org.hibernate.dialect.Oracle9Dialect
Progress - org.hibernate.dialect.ProgressDialect
FrontBase - org.hibernate.dialect.FrontbaseDialect
SAP DB - org.hibernate.dialect.SAPDBDialect
Sybase - org.hibernate.dialect.SybaseDialect
Sybase Anywhere - org.hibernate.dialect.SybaseAnywhereDialect
```



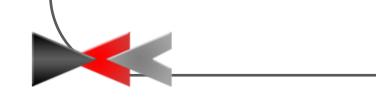








Hibernate.cfg.xml





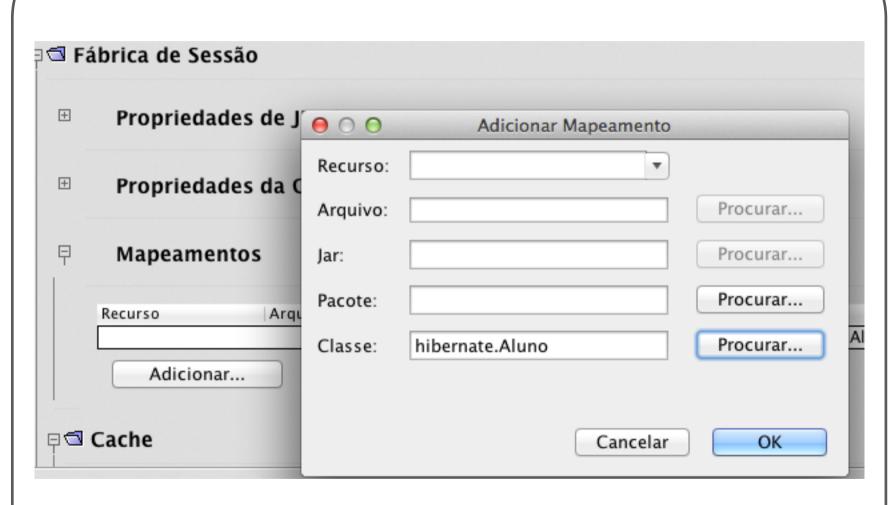
Mapeamento de Classes

- No final do arquivo de configuração devem ser informadas as classes mapeadas que o Hibernate deve processar
 - Se não for definida ela não poderá ser persistida

Recurso	Arquivo	Jar	Pacote	Classe
				hibernate.Aluno











Manipulando Objetos Persistentes

- Objetos Session
 - Sessão de comunicação com o banco de dados
 - É ele que abre a conexão e realiza a persistência
- Objeto SessionFactory
 - Deve ser chamado uma única vez durante a execução da aplicação
- Objeto Configuration
 - Acesso as configurações
 - Permite a conexão com o banco de dados

Objeto Configuration

```
import org.hibernate.*;
import org.hibernate.cfg.*;

public class Hibernate {

   public static void main(String[] args) {
        // TODO code application logic here
        //Cria objeto que receberá as configurações
        Configuration cfg = new Configuration();
        //Informe o arquivo XML que contém a configurações
        cfg.configure("hibernate.cfg.xml");
        //Cria uma fábrica de sessões.
        //Deve existir apenas uma instância na aplicação
```





Session

```
//Cria uma fábrica de sessões.
//Deve existir apenas uma instância na aplicação
SessionFactory sf = cfg.buildSessionFactory();
// Abre sessão com o Hibernate
Session session = sf.openSession();
//Cria uma transação
Transaction tx = session.beginTransaction();
// Cria objeto Aluno
Aluno aluno = new Aluno();
aluno.setNome("Jose da Silva");
aluno.setMatricula(100000001);
aluno.setCpf(100000001);
session.save(aluno); // Realiza persistência
tx.commit(); // Finaliza transação
session.close(); // Fecha sessão
```

Session - Métodos

- Save(object)
 - Inclui um objeto em uma tabela
- SaveorUpdate(object)
 - Inclui um objeto caso ele n\u00e3o exista ou atualiza o objeto caso ele exista
- Delete(object)
 - Apaga um objeto na tabela
- Get(Class, Serializable id)
 - Retorna um objeto a partir da sua chave primária. A classe é o primeiro parâmetro e o seu identificador o segundo

Atualização

```
// Busca objeto aluno da base de dados com chave primária = 1
aluno = (Aluno) session.get(Aluno.class, 1);
// Atualiza informação de matrícula.
aluno.setMatricula(200027807);
// Como o identificador do objeto aluno é diferente de 0,
// a sua matrícula é atualizada já que foi alterada
session.saveOrUpdate(aluno);
```





Remoção

```
Aluno aluno = new Aluno();
aluno.setId(1);
session.delete(aluno);
```

