

# Aplicaciones de Bases de Datos

(Práctica 12 – Análisis de consultas)



# UNIVERSIDAD DE BURGOS

Autores:

Mario Ubierna San Mamés

Jorge Navarro González



## Índice de Contenido

Método consultarPistas() .....	3
Método consultarReservasDePistasLibres .....	4
Método consultarReservasDePistasLibres .....	5

## Método consultarPistas()

```
Hibernate:
/* Pista.findAll */ select
  pista0_.nro as nro1_1_0_,
  reservas1_.fecha as fecha1_2_1_,
  reservas1_.hora as hora2_2_1_,
  reservas1_.PISTA as PISTA3_2_1_,
  jugador3_.nif as nif1_0_2_,
  pista0_.capacidad as capacidad2_1_0_,
  pista0_.descripcion as descripcion3_1_0_,
  reservas1_.PISTA as PISTA3_2_0_,
  reservas1_.fecha as fecha1_2_0_,
  reservas1_.hora as hora2_2_0_,
  jugador3_.apellido as apellido2_0_2_,
  jugador3_.nombre as nombre3_0_2_,
  jugadores2_.FECHA as FECHA1_3_1_,
  jugadores2_.HORA as HORA2_3_1_,
  jugadores2_.PISTA as PISTA3_3_1_,
  jugadores2_.NIF as NIF4_3_1_
from
  HR.Pista pista0_
left outer join
  HR.Reserva reservas1_
    on pista0_.nro=reservas1_.PISTA
left outer join
  HR.RESERVAJUGADOR jugadores2_
    on reservas1_.fecha=jugadores2_.FECHA
    and reservas1_.hora=jugadores2_.HORA
    and reservas1_.PISTA=jugadores2_.PISTA
left outer join
  HR.Jugador jugador3_
    on jugadores2_.NIF=jugador3_.nif
```

Para poder realizar este método hemos usado el método `findAllGraph`, con este método obtenemos todas las pistas en forma de grafo.

En los grafos se pueden cargar dos propiedades, en nuestro caso, hemos usado `javax.persistence.fetchgraph`, todos los atributos indicados en el grafo de entidades están como `EAGER`, los atributos no especificados se tratan como `LAZY`.

Las ventajas de utilizar grafos son :

- Soluciona el problema de las N+1 consultas.
- Trae toda la información relacionada, es decir, que aunque vamos a tener una colección muy grande (la cual va a ser almacenada), después si necesitamos

información no vamos a tener que volver a cargar todo, ya que en esa colección se encuentra toda la información que vamos a necesitar.

Las principales desventajas de los grados son:

- Al traernos toda la información que necesitamos desde el principio la carga de trabajo va a ser muy grande en ese momento inicial, además vamos a almacenar mucha información, la cual a lo mejor no nos va a hacer falta.

## Método consultarReservasDePistasLibres

```
Hibernate:
/* Reserva.findAll */ select
    reserva0_.fecha as fecha1_2_,
    reserva0_.hora as hora2_2_,
    reserva0_.PISTA as PISTA3_2_
from
    HR.Reserva reserva0_
```

Para realizar este método, hemos usado el método findAll de Reserva, dicho método nos devuelve todas las reservas, después comprobamos si los jugadores es menor que la capacidad, de ser así, lo introducimos a pistasLibres.

Las principales ventajas son:

- Como podemos observar en la captura, el código Hibernate es mucho menor que en el anterior método que usábamos grafos.
- La información que se trae es muy rápida, ya que hay que hacer una breve consulta sobre las reservas.

Por otro lado, la desventaja es:

- La principal desventaja es que no nos traemos toda la información de golpe, ya que cada vez que necesitamos la información para cada reserva tenemos que ejecutar la misma consulta.

## Método consultarReservasDePistasLibres

Hibernate:

```
/* Reserva.pistasLibres */ select
    reserva0_.fecha as fecha1_2_,
    reserva0_.hora as hora2_2_,
    reserva0_.PISTA as PISTA3_2_
from
    HR.Reserva reserva0_ cross
join
    HR.Pista pista2_
where
    reserva0_.PISTA=pista2_.nro
    and (
        select
            count(jugadors1_.FECHA)
        from
            HR.RESERVAJUGADOR jugadors1_
        where
            reserva0_.fecha=jugadors1_.FECHA
            and reserva0_.hora=jugadors1_.HORA
            and reserva0_.PISTA=jugadors1_.PISTA
    )<pista2_.capacidad
```

Finalmente, para hacer este método hemos realizado la consulta con una sola JPQL.

Las principales ventajas que podemos apreciar:

- Es más fácil, ya que se parece mucho al código SQL.
- Podemos navegar sobre objetos aunque no estén explícitamente en la cláusula from.

Las desventajas que nos encontramos son:

- Hay una falta de chequeos estáticos sobre el código, lo cual puede producir errores en tiempo de ejecución.