



Argentina
programa
4.0



UNIVERSIDAD
TECNOLOGICA
NACIONAL

ORM III

Etapas 2 - Desarrollador Java Intermedio



ORM III

- ## - Mapeo - Parte III

[illegible]

Impedance Mismatch - Mapeo de Herencia - Single Table

- Suponiendo que existe una única superclase y N clases hijas, el resultado de mapear la herencia con la estrategia Single Table generará una única tabla en la base de datos.
- Esta única tabla contendrá una columna por cada uno de los atributos persistentes de la superclase + una columna por cada atributo persistentes de cada una de las clases hijas.
- Además, tendrá una columna que actuará de “***campo discriminador***”, la cual indicará a qué clase pertenece la instancia/fila en cuestión.

Mapeo Objeto Relacional - *Annotations*

@Inheritance(strategy = InheritanceType.SINGLE_TABLE)

```
import javax.persistence.*;

@Entity
@Table(name="medio_de_transporte")
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
@DiscriminatorColumn(name="tipo")
public abstract class MedioDeTransporte {
    //atributos
}
```

Mapeo Objeto Relacional - *Annotations*

@Inheritance(strategy = InheritanceType.SINGLE_TABLE)

```
import javax.persistence.*;
```

```
@Entity
```

```
@DiscriminatorValue(value="aereo")
```

```
public class MedioAereo extends MedioDeTransporte {
```

```
    //atributos
```

```
}
```

Impedance Mismatch - Mapeo de Herencia - Single Table

Si consideramos una superclase con N atributos persistentes; una clase hija A con M atributos persistentes; y otra clase hija B con P atributos persistentes; entonces:

- Si se persiste una instancia de la clase A (con todos sus atributos seteados, inclusive los pertenecientes a la superclase), quedarán P columnas nulas.
- Si se persiste una instancia de la clase B (con todos sus atributos seteados, inclusive los pertenecientes a la superclase), quedarán M columnas nulas.

Impedance Mismatch - Mapeo de Herencia - Single Table



- Como consecuencia de la utilización de esta estrategia de mapeo de herencia, varias columnas de la tabla resultante podrían ser nulas.
- Pero, en contraparte, se obtiene una buena performance ya que solamente se debe consultar una única tabla.

Impedance Mismatch - Mapeo de Herencia - Joined

Suponiendo que existe una única superclase y N clases hijas, el resultado de mapear la herencia con la estrategia Joined generará, en la base de datos, una tabla por la superclase y N tablas más, una por cada clase hija.

Impedance Mismatch - Mapeo de Herencia - Joined

- La tabla que representa a la superclase contendrá, únicamente, tantas columnas como atributos persistentes existan en dicha clase. Además, puede contener opcionalmente el campo discriminador.
- Cada una de las tablas que representan a las clases hijas contendrán tantas columnas como atributos persistentes existan en dichas clases.

Impedance Mismatch - Mapeo de Herencia - Joined

- Cada una de las PKs de las tablas que representan a las clases hijas, a su vez serán una FK a la tabla que representa a la superclase.
- Para recuperar un objeto con todos sus atributos (los propios + los que están en la superclase), el ORM debe joinear las tablas.

Mapeo Objeto Relacional - *Annotations*

@Inheritance(strategy = InheritanceType.JOINED)

```
import javax.persistence.*;
```

```
@Entity
```

```
@Table(name="medio_de_transporte")
```

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

```
public abstract class MedioDeTransporte {
```

```
    @Id
```

```
    @GeneratedValue
```

```
    private int id;
```

```
}
```

Mapeo Objeto Relacional - *Annotations*

@Inheritance(strategy = InheritanceType.JOINED)

```
import javax.persistence.*;

@Entity
@Table(name="medio_de_transporte_aereo")
public class MedioAereo extends MedioDeTransporte {
    //atributos
}
```

Impedance Mismatch - Mapeo de Herencia - Table per class

Suponiendo que existe una única superclase y N clases hijas concretas, el resultado de mapear la herencia con la estrategia Table per Class generará, en la base de datos, N tablas: una por cada clase hija.

Impedance Mismatch - Mapeo de Herencia - Table per class

- Todos los atributos persistentes de la superclase serán persistidos en cada una de las tablas que mapean contra las clases hijas.
- Para recuperar polimórficamente todos los objetos, el ORM debe realizar unions entre todas las tablas que mapean contra las clases hijas.

Mapeo Objeto Relacional - *Annotations*

@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)

```
import javax.persistence.*;
```

```
@Entity
```

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

```
public abstract class MedioDeTransporte {
```

```
    @Id
```

```
    @GeneratedValue
```

```
    private int id;
```

```
}
```

Mapeo Objeto Relacional - *Annotations*

@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)

```
import javax.persistence.*;
```

```
@Entity
```

```
@Table(name="medio_de_transporte_aereo")
```

```
public class MedioAereo extends MedioDeTransporte {
```

```
    //atributos
```

```
}
```


Gracias!