# 03\_RELACIONES

## ASOCIACIONES HIBERNATE

# Uno a uno (unidireccional)

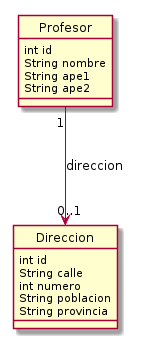
La relación uno a uno en Hibernate consiste simplemente en que un objeto tenga una referencia a otro objeto de forma que al persistirse el primer objeto también se persista el segundo.

En esta lección la relación va a ser unidireccional es decir que que la relación uno a uno va a ser en un único sentido.

**Ejemplo**

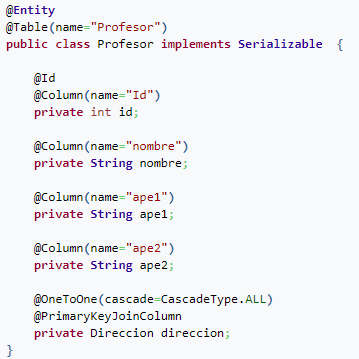
Para nuestro ejemplo vamos a usar las clases:

* Profesor
* Dirección



Entidad No Owner

Entidad Owner



A la propiedad direccion (línea 24) se han añadido dos anotaciones para indicar la relación *uno a uno* y que ésta relación se implemente mediante la clave primaria.

* **@OneToOne(cascade=CascadeType.ALL)**: Esta anotación indica la relación *uno a uno* de las 2 tablas. Tambien indicamos el valor de cascade al igual que en el fichero de hibernate.
* **@PrimaryKeyJoinColumn**: Indicamos que la relación entre las dos tablas se realiza mediante la clave primaria.

En código de **Direccion** no es necesario indicar nada sobre la relación tal y como hemos explicado en el caso del fichero de hibernate.

# Uno a uno (bidireccional)

Esta lección es muy similar a la anterior , pero en éste caso la relación entre las clases Profesor y Direccion va a ser bidireccional.

**EJEMPLO:**

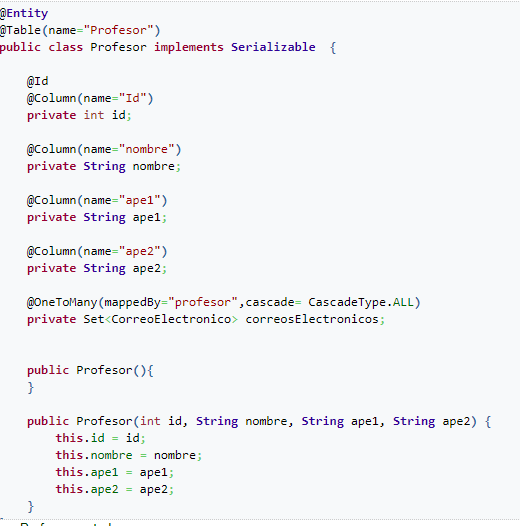
**Cuando por un correo electrónico quiero traer a una persona o viceversa**

# Uno a muchos (desordenada)

La relación uno a muchos consiste simplemente en que un objeto padre tenga una lista **sin ordenar** de otros objetos hijo de forma que al persistirse el objeto principal también se persista la lista de objetos hijo. Esta relación también suele llamarse maestro-detalle o padre-hijo.

**OneToMany**:Como su nombre indica le dice a Hibernate que esta propiedad contendrá la lista de hijos.

* **mappedBy**: Este atributo contendrá el nombre de la propiedad Java de la clase hija que enlaza con la clase padre. En nuestro ejemplo es el nombre de la propiedad profesor que se encuentra en la clase CorreoElectronico.
* **cascade**: Este atributo tiene el mismo significado que el del fichero de mapeo de Hibernate. Mas información en [Cascade](http://cursohibernate.es/doku.php?id=unidades:03_relaciones:06_cascade" \o "unidades:03_relaciones:06_cascade).



A la propiedad profesor se han añadido dos anotaciones para indicar la relación:

* **ManyToOne**:Al ser el otro lado de la relación indicamos que desde este lado es una relación *muchos a uno*.
* **JoinColumn**: Indicaremos el nombre de la columna que en la tabla *hija* contiene la clave ajena a la tabla *padre*. En nuestro ejemplo es la columna de la base de datos IdProfesor que se encuentra en la tabla CorreoElectronico la cual enlaza con la tabla Profesor.



# Uno a muchos (ordenada)

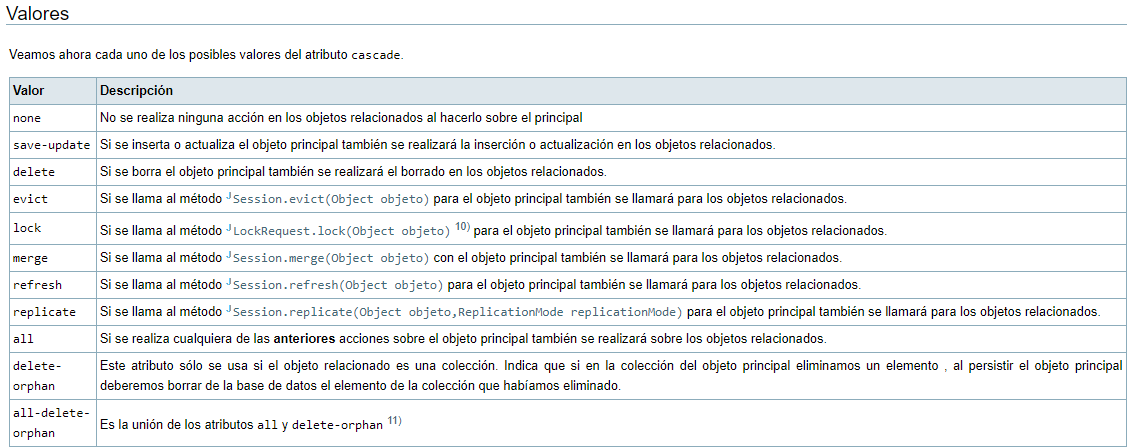
La relación uno a muchos consiste simplemente en que un objeto padre tenga una lista **Ordenar** de otros objetos hijo de forma que al persistirse el objeto principal también se persista la lista de objetos hijo. Esta relación también suele llamarse maestro-detalle o padre-hijo.

# Cascade

Ya hemos visto en esta unidad cómo crear relaciones entre objetos y que al definirlas existe el atributo cascade. Este atributo dispone de 11 posibles valores.

¿Cuál es el significado del atributo cascade? El significado es indicar qué debe hacer hibernate con las clases relacionadas cuando realizamos alguna acción con la clase principal.

Es decir, si borramos la clase principal, ¿debería borrarse la clase relacionada?. La respuesta a ésta y otras preguntas depende de nuestro modelo de clases, por ello existen 11 valores distintos y deberemos elegir entre todos ellos.



# Equals