

INTRODUCCIÓN AL PARADIGMA ORIENTADO A OBJETOS

Un **paradigma** es una forma de entender y representar la realidad: un conjunto de teorías, estándares y métodos que, juntos, representan un modo de organizar el pensamiento.

El **enfoque tradicional** del análisis y diseño estructurados, se descompone el problema en funciones o procesos y estructuras de datos.

En un **enfoque OO** se busca descomponer el problema, no en funciones, sino en unidades más pequeñas denominadas objetos.

El **paradigma OO** define los programas en términos de comunidades de objetos. Los objetos con características comunes se agrupan en clases.

El **diseño orientado a objetos** es fundamentalmente diferente a los enfoques de **diseño estructurado** tradicionales: requiere un modo distinto de pensar acerca de la descomposición, y produce arquitecturas de software muy alejadas del dominio de la cultura del diseño estructurado.

La **tecnología orientada a objetos** se apoya en los sólidos fundamentos de la ingeniería, cuyos elementos reciben el nombre global de modelo de objetos. El modelo de objetos abarca los principios de abstracción, encapsulación, modularidad, jerarquía, tipos, concurrencia y persistencia.

Ninguno de estos principios es nuevo por sí mismo. Lo importante del modelo de objetos es el hecho de conjugar todos estos elementos de forma sinérgica.

METODOLOGÍA ORIENTADA A OBJETOS

• Análisis orientado a objetos

Es un método de análisis que examina los requisitos desde la perspectiva de las clases y objetos que se encuentran en el vocabulario del dominio del problema.

• Diseño orientado a objetos

Es un método de diseño que abarca el proceso de descomposición orientada a objetos y una notación para describir los modelos lógico y físico, así como los modelos estático y dinámico.

• Programación orientada a objetos

Es un método de implementación en que los programas se organizan como colecciones cooperativas de objetos, cada uno de los cuales representa una instancia de alguna clase, y cuyas clases son, todas ellas, miembros de una jerarquía de clases unidas mediante relaciones de herencia.

BENEFICIOS DEL ENFOQUE OO

La diferencia explicada en el apartado anterior trae los siguientes beneficios de este enfoque:

- Disminución del bache semántico entre análisis y diseño proveyendo una representación consistente en todo el ciclo de vida
- Disminución del bache semántico entre análisis y diseño proveyendo una representación consistente en todo el ciclo de vida
- La transición del análisis al diseño es un refinamiento

ANÁLISIS VS DISEÑO

- Durante el análisis OO se presta especial atención a encontrar y describir los objetos (conceptos) del dominio del problema
- Durante el diseño OO se presta atención a la definición de los objetos software y en como colaboran para satisfacer los requisitos

ANÁLISIS OO

- La finalidad del análisis OO es crear una descripción del dominio desde una perspectiva de clasificación de objetos: identificación de conceptos, atributos e interrelaciones significativas

- El modelo del dominio NO es una descripción de los objetos software, es una visualización de los conceptos del mundo real y sus vinculaciones (se representan mediante diagrama de clases, sin operaciones)

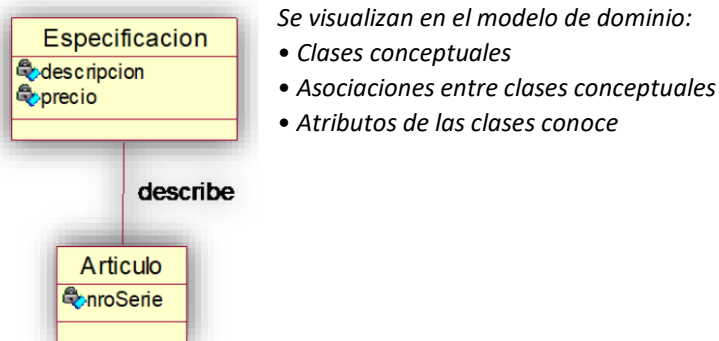
Clases conceptuales

- Una clase conceptual se puede considerar en términos de:
- Símbolo: palabras o imágenes que representan la clase conceptual
- Intensión: la definición de la clase conceptual
- Extensión: el conjunto de ejemplos a los que se aplica la clase conceptual

Modelo de Dominio

Análisis = descomposición de un dominio de interés en clases conceptuales

Modelo del dominio = representación visual de las clases conceptuales del mundo real



Clase

Definición 1: Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, etc.)

Definición 2: Una clase es un conjunto de objetos que comparten una estructura y un comportamiento comunes.

Definición 3: Una clase especifica una estructura de datos y las operaciones permisibles que se aplican a cada uno de sus objetos.

Los objetos se vinculan mediante enlaces enviando mensajes a operaciones que activan los métodos:

- Mensaje: es una solicitud para que se lleve a cabo la operación indicada y se produzca el resultado.
- Operaciones: es una función o transformación que se aplica a un objeto de una clase
- Métodos: es la implementación de una operación

Relaciones entre clases

Asociación

La asociación es una relación bidireccional. La relación entre clases conocida como Asociación, permite asociar objetos que colaboran entre sí. Cabe destacar que no es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.

Ej: dada una instancia de cliente podríamos encontrar el objeto que denota sus compras.

Posee CARDINALIDAD y esta puede ser:

- UNO A UNO
- UNO A MUCHOS
- MUCHOS A MUCHOS

Herencia

Capacidad por la cual una clase de orden inferior puede recibir estructura o acciones de una o más clases de orden superior. La subclase posee la capacidad de incorporar parte estructural y acciones propias.

Existen dos tipos de herencia:

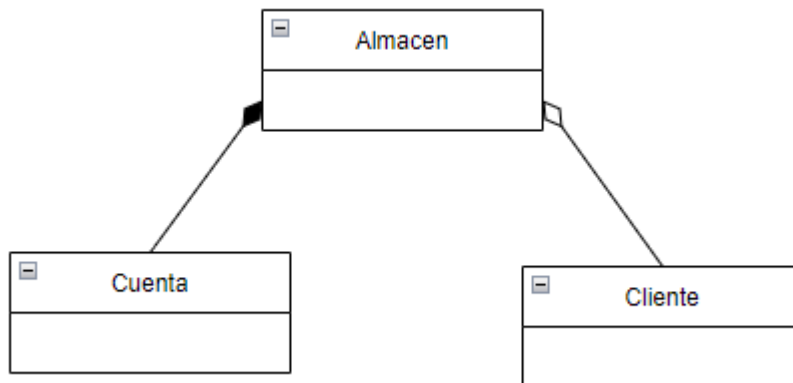
- Herencia Simple
- Herencia Múltiple
-

Polimorfismo

Capacidad por la cual una acción puede responder de distinta forma de acuerdo a la subclase que la implementa.

Agregación

La instancia se crea afuera, pero la agregación se hace dentro. Ej: Cuando se destruye el almacén, no se destruyen los clientes



Composición

También llamada “agregación con contención física”. La creación de la instancia se hace dentro. Cuando un objeto no existe sin el otro. Sus ciclos de vida están íntimamente relacionados. Ej: Cuando se destruye el almacén, también las cuentas

Uso

Es una asociación refinada, donde se establece que abstracción es cliente y cual servidor. El cliente hace USO del servidor

Diagrama de Clases

Es muy importante interpretar cada uno de los elementos del diagrama de clases, ya que esto va a ser la estructura del funcionamiento del sistema:

- Navegabilidad: describe la relación, que clase contiene el atributo
- Multiplicidad: describe la relación según la cantidad de instancias

Un diagrama de clases sirve para mostrar las relaciones entre las clases involucradas en el sistema. Está compuesto por:

- Clases
 - Atributos
 - Métodos
 - Visibilidad (público, privado, protegido, etc)
- Relaciones (Estructurales) y multiplicidad

OBJETOS

“Un objeto es cualquier cosa real o abstracta, acerca de la cual almacenamos datos y las operaciones que controlan dichos datos”

Se opone al análisis estructurado donde los datos y el comportamiento están débilmente relacionados. Los objetos son entidades que combinan un estado (es decir, datos) y un comportamiento (esto es, procedimientos o métodos). Estos objetos se comunican entre ellos para realizar tareas”.

Propiedades de los Objetos

El **estado** de un objeto abarca todas las propiedades (normalmente estáticas) del mismo, más los valores actuales (normalmente dinámicos) de cada una de esas propiedades.

El **comportamiento** nos muestra cómo actúa y reacciona un objeto, en términos de sus cambios de estado y paso de mensajes

La **identidad** es aquella propiedad de un objeto que lo distingue de todos los demás objetos

Abstracción

La abstracción es la propiedad que permite representar las características esenciales de un objeto, sin preocuparse de las restantes características (no esenciales).

Desde el punto de vista **Orientado a Objetos**, la abstracción:

- Expresa las características esenciales de un objeto, la cual distingue al objeto de los demás
- Provee límites conceptuales

Encapsulamiento

El Encapsulamiento es la propiedad que permite asegurar que el contenido de la información de un objeto está oculta al mundo exterior.

El encapsulamiento, al separar el comportamiento del objeto de su implantación, permite la modificación de éste sin que se tengan que modificar las aplicaciones que lo utilizan.

Si un objeto tiene más características de las necesarias los mismos resultarán difíciles de usar, modificar, construir y comprender.

¿QUÉ OBSERVAR PARA SABER SI UNA CLASE O UN OBJETO ESTÁN CONSTRUIDOS CON CALIDAD?

- **Acoplamiento**
 - Medida de la fuerza de la asociación establecida por una conexión entre dos objetos
- **Cohesión**
 - Establece el grado de conectividad interna. Indica que tan específico es un objeto.
- **Suficiencia**
 - Indica si se poseen las características necesarias de la abstracción como para permitir una interacción significativa y eficiente
- **Complejión**
 - Indica si la interfaz de la clase captura todas las características de la abstracción
- **Primitivas**
 - Denotan operaciones sencillas que se pueden acceder solo por medio de la representación interna básica de la abstracción

Referencias: Larman, Craig. **UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado**. 2a.ed.-- Madrid: Pearson Educación