Lo mismo pero con orientación a objetos…

```
Cliente c = new Cliente("1234567", "Manuel", ...);
Vehiculo v = new Vehiculo("1241-CKJ", "seat", "almera")
Averia averia = new Averia("Pre-revision ITV");

c.addVehiculo( v );
v.addAveria( averia );
Mecanico luis = findMecanico("Luis", "Pérez");
averia.asignarMecanico( luis );

Intervencion i = new Intervencion(averia, luis);
Repuesto r = findRepuesto("DKJ.1244");
new Sustitucion( i, r, 2 );

Factura f = new Factura( averia );

Tarjeta visa = new Tarjeta("visa", "1214-2154-2221", ...);
Metalico metalico = new Metalico();
c.addMedioPago( visa );
c.addMedioPago( metalico );

double importe = f.getImporte();
new Cargo(metalico, 100.0);
new Cargo(visa, importe - 100);
```

```
Cliente c = new Cliente("1234567", "Manuel", ...);
Vehiculo v = new Vehiculo("1241-CKJ", "seat", "almera")
Averia averia = new Averia("Pre-revision ITV");

c.addVehiculo( v );
v.addAveria( averia );
```

Llega un cliente nuevo

```
Mecanico luis = findMecanico("Luis", "Pérez");
averia.asignarMecanico( luis );
```

Se le asigna un mecánico

```
Intervencion i = new Intervencion(averia, luis);
Repuesto r = findRepuesto("DKJ.1244");
new Sustitucion( i, r, 2 );
```

El mecánico hace su trabajo

```
Factura f = new Factura( averia );
```

Cuando termina se crea la factura

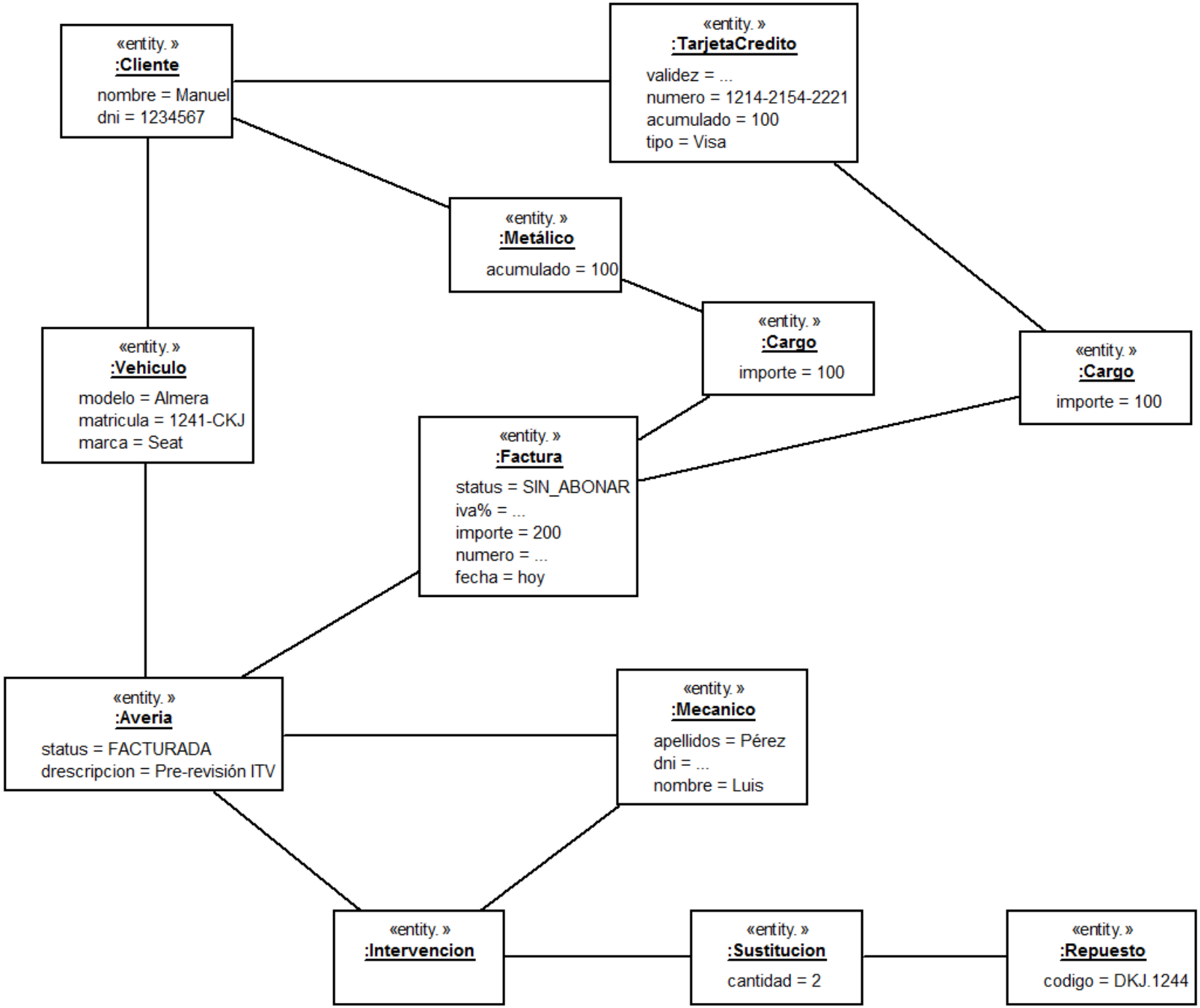
```
Tarjeta visa = new Tarjeta("visa", "1214-2154-2221", ...);
Metalico metalico = new Metalico();
c.addMedioPago( visa );
c.addMedioPago( metalico );
```

Cuando el cliente viene a recoger el coche…

… se registran sus medios de pago…

```
double importe = f.getImporte();
new Cargo(metalico, 100.0);
new Cargo(visa, importe - 100);
```

… y hace el pago


«entity.»
:Cliente
nombre = Manuel
dni = 1234567
«entity.»
:TarjetaCredito
validez = ...
numero = 1214-2154-2221
acumulado = 100
tipo = Visa
«entity.»
:Metálico
acumulado = 100
«entity.»
:Cargo
importe = 100
«entity.»
:Vehiculo
modelo = Almera
matricula = 1241-CKJ
marca = Seat
«entity.»
:Cargo
importe = 100
«entity.»
:Factura
status = SIN_ABONAR
iva% = ...
importe = 200
numero = ...
fecha = hoy
«entity.»
:Averia
status = FACTURADA
drescripcion = Pre-revisión ITV
«entity.»
:Mecanico
apellidos = Pérez
dni = ...
nombre = Luis
«entity.»
:Intervencion
«entity.»
:Sustitucion
cantidad = 2
«entity.»
:Repuesto
codigo = DKJ.1244

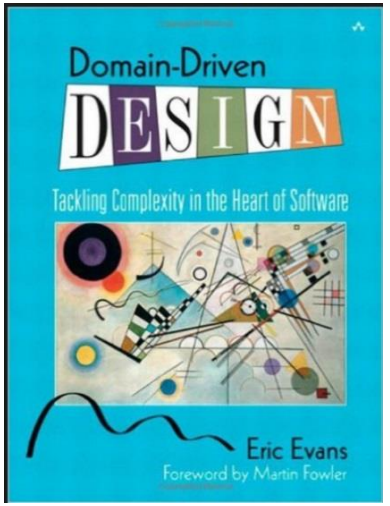
- ¿Es sencillo entender este código?
- ¿Qué hay en memoria?
- ¿Qué ocurrirá si el taller sigue funcionando, los clientes viviendo, los mecánicos reparando, etc…?
- ¿Y si se apaga el ordenador?

- La orientación a objetos y la persistencia no se llevan bien.
  - La persistencia es muy intrusiva…
  - … y me desbarata el diseño limpio del código que me daría la orientación a objetos

Patrón "modelo de dominio"

- Sin embargo quiero esa forma de programar
  - Es limpia
  - Más fácil de entender
  - Más fácil de mantener → más barato
  - Es reflejo directo del modelo del dominio

- Patrón "modelo de dominio"
  - Y recomendaciones de DDD



```
Cliente c = new Cliente("1234567", "Manuel", ...);
Vehiculo v = new Vehiculo("1241-CKJ", "seat", "almera")
Averia averia = new Averia("Pre-revision ITV");

c.addVehiculo( v );
v.addAveria( averia );
Mecanico luis = findMecanico("Luis", "Pérez");
averia.asignarMecanico( luis );

Intervencion i = new Intervencion(averia, luis);
Repuesto r = findRepuesto("DKJ.1244");
new Sustitucion( i, r, 2 );

Factura f = new Factura( averia );

Tarjeta visa = new Tarjeta("visa", "1214-2154-2221", ...);
Metalico metalico = new Metalico();
c.addMedioPago( visa );
c.addMedioPago( metalico );

double importe = f.getImporte();
new Cargo(metalico, 100.0);
new Cargo(visa, importe - 100);
```