



Análisis Orientado a Objetos

Qué es un Paradigma ?

Antes de adentrarnos en el AOO (Análisis Orientado a Objetos) necesitamos definir claramente que es un paradigma; Thomas Kuhn en su libro “*La estructura de las revoluciones científicas*” [Kuhn 70] define a un Paradigma de la siguiente forma:

“conjunto de teorías, estándares y métodos que juntos representan una forma de organizar el conocimiento, esto es, una forma de ver el mundo”.

Qué es entonces el Paradigma Orientado a Objetos ? Una serie de principios aplicables al desarrollo de software, basado en la visión del sistema o de la realidad, como un conjunto de abstracciones a la que se denomina objetos.

El objetivo del **AOO** es modelar la semántica del problema en términos de objetos distintos interrelacionados relacionados.

Definamos un objeto

“Una representación de casi cualquier información compuesta que un sistema (SOO) debe entender, conceptualizando como *información compuesta* a algo que tiene muchas propiedades o atributos diferentes.” Larman.

Técnica para identificar objetos

Distinguir de un texto determinado, los sustantivos y los verbos. Los sustantivos son “posibles” clases *candidatas* o atributos de las mismas. Los verbos son “posibles” responsabilidades que corresponden a esos posibles objetos.

Las clases candidatas pasaran a ser definitivas, cuando de acuerdo al contexto le asignemos los atributos y responsabilidades.

Conceptos básicos a tener en cuenta

Las técnicas orientadas a objetos funcionan bien en situaciones en las que los sistemas de información complicados, pasan a través de un proceso continuo de mantenimiento, adaptación y rediseño.

Cuando se introdujo la metodología orientada a objetos como herramienta de analisis, sus defensores citaban la reutilización de los objetos como su principal beneficio. Es lógico pensar que el reciclaje de las partes del programa reduce los costos del desarrollo en los sistemas basados en computadoras, y esto ha resultado un procedimiento muy efectivo en el desarrollo de GUI y bases de datos. Aunque la reutilización es el principal objetivo, también es muy importante el mantenimiento de los sistemas y, como la metodología orientada a objetos crea objetos que contienen

tanto datos como código de programa, si se modifica un objeto habrá un mínimo impacto en los demás objetos.

Diferencia de Análisis Diseño y Programación Orientada a Objetos

El **análisis** es la fase cuyo objetivo es estudiar y comprender el dominio del problema, una necesidad o una oportunidad de mejora es decir, el análisis se centra en responder a la pregunta **¿QUÉ HACER?**

El **diseño**, focaliza sus esfuerzos en la creación de la solución acorde a los requisitos planteados en el análisis, esto es, el diseño se haya centrado en el espacio de la solución, intentando dar respuesta a la cuestión **¿CÓMO HACERLO?**

Por último, la fase de **implementación** se encargara de la traducción del diseño de la aplicación al lenguaje de programación OO, adaptando por tanto la solución a un entorno concreto, para ello utiliza y examina los objetos que fueron analizados y posteriormente refinados en el diseño, que forman parte de un sistema, donde cada objeto es una representación de alguna cosa o evento real. **¿CÓMO IMPLEMENTARLO?**

Clases

Por lo general, los objetos forman parte de un grupo de elementos similares, conocidos como **clases**. La intención de colocar elementos en clases no es nuevo. Describir el mundo como algo compuesto de animales, vegetales y minerales es un ejemplo de clasificación. La metodología científica incluye clases de animales (como mamíferos) y después divide esas clases en subclases (como animales que ponen huevos, y mamíferos marsupiales).

Los objetos se representan y agrupan mediante clases, las cuales son óptimas para la reutilización y la facilidad de mantenimiento. Una clase define el conjunto de atributos compartidos y comportamientos que se encuentran en cada objeto de la clase. Por ejemplo, los registros para los estudiantes en la sección de un curso tienen información similar almacenada para cada estudiante. Se dice que los estudiantes conforman una clase. Los valores pueden ser distintos para cada estudiante, pero el tipo de información es el mismo.

RELACIONES:

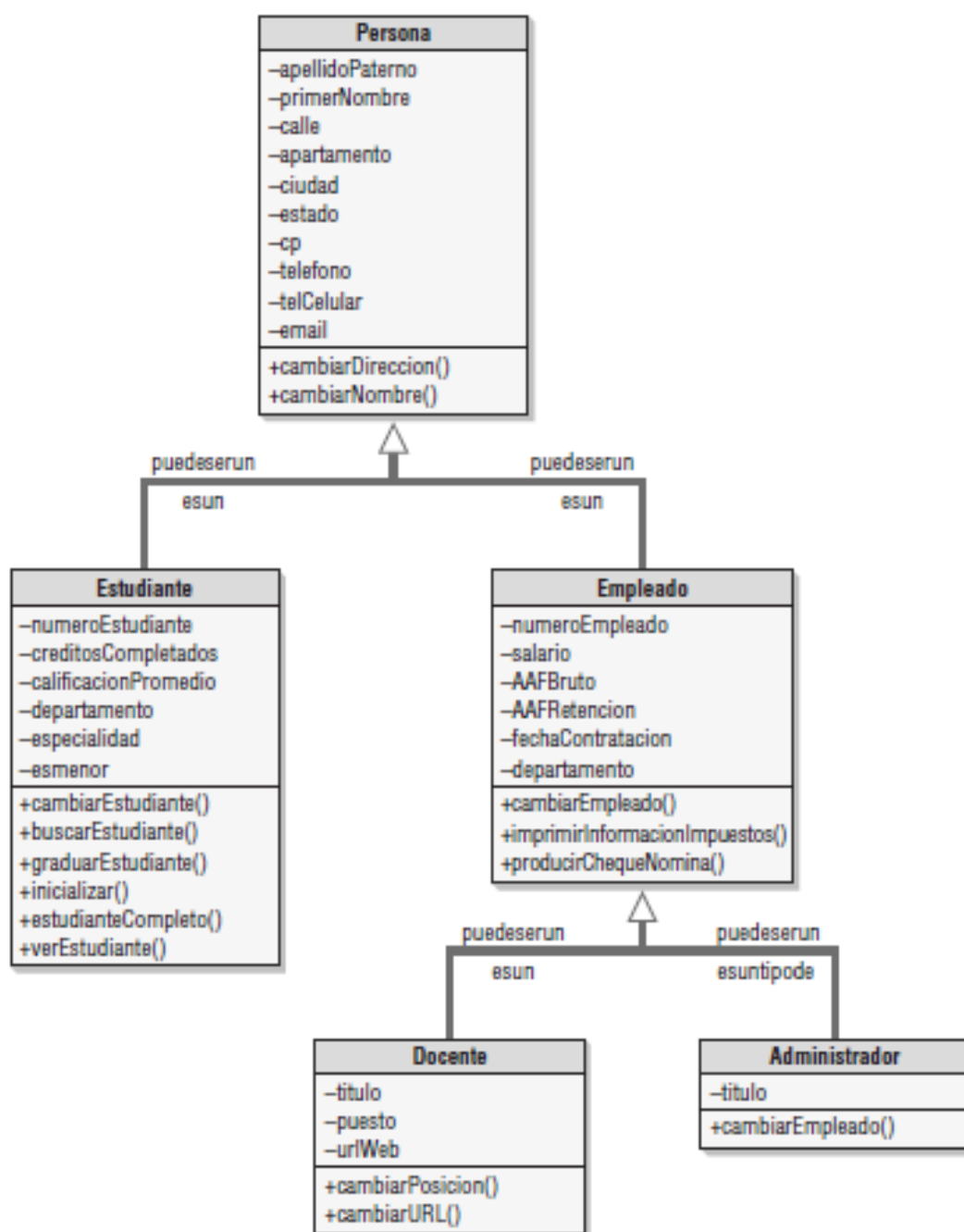
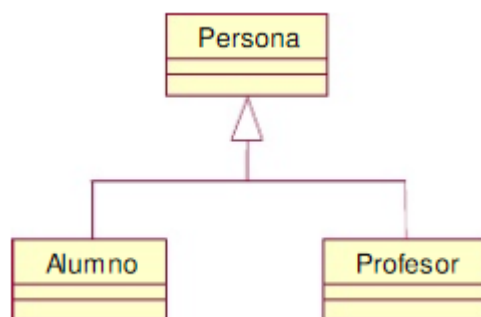
1) Herencia (Especialización/Generalización):

Indica que una clase (clase derivada/subclase) hereda los métodos y atributos especificados por una clase (clase base), por lo cual una clase derivada además de tener sus propios métodos y atributos, podrá acceder a las características y atributos visibles de su clase base es válida cuando las subclases también poseen comportamiento diferente entre sí además de compartir otros.



Símbolo de herencia

En la siguiente figura podrá observar un ejemplo de este tipo de relación:



Ventajas: Reutilización de código, ahorro de tiempo de desarrollo, posibilidad de agregar nuevas subclases sin afectar la clase principal.

Desventajas: Alto acoplamiento, clases poco portables a otro entorno/ contexto si quiero implementarlas sin herencia, debe reescribirse el código.

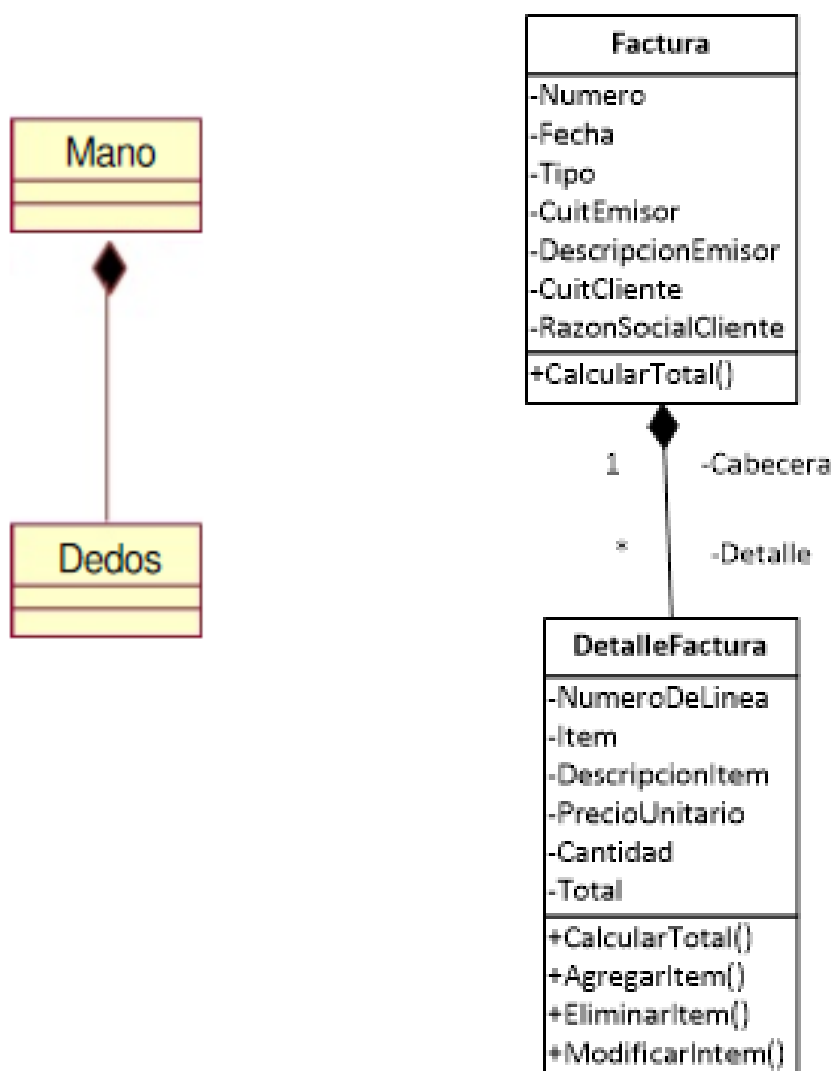
Se generaliza para ocultar diferencias y se especializa para resaltar las diferencias.

2) Asociación:

2.1) Composición:

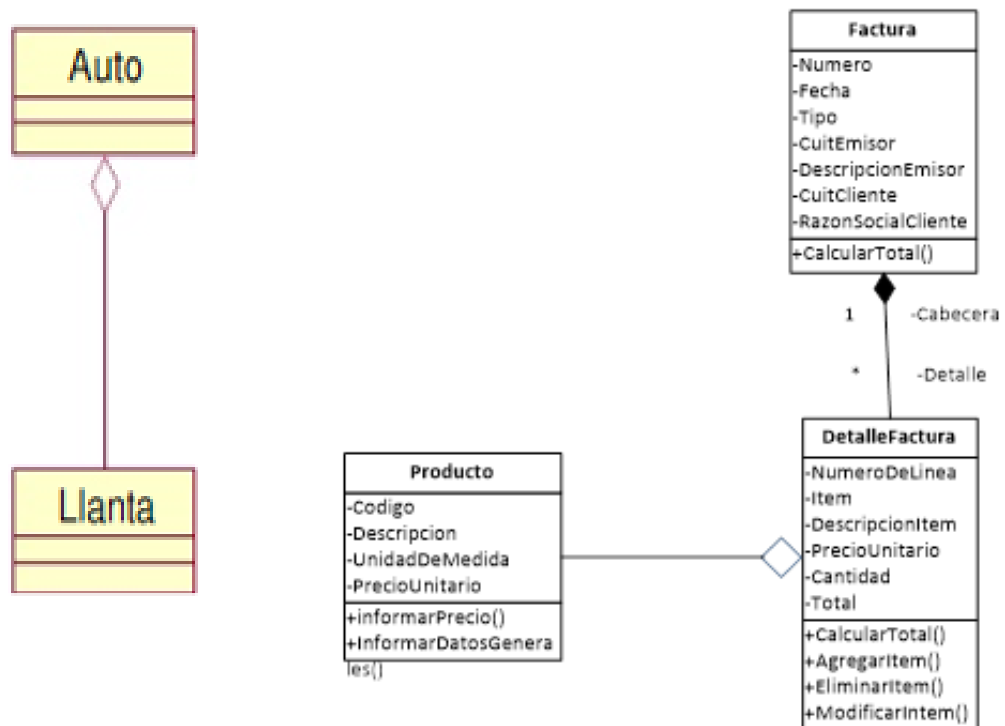
La composición es un tipo de relación estática, en donde el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del que lo incluye (el objeto base se construye a partir del objeto incluido, es decir, es parte/todo), “es parte de”. La no existencia del todo invalida la existencia de la parte, esta no tiene sentido si no está unida al todo.

En la siguiente figura se puede observar un ejemplo de este tipo de relación:



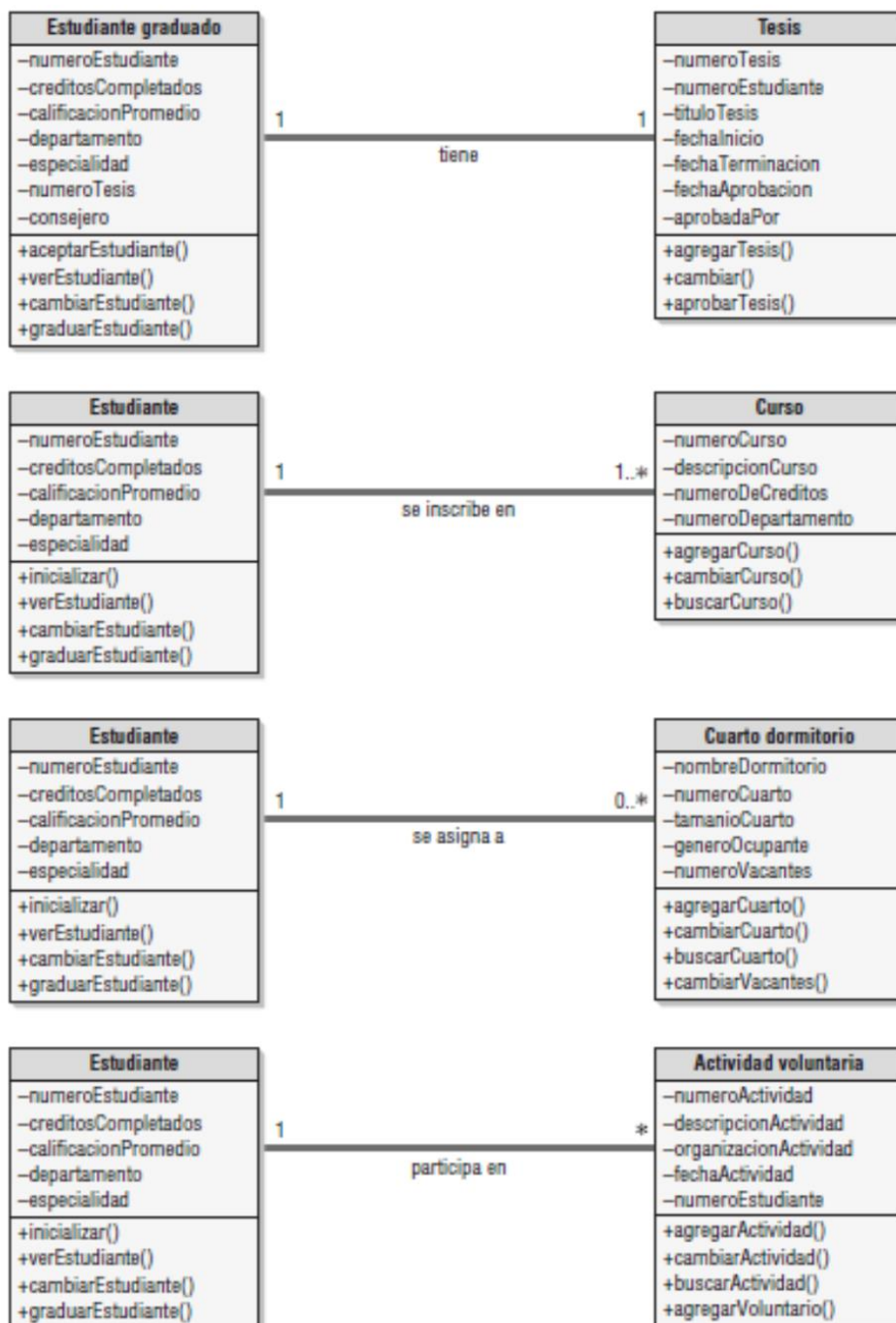
2.2) Agregación:

La agregación es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye (el objeto base utiliza al incluido para su funcionamiento). Una clase está formada por objetos de otra clase, la relación es “tiene un”. La no existencia del todo no invalida la parte.



2.3) Asociación Simple:

Es una relación semántica entre clase. A la misma se le pueden agregar signos tales como sentido de lectura nombre de la asociación, cardinalidad, etc.



TARJETAS CRC Y PENSAMIENTO EN OBJETOS

Ahora que hemos presentado los conceptos fundamentales del análisis y diseño orientados a objetos, necesitamos examinar formas de crear clases y objetos a partir de los problemas de negocios y los sistemas a los que nos enfrentamos.

Una forma de empezar a poner en práctica la metodología orientada a objetos es empezar a pensar y hablar de esta manera. Un método útil es desarrollar tarjetas CRC.

CRC representa a las clases, responsabilidades y colaboradores. El analista puede usar estos conceptos cuando empieza a hablar sobre el sistema o a modelarlo a partir de una perspectiva orientada a objetos. Las tarjetas CRC se utilizan para representar las responsabilidades de las clases y las interacciones entre ellas. Los analistas crean las tarjetas con base en escenarios que describen los requerimientos del sistema. Estos escenarios modelan el comportamiento del sistema que se está estudiando. Si se van a usar en un grupo, las tarjetas CRC se pueden elaborar en forma manual en pequeñas tarjetas de cartulina por cuestión de flexibilidad, o se pueden crear mediante el uso de una computadora.

Hemos agregado dos columnas a la plantilla de tarjeta CRC original: la columna Pensamiento en objetos y la columna de propiedades. Los enunciados de Pensamiento en objetos están escritos en español común; el nombre de la propiedad o atributo está escrito en su lugar apropiado. El propósito de estas columnas es aclarar el pensamiento y ayudar a pasar a la creación de diagramas de UML.

Interacción durante una sesión CRC

Las tarjetas CRC se pueden crear de manera interactiva con unos cuantos analistas que puedan trabajar en conjunto para identificar la clase en el dominio del problema presentado por la empresa. Una sugerencia es buscar todos los sustantivos y verbos en un enunciado del problema que se haya creado para capturarlo. Por lo general, los sustantivos indican las clases en el sistema; para encontrar las responsabilidades hay que identificar los verbos.

Haga una sesión de lluvia de ideas con su grupo de analistas para identificar todas las clases que pueda. Siga el formato estándar para la lluvia de ideas, que consiste en no criticar la respuesta de ninguno de los participantes en este punto, sino solicitar todas las respuestas posibles. Una vez identificadas todas las clases, los analistas pueden empezar a compilarlas, eliminar las ilógicas y escribir cada una en su tarjeta. Asigne una clase a cada persona en el grupo, que será su “propietaria” durante la sesión CRC.

A continuación, el grupo crea escenarios que en realidad son recorridos de las funciones del sistema, para lo cual se toma la funcionalidad deseada del documento de requerimientos creado con anterioridad. Primero hay que considerar los métodos de sistemas típicos, con excepciones tales como la recuperación de errores que se lleva a cabo después de haber cubierto los métodos de rutina.

Mientras el grupo decide sobre la clase que será responsable de una función específica, el analista propietario de la misma durante la sesión elige esa tarjeta y declara: “Necesito cumplir con mi responsabilidad”. Cuando alguien sostiene una tarjeta en el aire, se considera un objeto y puede hacer cosas. Después el grupo procede a refinar la responsabilidad en tareas cada vez más pequeñas, de ser posible. Si es apropiado, el objeto puede cumplir con estas tareas o el grupo puede

decidir que se cumplan mediante la interacción con otras cosas. Si no existen otras clases apropiadas, tal vez el grupo tenga que crear una.

Nombre de la clase: Departamento			
Superclases:			
Subclases:			
Responsabilidades	Colaboradores	Pensamiento en objetos	Propiedad
Agregar un nuevo departamento	Curso	Conozco mi nombre	Nombre del departamento
Proveer información del departamento		Conozco mi silla del departamento	Nombre de la silla

Nombre de la clase: Curso			
Superclases:			
Subclases:			
Responsabilidades	Colaboradores	Pensamiento en objetos	Propiedad
Agregar un nuevo curso	Departamento	Conozco el número de mi curso	Número de curso
Modificar información del curso	Libro de texto	Conozco mi descripción	Descripción del curso
Mostrar información del curso	Asignación	Conozco mi número de créditos	Créditos
	Examen		

Nombre de la clase: Libro de texto			
Superclases:			
Subclases:			
Responsabilidades	Colaboradores	Pensamiento en objetos	Propiedad
Agregar un nuevo libro de texto	Curso	Conozco mi ISBN	ISBN
Cambiar la información del libro de texto		Conozco mi autor	Autor
Buscar información del libro de texto		Conozco mi título	Título
Eliminar libros de texto obsoletos		Conozco mi edición	Edición
		Conozco mi editorial	Editorial
		Sé si soy obligatorio	Obligatorio

Nombre de la clase: Asignación			
Superclases:			
Subclases:			
Responsabilidades	Colaboradores	Pensamiento en objetos	Propiedad
Agregar una nueva asignación	Curso	Conozco mi número de asignación	Número de tarea
Cambiar una asignación		Conozco mi descripción	Descripción de tarea
Ver una asignación		Sé cuántos puntos valgo	Puntos
		Sé cuando me deben entregar	Fecha de entrega