

**Subsecretaría de Educación Superior**



***Dirección General de Educación Superior Tecnológica***

**INSTITUTO TECNOLÓGICO DE SALINA CRUZ**

**CARRERA**

**INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES**

**MATERIA**

**FUNDAMENTOS DE PROGRAMACIÓN**

**TRABAJO QUE PRESENTA**

**FLORES PACHECO LUIS ALBERTO**

**INVESTIGACIÓN**

**CARACTERÍSTICAS DEL MODELO ORIENTADO A OBJETOS Y ELEMENTOS PRIMORDIALES EN EL MODELO DE OBJETOS.**

**NÚMERO DE CONTROL:**

**121020158**

**GRADO**

**I SEMESTRE GRUPO E-1**

**PROFESOR**

**MC. SUSANA MÓNICA ROMAN NAJERA**

**SALINA CRUZ, OAXACA., A OCTUBRE DE 2012**

Índice

[Introducción 1](#_Toc338886460)

[El modelo Orientado a Objetos 2](#_Toc338886461)

[Objetos 2](#_Toc338886462)

[Clase 2](#_Toc338886463)

[Herencia 3](#_Toc338886464)

[Elementos del modelado orientados objetos 3](#_Toc338886465)

[Abstracción 3](#_Toc338886466)

[Modularidad 4](#_Toc338886467)

[Encapsulación 4](#_Toc338886468)

[Jerarquización 5](#_Toc338886469)

[Conclusiones 6](#_Toc338886470)

[Referencias bibliográficas 7](#_Toc338886471)

# Introducción

Los sistemas basados en modelos de datos orientados a objeto fueron inspirados a partir del paradigma de programación orientada a objeto. Entre los primeros lenguajes orientados a objetos podemos citar Smalltalk y Simula. Ejemplos más recientes son C++, CIOS, CLU y JAVA [Berard 96].

La programación orientada a objetos es una de las formas más populares de programar y viene teniendo un gran desarrollo de proyectos de software desde los últimos años. Este desarrollo se debe a sus grandes capacidades y ventajas frente a las antiguas formas de programar.

La presente investigación tiene como objetivo, proporcionar la información adecuada y pertinente, correspondiente a la cuarta unidad de la materia “Fundamentos de Programación”, donde se da a conocer dos de puntos de la cuarta unidad “Programación orientada a objetos y modelado”.

Es importante esta investigación, pues esto permitirá al lector entender el modelado orientado a objetos, sus características y sus elementos primordiales, analizando cada uno de sus partes. Esta recopilación fue desarrollada mediante la consulta de páginas web, blogs especializados y en algunas fuentes bibliográficas analizando el contenido de cada uno de ellas.

En el subtema 4.1 Características del modelo orientado a objetos, analizaremos cada uno de sus partes, definiremos lo que es objeto, clase y herencia. En el subtema 4.2 Elementos primordiales en el modelo de objetos, explicaremos cada uno de los elementos que conforman el modelado a objetos.

Ojala que esta investigación cumpla con los requerimientos establecidos por la profesora para la presentación de la investigación y queda abierto para críticas y aportaciones por parte de mis compañeros y la profesora.

# El modelo Orientado a Objetos

## Objetos

Un objeto es una entidad, que posee un estado, comportamiento e identidad.

* Estado: abarca todas las propiedades del objeto.
* Comportamiento: consiste en la forma de actuar y reaccionar un objeto.
* Identidad: son las propiedades de los objetos que los distinguen de los demás.

Por ejemplo, los automóviles tienen características (marca, modelo, color, velocidad máxima, etc.) y comportamiento (frenar, acelerar, retroceder, llenar combustible, cambiar llantas, etc.).

Los Objetos de Software, al igual que los objetos del mundo real, también tienen características y comportamientos. Un objeto de software mantiene sus características en una o más "variables", e implementa su comportamiento con "métodos". Un método es una función o subrutina asociada a un objeto.

## Clase

La clase es un modelo o prototipo que define las variables y métodos comunes a todos los objetos de cierta clase. También se puede decir que una clase es una plantilla genérica para un conjunto de objetos de similares características.

En el mundo real, normalmente tenemos muchos objetos del mismo tipo. Por ejemplo, nuestro teléfono celular es sólo uno de los miles que hay en el mundo. Si hablamos en términos de la programación orientada a objetos, podemos decir que nuestro objeto celular es una instancia de una clase conocida como "celular". Los celulares tienen características (marca, modelo, sistema operativo, pantalla, teclado, etc.) y comportamientos (hacer y recibir llamadas, enviar mensajes multimedia, transmisión de datos, etc.).

## Herencia

La herencia es uno de los conceptos más cruciales en la POO. La herencia básicamente consiste en que una clase puede heredar sus variables y métodos a varias subclases (la clase que hereda es llamada superclase o clase padre). Esto significa que una subclase, aparte de los atributos y métodos propios, tiene incorporados los atributos y métodos heredados de la superclase. De esta manera se crea una jerarquía de herencia.

# Elementos del modelado orientados objetos

## Abstracción

La abstracción consiste en captar las características esenciales de un objeto, así como su comportamiento.

Denota las características esenciales que distinguen a un objeto de otros tipos de objetos, definiendo precisas fronteras conceptuales, relativas al observador.

* Surge del reconocimiento de similaridades entre ciertos objetos, situaciones o procesos en el mundo real.
* Decide concentrarse en estas similaridades e ignorar las diferencias.
* Enfatiza detalles con significado para el usuario, suprimiendo aquellos detalles que, por el momento, son irrelevantes o distraen de lo esencial.
* Deben seguir el "principio de mínimo compromiso", que significa que la interface de un objeto provee su comportamiento esencial, y nada más que eso. Pero también el "principio de mínimo asombro": capturar el comportamiento sin ofrecer sorpresas o efectos laterales.

## Modularidad

Es la propiedad que tiene un sistema después de ser descompuesto en módulos con alta cohesión y acoplamiento.

* Cada módulo se puede compilar separadamente, aunque tengan conexiones con otros módulos.
* En un diseño estructural, modularización comprende el agrupamiento significativo de subprogramas. En diseño orientado a objetos, la modularización debe ceñirse a la estructura lógica elegida en el proceso de diseño.
* Dividir un programa en componentes individualizados reduce en alguna manera su complejidad.
* **C++** Se separan los módulos interface de los módulos con implementación, estando los primeros en ficheros con extensión .h llamados *header files*, mientras que los segundos se almacenan en ficheros con extensión .c, .cc, .cp o .cpp. La dependencia entre ficheros se realiza a través de la macro #include.

## Encapsulación

Es el proceso de compartimentalización de los elementos de una abstracción que constituyen su estructura y comportamiento. La encapsulación sirve para separar la interface de una abstracción y su implementación.

* Es un concepto complementario al de abstracción.
* La encapsulación esconde la implementación del objeto que no contribuye a sus características esenciales.
* La encapsulación da lugar a que las clases se dividan en dos partes:
  1. Interface: captura la visión externa de una clase, abarcando la abstracción del comportamiento común a los ejemplos de esa clase.
  2. Implementación: comprende la representación de la abstracción, así como los mecanismos que conducen al comportamiento deseado.

Se conoce también como ocultamiento o privacidad de la información.

## Jerarquización

Es una clasificación u ordenación de las abstracciones.

* Por jerarquía denotamos el orden de relación que se produce entre abstracciones diferentes.
* Los tipos de jerarquía más útiles:
  1. **Herencia** (generalización/especialización, padre/hijo, jerarquía del tipo "es un"...). Una clase (subclase) comparte la estructura o comportamiento definido en otra clase, llamada superclase.
  2. **Herencia múltiple** Una clase comparte la estructura o comportamiento de varias superclases.
  3. **Agregación** Comprende relaciones del tipo "es parte de" al realizar una descomposición.

# Conclusiones

El reporte de investigación en esta unidad es de gran importancia ya que adquirimos nuevos conocimientos sobre el modelado orientados a objetos, pues esto nos servirá como futuros Ingenieros en Tecnologías de la Información y de las Comunicaciones, por que es de gran importancia tener un conocimiento sobre las características del modelado a objetos así como sus características.

Como vimos, el modelado orientado a objetos se define como una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación.

Con la POO tenemos que aprender a pensar las cosas de una manera distinta, para escribir nuestros programas en términos de objetos, propiedades, y métodos.

A través de esta investigación hemos adquirido una nueva experiencia que emplearemos como futuros ITIC’s, conociendo la manera de como realizar las cosas de manera que otras personas puedan utilizarlas y adelantar su trabajo, de manera que consigamos que el código se pueda reutilizar.

Toda la información recaba en el presente trabajo será de gran utilizad para el ingeniero en TIC´s, pues como ingenieros debemos de estar analizando los problemas que se nos puedan presentar.

# Referencias bibliográficas

Álvarez Pol Héctor. El modelo de objetos. Internet. En línea. Página consultada el día 24 de octubre de 2012. Disponible en: <http://fpsalmon.usc.es/genp/doc/cursos/poo/modelo.html#MOD>

Ciberaula. Programación orientada a objetos. Internet. En línea. Página consultada el día 24 de octubre de 2012. Disponible en: <http://www.ciberaula.com/articulo/tecnologia_orientada_objetos/>

Universidad Autónoma Metropolitana. El modelo orientado a objetos. Internet. Fuera de línea. Página consultada el día 24 de octubre de 2012. Disponible en: <http://uamisoft.izt.uam.mx/uamisoft/lib/exe/fetch.php?media=amm:cursos:psu:07-o:el_modelo_oo.pdf>