# Programación orientada a objetos

Par atener un objeto debo tener:

* Atributos(características)
* Tangible
* Comportamiento “Puede realizar funciones”

La manera de describir un objeto es mediante sus atributos y su comportamiento. Se comprenden mediante la observación, e intentar entender las funciones.

Con varios objetos se pueden analizar para encontrar una relación entre ellos.

Pasos para interactuar con objetos:

1. Identifico los objetos (análisis)
2. Identifico las características/funciones de los objetos (diseñar)
3. Identifico la relación entre los objetos (construir)

Para que algo sea un atributo debe ser una variable (ejemplo de las puertas). Se les puede asignar un valor. Si no es un sub-objeto (que forma parte de una composición).

Paradigma

Es una forma en que se interpretan las teorías y metodologías sobre un aspecto u objeto particular científico.

🡪 Una forma de resolver problemas mediante le desarrollo de software

1. Un paradigma puede ser considerado como un estilo fundamental de programación.
2. Es una forma de construir estructuras y elementos de los programas
3. Es como el lenguaje ve al mundo (de que manera resuelve el problema. Si es orientado a objetos, trabaja con objetos)

## Tipos de paradigmas

* Funcional (Utiliza funciones) (Lisp, Scheme, F#)
* Imperativo (Utiliza estructuras) (C, C++, PHP)
* Lógico (a través de reglas) (Prolog)
* Orientado a objetos (Utiliza objetos, y las relaciones entre ellos) (Java, C++, SmallTalk, C#)

El paradigma orientado a objetos permite analizar, diseñar y construir programas mediante objetos. Identificando, caracterizando y relacionando.

## Conceptos relacionados

* Abstracción

Representación no exacta de la realidad. (La forma en que se transmite la realidad subjetiva) (Capacidad del humano para crear representaciones de la realidad)

Se omiten detalles. Para dar énfasis a lo importante. Y además disminuir la complejidad del problema.

Crear objetos acordes a la percepción de la realidad, que poseen las características abstraídas.

* Encapsulamiento (Encapsulación)

En la POO se pueden ocultar los atributos y métodos de un objeto. Este grupo va a representar un espacio en memoria.

Es la propiedad de ocultar la información de un objeto o “Estas funcionalidades mías solo yo las puedo usar”. (Ejemplo del microondas)

El objetivo del encapsulamiento es simplificar. Dar acceso restringido a los métodos de un proceso.

* Clases

Es un molde/plantilla para definir los atributos y los métodos (el comportamiento que va a tener un objeto). Pero la clase no me va a definir el objeto. Solo nos da la información de que sea “cupcake”, no sus características.

Con el mismo molde podemos crear distintos tipos de “cupcake”. No nos comemos el molde.

Se crea hasta tener abstraídos los atributos y los métodos de los objetos que necesitamos.

* Atributos

Cada uno es independiente al objeto que se aplica. Son aquellas características que diferencian un objeto de otro.

* Métodos

Son las operaciones que nos permiten cambiar o modificar los atributos de una clase. O bien para interactuar con estos atributos.

Esto es declarar la interfaz de un objeto (son aquellos métodos que otros objetos pueden utilizar para interactuar con el objeto).

La interfaz se puede establecer por medio de comandos. Estos serían métodos públicos.

* Instancia

“Hacer el cupcake”.

Instanciación🡪 Proceso mediante el cual creo objetos o instancias a través de las clases que he definido. No se puede crear un objeto si no hay una clase.

Básicamente objeto = instancia

El cambio en una instancia no va a afectar a otras instancias (a no ser que explícitamente se programe de esa forma).

* Herencia

Son los atributos y los métodos que crean una relación (provenientes) con una clase padre. Pero se hacen modi-

ficaciones para suplir mis

necesidades específicas.

Clase base/Superclase

Subclase/ Clase hijo

No es buena idea crear instancias desde clases abstractas, es decir no crear “pajaros”, sino “pingüinos” o “loras”.

Permite reutilizar código, pues hay métodos que se van a encontrar en común. Y solo se modifican aquellos métodos que se necesiten cambiar.

* Polimorfismo

Tratar a todos los objetos de una misma clase como uno solo. Pues todos van a tener un comportamiento en común. “la acción comer, en los animales”. Esto puede darse justamente por la existencia de una herencia.

Capacidad de tratar objetos distintos como uno mismo.

Es la suma de herencia + sobreescritura.

* Sobreescritura

Cambiar el comportamiento heredado de una clase padre. (@overwrite)

* Sobrecarga

Métodos con el mismo nombre pueden ser identificados (diferir) en la cantidad o tipo de parámetros.

* Modularidad

Modulo 🡪 un agrupamiento lógico de abstracciones que yo realizo. Es decir, un agrupamiento de clases.

Se puede ver en la existencia de varias clases, por lo tanto, la existencia de varios archivos.

“Librería”

Se espera que las clases dentro de un modulo estén altamente relacionadas.

(package)

Abstracto

“Mostrar su propia representación”