**Programación Orientada a Objetos (Conceptos)**

**Objetos:** Entidad que tiene estado y comportamiento. El Estado de un objeto se almacena en sus atributos, y el comportamiento de un objeto se define en sus métodos.

**Clases:** Una clase es una plantilla para crear objetos. Una clase define los atributos y métodos de un objeto.

Una clase es un tipo de datos que define un conjunto de atributos y métodos. Los atributos son variables que se asocian con una clase, y los métodos son funciones que se asocian con una clase.

**Herencia:** La herencia es la capacidad de una clase para heredar los atributos y métodos de otra clase. Esto permite crear clases mas complejas a partir de clases más simples.

**Polimorfismo:** El polimorfismo es la capacidad de una función o método para tener diferentes significados dependiendo del tipo de objeto en el que se llama. Esto permite crear código más general y flexible.

**Encapsulamiento:** Es la practica de ocultar los detalles de implementación de un objeto de sus usuarios. Esto hace que los objetos sean más fáciles de usar y más fáciles de mantener.

**Notas de pdf [Tema1: ¿Que es el análisis orientado a objetos?]**

**Tema 1. ¿Qué es el análisis orientado a objetos?**

* 1. **Concepto de análisis:**

Es el estudio de una situación específica, usando algunas técnicas para determinar la frontera y características del problema, y proyectar posibles soluciones.

Cuando este análisis es realizado por un informático, se aplican técnicas para establecer cual es el problema que genera una mala administración de la información (problema informático) y cuales son las dimensiones del área afectada (limites)

Al comprender el problema, el informático esta en una mejor posición para proponer alternativas de soluciones a este. Lo mas importante en el análisis **es investigar el problema, entender sus causas y posibles consecuencias.**

Según IEEE, el análisis es: “proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de hardware o de software, así como el proceso de estudio y refinamiento de dichos requisitos”

* 1. **Concepto de Orientación a Objetos:**

Los objetos son todos aquellos elementos o conceptos que pertenecen a la situación que se esta analizando y que tienen una participación fundamental.

**Ejemplo:**En la situación **Partido de Futbol**, los objetos que se deben tener en cuenta son:   
Jugadores, balón y arbitro.

La manera en que estos objetos se relacionan e interactúan entre si es importante porque permiten identificar como se da la dinámica del partido.

* 1. **Análisis orientado a objetos**

Plantea el análisis de una situación a través del reconocimiento de los objetos presentes y la manera como estos interactúan entre sí.

Cuando analizamos un sistema bajo la mirada de la orientación a objetos, podemos describirlo tan cual es en el mundo real y comprenderlo más claramente. Esta comprensión nace a partir de la realidad o dominio del problema.

Cuando se trata de explicar un problema complejo, usando herramientas y métodos simples, se dice que hemos aplicado abstracción.

**Otra explicación:**

Cuando analizamos un sistema bajo la mirada de la orientación a objetos, lo dividimos en objetos. Cada objeto representa una parte del sistema y tiene su propio estado y comportamiento. Esto nos permite entender el sistema de una manera más clara y concisa.

La abstracción es el proceso de eliminar los detalles irrelevantes de un problema. Esto nos permite enfocarnos en los aspectos mas importantes del problema y resolverlo de una manera más eficiente.

**Ejemplo:**Si queremos analizar un sistema de gestión de inventario, podemos dividirlos en objetos como productos, clientes y pedidos. Cada objeto tiene su propio estado y comportamiento.  
Por ejemplo, un producto puede tener un nombre, una dirección y un número de teléfono. Un cliente puede tener un nombre, una dirección y un número de teléfono Un pedido puede tener un numero de pedido, una fecha de pedido y una lista de productos.

Al analizar el sistema en términos de objetos, podemos entenderlo de una manera mas clara y concisa. También podemos identificar los aspectos mas importantes del sistema y resolver los problemas de una manera más eficiente.

* 1. **Fases del análisis orientado a objetos**

Para establecer el análisis se hace necesario estudiar al sistema desde dos aspectos estratégicos.

* Análisis de la estructura de objetos
* Análisis del comportamiento de objetos

**Análisis de la estructura de objetos:** Se identifican a todos los tipos y categorías de estos, y como se asocian.

**Ejemplo:** En un colegio podemos encontrar a los alumnos y profesores. Es posible describir las características de los objetos y, además, establecer que los objetos se relacionan de la siguiente manera: el profesor hace clase a los alumnos.

**Análisis del comportamiento de objetos:** Se establece la secuencia de interacción y la manera en que estos van cambiando de estado.

Las fases del análisis orientado a objetos están fuertemente relacionadas y son útiles para crear el modelo de un sistema bien integrado.

**[Tema2: ¿Cómo se configura la orientación a objetos?]**

Primero, es necesario comprender los siguientes conceptos:

**2.1 Clase**

Corresponde a una categorización de varios objetos que poseen características y comportamientos semejantes, además de un mismo objetivo.

**Ejemplo:**

Los relojes, a pesar de que las imágenes nos muestran tres tipos diferentes, todos son categorizados dentro del conjunto **relojes**, cuyo objetivo es común es medir el tiempo.

**2.2 Atributo**

Corresponden a las características que permiten describir a un conjunto de objetos que pertenecen a una misma clase. Los atributos son características estáticas, pues solo representan algún dato que pertenece al objeto. Para los relojes del caso anterior, podemos establecer como características: marca, modelo, tipo, y color, entre otras características.

**2.3 Método**

Corresponde al comportamiento o lo que puede hacer un objeto. Los métodos son las funciones que puede ejecutar la clase que representa a todos los objetos. El método es dinámico, pues son actividades que el objeto puede ejecutar.

En el ejemplo anterior, los métodos de la clase reloj son mostrar y cambiar la hora.

-

El símbolo que utilizaremos para esquematizar una clase será un rectángulo con tres subdivisiones, donde agregaremos su nombre, lista de atributos y lista de métodos.

**2.4 Encapsulamiento**

Corresponde a una propiedad de los objetos a través de la cual pueden contener toda su estructura (atributos y comportamientos) aislada u oculta de los elementos del entorno. De esta manera, protegen su estructura.

**2.5 Polimorfismo**

Corresponde a la característica de algunos objetos de ejecutar la misma tarea(método), pero llevandola a cabo de maneras diferentes. Lo que se debe destacar es que el objetivo final se cumple.

**2.6 Herencia**

Corresponde a la capacidad de subclasificar los objetos que pertenecen a la misma clase para manejar mejor la complejidad.

**Ejemplo1:**

Es posible establecer la clase llamada **persona** en la cual se encuentran todos a quienes considera una empresa. Dentro de este gran conjunto de personas, podemos establecer subclasificaciones para reagruparlas en grupos mas pequeños, pero con características que las diferencian. De esta manera, tenemos dos subgrupos: **empleados y clientes,** que siguen siendo **personas**, pero que poseen atributos y comportamientos que los diferencian. Todas las personas tienen atributos como rut y nombre completo, pero solo los clientes tienen un numero de cliente asignado.

La herencia se basa en la jerarquización de clases

**Ejemplo2:**

Se muestra la herencia de un destacado perfil desde el abuelo (superclase) hacia su hijo y su nieta, que son subclases de la clase abuelo.

Una subclase puede heredar la estructura y comportamiento de su superclase.

**Requerimiento:** condición o capacidad que debe poseer un sistema o un componente de un sistema para satisfacer un contrato, un estándar, una especificación u otro documento, formalmente.

**[Tema3: Técnicas de captación de requerimientos]**

Los requerimientos son el resultado final de la fase de análisis y representan a todas las características que el sistema debe tener a futuro para resolver los problemas o carencias que actualmente aquejan a la empresa u organización.

Los requerimientos deben ser claros, sin dobles interpretaciones, y deben responder a las necesidades de las personas que usaran el sistema.

Descubrir una deficiencia en los requerimientos implica corregir lo que ya se ha avanzado en el proyecto y perder todo lo realizado hasta ese momento, trayendo consecuencias negativas como perdida de tiempo, dinero y esfuerzos ejecutados.

**En otras palabras**

Los requisitos son las características que debe tener un sistema para cumplir con sus objetivos. Son el resultado de la fase de análisis, donde se recopila información sobre las necesidades de los usuarios y se identifica el problema que el sistema debe resolver.

Los requisitos deben ser claros y concisos, y deben estar escritos en un lenguaje que los usuarios puedan entender. También deben ser completos y precisos, y deben cubrir todas las necesidades de los usuarios.

**[Tema4: Clasificación de requerimientos funcionales y no funcionales]**

Dos tipos de requerimientos iniciales conocidos como funcionales y no funcionales.

Para encontrarlos se deben aplicar diversas técnicas y herramientas de investigación para recoger la información que permitirá comprender la realidad y resolver los problemas detectados.

**Google Bard:**

**Requisitos funcionales:**

Los requisitos funcionales son los requisitos que describen lo que el sistema debe hacer. Describen las funciones que el sistema debe cumplir y los resultados que debe producir. Los requisitos funcionales suelen ser especificados por los usuarios del sistema o por los analistas de requisitos.

**Requisitos no funcionales:**

Son los requisitos que describen como debe comportarse el sistema. Describen las cualidades del sistema, como su rendimiento, su seguridad, su usabilidad y su mantenibilidad. Los requisitos no funcionales suelen ser especificador por los arquitectos de sistemas o por los ingenieros de software.

Los requisitos funcionales garantizan que el sistema satisfaga las necesidades de los usuarios, mientras que los requisitos no funcionales garantizan que el sistema sea capaz de cumplir su propósito y cumplir con los requisitos de la organización.

**4.1 Requerimientos funcionales**

Corresponden a las especificaciones de las tareas que el sistema debe realizar para el usuario bajo condiciones específicas. En otras palabras, es lo que el usuario necesita que haga el sistema para hacer su trabajo.

**4.2 Requerimientos no funcionales**

Representan todo el conjunto de características que deben estar presentes para que el sistema pueda trabajar plenamente. Estas características tienen relación directa con aspectos tales como:

* La fiabilidad del software (operación libre de fallos) bajo conficiones establecidas.
* Respuestas adecuadas en el tiempo con alta calidad y excelente rendimiento.
* La capacidad de almacenamiento que necesita el sistema para sus datos.

**4.3 Técnicas de recopilación de requerimientos**

**a) Entrevistas**

**b) Observación**

**c) Encuestas**

**[Tema5: Análisis gramatical de problemas]**

Analizar situaciones es primordial para entender a lo que nos enfrentamos. Las personas (usuarios) a quienes se va a entrevistar, encuestar u observar nos mostraran los escenarios que representan la manera en que ellos trabajan.

Luego de investigar y profundizar en la situación actual, se establece el origen de los problemas y se determinan los requerimientos para solucionarlos. Todo lo anterior es análisis, mas un poco de sentido común.

**Recordar:**Análisis: “proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de hardware o de software, así como el proceso de estudio y refinamiento de dichos requisitos”