



## Práctica 7

### Objetivos

- Mapear con anotaciones JPA el modelo de dominio desarrollado en la práctica anterior
- Mapear en *Green Field* y en *Legacy*
- Ajustar la implementación para poder ser mapeado según la especificación JPA
- Verificar la funcionalidad y mapeo con test unitarios

### Ejercicios

#### ***Mapear el modelo de dominio generando la BDD***

Instrucciones:

- Copia tus clases de modelo del dominio desarrolladas en la práctica anterior al paquete `uo.ri.cws.domain`.
- Levanta la base de datos HSQLDB adjunta. Está vacía no tiene tablas.
- En el fichero `persistence.xml` mantén descomentada la línea:  

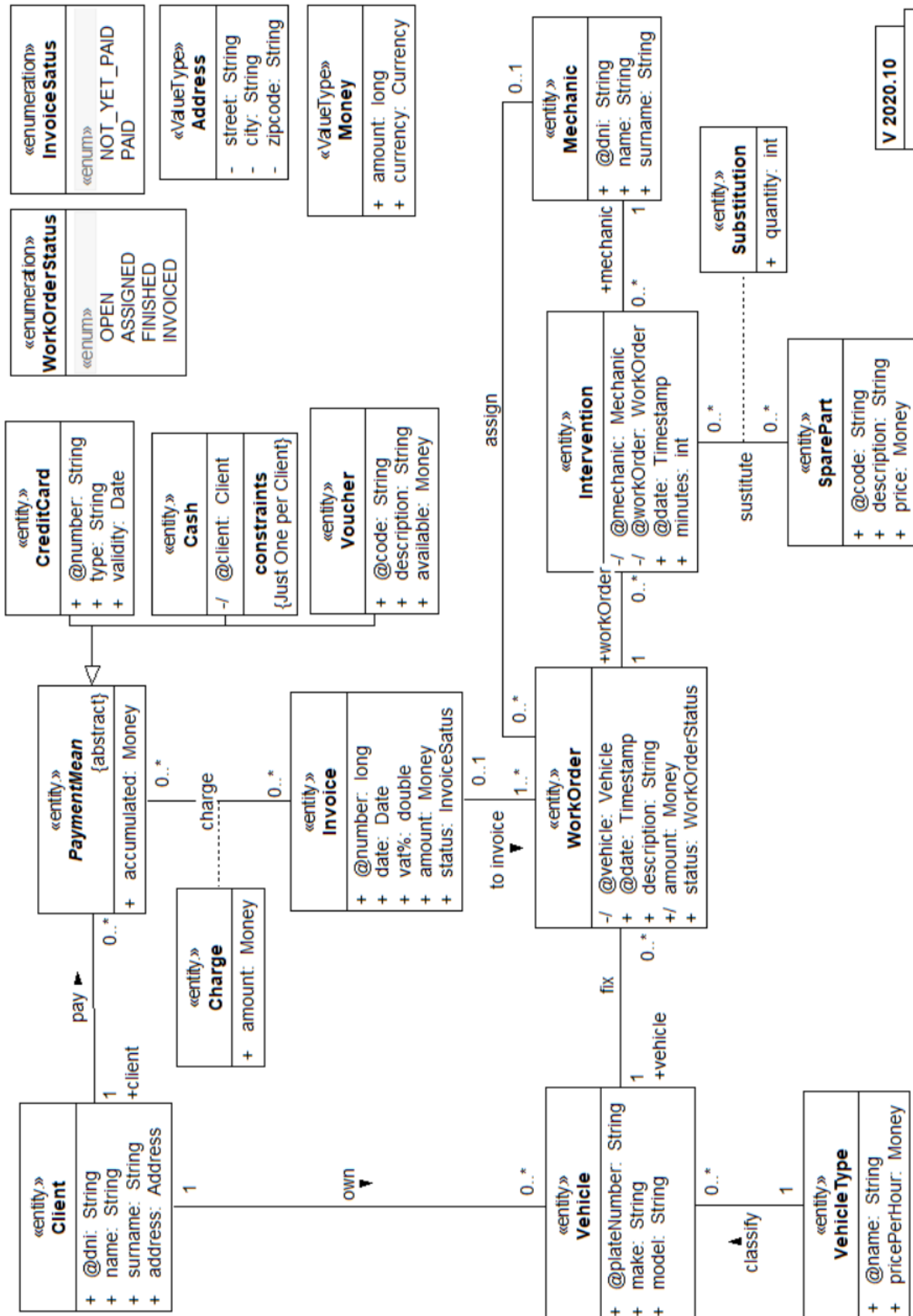
```
<property name="eclipselink.ddl-generation" value="drop-and-create-tables"/>
```
- Anota todos los atributos que son extremo de asociación con `@Transient`.
- Añade la clase *BaseEntity* como base de todas las entidades del dominio
  - \* Usaremos identidad artificial basada en UUID (String)
  - \* Añade un Long para campo de versión
  - \* Anótala con `@MappedSuperclass`, `@Id` y `@Version`.
- Mapea las entidades, una a una, añadiendo `@Entity` según proceda.
  - \* Tras cada clase anotada ejecuta la clase `JustLoadEntityManagerFactory.java` suministrada. Esto fuerza al mapeador a cargar la unidad de persistencia y a regenerar las tablas.
  - \* Observa las tablas generadas con el administrador de la base de datos.
- Mapea las asociaciones una a una y observa los cambios en las tablas añadiendo anotaciones en ambos extremos.
- Mapea la herencia de las clases *PaymentMean*. Prueba con las distintas estrategias y observa las tablas que se generan.
- Añade anotaciones a las entidades de forma la base de datos añada un índice único sobre las identidades naturales de las entidades. Usa las anotaciones: `@Column(unique=true)` y/o `@Table(uniqueConstraints...)` según necesites.
- Ejecuta los test suministrados como verificación de que el mapeo es correcto.



### ***Mapear el modelo de dominio a un esquema de BDD existente***

- Con la base de datos parada, ejecuta el script `setLegacy.bat`. Esto carga el fichero que contiene la base de datos CarWorkshop con datos y nombres de tablas y campos **diferentes** de los que nos ha generado el mapeador.
- Comenta en el fichero `persistence.xml` la línea  

```
<property name="eclipselink.ddl-generation" value="drop-and-create-tables"/>
```
- Hay estas diferencias entre el modelo de objetos y el de tablas:
  - \* Las tablas siguen el patrón de nombres *T<Entidad>s*, p.e. *TClientes*, *TVehiculos*, etc.
  - \* Algunos campos fecha contienen un *timestamp* y otros sólo la fecha.
  - \* La jerarquía de medios de pago tiene una tabla para cada clase.
  - \* Los campos de tipo enumerado están en formato VARCHAR en las tablas (String y no Ordinal).
  - \* El atributo *Vehicle.make* se mapea a una columna llamada "BRAND".
- Realiza los ajustes de mapeo necesarios para que el mapeador sea capaz de adaptarse al nuevo esquema de tablas y los tests vuelvan a pasar correctamente.



V 2020.10