
Prog. Orientada a Objetos e Mapeamento Objeto-Relacional – IMD0104

Herança

João Carlos Xavier Júnior

jcxavier@imd.ufrn.br

Mapeamento de Herança em JPA

- ❑ Existem **3 maneiras** de realizar herança com o JPA.
- ❑ Uso da anotação **@Inheritance**:
 - ❖ `@Inheritance(strategy=InheritanceType....)`
- ❑ O **InheritanceType** define 3 valores para declarar a estratégia a ser usada:
 - ❖ `InheritanceType.SINGLE_TABLE`
 - ❖ `InheritanceType.JOINED`
 - ❖ `InheritanceType.TABLE_PER_CLASS`
- ❑ O valor padrão é `InheritanceType.SINGLE_TABLE`.

Estratégias de Herança

❑ Single Table:

- ❖ A estratégia **Single Table** mapeia todas as classes da hierarquia na **superclasse**.
- ❖ A anotação que define a estratégia de mapeamento é colocada na superclasse.
- ❖ Na tabela de mapeamento é criada **uma coluna** para **cada campo** de dados de cada classe da hierarquia.
- ❖ Quanto maior for a hierarquia, mais larga será a tabela mapeada (**colunas vazias**).

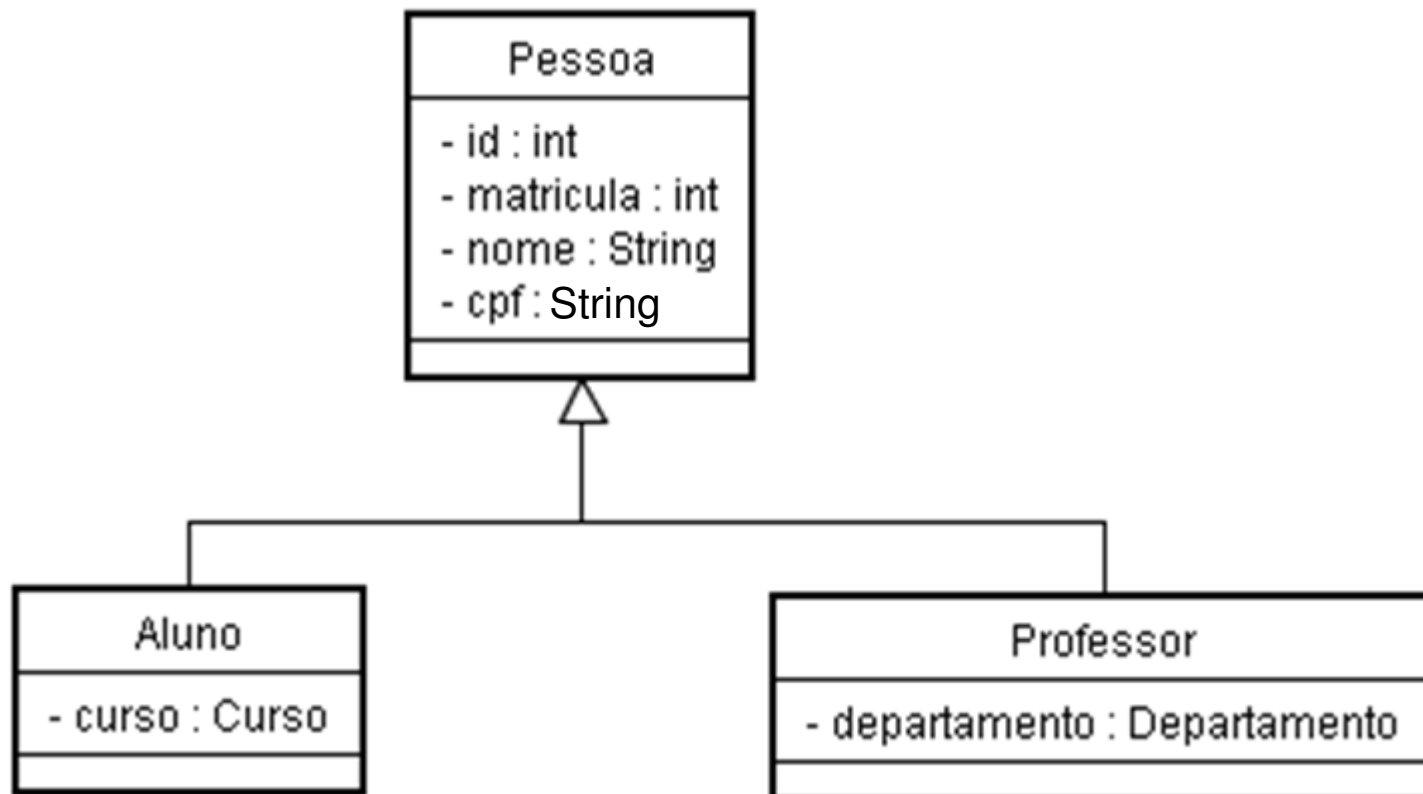
Estratégias de Herança

❑ Single Table:

- ❖ Mapeamento **single table** é o **mais rápido** de todos os modelos de mapeamento de herança (**ausência de join, uso de insert e update**).
- ❖ Neste mapeamento, **todos os objetos** da hierarquia são mapeados na **tabela da superclasse**.
- ❖ Também é chamado de **mapeamento de hierarquia plano**.

Estratégias de Herança

❑ Single Table:



Single Table

❑ Anotações:

- ❖ `@Inheritance`: para informar o **tipo de herança**.

```
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
```

- ❖ `@DiscriminatorColumn`: define a coluna que vai diferenciar as entidades:

- `name`: nome da coluna;
- `discriminatorType`: tipo de dado da coluna.

```
@DiscriminatorColumn(name = "tipo_pessoa",  
discriminatorType = DiscriminatorType.INTEGER)
```

Single Table

❑ Tabela por Hierarquia:

<<Table>> PESSOA	
- ID_PESSOA	<<PK>>
- MATRICULA	
- NOME	
- CPF	
- TIPO_PESSOA	
- ALUNO_ID_CURSO	
- PROFESSOR_ID_DEPARTAMENTO	

Single Table

❑ Criando a classe **Pessoa**:

```
@Entity
@SequenceGenerator(name="SEQ_PESSOA", initialValue=1, allocationSize=1,
sequenceName="seq_pessoa")
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
@DiscriminatorColumn(name = "tipo_pessoa",
discriminatorType = DiscriminatorType.INTEGER)
@DiscriminatorValue("0")
public class Pessoa implements Serializable{

    @Id
    @GeneratedValue(strategy=GenerationType.SEQUENCE,generator="SEQ_PESSOA")
    @Column(name="id_pessoa")
    private int id;

    private int matricula;
    private String nome;
    private String cpf;

    // Getters and Setters
```


Single Table

- ❑ Criando a classe **Aluno**:

```
@Entity
@DiscriminatorValue("1")
public class Aluno extends Pessoa{
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "aluno_id_curso")
    private Curso curso;

    // Getters and Setters
}
```

Single Table

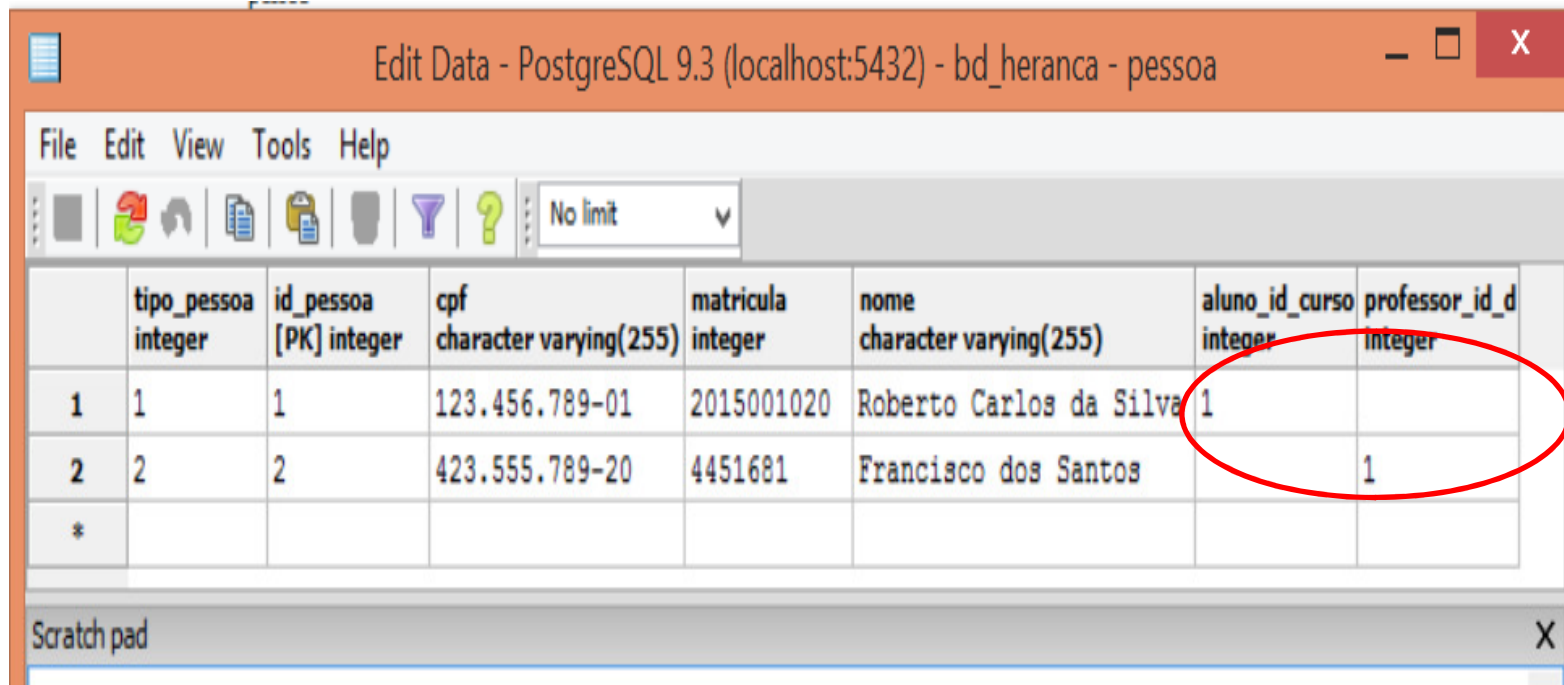
- ❑ Criando a classe **Professor**:

```
@Entity
@DiscriminatorValue("2")
public class Professor extends Pessoa{
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "professor_id_departamento")
    private Departamento departamento;

    // Getters and Setters
}
```

Single Table

❑ Tabela Pessoa:



File Edit View Tools Help

No limit

	tipo_pessoa integer	id_pessoa [PK] integer	cpf character varying(255)	matricula integer	nome character varying(255)	aluno_id_curso integer	professor_id_d integer
1	1	1	123.456.789-01	2015001020	Roberto Carlos da Silva	1	
2	2	2	423.555.789-20	4451681	Francisco dos Santos		1
*							

Scratch pad

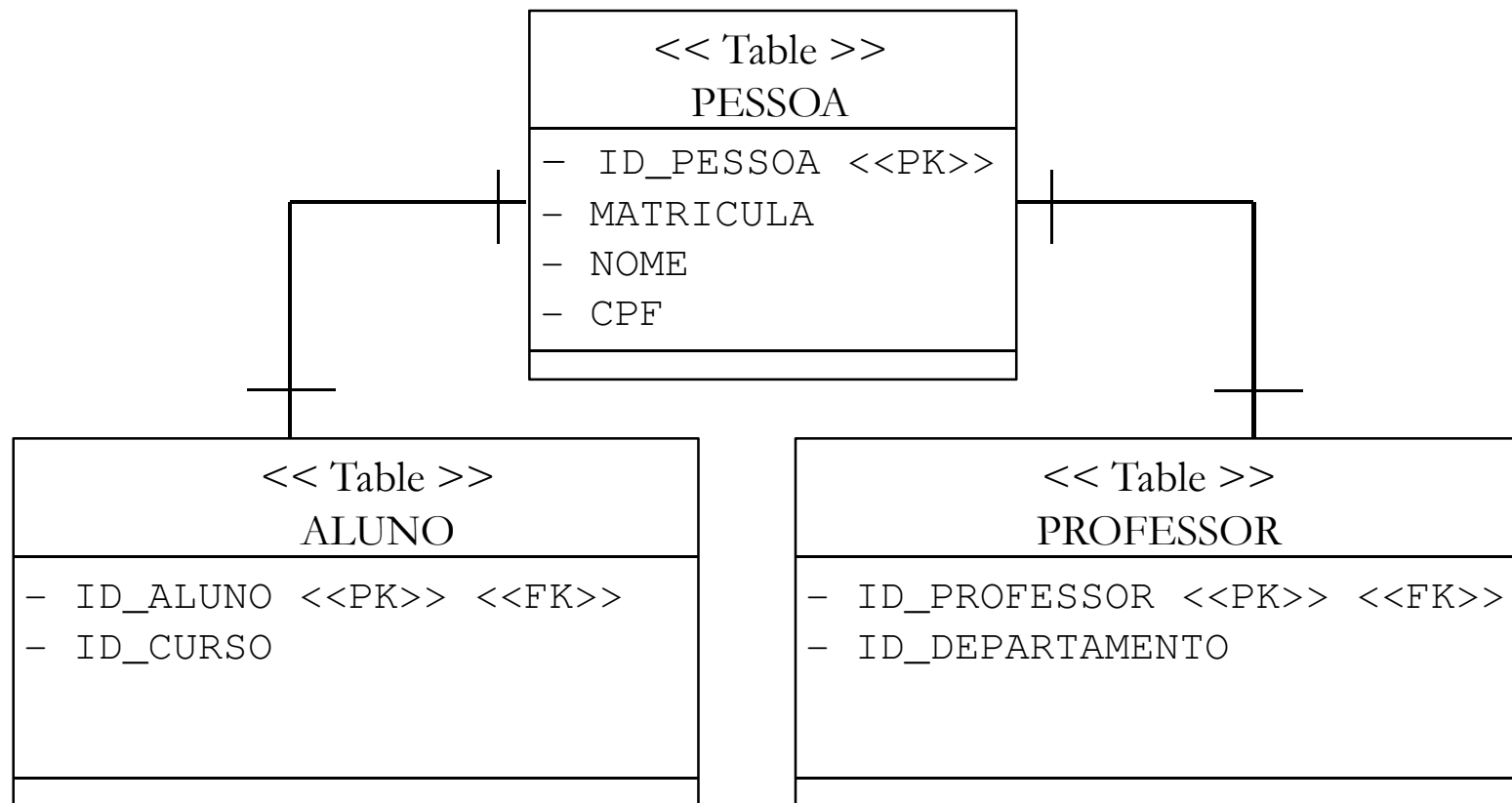
Estratégias de Herança

❑ Joined:

- ❖ A estratégia **Joined** usa uma tabela diferente para cada classe da hierarquia.
- ❖ A raiz da hierarquia de classes é representada por uma única tabela e cada subclasse é representada por uma tabela separada.
- ❖ Cada tabela de uma subclasse contém apenas os campos definidos na subclasse e chaves.

Estratégias de Herança

❑ Joined:



Joined

- ❑ Tabela por subclasse:
 - ❖ Tabelas filhas apenas com **colunas não herdadas**.
 - ❖ PK de tabelas filhas são FK para a tabela mãe.
 - ❖ Recuperação de classe filha => **JOIN** entre tabela filha e tabela mãe.

Joined

❑ Tabela por subclasse:

```
@Entity
@Table(name = "pessoa02")
@SequenceGenerator(name="SEQ_PESSOA02", initialValue=1, allocationSize=1,
sequenceName="seq_pessoa02")
@Inheritance(strategy = InheritanceType.JOINED)

public class Pessoa02 implements Serializable{
    @Id
    @GeneratedValue(strategy=GenerationType.SEQUENCE,generator="SEQ_PESSOA02")
    @Column(name="id_pessoa02")
    private int id;
    private int matricula;
    private String nome;
    private String cpf;

    // Getters and Setters
}
```

Joined

❑ Tabela por subclasse:

```
@Entity
@Table(name="aluno02")
@PrimaryKeyJoinColumn(name = "id_aluno02")
public class Aluno02 extends Pessoa02{
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "aluno02_id_curso")
    private Curso curso;

    // Getters and Setters
}
```


Joined

❑ Tabela por subclasse:

```
@Entity
@Table(name="professor02")
@PrimaryKeyJoinColumn(name = "id_professor02")
public class Professor02 extends Pessoa02{
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "professor02_id_departamento")
    private Departamento departamento;

    // Getters and Setters
}
```

Joined

❑ Tabela Pessoa:

Edit Data - PostgreSQL 9.3 (localhost:5432) - bd_heranca - pessoa02

	id_pessoa02 [PK] integer	cpf character varying(255)	matricula integer	nome character varying(255)
1	1	456.777.799-00	12345	Francisco das Chagas
2	2	123.456.789-01	2015001020	Roberto Carlos da Silva
3	3	423.555.789-20	4351681	João Carlos Xavier Júnior
*				

Joined

❑ Tabelas Aluno e Professor:

Edit Data - PostgreSQL 9.3 (localhost:5432) - bd_heranca - aluno02		
File Edit View Tools Help		
No limit		
	id_aluno02 [PK] integer	aluno02_id_curso integer
1	2	1
*		

Edit Data - PostgreSQL 9.3 (localhost:5432) - bd_heranca - professor02		
File Edit View Tools Help		
No limit		
	id_professor02 [PK] integer	professor02_id_departamento integer
1	3	1
*		

Estratégias de Herança

❑ Table per class:

- ❖ A estratégia **table-per-class** usa uma tabela diferente para cada classe da hierarquia.
- ❖ Parecida com a estratégia **joined**.
- ❖ Porém, cada tabela inclui todo o **estado de uma instância** (propriedades herdadas ou não são mapeadas para colunas na tabela correspondente)

Estratégias de Herança

❑ Table per class:

<< Table >> ALUNO	
-	ID_ALUNO <<PK>>
-	MATRICULA
-	NOME
-	CPF
-	ID_CURSO

<< Table >> PROFESSOR	
-	ID_PROFESSOR <<PK>>
-	MATRICULA
-	NOME
-	CPF
-	ID_DEPARTAMENTO

Table per class

- ❑ Tabela por classe concreta:

```
@MappedSuperclass
public class Pessoa03 {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    @Column(name="id_pessoa03")
    private int id;
    private int matricula;
    private String nome;
    private String cpf;

    // Getters and Setters
}
```

Table per class

- ❑ Tabela por classe concreta:

```
@Entity
@Table(name="aluno03")
@Inheritance(strategy = InheritanceType.TABLE PER CLASS)
public class Aluno03 extends Pessoa03{
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "aluno03_id_curso")
    private Curso curso;

    // Getters and Setters
}
```

Table per class

- ❑ Tabela por classe concreta:

```
@Entity
@Table(name="professor03")
@Inheritance(strategy = InheritanceType.TABLE PER CLASS)
public class Professor03 extends Pessoa03{
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "professor03_id_departamento")
    private Departamento departamento;

    // Getters and Setters
}
```


Table per class

□ Tabelas Aluno e Professor:

Edit Data - PostgreSQL 9.3 (localhost:5432) - bd_heranca - aluno03						
File Edit View Tools Help						
						No limit
	id_pessoa03 [PK] integer	cpf character varying(255)	matricula integer	nome character varying(255)	aluno03_id_curso integer	
1	10	123.456.789-01	2015001020	Roberto Carlos da Silva	1	
*						

Edit Data - PostgreSQL 9.3 (localhost:5432) - bd_heranca - professor03						
File Edit View Tools Help						
						No limit
	id_pessoa03 [PK] integer	cpf character varying(255)	matricula integer	nome character varying(255)	professor03_id_departamento integer	
1	20	423.555.789-20	4351681	João Carlos Xavier Júnior	1	
*						

Dúvidas...

