

JPA

Java Persistence Query Language (JPQL)

- Lenguaje similar a SQL
- Recuperación de información
 - `find()`
 - JPQL. Métodos de EntityManager
 - *createQuery. Dinámicas*
 - *createNamedQuery. Estáticas, almacenadas en XML o con anotaciones en los fuentes, junto a la entidad*
 - SQL
 - *createNativeQuery*
- JPQL maneja entidades (en lugar de tablas) y atributos de objetos (en lugar de campos)
- Tipos de sentencias
 - *SELECT* para consultar
 - *UPDATE* o *DELETE* para actualizar

JPA

JPQL. Identificadores

<i>SELECT</i>	<i>FROM</i>	<i>WHERE</i>	<i>UPDATE</i>	<i>DELETE</i>
<i>JOIN</i>	<i>OUTER</i>	<i>INNER</i>	<i>LEFT</i>	<i>GROUP</i>
<i>BY</i>	<i>HAVING</i>	<i>FETCH</i>	<i>DISTINCT</i>	<i>OBJECT</i>
<i>NULL</i>	<i>TRUE</i>	<i>FALSE</i>	<i>NOT</i>	<i>AND</i>
<i>OR</i>	<i>BETWEEN</i>	<i>LIKE</i>	<i>IN</i>	<i>AS</i>
<i>UNKNOWN</i>	<i>EMPTY</i>	<i>MEMBER</i>	<i>OF</i>	<i>IS</i>
<i>AVG</i>	<i>MAX</i>	<i>MIN</i>	<i>SUM</i>	<i>COUNT</i>
<i>ORDER</i>	<i>BY</i>	<i>ASC</i>	<i>DESC</i>	<i>MOD</i>
<i>UPPER</i>	<i>LOWER</i>	<i>TRIM</i>	<i>POSITION</i>	<i>CHARACTER_LENGTH</i>
<i>CHAR_LENGTH</i>	<i>BIT_LENGTH</i>	<i>CURRENT_TIME</i>	<i>CURRENT_DATE</i>	<i>CURRENT_TIMESTAMP</i>
<i>NEW</i>	<i>EXISTS</i>	<i>ALL</i>	<i>ANY</i>	<i>SOME</i>

JPA

JPQL. SELECT

- Consultas básicas

- *SELECT c FROM Clientes c.* Se obtiene una lista con todos los clientes
- *SELECT c.dni FROM Clientes c.* Se obtiene una lista con los DNI de todos los clientes
- *SELECT DISTINCT c.nombre FROM Cliente c.* Se obtiene una lista con todos los nombres diferentes entre los clientes
- *SELECT c.nombre FROM Cliente c WHERE c.nombre LIKE 'j%'* Lista con los nombres que empiecen por j
- *SELECT c.nombre FROM Cliente c WHERE c.id > 2 AND c.id < 5.* Lista con los nombres de clientes, cuyo id esté entre 3 y 4
- *SELECT c.nombre FROM Cliente c WHERE c.dni = '42544422'.* Lista con el nombre , cuyo DNI es...
- *SELECT c.nombre FROM Cliente c WHERE c_pedidos IS EMPTY.* Lista los nombres que no tienen pedidos

JPA

JPQL. SELECT

- Procesamiento de resultados
 - Query.getResultList()
 - Query.getSingleResult()
- Añadir a UserDAO el método:
 - public List<User> getByLevel(int level); Devuelve una lista con todos los usuarios con un nivel igual o superior al fijado

```
String jpql1 = "SELECT u.id FROM User u";
List<?> res = em.createQuery(jpql1).getResultList();

String jpql2 = "SELECT u FROM User u WHERE u.id='1'";
Query query = em.createQuery(jpql2);
User res2 = (User)query.getSingleResult();

query = em.createNamedQuery("getByLevel");
query.setParameter("level", 3);
List<?> res3 = query.getResultList();

@NamedQuery(name = "getByLevel", query = "SELECT u FROM User u WHERE u.level >= :level")
@Entity public class User {}
```

