

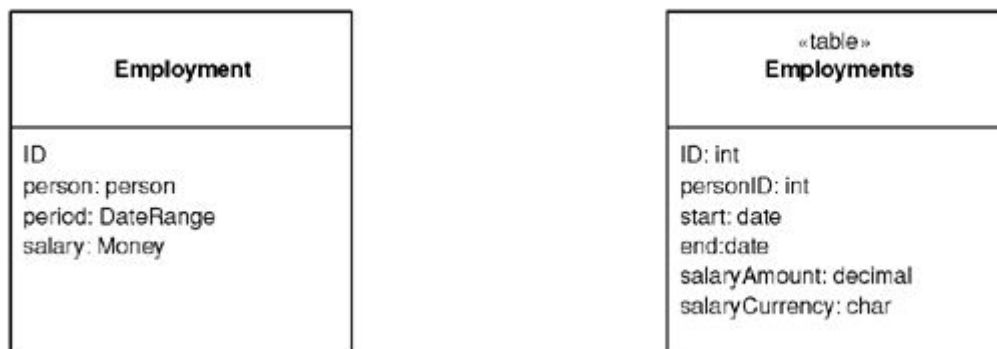
Mapeo objeto-relacional: Embedded Value

¿Porque Aparece este patrón?

Muchos objetos pequeños tienen sentido en un sistema orientado a objetos que no tienen sentido como tablas en una base de datos. Un ejemplo de esto pueden ser objetos para representar una moneda en específico u objetos para representar rango entre fechas.

¿Cuál es su solución?

Embedded Value asigna los valores de un objeto a distintos campos de un determinado registro. Un solo concepto del dominio se expresa como dos o más clases, con el objeto de lograr un modelado más expresivo. Sin embargo, a la hora de guardar estos objetos en la base de datos, es posible aplanarlos y guardar varios objetos de manera conjunta en una sola tabla.



¿Cómo funciona?

Cuando el objeto principal se quiere guardar, los objetos dependientes también se guardan en el mismo momento. Las clases dependientes no tienen sus propios métodos de persistencias ya que toda la persistencia la realiza el objeto principal. En el caso del ejemplo tanto la clase **DateRange** como la clase **Money** no tienen ningún método para guardar los datos, sino que cualquier método de persistencia es manejado por la clase **employment**.

¿Cuándo utilizarlo?

El principal caso para la utilización de Embedded Value es cuando se esta en presencia de objetos de valor, los objetos de valor son objetos simples y pequeños, los cuales por lo general son pasados por su valor en vez de por referencia, es decir, no están interesados en tener un valor único que los identifique, esto se debe a que al ser objetos relativamente pequeños y fácil de crear no importa si existen dos objetos con los mismos valores.

Se puede utilizar Embedded Value para objetos con referencia cuando existe una asociación uno a uno entre ellos, un ejemplo puede ser una orden de compra y su orden de envío correspondiente.

Ventajas

La gran ventaja de Embedded Value es que permite realizar consultas SQL del objeto principal obteniendo los valores del objeto dependiente.

Tiene mejoras en el rendimiento a la hora de realizar consultas, ya que existe una menor cantidad de relaciones entre tablas.

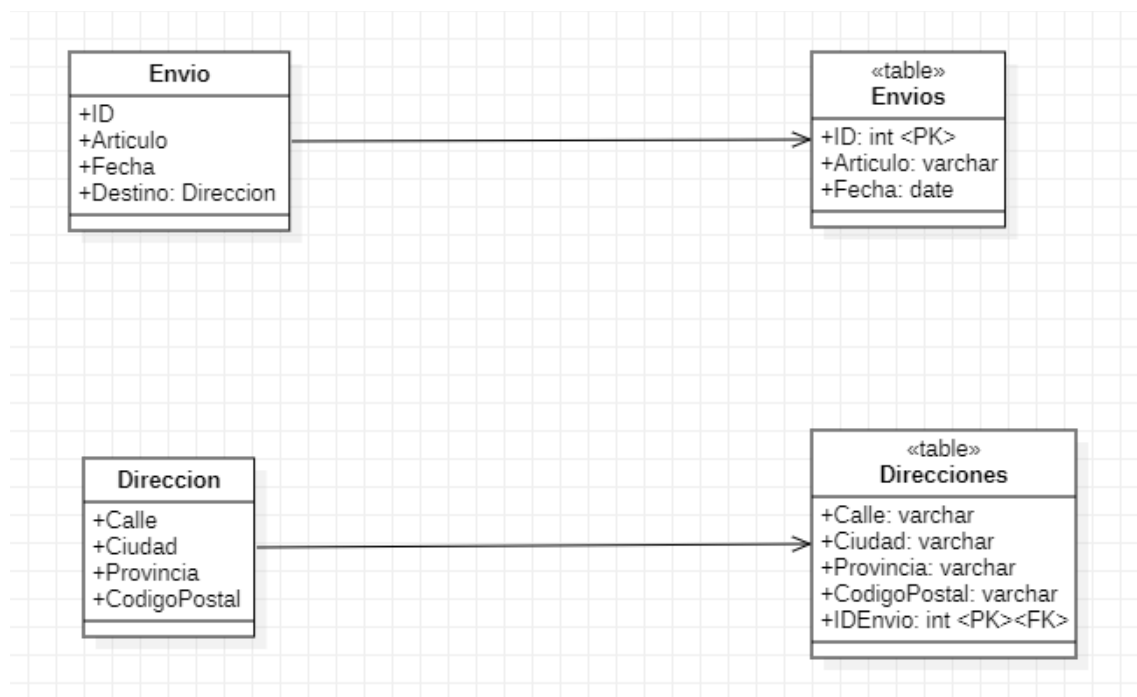
Desventajas

Si se utiliza Embedded Value en ocasiones erróneas, como por ejemplo en objetos con referencia con asociación uno a muchos o muchos a muchos, puede generar desorden a la hora de realizar consultas en la base de datos.

Ejemplo

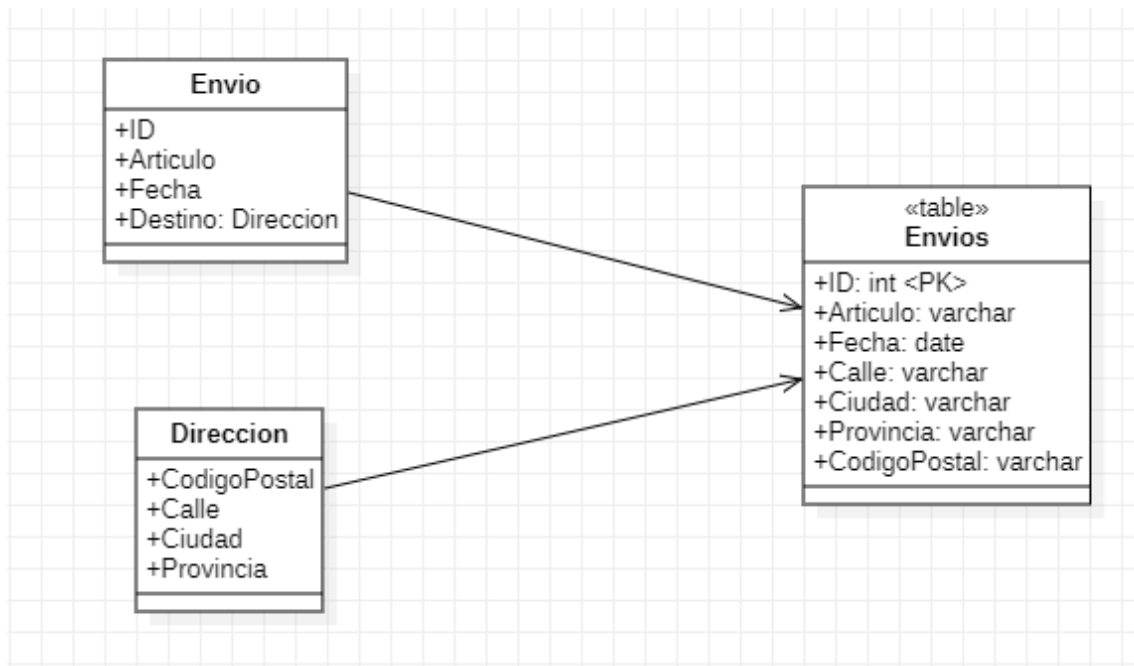
En este ejemplo vamos a ver la compra de un artículo el cual tiene una orden de envío con sus respectivos detalles. Se cuentan con dos clases persistentes envío y dirección, las cuales tienen una relación uno a uno.

Esquema sin el patrón aplicado



Si quisiéramos obtener toda la información del envío debemos realizar una sentencia SQL utilizando join, la manipulación del join va a influir en el rendimiento de la aplicación, por lo que se podría unir ambos objetos en una sola tabla.

Esquema con el patrón aplicado



Si se quiere obtener toda la información del envío solo se debe realizar una sentencia SQL hacia la tabla envíos.