

Persistencia	Fecha edición: 26/06/01 14:16:34	
Rev. 1.1		
Modelo de persistencia		
Diagrama de Persistencia	Pág. 1 de 6	jvilalta@vico.org
Fichero: ABC persistencia.vsd		www.vico.org

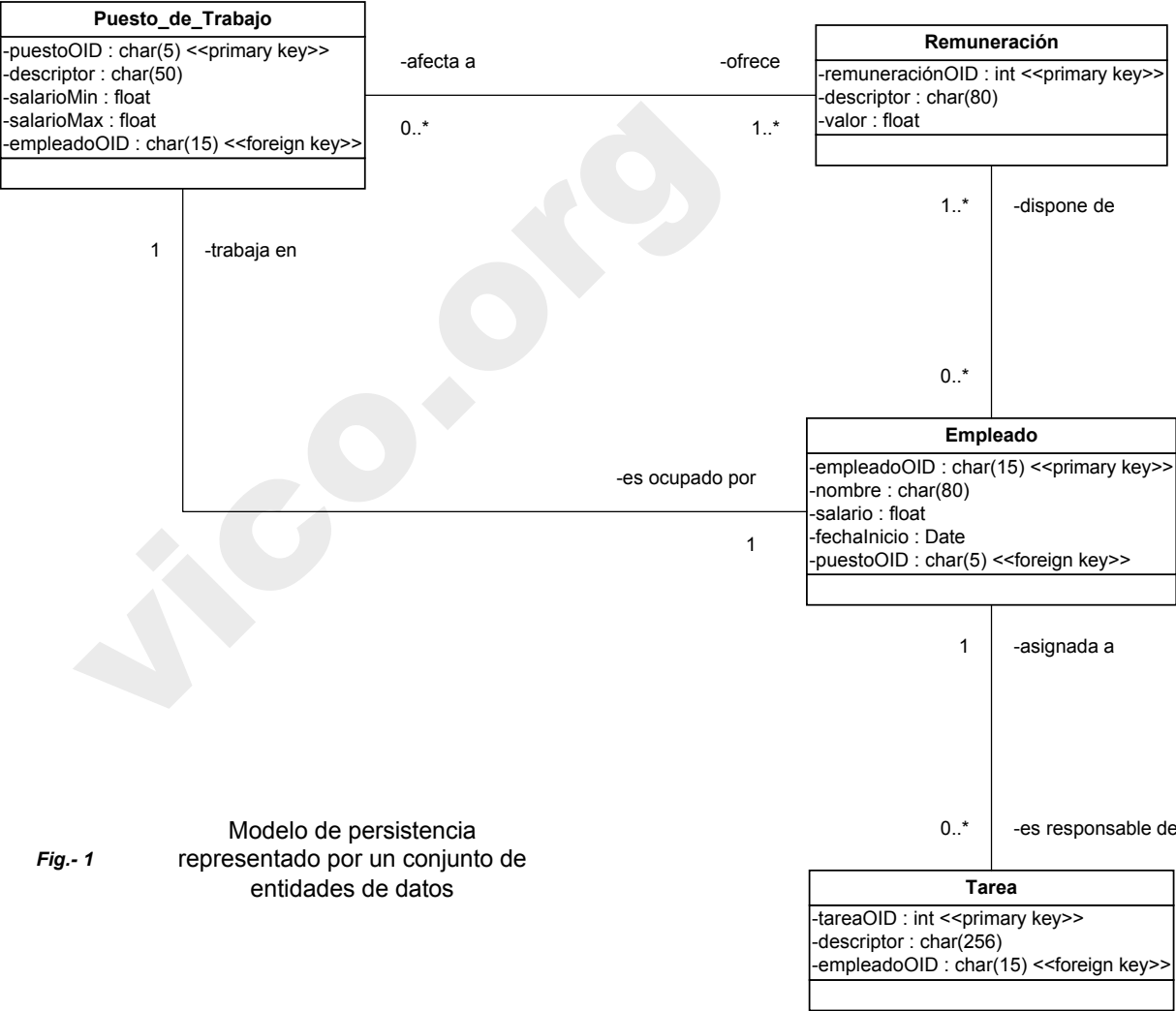


Fig.- 1 Modelo de persistencia
representado por un conjunto de
entidades de datos

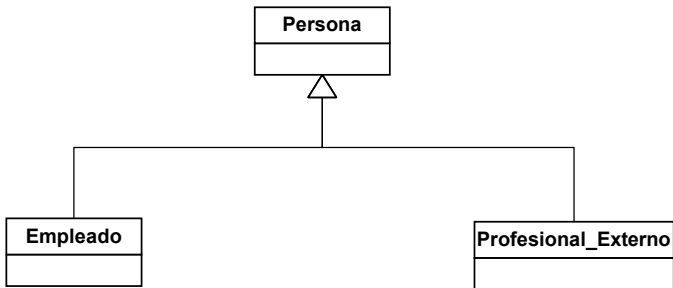


Fig.- 2 Jerarquía simple de una clase

Persona
-personaOID : char(15) <<primary key>>
-objectType : char(15)
-nombre : char(80)
-salario : float
-fechaInicio : Date
-precioHora : float
-contratoTipo : char(15)
-consultorTipo : char(15)
-responsabilidad : char(15)

Fig.- 3 Una entidad mapea una jerarquía completa de clases

Empleado
-empleadoOID : char(15) <<primary key>>
-nombre : char(80)
-salario : float
-fechaInicio : Date
-puestoOID : char(5) <<foreign key>>

Profesional_Externo
-profExternoOID : char(15) <<primary key>>
-nombre : char(80)
-precioHora : float
-fechaInicio : Date
-contratoTipo : char(15)

Fig.- 4 Una entidad mapea cada clase concreta de la jerarquía de clases

Persistencia		Fecha edición: 26/06/01 14:16:34	
Rev. 1.1 Modelo de persistencia			
Diagrama de Persistencia		Pág.3 de 6	jvilalta@vico.org
Fichero: ABC persistencia.vsd			www.vico.org

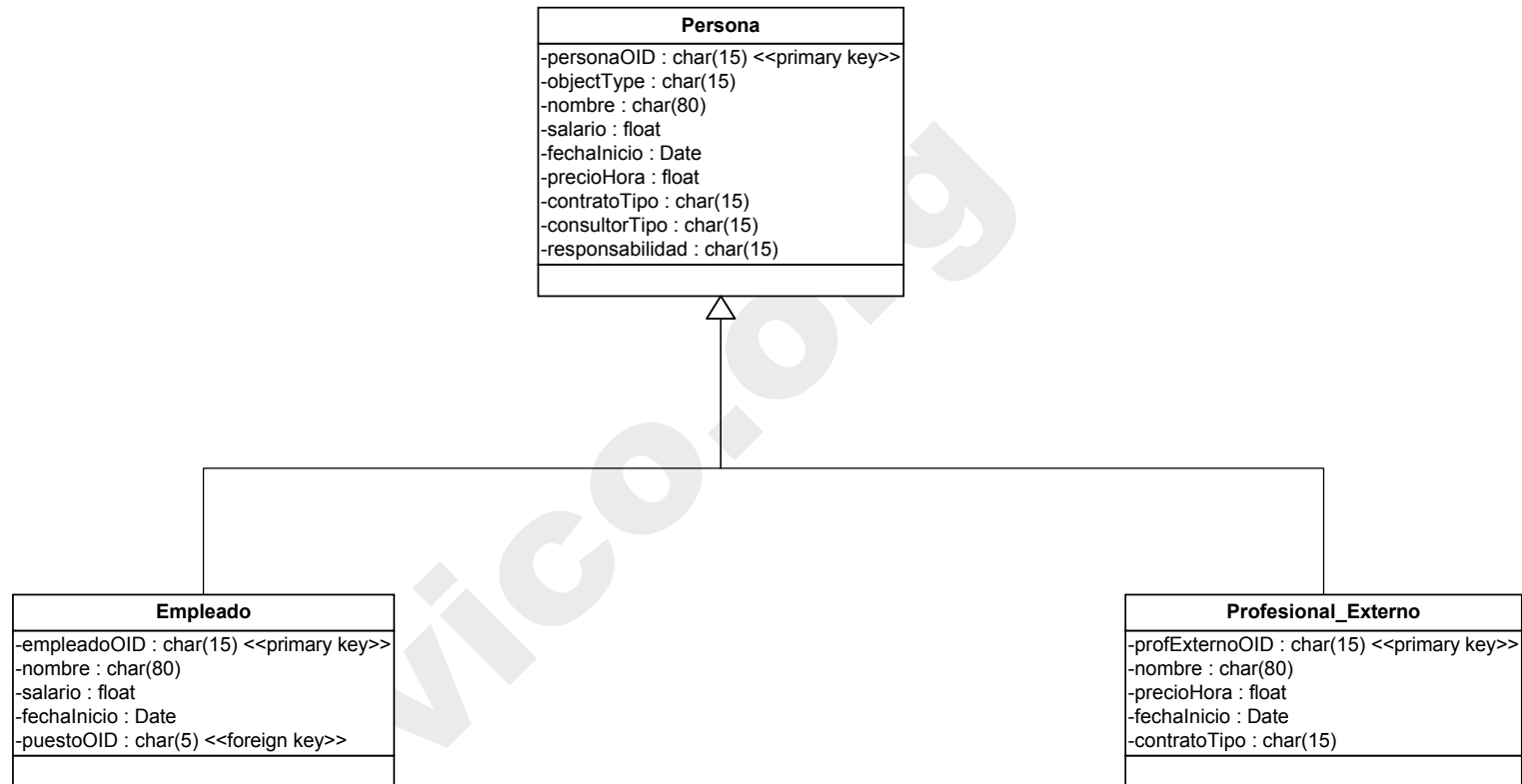


Fig.- 5 Una entidad mapea cada una de las clases de la jerarquía

Revisar qué atributos son específicos de cada entidad

Persistencia	Fecha edición: 26/06/01 14:16:34		
Rev. 1.1 Modelo de persistencia			
Diagrama de Persistencia	Pág.4 de 6	jvilalta@vico.org	
Fichero: ABC persistencia.vsd		www.vico.org	

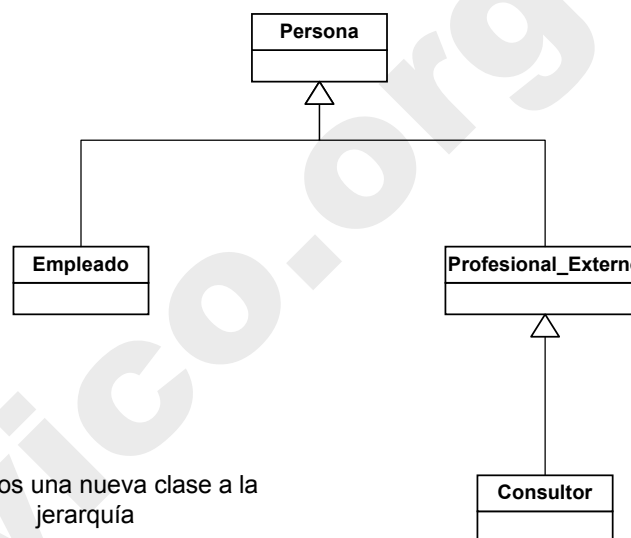


Fig.- 6

Añadimos una nueva clase a la jerarquía

Persistencia	Fecha edición: 26/06/01 14:16:34		
Rev. 1.1 Modelo de persistencia			
Diagrama de Persistencia	Pág.5 de 6	jvilalta@vico.org	
Fichero: ABC persistencia.vsd		www.vico.org	

Persona
-personaOID : char(15) <<primary key>>
-objectType : char(15)
-nombre : char(80)
-salario : float
-fechaInicio : Date
-precioHora : float
-contratoTipo : char(15)
-consultorTipo : char(15)
-responsabilidad : char(15)

Fig.- 7 Toda la jerarquía de clases se ha mapeado en sola entidad

Empleado
-empleadoOID : char(15) <<primary key>>
-nombre : char(80)
-salario : float
-fechaInicio : Date
-puestoOID : char(5) <<foreign key>>

Profesional_Externo
-profExternoOID : char(15) <<primary key>>
-nombre : char(80)
-precioHora : float
-fechaInicio : Date
-contratoTipo : char(15)

Fig.- 8 Cada clase concreta de la jerarquía es mapeada contra una entidad

Consultor
-consultorOID : char(15)
-nombre : char(80)
-precioHora : float
-fechaInicio : Date
-contratoTipo : char(15)
-responsabilidad : char(15)

Persistencia	Fecha edición: 26/06/01 14:16:29	
Rev. 1.1		
Modelo de persistencia		
Diagrama de Persistencia	Pág. 1 de 2	jvilalta@vico.org
Fichero: ABC persistencia_b.vsd		www.vico.org

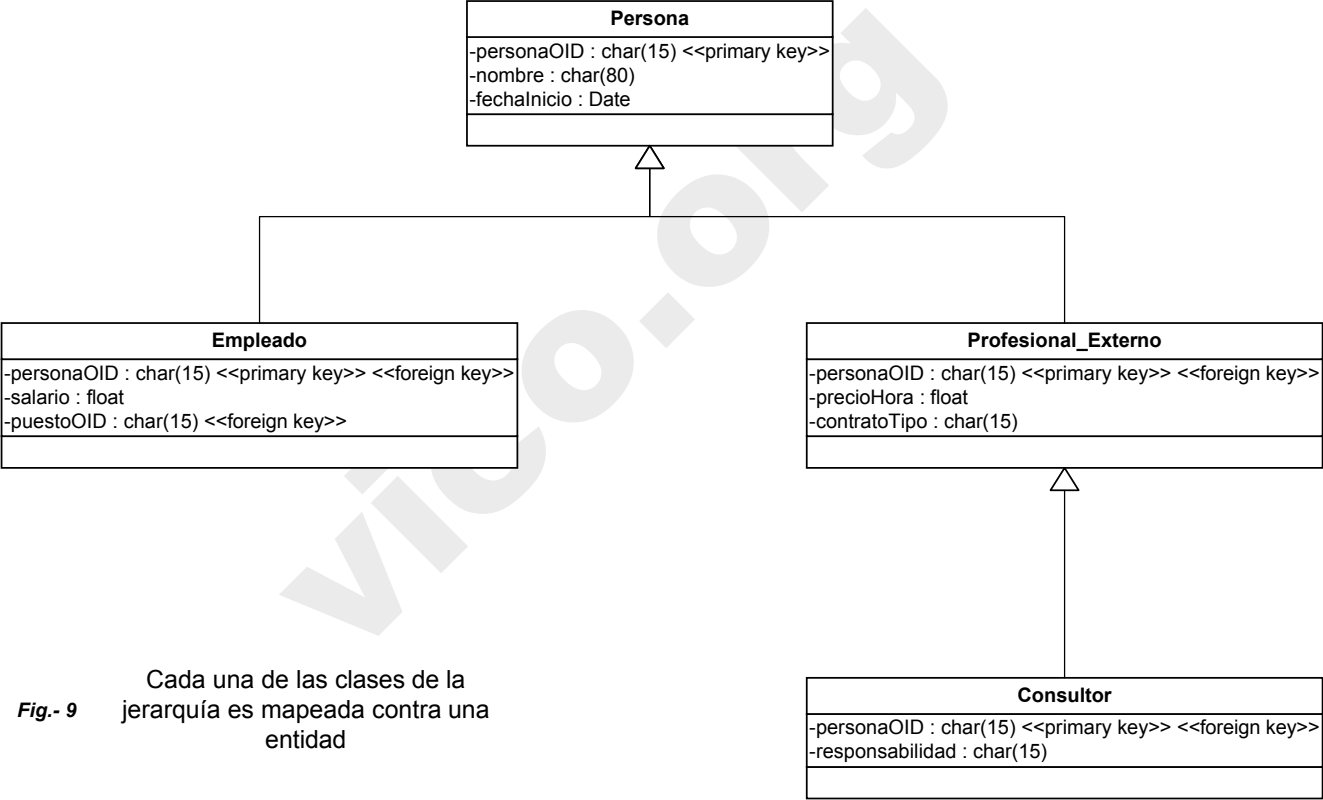


Fig.- 9 Cada una de las clases de la jerarquía es mapeada contra una entidad

Persistencia		Fecha edición: 26/06/01 14:16:29
Rev. 1.1 Modelo de persistencia		
Diagrama de Persistencia	Pág.2 de 2	jvilalta@vico.org
Fichero: ABC persistencia_b.vsd		www.vico.org

Comparación de las distintas estrategias de mapping

Para poder analizar los pros y contras de las tres opciones consideremos la jerarquía de clases de la fig. 6 a la que hemos añadido una nueva clase **Consultor** que hereda de **Profesional_Externo**.

Opción 1.-

La fig 7 presenta la actualización del modelo de persistencia en la que toda la jerarquía de clases se ha mapeado en una sola entidad.

La actualización es rápida y sencilla, si bien incrementamos el espacio ocupado por la base de datos.

Opción 2.-

La fig 8 presenta el modelo de persistencia actualizado en donde cada clase concreta es mapeada a una entidad de datos. Con esta estrategia solo necesitamos añadir una tabla más, aunque la manera de tratar los objetos que pueden cambiar su asociación es ahora más compleja.

Opción 3.-

La fig 9 presenta la solución de la tercera estrategia, mapear cada una de las clases de la jerarquía a una entidad de datos. Necesitamos añadir una nueva tabla que incluya los atributos de la clase **Consultor**. El problema de esta estrategia es que necesitamos diferentes accesos a la base de datos para trabajar con las instancias de la nueva clase.

Factores a considerar	Una tabla por jerarquía	Una tabla por Clase concreta	Una tabla por Clase
Generación de reports	Sencillo	Medio	Medio/Difícil
Facilidad implementación	Simple	Medio	Difícil
Facilidad acceso de datos	Sencillo	Sencillo	Medio/Sencillo
Acoplamiento	Muy alto	Alto	Bajo
Rapidez acceso de datos	Rápido	Rápido	Medio/Rápido
Soporta polimorfismo	Medio	Bajo	Alto

No existe una estrategia perfecta, tenemos que conciliar nuestra opción con los distintos requerimientos no funcionales que nos condiciona. Número de registros a procesar, capacidad del motor de la base de datos, complejidad de la estructura de clases, etc.

Persistencia	Fecha edición: 26/06/01 14:16:34		
Rev. 1.1			
Modelo de persistencia			
Diagrama de Persistencia	Pág.6 de 6	jvilalta@vico.org	
Fichero: ABC persistencia.vsd		www.vico.org	

Fig.- 10 Asociación Muchos-a-Muchos

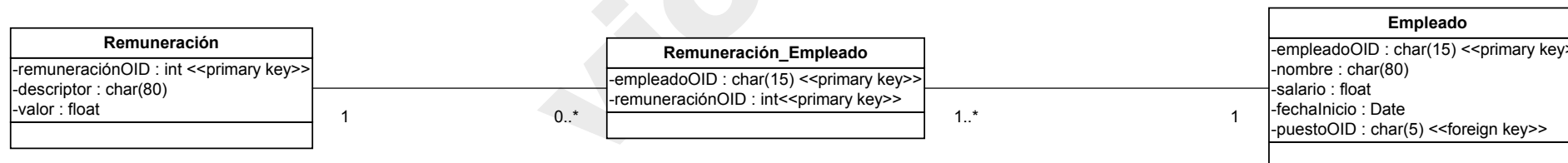
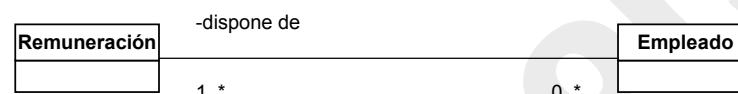


Fig.- 11 Implementación de una asociación Muchos-a-Muchos con una tabla asociativa