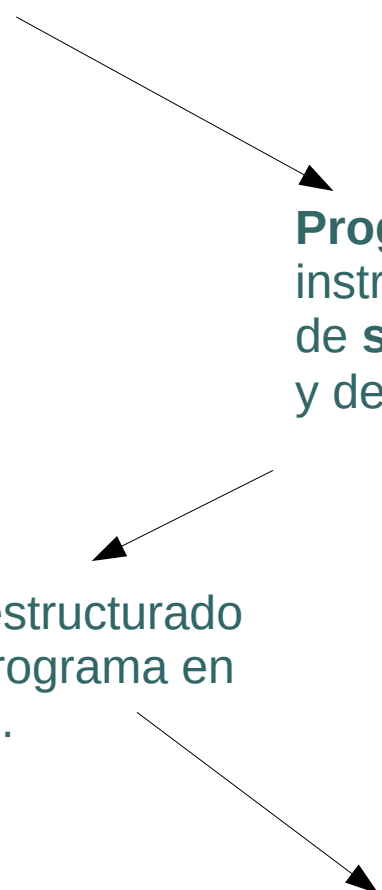


Programación Orientada a Objetos

¿¿¿Qué es un **programa** ???
Secuencia de instrucciones



Programación estructurada:
instrucciones **secuenciales**,
de **selección** (`if then else`, `switch case`)
y de **repetición** (`for`, `while`, `do while`).

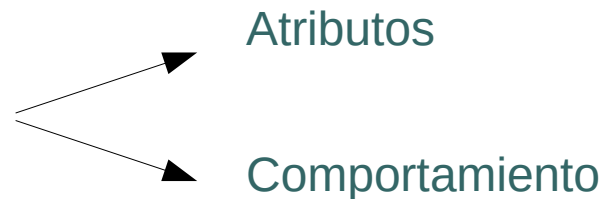
El diseño de un programa estructurado
se consigue dividiendo el programa en
subprogramas o **funciones**.

Las funciones utilizan datos globales
y locales y los procesan.

¿Cómo se modela el mundo real?

Muchas veces, la separación de datos y funciones que manipulan dichos datos resultan en que modelar objetos del mundo real sea difícil.

En el mundo real se tratan con objetos tales como personas, casas o motocicletas



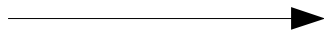
Atributos



DATOS

(tienen un valor determinado)
Persona = nombre (Nacha Guevara)
color de ojos (castaño)
estatura (1,83m)

Comportamiento
(acciones)



Funciones
(métodos)

Esta es la idea de TAD, ya que en un TAD se describe tanto los atributos de un objeto como así también su comportamiento (operaciones o funciones).

Pero la POO (Programación Orientada a Objetos) va mucho más allá...

La idea fundamental de la POO es combinar en una sola entidad tanto los datos como las funciones que actúan sobre los datos.

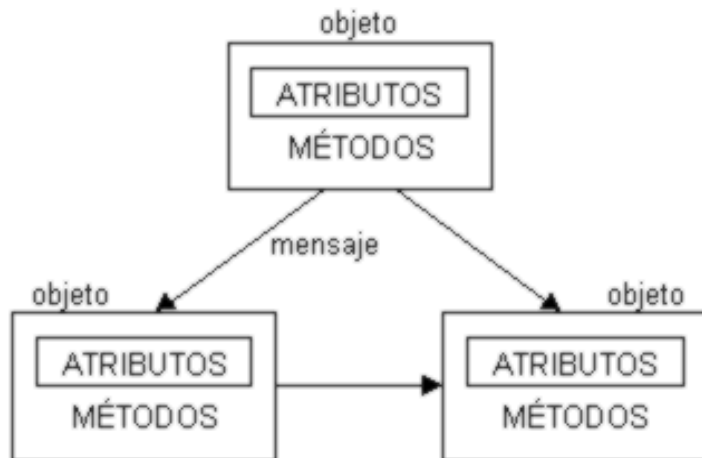
Clase: es una implementación de un tipo abstracto de datos (TAD) y describe no sólo los atributos de un objeto sino también sus operaciones.

Ejemplo: la clase Auto tiene como atributos marca, modelo, color, cantidad de puertas y se puede arrancar(), acelerar(), frenar().

Objeto: es una instanciación de una clase. Es una variable de una clase particular.

Ejemplo: variable miAuto es de tipo Auto. Por esto, miAuto tiene marca, modelo, color, cantidad de puertas determinadas y se puede arrancar(), acelerar(), frenar().

Organización típica de un programa orientado a objetos



Objeto:

Funciones miembro o métodos
Atributos o variables de instancia (datos)

Envío de un mensaje a un objeto:

Llamada a la función miembro o método
(mensaje)



Conceptos importantes relacionados a la POO:

- **Abstracción:** se enfoca en la visión externa de un objeto, separa el comportamiento específico de un objeto.

Ejemplo: puedo usar un televisor desde el panel o desde el control remoto. No sé cómo se obtiene la imagen, etc pero lo uso igual. Conozco la interfaz, no su funcionamiento. ¿Qué pasa si cambia el funcionamiento del televisor por dentro? Nada!! (siempre y cuando no cambie la interfaz). Veo las clases como una “caja negra”, importa qué hace la clase, pero no cómo.

- **Encapsulación o encapsulamiento:** reunión de todos los elementos que se pueden considerar pertenecientes a una misma entidad (a un cierto nivel de abstracción). Agrupamiento de datos y operaciones relacionadas bajo una misma entidad de programación.
- **Herencia:** el concepto de clase conduce al concepto de herencia. Usamos clases y sub clases. La clase animal se divide en mamíferos, anfibios y los otros. Las subclases comparten (heredan) las características de la clase de la que derivan.
- **Reusabilidad o reutilización:** una vez creada, escrita y depurada una clase, se puede poner a disposición de otros programadores. Además, la herencia permite reutilizar código, ya que la clase derivada le puedo agregar atributos o comportamiento, a partir de la clase base.
- **Polimorfismo:** una operación (función) puede tener el mismo nombre para distintas clases, actuando de modo distinto para cada clase.

- Clase Pila
 - ¿Atributos?
 - ¿Métodos?

- Clase Cola
 - ¿Atributos?
 - ¿Métodos?

- Clase Persona
 - ¿Atributos?
 - ¿Métodos?