

# Hibernate

## Queries

- Hibernate tiene varias formas de consultar a la base.
- Existen tres formas de escribir un query:
  - En HQL (Hibernate Query Language)
    - `session.createQuery("from Persona p where p.nombre like \"%Pablo%\");`
  - Con un Criteria (consultas dinámicas)
    - `session.createCriteria(Persona.class).add(Expression.like("nombre", "%Pablo%"));`
  - Con SQL
    - `session.createSQLQuery("select {p.*} from PERSONA {p} where nombre like \"%Pablo%\"", p, Persona.class)}`

## Interfaces Query y Criteria

- Las interfaces `Query` y `Criteria` definen varios métodos para controlar la ejecución de una consulta.
- `Query` provee métodos para hacer binding entre parámetros concretos y los parámetros de la consulta.
- Para ejecutar una `Query` es necesario obtener una instancia de estas interfaces usando `Session`.
- `Criteria` se utiliza para generar consultas dinámicas.

## Instanciando Query

- Para crear una nueva instancia de `Query`, se debe invocar a:

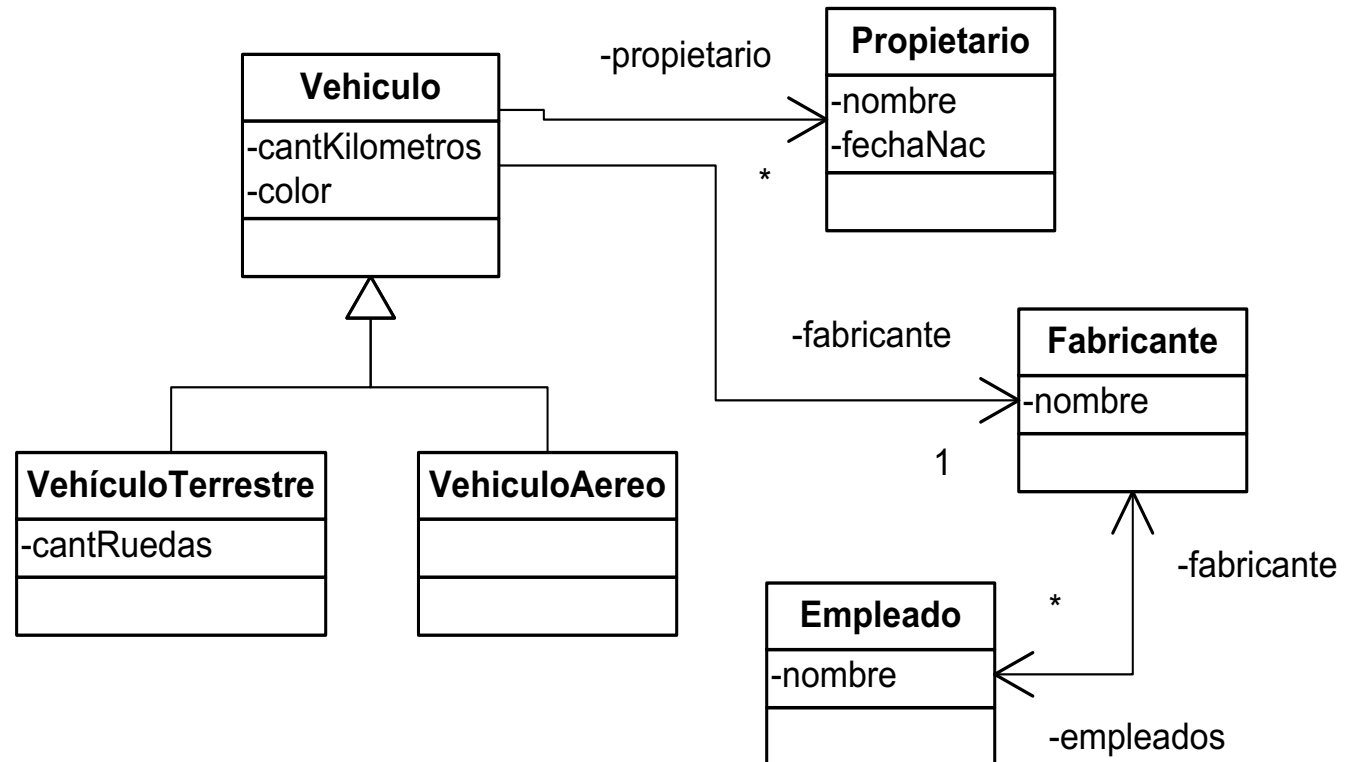
- `createQuery()` : prepara una consulta con HQL

- `Query hqlQuery = session.createQuery("from Persona");`

- `createSQLQuery()` : crea una consulta SQL usando la sintaxis de la base subyacente

- `Query sqlQuery = session.createSQLQuery("select {p.*} from PERSONA {p}", "p", Persona.class)`

## Ejemplos



## Consultas

- Listar todos los vehículos

```
from Vehiculo v
```

- Listar todos los vehículos rojos

```
from Vehiculo v
```

```
where v.color like "rojo"
```

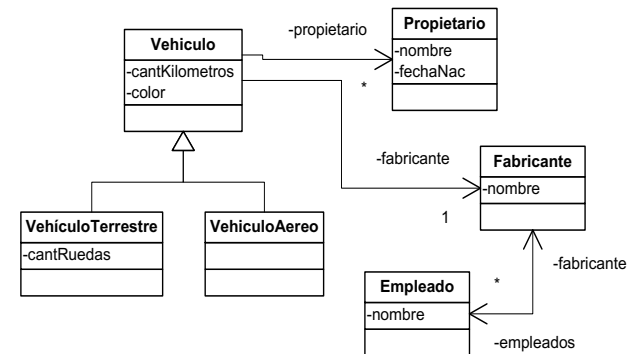
- Listar todos los vehículos del fabricante con nombre honda

```
select v
```

```
from vehiculo v
```

```
    join v.fabricante as f
```

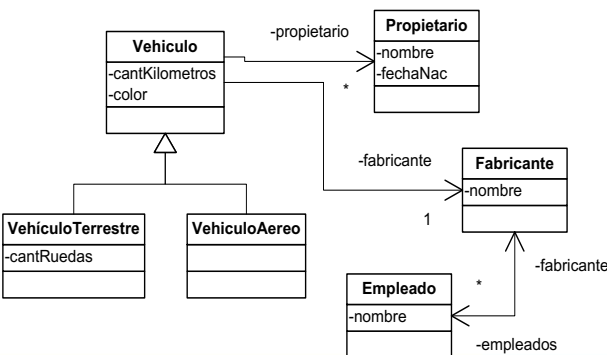
```
where f.nombre like "honda"
```



## Consultas

- Listar todos los vehículos rojos de más de 20000 kilómetros que pertenecen al fabricante honda y cuyo propietario se llama Juan.

```
select v
from Vehiculo as v
      join v.propietario as p
      join v.fabricante as f
where v.color like "rojo" and
      v.cantKilometros > 20000 and
      f.nombre like "honda" and
      p.nombre like "juan"
```



## Consultas

- Listar todos los vehículos terrestres rojos cuyo kilometraje es mayor al promedio.

```
select v
from VehiculoTerrestre as v
where
    v.color like "rojo"
    v.cantKilometros > (
        select avg(v1.cantKilometros)
        from Vehiculo as v1 )
```



## Externalización de consultas

- Hibernate permite guardar consultas en un archivo xml y recuperarlas por un nombre.
- Esto se conoce como consultas nombradas
  - Ejemplo:

```
session.getNamedQuery("buscarPersonasPorNombre")  
    .setString("nombre", nombre)  
    .list();
```

- Donde “buscarPersonasPorNombre” está en Persona.hbm.xml usando el elemento <query>

```
<query name="buscarPersonasPorNombre">  
    <![CDATA[from Persona persona  
    where persona.nombre like :nombre  
    ]]>  
</query>
```

No necesariamente tiene que ser HQL, puede ser SQL.

## Tipos de retorno posibles

- Array de Object[]

```
select mother, offspr, mate.name
from DomesticCat as mother
    inner join mother.mate as mate
    left outer join mother.kittens as offspr
```

## Funciones de agregación

- Ejemplos

```
select avg(cat.weight), sum(cat.weight), max(cat.weight), count(cat)
from Cat cat
```

```
select cat.weight + sum(kitten.weight)
from Cat cat
      join cat.kittens kitten
group by cat.id, cat.weight
```

```
select count(distinct cat.name), count(cat) from Cat cat
```

```
select firstName||' '||initial||' '||upper(lastName) from Person
```

## Campos especiales

- Id

```
from Cat as cat where cat.id = 123
```

- Class

```
from Cat cat where cat.class = DomesticCat
```

## Otras funciones (any, some, all, exists)

```
select mother from Cat as mother, Cat as kit  
where kit in elements(foo.kittens)
```

```
select p from NameList list, Person p  
where p.name = some elements(list.names)
```

```
from Cat cat where exists elements(cat.kittens)
```

```
from Player p where 3  
> all elements(p.scores)
```

```
from Show show where 'fizard' in indices(show.acts)
```

## Subconsultas

```
from Cat as fatcat
where fatcat.weight
> (
    select avg(cat.weight) from DomesticCat cat
)
```

```
from DomesticCat as cat
where cat.name = some (
    select name.nickName from Name as name
)
```

```
from Cat as cat
where not exists (
    from Cat as mate where mate.mate = cat
)
```

```
from DomesticCat as cat
where cat.name not in (
    select name.nickName from Name as name
)
```

```
select cat.id, (select max(kit.weight) from cat.kitten kit)
from Cat as cat
```