



ESCUELA DE  
INGENIERÍA EN CIENCIAS Y SISTEMAS  
FACULTAD DE INGENIERÍA  
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



**Día, Fecha:**


Miércoles, 15/02/2023

**Hora de inicio:**

07:10

# Introducción a la programación y computación 1 [D]

*José Eduardo Morales García*

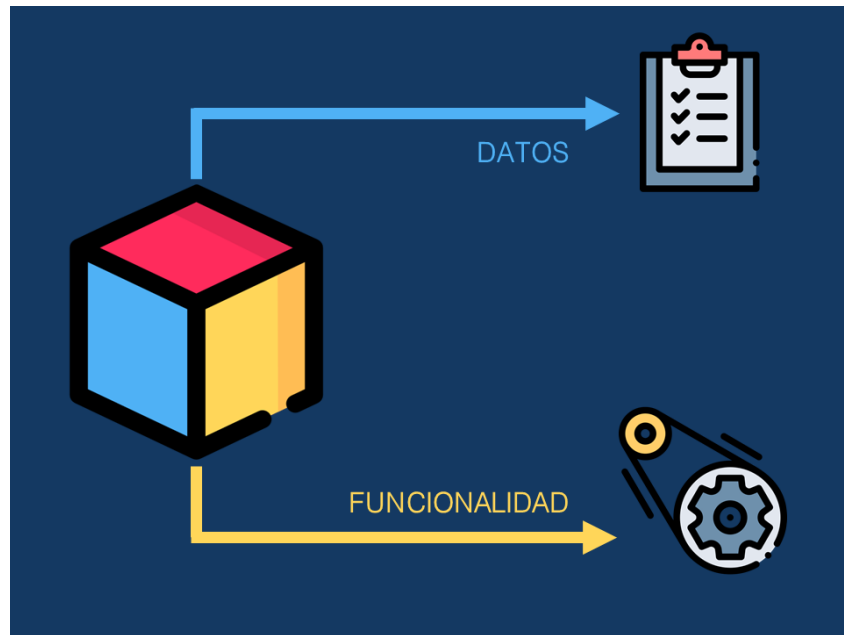


# Introducción a la Programación y Computación 1

# 1. Introducción

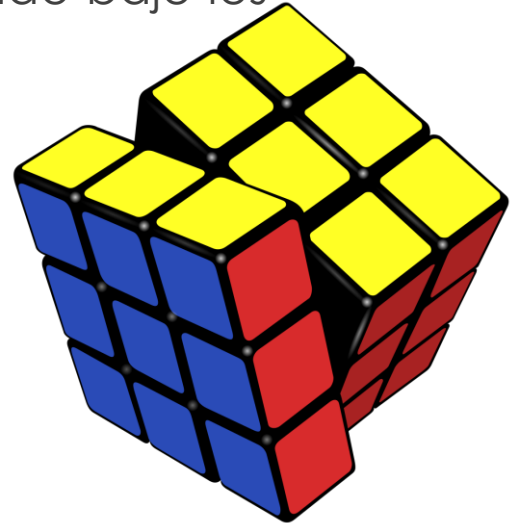
Dentro de cualquier lenguaje de programación se encuentran ligados nuestros paradigmas, entre ellos el paradigma de programación POO, el cual nos ayuda a definir una estructura real y estable del programa.

## 2. POO



## 2. POO

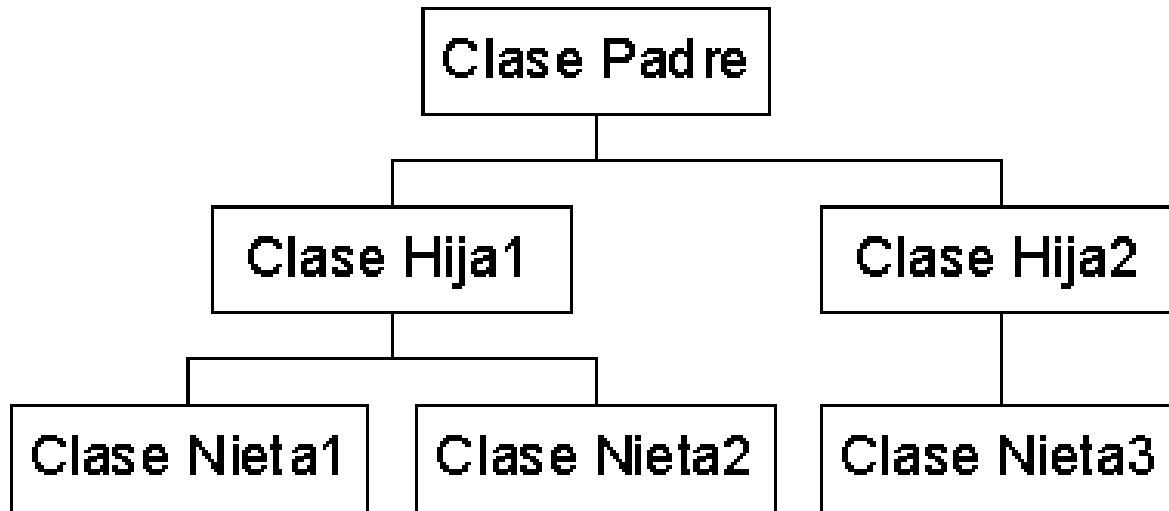
Es un paradigma de programación, se basa bajo un modelo para poder trabajar con el, esta basado bajo los conceptos de clase y objeto.



## 2. POO

```
public class perrito {  
  
    public String nombre;  
    public int anios;  
    public Boolean sano;  
  
    public perrito(){  
        this.nombre = "";  
        this.anios = -1;  
        this.sano = false;  
    }  
  
    public String ladrar()  
    {  
        return "woof! woof";  
    }  
}
```

### 3. Pilares de la programación



### 3. Encapsulamiento

El término encapsulamiento en Java, consiste en ocultar atributos de un objeto de manera que solo se pueda cambiar mediante operaciones definidas en ese objeto. Está estrechamente relacionado con la visibilidad.



# Visibilidad

- ▶ public: nos indica que es accesible desde cualquier clase [interface].
- ▶ private: solo es accesible desde la clase actual.
- ▶ protected: accesible desde la clase actual, sus descendientes o el paquete del que forma parte.
- ▶ sin ninguna palabra: accesible desde cualquier clase del paquete.

# Forma de acceder a atributos privados

- ▶ Los atributos en Java, deben tener visibilidad `private`. Entonces, ¿cómo accedemos a los atributos? La respuesta está en los métodos Getters y Setters.

### 3. Abstracción

En determinada definición, la abstracción es obtener los elementos relacionados con un objeto en la naturaleza para simplificar de determinada forma su estructura.

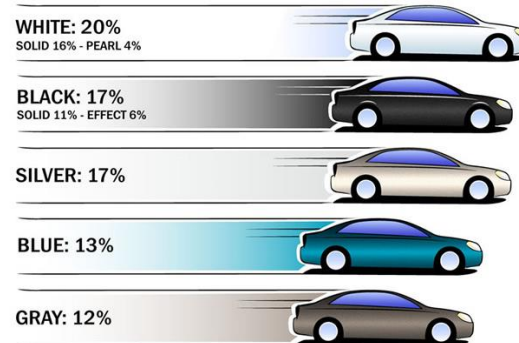
### 3. Herencia

Dentro de su concepto, herencia esta relacionada con todas las subpartes conforman una en especifico, desde un punto de vista jerárquico, entre las cuales comparten atributos.



### 3. Polimorfismo

Abarca todo lo relacionado entre sus partes, como ambas comparten ciertas funcionalidades que se implementan para poder realizar ciertas operaciones, pero de diferente manera.



# Contenido

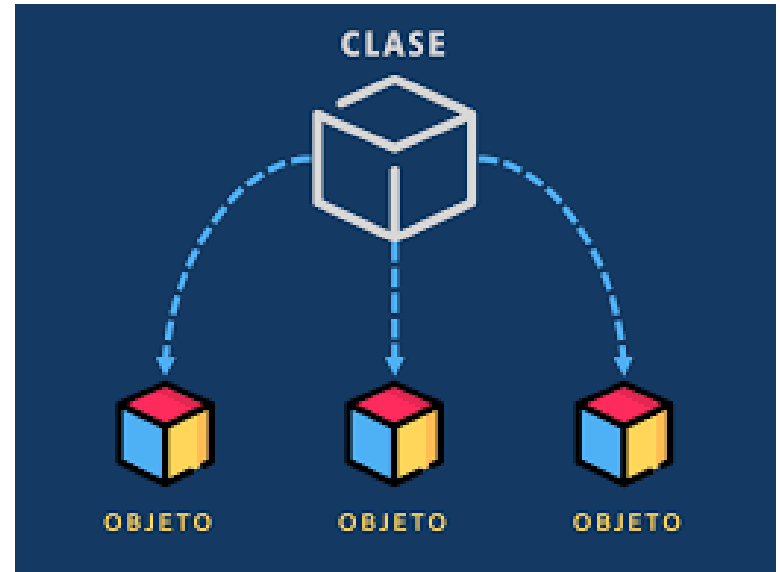
1. Introducción
2. POO
3. 4 Pilares de POO
4. **Datos abstractos (Clases)**

## 4. Clase

```
if(item_Event == "TDH_EVENT_FOLDER"):
    #find RICname and opaqueV to compose to be a
    for i in searchlines:
        if "<Name value=" in i and flagCheckRicna
            print("yoyo2",i)
            ricName = i
            flagCheckRicname = 0
            print ("flagCheckRicname_TDH: ", flagC
            #Find out the RICname.
            searchObj1 = re.search( r'"(.*)"', ric
            if searchObj1: RDHTDHRicName = searchOb
            print ("TDH: Name = ", RDHTDHRicName)
```

## 4. Clases

Es un esquema de programación, que define un objeto, dentro del cual puede abarcar características y operaciones especiales, que diferencian del resto.





## 4. Objeto

Representa un elemento, dentro de nuestra aplicación o en un concepto más natural un objeto es una representación de un algo, con objetivo y diferencias.



## 4. Instancia

Representa un objeto, dentro del conjunto de elementos, el cual le da la singularidad de diferencia.



# 4.

```
public class perrito {  
  
    public String nombre;  
    public int anios;  
    public Boolean sano;  
  
    public perrito(){  
        this.nombre = "";  
        this.anios = -1;  
        this.sano = false;  
    }  
  
    public String ladrar()  
    {  
        return "woof! woof";  
    }  
}
```



Parte practica