

Lenguaje de Consulta

JPQL

--

Introducción

JPQL ?

El lenguaje query de JPA puede ser considerado como la version de [SQL orientada a objetos](#).

JPA provee una API [Criteria](#), que permite construir consultas (query) basados en JPQL.

Ejemplo:

```
/** Listado Ordenado por nombre de Alumno */  
// Clase AlumnoFacade  
  
@Override  
public List<Alumno> findAll() {  
  
    EntityManager em = getEntityManager();  
  
    CriteriaBuilder cb = em.getCriteriaBuilder();  
  
    javax.persistence.criteria.CriteriaQuery cq =  
        cb.createQuery();  
  
    .....  
  
}
```

Ejemplo:

```
/** Listado Ordenado por nombre de Alumno */  
@Override  
public List<Alumno> findAll() {  
    EntityManager em = getEntityManager();  
    CriteriaBuilder cb = em.getCriteriaBuilder();  
    javax.persistence.criteria.CriteriaQuery cq =  
        cb.createQuery();  
    Root<Alumno> c = cq.from(Alumno.class);  
    cq.select(c).orderBy(cb.asc(c.get("nombre")));  
    Query q = em.createQuery(cq);  
    return q.getResultList();  
}
```

Ejemplo:

```
/** Listado de Alumno de la LCC **/
```



/** Listado de Alumno de la LCC **/

```
3 public class Alumno implements Serializable {
4
5     private static final long serialVersionUID = 1L;
6     @Id
7     @GeneratedValue(strategy = GenerationType.IDENTITY)
8     @Basic(optional = false)
9     @Column(name = "idalumno")
0     private Integer idalumno;
1     @Column(name = "nombre")
2     private String nombre;
3     @Column(name = "registro")
4     private String registro;
5     @JoinColumn(name = "carrera_idcarrera", referencedColumnName=
6     @ManyToOne(optional = false)
7     private Carrera carreraIdcarrera;
8
9     ... ..
10 }
```

Ejemplo:

```
/** Listado de Alumno de la LCC */  
  
public List<Alumno> findAlumnosLCC() {  
    EntityManager em = getEntityManager();  
    CriteriaBuilder cb = em.getCriteriaBuilder();  
    javax.persistence.criteria.CriteriaQuery cq = cb.createQuery();  
    Root<Alumno> c = cq.from(Alumno.class);  
    cq.select(c);  
    cq.where(cb.equal(c.get(" carreraldcarrera ").get("nombre"),  
        "LCC"));  
  
    Query q = em.createQuery(cq);  
    return q.getResultList();  
}
```

Ejemplo:

```
/** Listado de Alumno de la LCC y LSI */  
  
public List<Alumno> findAlumnos() {  
    EntityManager em = getEntityManager();  
    CriteriaBuilder cb = em.getCriteriaBuilder();  
    javax.persistence.criteria.CriteriaQuery cq = cb.createQuery();  
    Root<Alumno> c = cq.from(Alumno.class);  
    cq.select(c);  
    cq.where(cb.or(  
        cb.equal(c.get("carreraldcarrera").get("nombre"), "LCC"),  
        cb.equal(c.get("carreraldcarrera").get("nombre"), "LSI")));  
  
    Query q = em.createQuery(cq);  
    return q.getResultList();  
}
```


Ejemplo:

```
/** Listado de Alumno con parámetros */
```

```
public List<String> findAlumnos(String unaCarrera) {  
    EntityManager em = getEntityManager();  
    CriteriaBuilder cb = em.getCriteriaBuilder();  
    javax.persistence.criteria.CriteriaQuery cq = cb.createQuery();  
    Root<Alumno> c = cq.from(Alumno.class);  
    cq.select(c);  
  
    ParameterExpression<String> param =  
        em.getCriteriaBuilder().parameter(String.class);  
  
    cq.where(cb.equal(c.get("carreraIdcarrera").get("nombre"), param));  
  
    Query q = em.createQuery(cq);  
  
    q.setParameter(param, unaCarrera);  
    return q.getResultList();  
}
```

Ejemplo:

```
/** Obtener productos de un tipo */  
/** Sistema de facturación */  
public List<String> findProductosTipos(String tipo) {  
    EntityManager em = getEntityManager();  
    CriteriaQuery cq = em.getCriteriaBuilder().createQuery();  
    Root<Productos> rt = cq.from(Productos.class);  
    cq.select(rt.get("nombre")).distinct(true);  
    ParameterExpression<String> p =  
        em.getCriteriaBuilder().parameter(String.class);  
    cq.where(em.getCriteriaBuilder().equal(rt.get("tipo"), p));  
    Query q = em.createQuery(cq);  
    q.setParameter(p, tipo);  
    return q.getResultList();  
}
```

Ejemplo:

/ Buscar facturas por fecha **/**

/ Sistema de facturación **/**

```
public List<Facturas> findCuentas(Date desde, Date hasta) {  
    EntityManager em = getEntityManager();  
    CriteriaBuilder cb = em.getCriteriaBuilder();  
    CriteriaQuery cq = em.getCriteriaBuilder().createQuery();  
    Root<Facturas> rt = cq.from(Facturas.class);  
    cq.select(rt);  
    cq.where(cb.between(rt.<Date> get("fecha"), desde, hasta));  
    Query q = em.createQuery(cq);  
    return q.getResultList();  
}
```

Referencias:

- <http://wiki.eclipse.org/EclipseLink/FAQ/JPA>
- <http://www.objectdb.com/java/jpa/query>
- <http://www.eclipse.org/eclipselink/documentation/>