



CURSO DE **HIBERNATE 5**


OpenWebinars



HIBERNATE

(2)

HIBERNATE



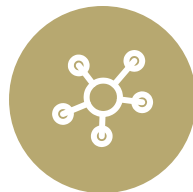
Más que un ORM.
Comparativa con
otros productos. JPA.
Maven. Módulos

(4)

ENTIDADES



Definición del modelo
del dominio. Entidades
y ciclo de vida. XML y
anotaciones. Tipos de
datos.



(1)

INTRODUCCION

Persistencia, desfase
objeto-relacional,
ORM. Productos y
estándares



(3)

PRIMER PROYECTO

Hibernate.cfg.xml,
EntityManager y
persistence.xml



(5)

ASOCIACIONES

ManyToOne, OneToMany,
OneToOne, ManyToMany



HIBERNATE

(7)

COLECCIONES



Mapeo de colecciones.
Tipos (list, set, map).
Colecciones ordenadas (sorted vs. ordered).

(9)

CONTEXTO DE PERSISTENCIA



Almacenamiento, recuperación y borrado de entidades.



(6)

ELEMENTOS AVANZADOS

Campos calculados, herencia.



(8)

GENERACION DEL ESQUEMA

Customización del proceso de generación del esquema.



(10)

TRANSACCIONES

Control de concurrencia.
Patrones y antipatrones.



HIBERNATE

(12) ENVERS



Introducción a la
auditoria de entidades.



(11) CONSULTAS HPQL VS JPQL

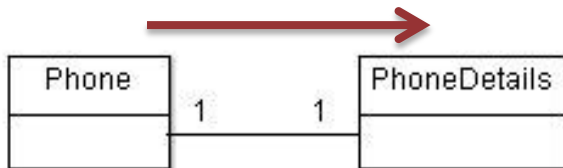
Consultas con
parámetros,
Anotaciones. SQL nativo



1.

**ASOCIACIONES UNO
A UNO
UNIDIRECCIONALES**

ASOCIACIONES UNO A UNO UNIDIRECCIONALES



```
@OneToOne
@JoinColumn(name = "details_id")
private PhoneDetails details;
```

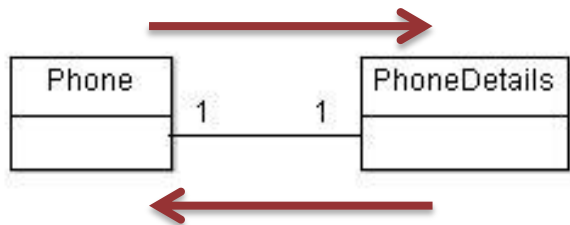
En el tratamiento parece una asociación *ManyToOne*, pero es diferente. Aquí, el lado propietario es **Phone**, que es quien marca la dependencia de existencia con **PhoneDetails** (no tiene sentido una instancia de **PhoneDetails** sin una de **Phone**)



2.

**ASOCIACIONES UNO
A UNO
BIDIRECCIONALES**

ASOCIACIONES UNO A UNO BIDIRECCIONALES



```
@OneToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "phone_id")
private Phone phone;
```

```
@OneToOne(mappedBy = "phone", cascade = CascadeType.ALL, orphanRemoval = true, fetch = FetchType.LAZY)
private PhoneDetails details;
```

```
public void addDetails(PhoneDetails details) {
    details.setPhone(this);
    this.details = details;
}
```

```
public void removeDetails() {
    if (details != null) {
        details.setPhone(null);
        this.details = null;
    }
}
```