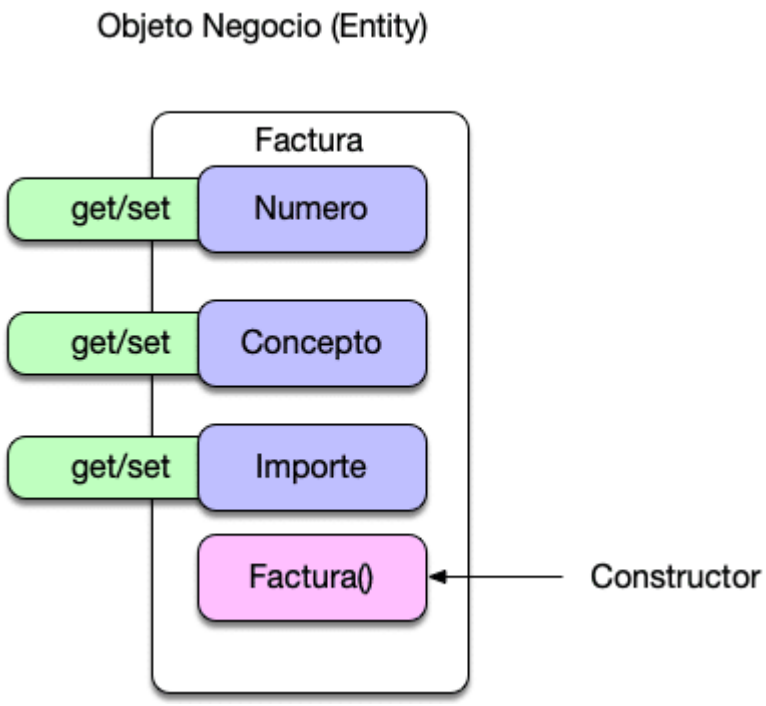
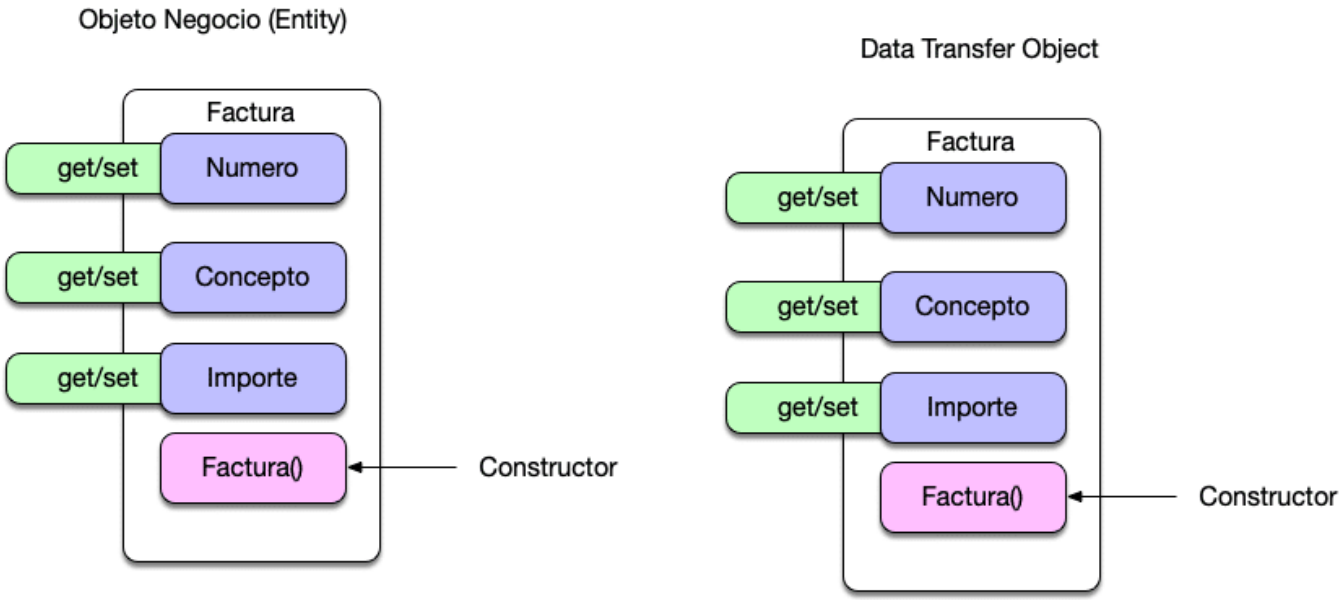
Tabla de Contenidos

¿DTO vs Entity? . Es una pregunta muy muy habitual de cualquier desarrollador. ¿Que diferencia existe entre un Data Transfer Object y un Objeto de Negocio o Entidad (Entity a nivel de JPA) . Es una pregunta interesante. Vamos a verlo paso a paso.[ihc-hide-content ihc_mb_type="show" ihc_mb_who="4" ihc_mb_template="1"]

Una Entidad u objeto de negocio define un concepto de nuestro programa puede ser un Cliente una Factura, un Rectángulo etc . Así pues en principio es muy sencilla de construir ya que dispone de propiedades métodos get /set y constructores .

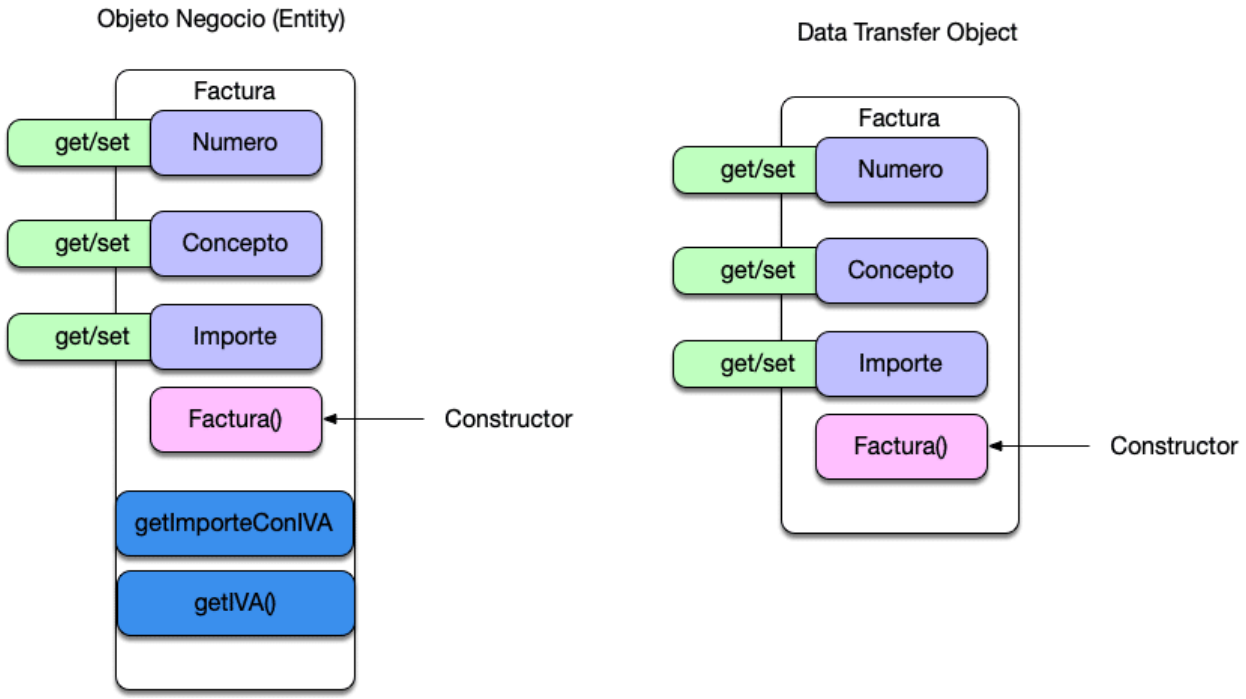


Hasta aquí todo correcto ya que esto define el concepto de Entity o Entidad. Sin embargo nos encontramos que un Data Transfer Object también contiene la misma información así que en un primer momento son prácticamente iguales.



Entity vs DTO y métodos de negocio

Una de las características fundamentales que tienen las entidades es que disponen de métodos adicionales de negocio con el cual hacen algunos calculos. Por ejemplo una factura puede contar con el método getImporteConIVA que calcula el importe con el IVA que le corresponda .

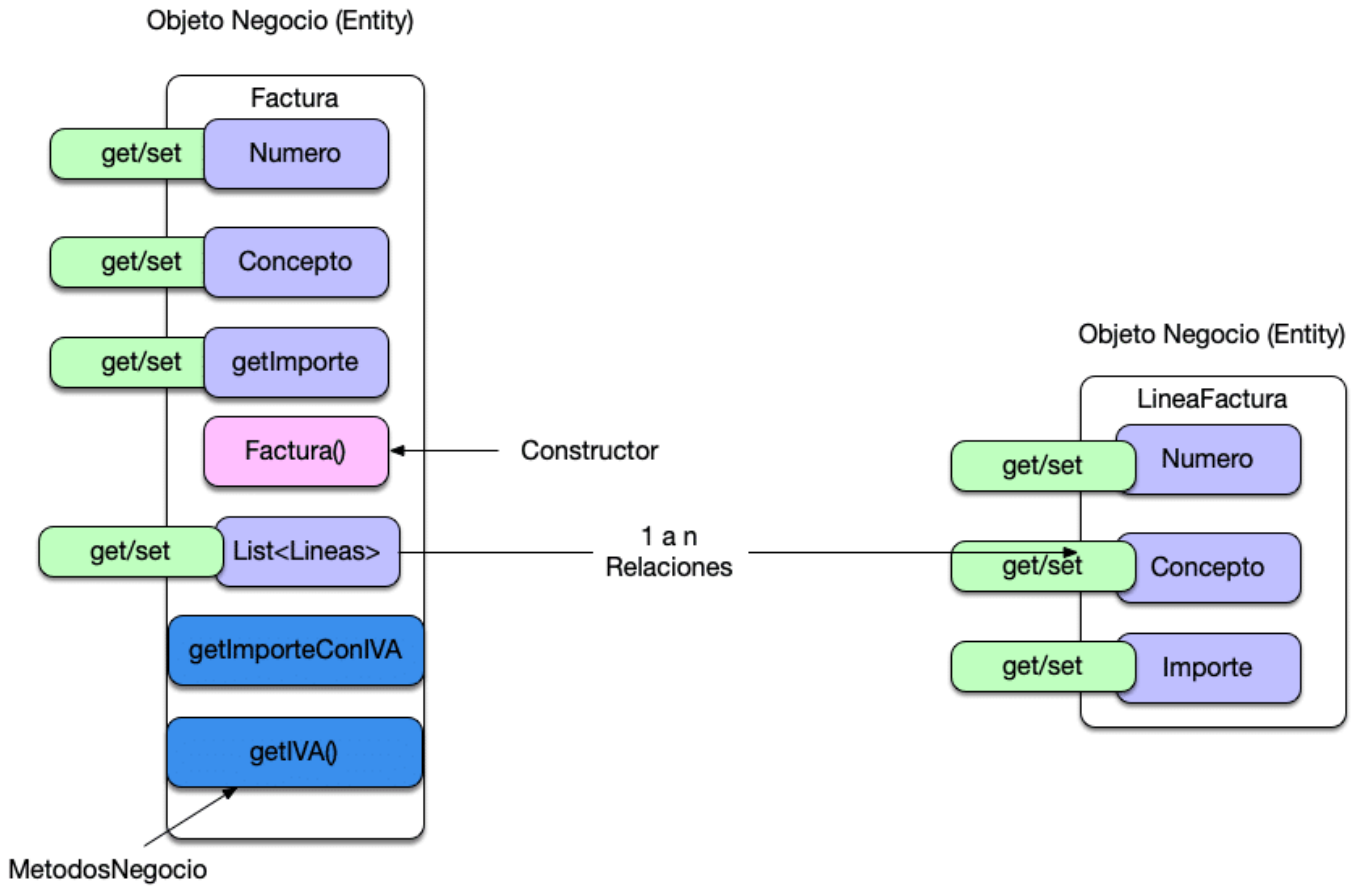


Este tipo de métodos no existe en un DTO ya que el DTO únicamente se usa para compartir propiedades entre sistemas diversos.

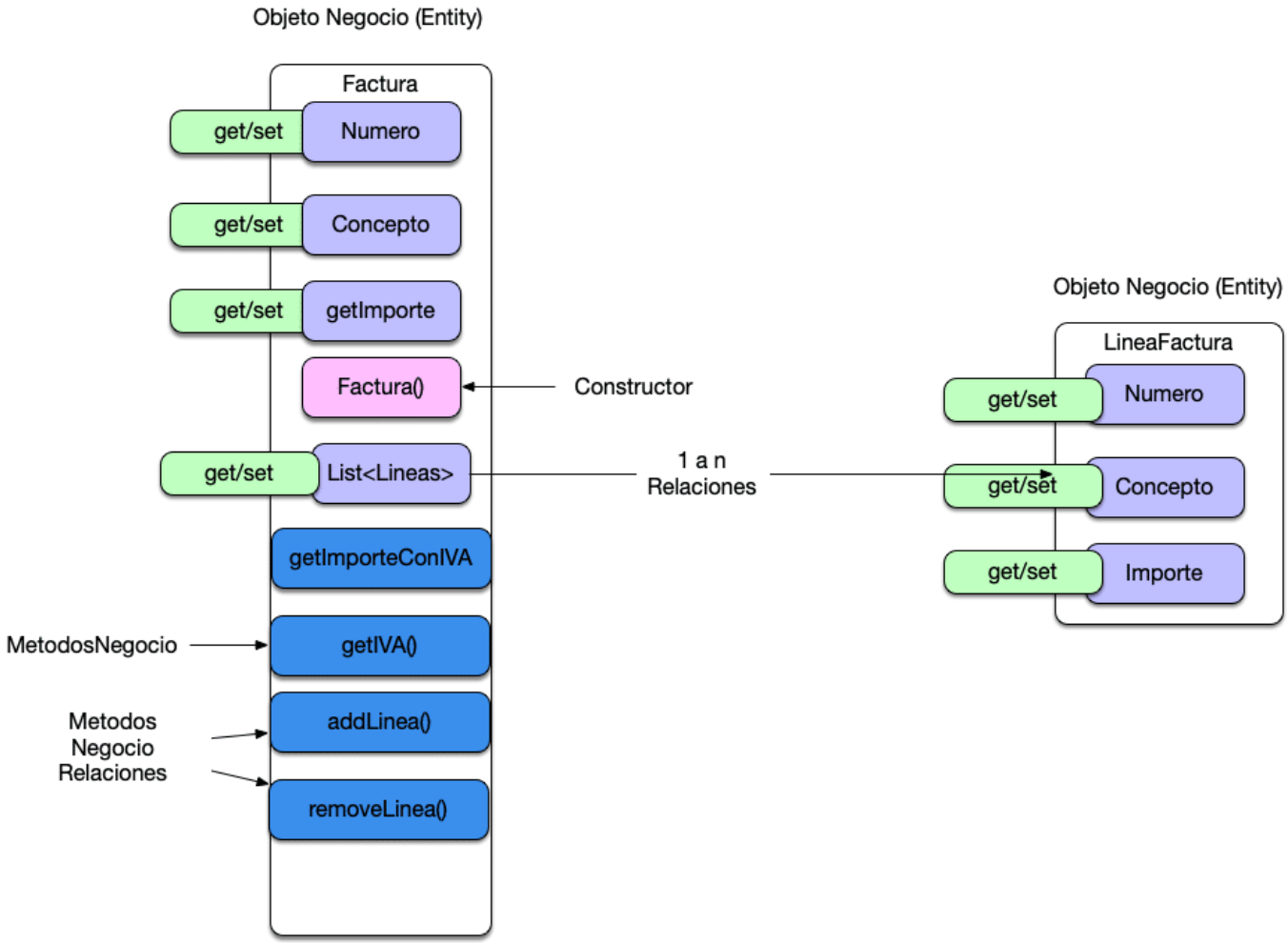
Entidades y Relaciones

Normalmente cuando trabajamos con Entidades como Cliente o Factura estas Entidades se

relacionan entre sí de tal manera que un Cliente tiene varias Facturas o una Factura tiene varias LineasFacturas . Por lo tanto hay relaciones entre las diferentes entidades que se presentan .



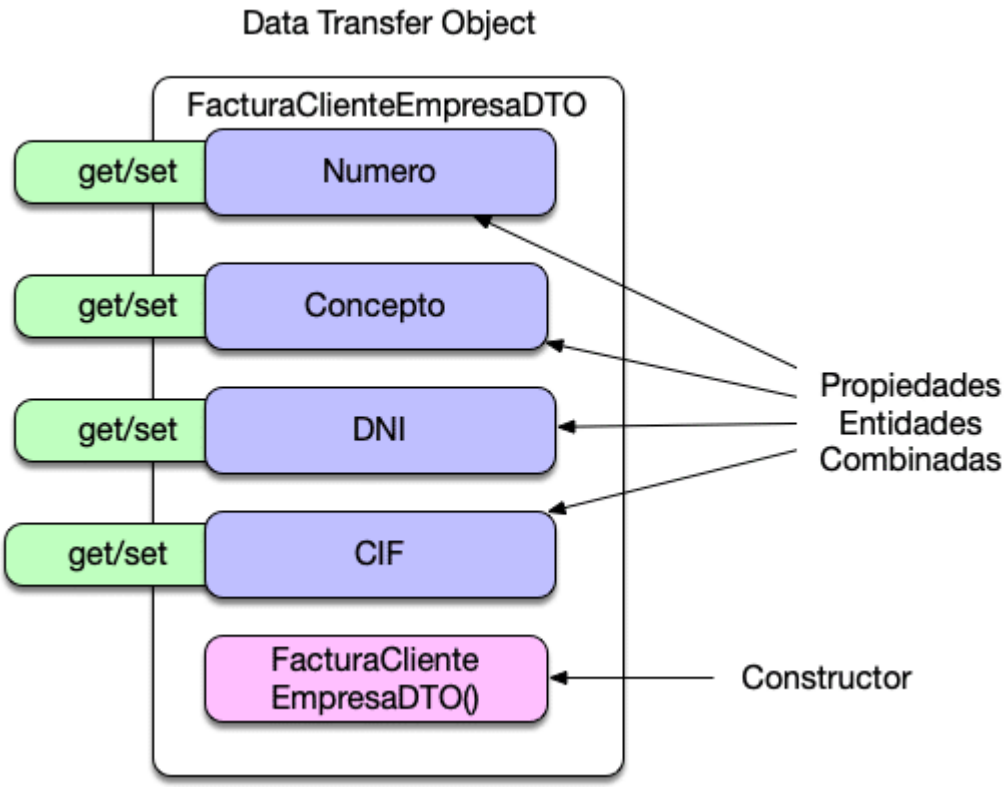
Los DTOs no suelen tener relaciones y son bastante planos. Eso sí a veces pueden tener algunas relaciónes muy básicas entre ellos ya que podemos tener que devolver entre sistemas estructuras agregadas . Ahora bien junto con las relaciones las Entidades disponen de métodos add y remove para añadir nuevas Lineas a la Factura o eliminarlas . Estos métodos de negocio no existen a nivel del DTO.



Por lo tanto como podemos ver los DTO se mantienen prácticamente inmutables desde su primera definición.

DTO vs Entity y propiedades

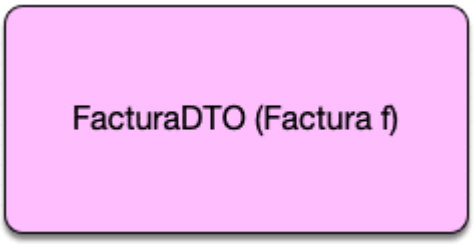
Un DTO no tiene porque tener los campos de una única entidad ya que en muchas ocasiones combina propiedades que se encuentran disponibles en un conjunto de entidades amplio.



Este es una de las características típicas de los DTO que los diferencia de las entidades . Muchas veces los DTO hacen referencias a consultas complejas que devuelven un conjunto de datos que las aplicaciones clientes necesitan.

DTOS y Constructores

Otra de las características muy típicas de los DTO es el uso de constructores que reciben objetos de tipo Entidad que simplifiquen la creación de estos:



Estas son algunas de las diferencias más importantes entre DTO vs Entity a nivel de cualquier lenguaje de programación:

- Data Transfer Object (DTO) un concepto clave
- DTO Assembler, un patrón de diseño
- Entity to DTO y Java 8

[/ihc-hide-content]