Entendiendo el Problema

TDD

WHY - WHAT - HOW

¿Por qué?

Busca establecer el propósito y los objetivos de un proyecto de software. ¿Por qué se usará el sistema? Por qué será útil para el usuario, todos los por qué y toda pregunta estratégica.

Al hacer esta pregunta, se busca entender las razones detrás del proyecto, qué problema se está tratando de resolver y cómo se beneficiarán los usuarios del software.

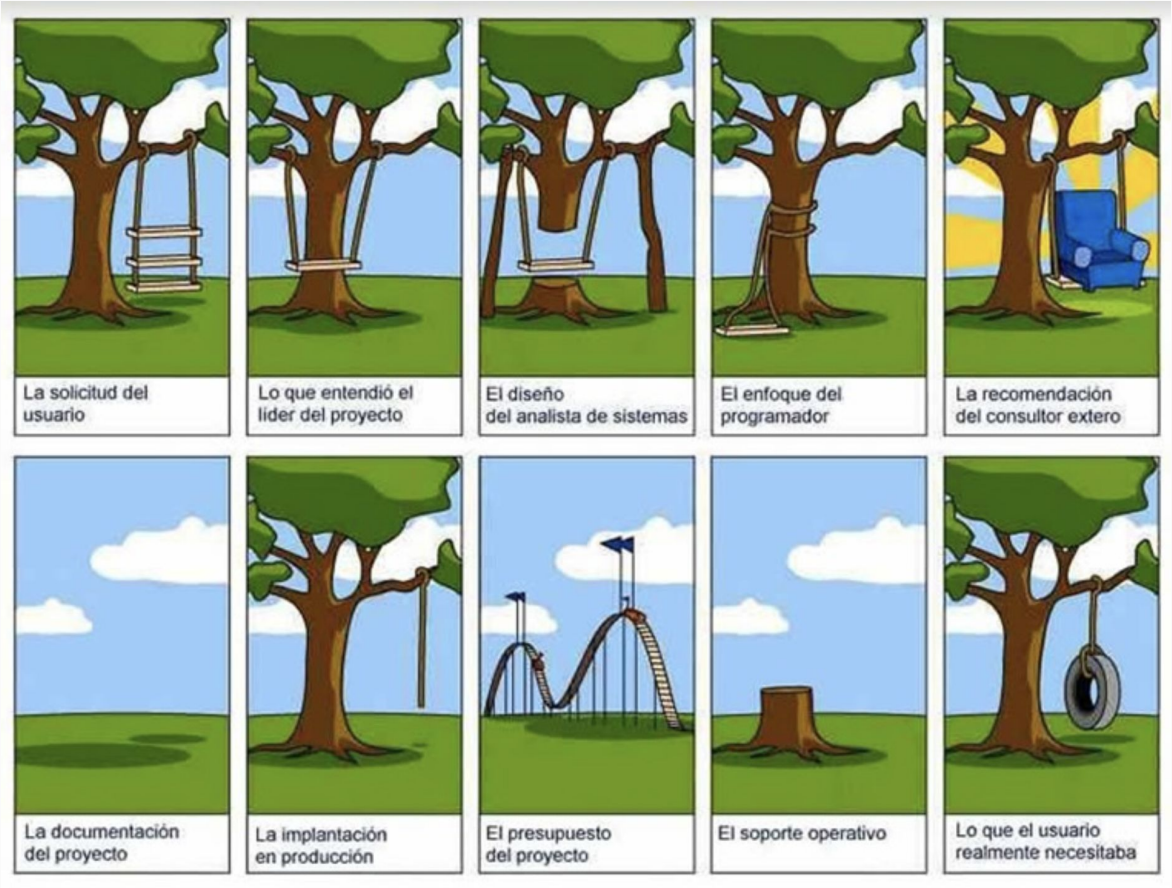
¿Qué?

Busca definir los requisitos del software. Al hacer esta pregunta, se busca entender qué funcionalidades se deben incluir en el software, qué datos se deben procesar y cómo se debe interactuar con el usuario.

¿Cómo?

Busca definir la implementación del software. Al hacer esta pregunta, se busca entender cómo se construirá el software, qué tecnologías se utilizarán y cómo se integrarán las diferentes partes del software.

WHY - WHAT - HOW


La solicitud del usuario
Lo que entendió el líder del proyecto
El diseño del analista de sistemas
El enfoque del programador
La recomendación del consultor extero
La documentación del proyecto
La implantación en producción
El presupuesto del proyecto
El soporte operativo
Lo que el usuario realmente necesitaba

Complejidad del Desarrollo de Software

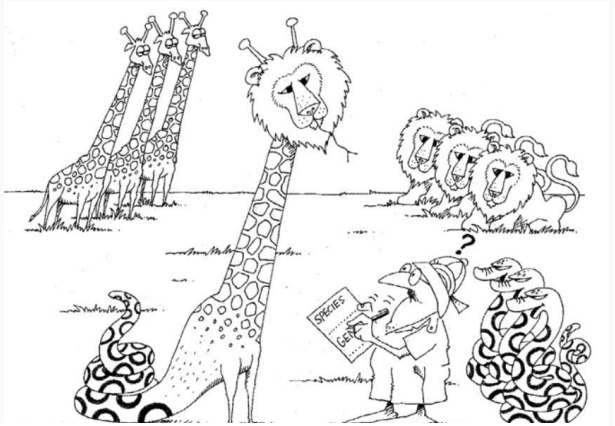
Complejidad = Complejidad del Problema + Complejidad de la Solución

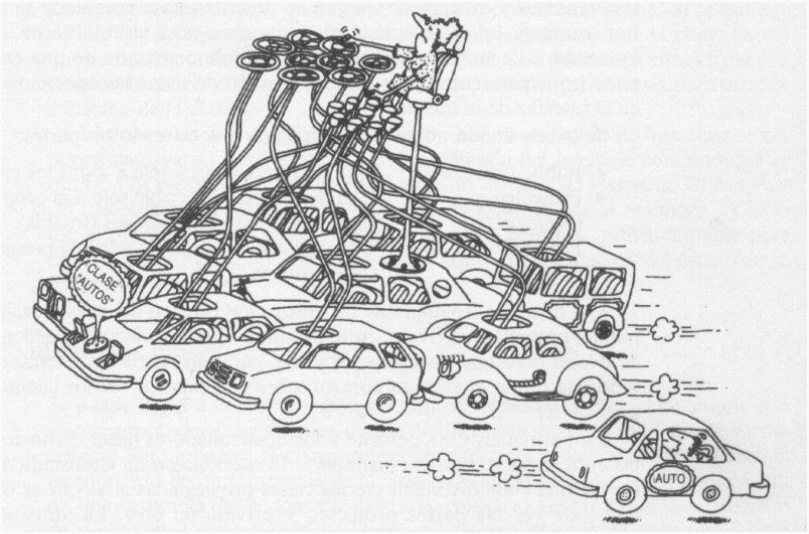
Las soluciones son mayoritariamente simples, lo difícil o complicado en todo caso es llegar a ellas

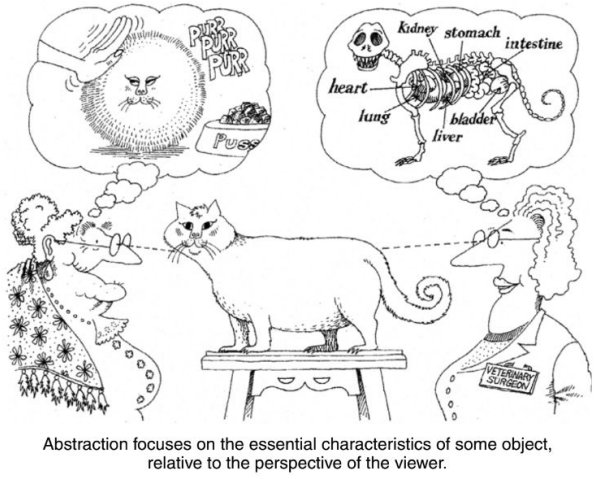
Para resolver el problema es necesario entenderlo antes de empezar a pensar una solución

Mecanismos para atacar la complejidad

- Descomposición → Descubrir
- Abstracción → Pensar
- Establecer Jerarquías → Inventar







Abstraction focuses on the essential characteristics of some object, relative to the perspective of the viewer.

Modelo de Dominio

Objetivo: Entender en detalle el negocio y sus reglas

Mecanismo utilizado: Patrones de Análisis o Colaboración

Modelo de Diseño

Objetivo: Implementar una solución al modelo planteado en el análisis teniendo en cuenta las restricciones impuestas por los requerimientos no funcionales.

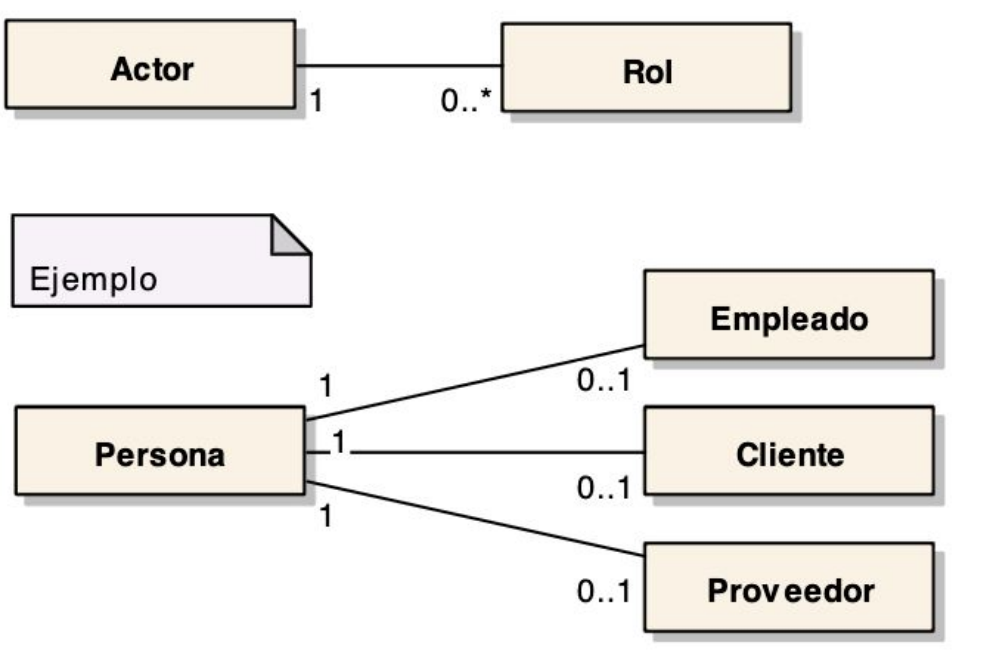
Mecanismo utilizado: Patrones de Diseño

Patrones de Colaboración

Conceptos a buscar	Instancias
Gente	Actor
	Rol
Lugares	Lugar
	Gran Lugar
Cosas	Ítem
	Ítem Específico
	Ensamble
	Parte
	Contenedor
	Contenido
	Grupo
	Miembro
Eventos	Transacciones
	Transacciones Compuestas
	Transacciones Cronológicas
	Line Ítem

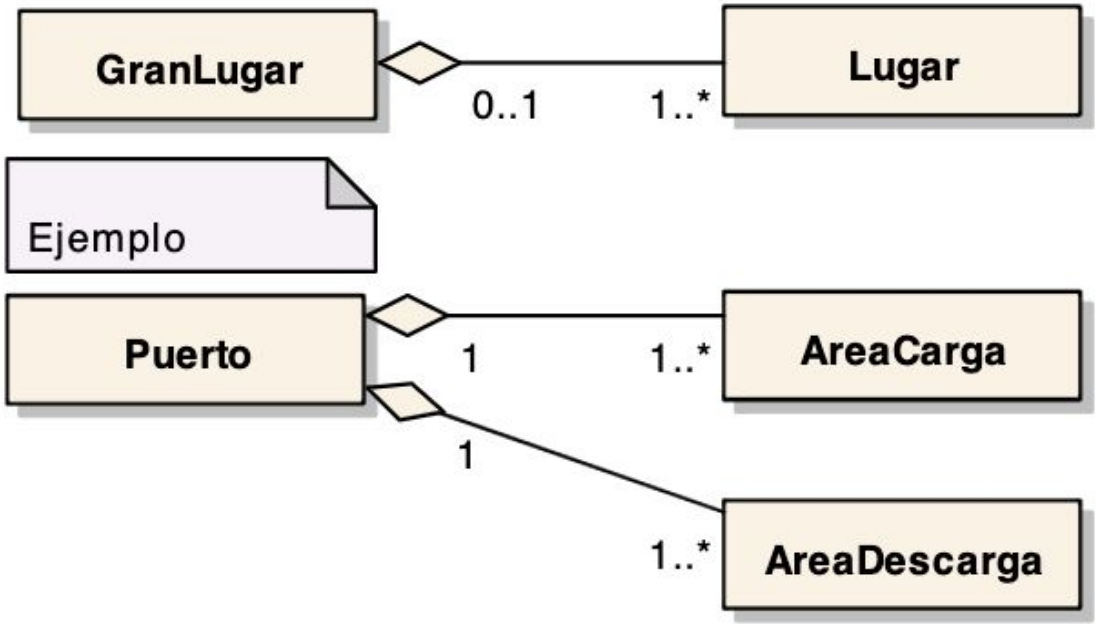
ACTOR - ROL PATTERN

Un actor puede conocer varios roles, pero solo puede tomar uno de cada tipo.



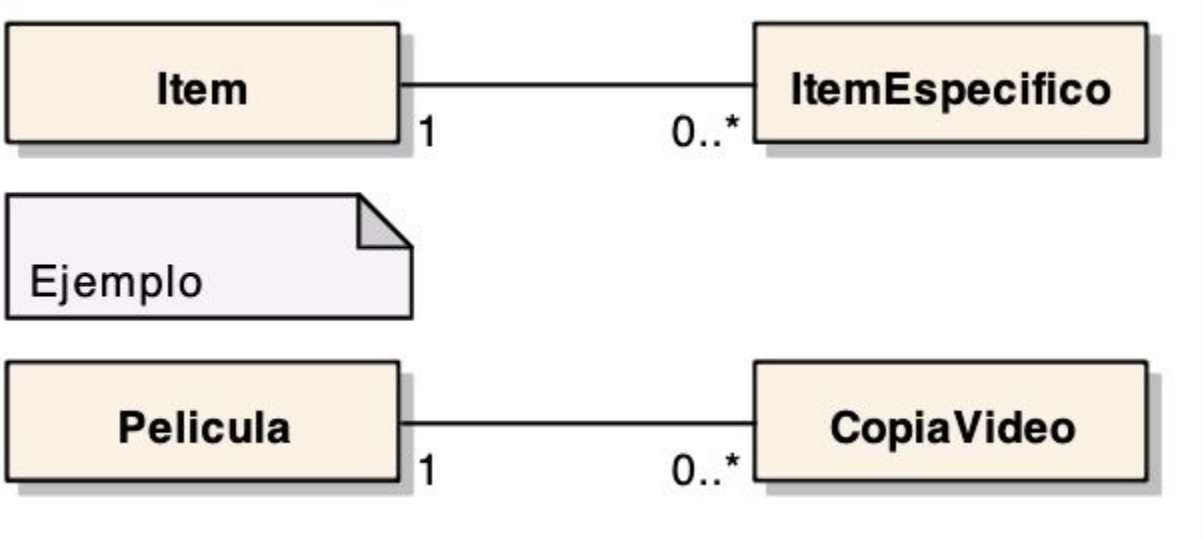
GRAN LUGAR - LUGAR PATTERN

Un gran lugar conoce al menos un lugar, y sirve como contenedor de sus lugares.



ITEM - ITEM ESPECIFICO PATTERN

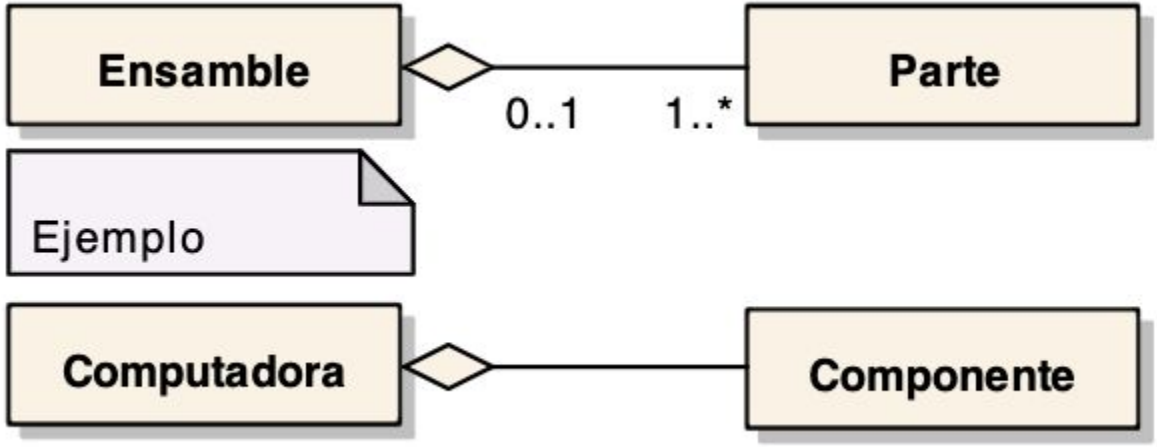
El ítem describe la información que es común en todas las variaciones.



ENSAMBLE - PARTE PATTERN

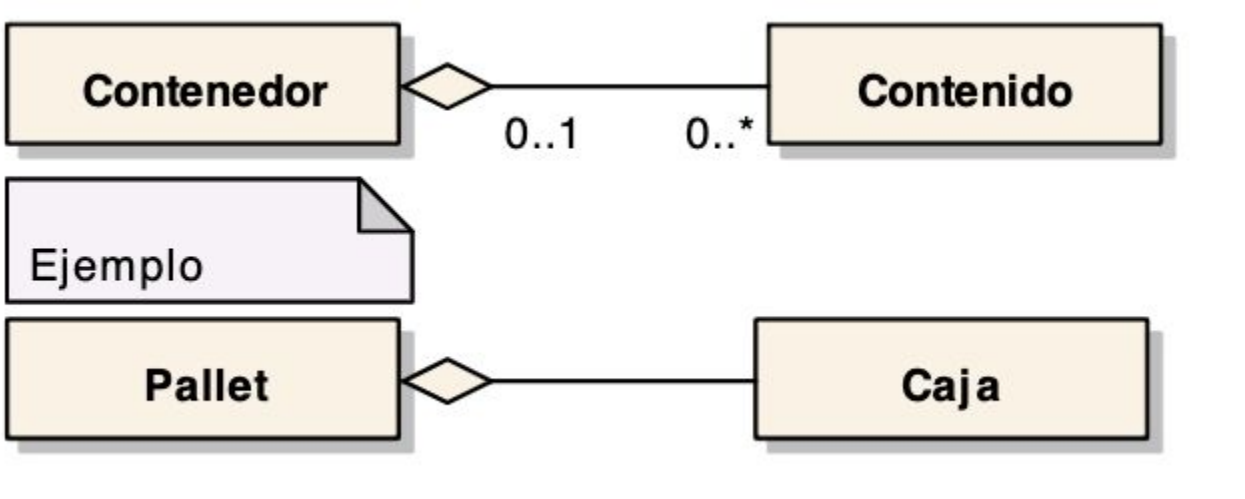
El patrón de ensamblaje-parte es uno de los patrones para modelar cosas con estructuras complejas.

Se utiliza para modelar una cosa que está construida a partir de otras cosas. Este patrón difiere de las otras estructuras complejas, contenedor y grupo, porque no puede existir sin al menos una parte.



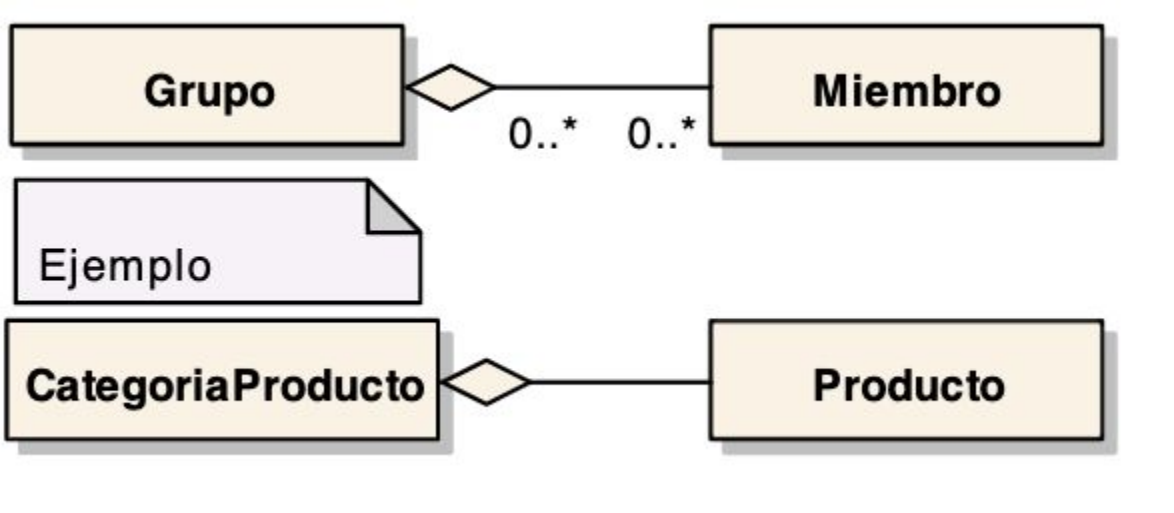
CONTENEDOR - CONTENIDO PATTERN

Utiliza el patrón contenedor-contenido cuando una cosa es un recipiente o lugar de almacenamiento para otras cosas.



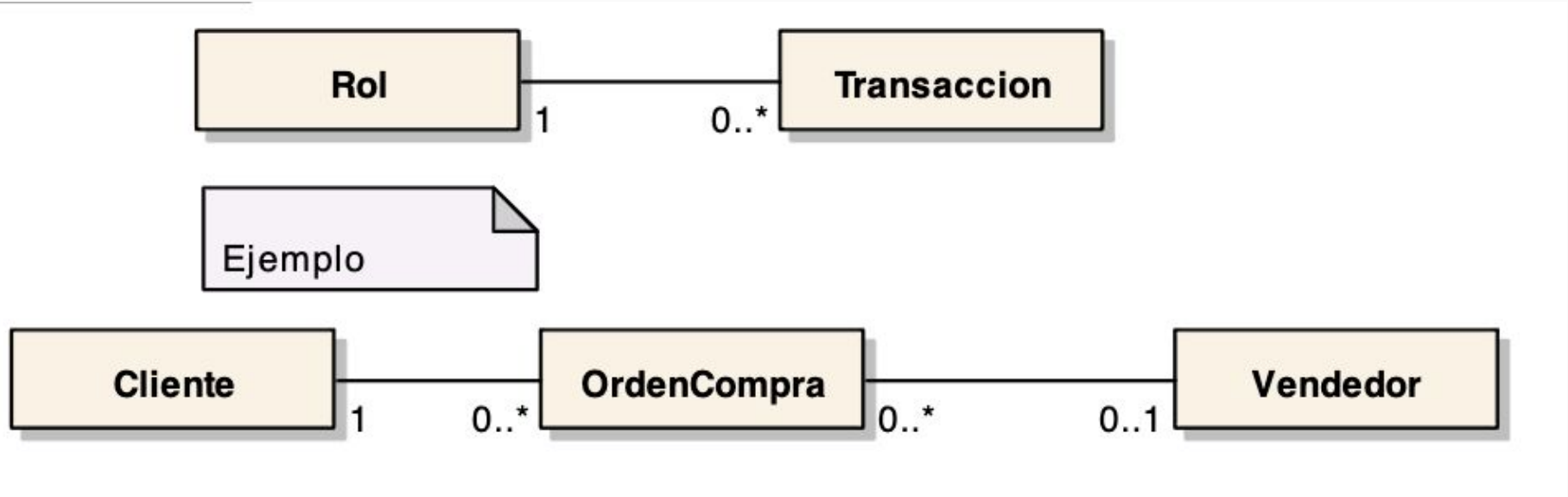
GRUPO - MIEMBRO PATTERN

Se utiliza con frecuencia para modelar colecciones y clasificaciones de cosas, y también puede ser utilizado para colecciones de personas y lugares.



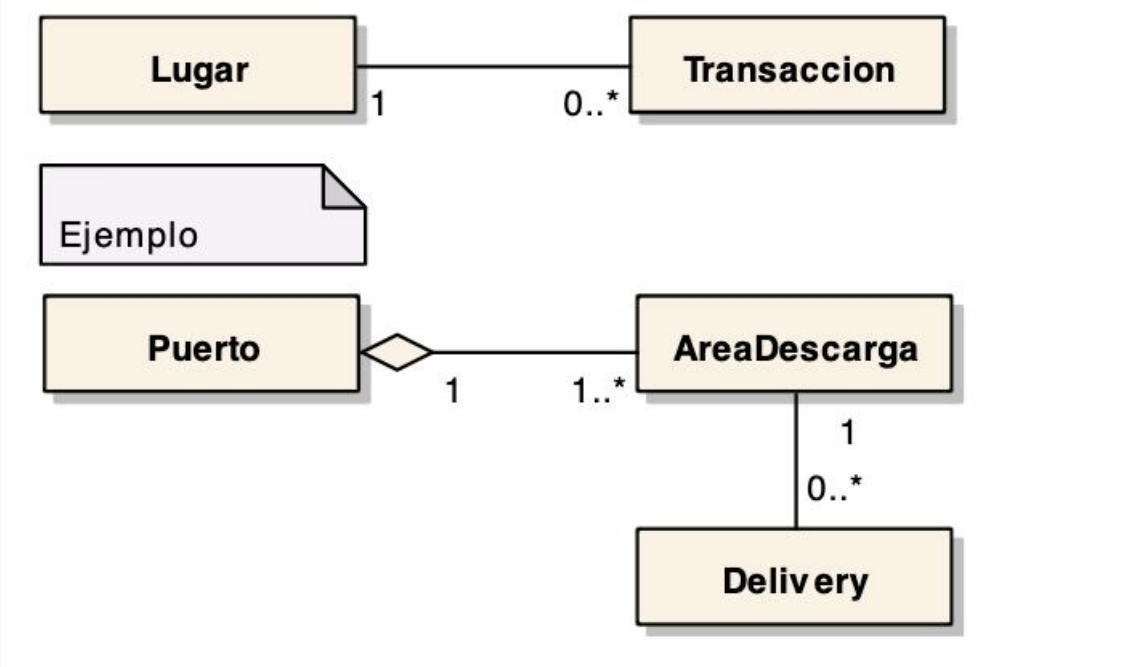
ROL - TRANSACCION PATTERN

El patrón transacción-rol modela una entidad interactuando con cosas en lugares. El jugador del patrón de rol se utiliza en lugar de actor porque el rol describe la participación de una entidad dentro de un contexto y una interacción siempre ocurre dentro de un contexto.



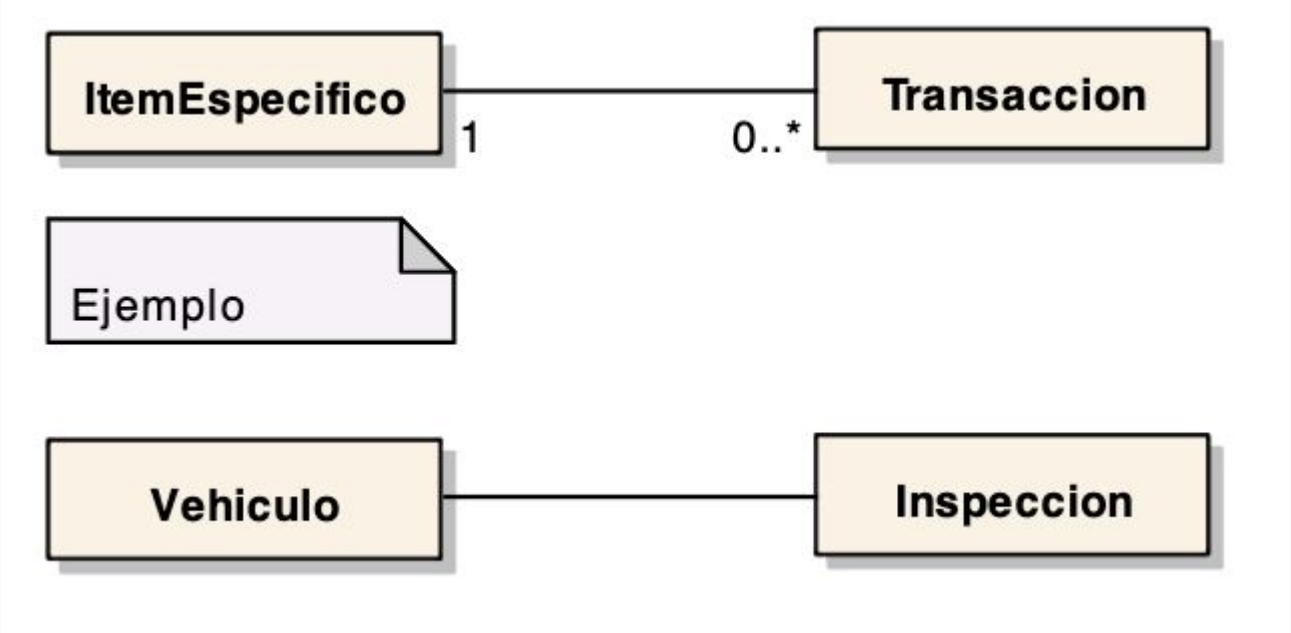
TRANSACCION - LUGAR PATTERN

Una transacción conoce acerca de un lugar donde ocurre



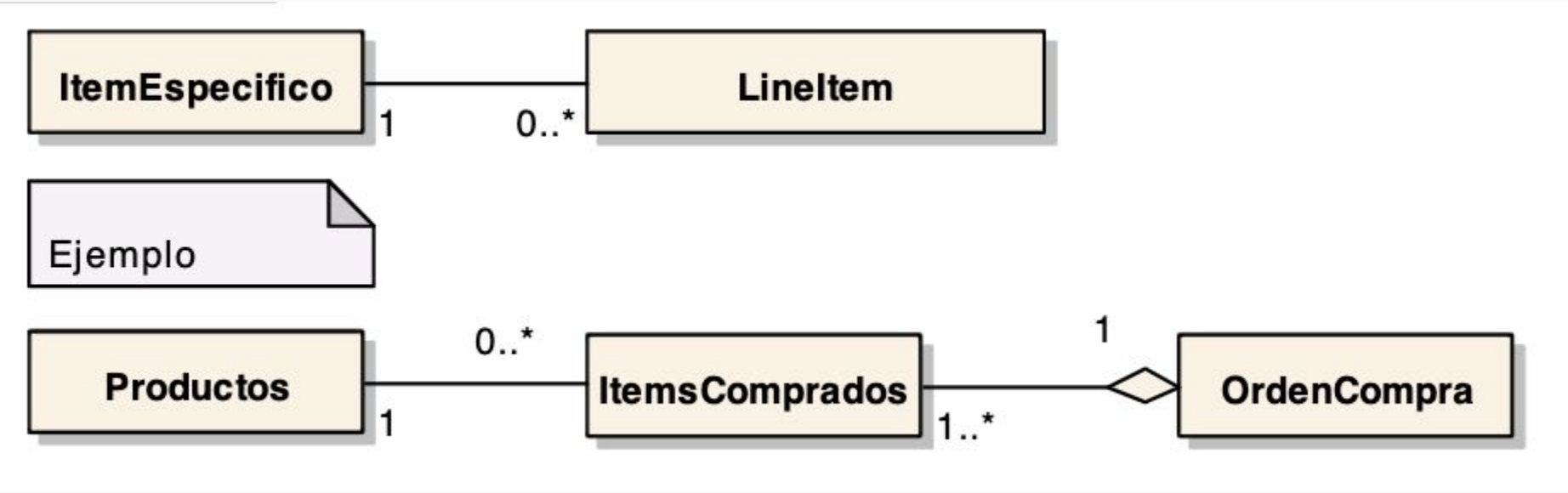
ITEM ESPECÍFICO - TRANSACCIÓN PATTERN

Modela la participación de una cosa, un ítem, en una interacción.



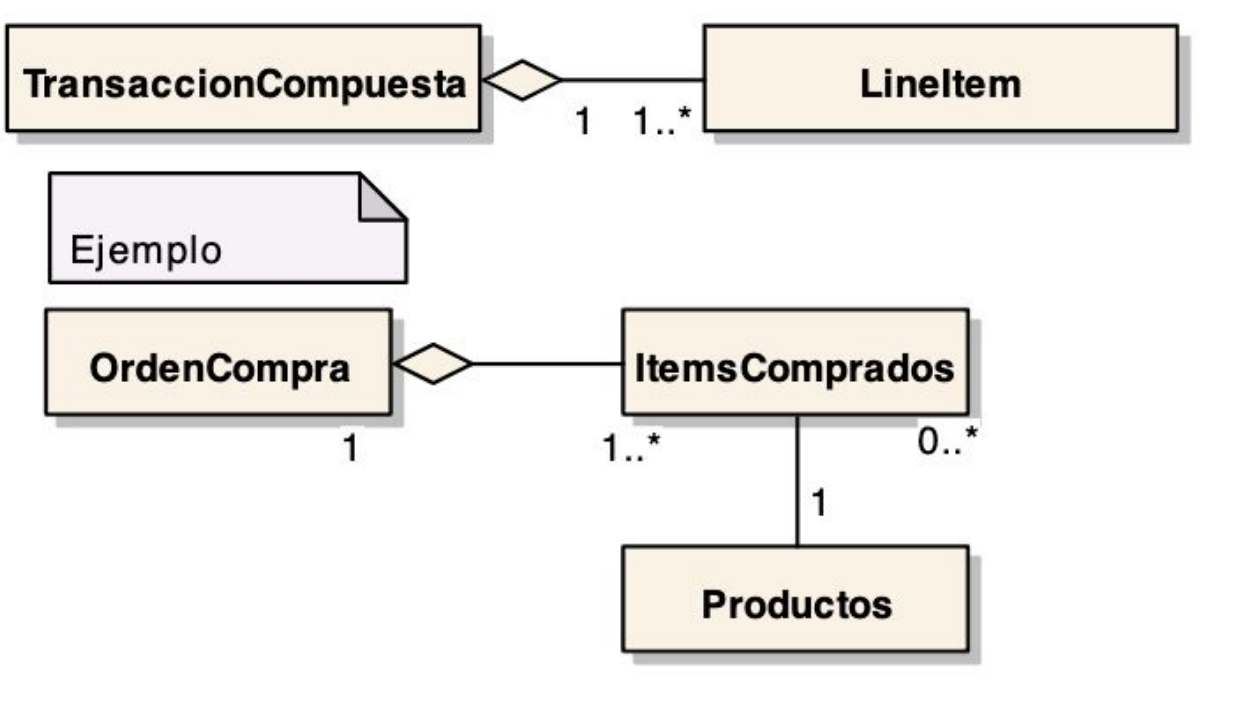
ITEM ESPECÍFICO - LINE ITEM PATTERN

Utiliza el patrón de ítem específico - line item para describir la interacción de una cosa en un evento que involucra muchas cosas.
Un elemento de línea conoce exactamente un artículo específico. El elemento de línea captura detalles sobre la interacción del artículo específico con una transacción compuesta.



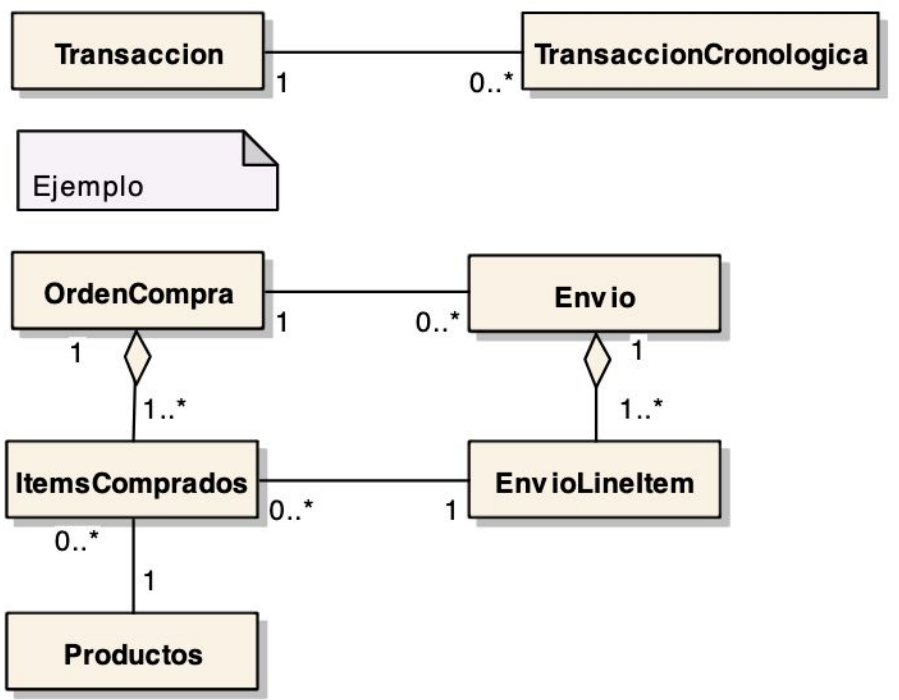
TRANSACCIÓN COMPUESTA - LINE ITEM PATTERN

Utiliza el patrón cuando una transacción involucra muchos line items



TRANSACCIÓN - TRANSACCIÓN CRONOLÓGICA PATTERN

Utiliza el patrón de transacción - transacción cronológica para modelar interacciones que siguen a interacciones anteriores.



REGLAS DE NEGOCIO

Son las restricciones que gobiernan las acciones dentro de un dominio de negocio

- En el modelo se traducen en reglas de colaboración.
- La forma de incorporarlas al modelo consiste en restricciones a ser probadas en la colaboración entre los distintos objetos del modelo.
- En el modelo se traduce por ejemplo en si dos objetos pueden crear una nueva relación o remover una existente.
- ¿Donde ubicarlas?
- Si no se ubican en el modelo, el mismo es incompleto. La dinámica de los objetos será externa al mismo.
- ¿Cómo expresarlas? ¿Qué colaborador (objeto) prueba cada regla en función de la información que conoce o puede consultar?

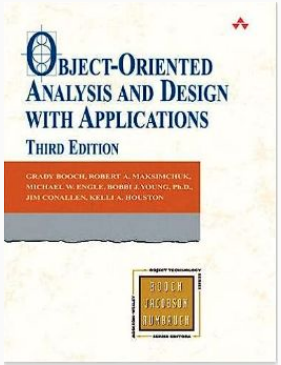
REGLAS DE NEGOCIO

Tipos de reglas

- Tipo: un medicamento puede ser cargado solo en un container refrigerado
- Multiplicidad: un pallet refrigerado puede contener hasta 10 cajas
- Propiedad (validación, comparación): un pago debe registrar un número válido de tarjeta de crédito, la temperatura de un container refrigerado debe ser menor a 0 grados centígrados
- Estado: una orden no debe ser entregada si antes fue cancelada
- Conflicto: un vuelo no puede ser programado en una puerta en un mismo horario que otro vuelo, un producto no puede ser sumado a una orden de compra de un menor de edad si la venta está prohibida a menores.

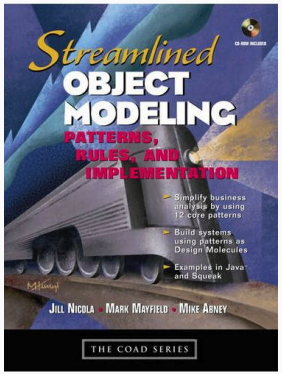
Reglas de colaboración: chequeo de reglas de negocio entre objetos participantes de las relaciones.
El cambio de estado, el establecimiento de una relación o la ruptura de la misma requiere el chequeo de una regla de negocio. Este chequeo lo vemos como una colaboración entre objetos.

BIBLIOGRAFIA



Object-Oriented Analysis and Design with Applications

by Grady Booch, Robert A. Maksimchuk, et al. | Apr 30, 2007



Streamlined Object Modeling: Patterns, Rules and Implementation

by Jill Nicola / Mark Mayfield / Mike Abney Nicola / Mayfield / Abney (Author)