

Consultas con JPQL

Sencillas

1. Todos los clientes.

```
select c
from Client c
```

2. Todos los clientes de Oviedo.

```
select c
from Client c
where c.address.city = 'Oviedo'
```

3. Todos los vehículos de marca “NISSAN”.

```
select v
from Vehicle v
where lower(v.make) like '%nissan%'
```

4. Todas las órdenes de trabajo terminadas (no facturadas). [Nota: las comparaciones con enumerados requieren el nombre de paquete; p.e uo.ri.model.InvoiceStatus.NOT_YET_PAID]

```
select a
from WorkOrder a
where a.status = uo.ri.model.types.WorkOrderStatus.FINISHED
```

5. Todas las órdenes de trabajo reparadas entre dos fechas. [Nota: constantes fecha con formato '2012-10-01']

```
select a
from WorkOrder a
where a.date >= '2012-11-01'
and a.date < '2012-12-01'
```

Navegación (join implícito)

6. Las órdenes de trabajo asignadas a un mecánico determinado.

```
select a
from WorkOrder a
where a.mechanic.dni = ?1
```

7. Todas las órdenes de trabajo de los vehículos de un cliente determinado por su DNI.

```
select a
from WorkOrder a
where a.vehicle.client.dni = ?1
```

8. Todos los vehículos de un cliente indicado por su DNI.

```
select v
from Vehicle v
where v.client.dni = ?1
```

9. Los clientes con saldo en sus Bonos.

```
select b.client
from Voucher b
where b.available > 0
```

10. Todos los vehículos en los que se ha cambiado un determinado repuesto indicado por su nombre.

```
select s.intervention.workOrder.vehicule
from Substitution s
where lower(s.sparePart.name) like '%?1%'
```

Navegación (join explícito)

11. Datos de los vehículos involucrados en una factura determinada por su número.

```
select a.vehicule
from Invoice f
      join f.workOrders a
where f.number = ?1
```

12. Los medios de pago empleados para pagar una determinada orden de trabajo, indicada por su ID.

```
select c.paymentMean
from Charge c
      join c.invoice f
      join f.workOrders a
where a.id = ?1
```

13. Todas las órdenes de trabajos que han sido pagadas por una tarjeta de crédito especificada por su número.

```
select a
from CreditCard tc
      join tc.charges c
      join c.invoice f
      join f.workOrders a
where tc.number = ?1
```

14. Se desea comprobar si hay inconsistencias en los datos: obtén las órdenes de trabajo que han sido abonadas por clientes que no son los dueños del vehículo.

```
select a
from WorkOrder a
      join a.invoice f
      join f.charges c
      join c.paymentMeans mp
where a.vehicule.client <> mp.client
```

Agregados

15. El tiempo medio de intervención.

```
select avg(i.minutes)
from Intervention i
```

16. El tiempo medio (en días) que tardan los clientes en recoger sus vehículos.
[Nota: la resta de dos fechas devuelve días].

```
select avg(a.invoice.date - a.date)
from WorkOrder a
```

17. La orden de trabajo más cara.

```
select a
from WorkOrder a
where a.amount = (
  select max(aa.amount)
  from WorkOrder aa
)
```

18. El último número de factura generado.

```
select max(f.number)
from Invoice f
```

19. La cantidad de órdenes de trabajo en las que trabajó un mecánico entre dos fechas.

```
select count(distinct i.workOrder)
from Intervention i
where i.mechanic.dni = ?1
      and i.workOrder.date >= ?2
      and i.workOrder.date < ?3
```

Con agrupación

20. La cantidad de dinero gastada en nuestro taller por cada cliente de más a menos.

```
select sum(mp.acummulated)
from PaymentMean mp
group by mp.client.dni
order by sum(mp.acummulated) desc
```

21. ¿Cuántas órdenes de trabajo tiene asignadas cada mecánico? Proyecta el id del mecánico y la cantidad. [Nota: los mecánicos tienen asignadas las órdenes de trabajo del día; una vez terminada ya no tiene mecánico asignado]

```
select a.mechanic.id, count(a)
from WorkOrder a
group by a.mechanic.id
having a.mechanic.id is not null
```

22. Listado de la cantidad pagada en nuestro taller con cada tipo de medio de pago ordenado de más a menos. [Nota: La función type(o) devuelve el tipo del objeto o.]

```
select sum(mp.acummulated), type(mp)
from PaymentMean mp
group by type(mp)
order by sum(mp.acummulated) desc
```

23. Listado de gasto por repuesto de más a menos.

```
select s.sparePart.id, sum(s.quantity * s.sparePart.price)
from Substitution s
group by s.sparePart
order by sum(s.quantity * s.sparePart.price) desc
```

24. Cantidad de tiempo trabajado en órdenes de trabajo de cada mecánico entre fechas

```
select i.mechanic.id, sum(i.minutes)
from Intervention i
where i.workOrder.date >= ?1
      and i.workOrder.date < ?2
group by i.mechanic
```

25. Todas las órdenes de trabajo con tiempo total de intervención mayor de 4 horas (una orden de trabajo puede tener varias intervenciones).

```
select a
from WorkOrder a
where a.id in (
  select i.workOrder.id
  from Intervention i
  group by i.workOrder.id)
```

```

        having sum(i.minutes) > 240
    )

```

26. El cliente que más ha gastado en nuestro taller.

```

select distinct mp.client
from PaymentMean mp
where (
    select sum(mp2.acummulated)
    from PaymentMean mp2
    where mp2.client = mp.client
) >= ALL (
    select sum(mp3.acummulated)
    from PaymentMean mp3
    group by mp3.client
)

```

27. El repuesto más usado (del que se ha empleado más cantidad).

```

select distinct s.sparePart
from Substitution s
where (
    select sum(ss.quantity)
    from Substitution ss
    where ss.sparePart = s.sparePart
) >= ALL (
    select sum(sss.quantity)
    from Substitution sss
    group by sss.sparePart
)

```

28. El gasto medio por cliente en el taller.

```

select sum(mp.acummulated) / count(distinct c)
from Client c
left join c.paymentMeans mp

```

29. El número medio de sustituciones por orden de trabajo.

```

select (count(s) + 0.0) / count(distinct a)
from WorkOrder a
left join a.interventions i
left join i.substitutions s

```

"count(s) + 0.0" es una forma rápida de hacer cast a double. Sin ello, el resultado es la división entera (trunca decimales).

30. El número medio de repuestos empleados por orden de trabajo.

```

select sum(s.quantity + 0.0) / count(distinct a)
from WorkOrder a
left join a.interventions i
left join i.sustitutions s

```

s.quantity es un entero, s.quantity + 0.0 es una forma sencilla de hacer un cast a double para que la división devuelva decimales. En caso contrario devolvería la parte entera.