

Rubén Agra Casal**// 1.1 Todos los empleados**

```
String hql = "FROM Empleado";
List<Empleado> empleados = em.createQuery(hql,
Empleado.class).getResultList();
System.out.println("----- Lista de empleados -----");
empleados.forEach(System.out::println);
```

// 1.2 Todos los departamentos

```
hql = "FROM Departamento";
List<Departamento> departamentos = em.createQuery(hql,
Departamento.class).getResultList();
System.out.println("----- Lista de departamentos
-----");
departamentos.forEach(System.out::println);
```

// 2.1 Nombres de sedes

```
hql = "SELECT s.nombre FROM Sede s";
List<String> nombresSedes = em.createQuery(hql,
String.class).getResultList();
System.out.println("----- Nombres de sedes -----");
nombresSedes.forEach(System.out::println);
```

// 2.2 Nombres de empleados

```
hql = "SELECT e.nombre FROM Empleado e";
List<String> nombresEmpleados = em.createQuery(hql,
String.class).getResultList();
System.out.println("----- Nombres de empleados
-----");
nombresEmpleados.forEach(System.out::println);
```

// 2.3 Nombres de proyectos

```
hql = "Select p.nombre FROM Proyecto p";
List<String> nombresProyectos = em.createQuery(hql,
String.class).getResultList(); System.out.println("-----
Nombres de proyectos -----");
nombresProyectos.forEach(System.out::println);
```

// 3.1 Empleados where dni

```
hql = "FROM Empleado e WHERE e.dni = :dni";
List<Empleado> empleadosConDNI = em.createQuery(hql,
Empleado.class).setParameter("dni", "12345678A").getResultList();
System.out.println("----- Nombres de Empleados WHERE DNI
-----");
empleadosConDNI.forEach(System.out::println);
```

// 3.2 Proyectos where fecha inicio

```
hql = "FROM Proyecto p WHERE p.fInicio > :fInicio";
Date fInicio = format.parse("2024-01-01");
List<Proyecto> proyectosFechaInicio = em.createQuery(hql,
Proyecto.class).setParameter("fInicio", fInicio).getResultList();
System.out.println("----- Nombres de Empleados WHERE Fecha
inicio -----");
proyectosFechaInicio.forEach(System.out::println);
```

// 3.3 Proyectos where fecha fin

```
hql = "FROM Proyecto p WHERE p.fFin < :fFin";
Date fFin = format.parse("2025-01-01");
List<Proyecto> proyectosFechaFin = em.createQuery(hql,
Proyecto.class).setParameter("fFin", fFin).getResultList();
System.out.println("----- Nombres de Empleados WHERE Fecha
fin -----");
proyectosFechaFin.forEach(System.out::println);
```

// 4.1 Alias para Sede

```
hql = "FROM Sede s";
List<Sede> sedes = em.createQuery(hql,
Sede.class).getResultList();
System.out.println("----- Nombres de Sedes -----");
sedes.forEach(System.out::println);
```

// 4.2 Alias para departamentos

```
hql = "FROM Departamento d";
List<Departamento> departamentosList = em.createQuery(hql,
Departamento.class).getResultList();
System.out.println("----- Nombres de Departamentos
-----");
departamentosList.forEach(System.out::println);
```

// 5.1 Ordenar sedes por nombre

```
hql = "FROM Sede s ORDER BY s.nombre";
List<Sede> sedesNombre = em.createQuery(hql,
Sede.class).getResultList();
System.out.println("----- Sedes ordenados por nombre
-----");
sedesNombre.forEach(System.out::println);
```

// 5.2 Ordenar departamentos por sede

```
hql = "FROM Departamento d ORDER BY d.sede";
List<Departamento> departamentosSede = em.createQuery(hql,
Departamento.class).getResultList();
System.out.println("----- Departamentos ordenados por Sedes
-----");
departamentosSede.forEach(System.out::println);
```

// 5.3 Ordenar empleados por categoría

```
hql = "FROM DatosProfEmpleado d ORDER BY d.categoria";
List<DatosProfEmpleado> datosProfEmpleadoCategoria =
em.createQuery(hql, DatosProfEmpleado.class).getResultList();
```

```
System.out.println("----- Empleados por categoría  
-----");  
datosProfEmpleadoCategoría.forEach(System.out::println);
```

// 5.4 Ordenar Proyectos por fecha de Inicio

```
hql = "FROM Proyecto p ORDER BY p.fInicio";  
List<Proyecto> proyectosfInicio = em.createQuery(hql,  
Proyecto.class).getResultList();  
System.out.println("----- Proyectos ordenados por fecha de  
inicio -----");  
proyectosfInicio.forEach(System.out::println);
```

// 5.5 Ordenar Proyectos por fecha de Inicio

```
hql = "FROM Proyecto p ORDER BY p.fFin";  
List<Proyecto> proyectosfFin = em.createQuery(hql,  
Proyecto.class).getResultList();  
System.out.println("----- Proyectos ordenados por fecha de  
fin -----");  
proyectosfFin.forEach(System.out::println);
```

// ----- BLOQUE II -----

// 6.1 Listar empleados que empieza por M

```
hql = "FROM Empleado e WHERE e.nombre LIKE :nombre";  
List<Empleado> empleadosNombre = em.createQuery(hql,  
Empleado.class).setParameter("nombre", "M%").getResultList();  
System.out.println("----- Empleados que empiecen por M  
-----");  
empleadosNombre.forEach(System.out::println);
```

// 6.2 Listar proyectos que empiecen por A

```
hql = "FROM Proyecto p WHERE p.nombre LIKE :nombre";  
List<Proyecto> proyectosNombre = em.createQuery(hql,  
Proyecto.class).setParameter("nombre", "A%").getResultList();  
System.out.println("----- Proyectos que empiecen por A  
-----");  
proyectosNombre.forEach(System.out::println);
```

// 7.1 Contar número total de empleados

```
hql = "SELECT COUNT(e) FROM Empleado e";  
Long empleadoCount = em.createQuery(hql,  
Long.class).getSingleResult();  
System.out.println("----- Total de empleados: " +  
empleadoCount + " -----");
```

// 7.2 Contar número total de proyectos

```
hql = "SELECT COUNT(p) FROM Proyecto p";  
Long proyectoCount = em.createQuery(hql,  
Long.class).getSingleResult();  
System.out.println("----- Total de proyectos: " +  
proyectoCount + " -----");
```

// 7.3 Contar número total de sedes

```
hql = "SELECT COUNT(s) FROM Sede s";
Long sedeCount = em.createQuery(hql,
Long.class).getSingleResult();
System.out.println("----- Total de sedes: " + sedeCount + "
-----");
```

// 8.1 Número total de empleados en un departamento

```
hql = "SELECT COUNT(e) FROM Empleado e WHERE e.departamento.nombre
= :nombreDepto GROUP BY e.departamento.nombre";
Long empleadoDepartamentoCount = em.createQuery(hql,
Long.class).setParameter("nombreDepto",
"Ventas").getSingleResult();
System.out.println("----- Total de empleados en
departamento (Ventas): " + empleadoDepartamentoCount + "
-----");
```

// 8.2 Numero total de proyectos en una sede

```
hql = "SELECT COUNT(ps) FROM ProyectoSede ps WHERE ps.sede.nombre
= :nombreSede GROUP BY ps.sede.nombre";
Long proyectoSedeCount = em.createQuery(hql,
Long.class).setParameter("nombreSede", "Sucursal
Norte").getSingleResult();
System.out.println("----- Total de proyectos en sede
(Sucursal Norte): " + proyectoSedeCount + " -----");
```

// 8.3 Número total de departamentos en una sede

```
hql = "SELECT COUNT(d) FROM Departamento d WHERE d.sede.nombre
= :nombreSede GROUP BY d.sede.nombre";
Long departamentoSedeCount = em.createQuery(hql,
Long.class).setParameter("nombreSede", "Sede
principal").getSingleResult();
System.out.println("----- Total de departamentos en sede
(Sede principal): " + departamentoSedeCount + " -----");
```

// 9.1 Departamentos con más de x número de empleados

```
hql = "SELECT e.departamento.nombre FROM Empleado e GROUP BY
e.departamento.nombre HAVING COUNT(e) > :numEmpleados";
List<String> departamentoMasEmpleados = em.createQuery(hql,
String.class).setParameter("numEmpleados", 2).getResultList();
System.out.println("----- Departamentos con más de 2
empleados -----");
departamentoMasEmpleados.forEach(System.out::println);
```

// 9.2 Proyectos con menos de x sedes

```
hql = "SELECT ps.proyecto.nombre FROM ProyectoSede ps GROUP BY
ps.proyecto.nombre HAVING COUNT(ps) < :numSedes";
List<String> proyectosMenosSedes = em.createQuery(hql,
String.class).setParameter("numSedes", 2).getResultList();
System.out.println("----- Proyectos con menos de 2 sedes
-----");
proyectosMenosSedes.forEach(System.out::println);
```

// 9.3 Departamentos con un número de x sedes

```
hql = "SELECT d.sede.nombre FROM Departamento d GROUP BY d.sede.id
HAVING COUNT(d.id) = :numDepartamentos";
List<String> departamentoIgualSedes = em.createQuery(hql,
String.class).setParameter("numDepartamentos", 2).getResultList();
System.out.println("----- Sedes con 2 departamentos
-----");
departamentoIgualSedes.forEach(System.out::println);
```

// 10.1 Fecha de inicio y fecha de fin de los proyectos

```
hql = "SELECT p.id, p.fInicio, p.fFin FROM Proyecto p";
List<Object[]> proyectosFechas = em.createQuery(hql,
Object[].class).getResultList();

System.out.println("----- Fechas de inicio y de fin de los
proyectos -----");
for (Object[] row : proyectosFechas) {
    int id = (int) row[0];
    Date fechaInicio = (Date) row[1];
    Date fechaFin = (Date) row[2];
    System.out.println("Id: " + id + " | Fecha Inicio: " +
    fechaInicio + " | Fecha Fin: " + fechaFin);
}
```

// 10.2 DNI y nombre de empleado

```
hql = "SELECT e.dni, e.nombre FROM Empleado e";
List<Object[]> empleadosNombres = em.createQuery(hql,
Object[].class).getResultList();

System.out.println("----- DNIs y Nombres de los empleados
-----");
for (Object[] row : empleadosNombres) {
    String dni = (String) row[0];
    String nombre = (String) row[1];
    System.out.println("DNI: " + dni + " | Nombre: " +
nombre);
}
```

// 10.3 Categoría y sueldos de los empleados

```
hql = "SELECT d.categoria, d.sueldoBrutoAnual FROM
DatosProfEmpleado d";
List<Object[]> categoriasSueldos = em.createQuery(hql,
Object[].class).getResultList();

System.out.println("----- Categorías y sueldos brutos
anuales -----");
for (Object[] row : categoriasSueldos) {
    String categoria = (String) row[0];
    float sueldoBruto = (Float) row[1];
    System.out.println("Categoría: " + categoria + " |
Sueldo bruto anual: " + sueldoBruto);
}
```

// ----- BLOQUE III -----

// 12.1 Lista de empleados junto a su departamento

```
hql = "SELECT e, d FROM Empleado e LEFT JOIN e.departamento d";
List<Object[]> empleadosDepartamentos = em.createQuery(hql,
Object[].class).getResultList();
```

```
System.out.println("----- Empleados y departamentos
-----");
```

```
for(Object[] row : empleadosDepartamentos) {
    Empleado emp = (Empleado) row[0];
    Departamento dpto = (Departamento) row[1];

    String nombreEmpleado = emp.getNombre();
    String nombreDepartamento = (dpto !=null) ?
dpto.getNombre() : "SIN DEPARTAMENTO";
```

```
        System.out.println("Empleado: " + nombreEmpleado +
" | Departamento: " + nombreDepartamento);
    }
```

// 13.1 Obtener empleados con un salario mayor al promedio

```
hql = "SELECT e FROM Empleado e LEFT JOIN e.datosProf dp WHERE
dp.sueldoBrutoAnual > (SELECT AVG(dp2.sueldoBrutoAnual) FROM
DatosProfEmpleado dp2)";
```

```
List<Empleado> empleadosMasSueldo = em.createQuery(hql,
Empleado.class).getResultList();
```

```
System.out.println("----- Empleados con salarios superiores al
promedio -----");
```

```
for (Empleado emp : empleadosMasSueldo) {
    System.out.println(emp.getNombre() + "| Sueldo: " +
emp.getDatosProf().getSueldoBrutoAnual());
}
```

// 14.1 Obtener empleados con un salario mayor al ingresado

```
float salario = 20000;
```

```
hql = "SELECT e FROM Empleado e LEFT JOIN e.datosProf dp WHERE
dp.sueldoBrutoAnual > :sueldo";
List<Empleado> empleadoMasSueldoIngresado = em.createQuery(hql,
Empleado.class).setParameter("sueldo", salario).getResultList();
```

```
System.out.println("----- Empleados con salarios superiores a " +
salario + "-----");
```

```
for(Empleado emp : empleadoMasSueldoIngresado) {
    System.out.println("DNI: " + emp.getDni() + " | Nombre: " +
emp.getNombre() + " | Sueldo: " +
emp.getDatosProf().getSueldoBrutoAnual());
}
```

// 15.1 Aumentar salario un 10% a empleados de un determinado departamento

```
hql = "UPDATE DatosProfEmpleado dp " +  
"SET dp.sueldoBrutoAnual = dp.sueldoBrutoAnual * 1.1 " +  
"WHERE dp.empleado.departamento.nombre = :nombreDepto";  
  
int actualizados = em.createQuery(hql)  
    .setParameter("nombreDepto", "Recursos Humanos")  
    .executeUpdate();  
  
System.out.println("----- Empleados actualizados: " + actualizados  
+ "-----");
```

// 16.1 Eliminar empleados cuyo salario sea menor que 20000€

```
hql = "DELETE FROM DatosProfEmpleado dp WHERE dp.sueldoBrutoAnual  
< :sueldo";  
  
actualizados = em.createQuery(hql).setParameter("sueldo",  
22000).executeUpdate();  
  
System.out.println("----- Empleados eliminados: " + actualizados +  
"-----");
```

// 16.2 Eliminar proyectos cuya fecha sea inferior a una dada

```
hql = "DELETE FROM ProyectoSede p WHERE p.fFin < :fecha";  
actualizados = em.createQuery(hql).setParameter("fecha",  
java.sql.Date.valueOf("2024-01-01")).executeUpdate();  
  
System.out.println("----- Proyectos eliminados: " + actualizados +  
"-----");
```