



El futuro digital  
es de todos

MinTIC

«Misión  
TIC2022»

Fund. de Programación

Grupos 79,80,81

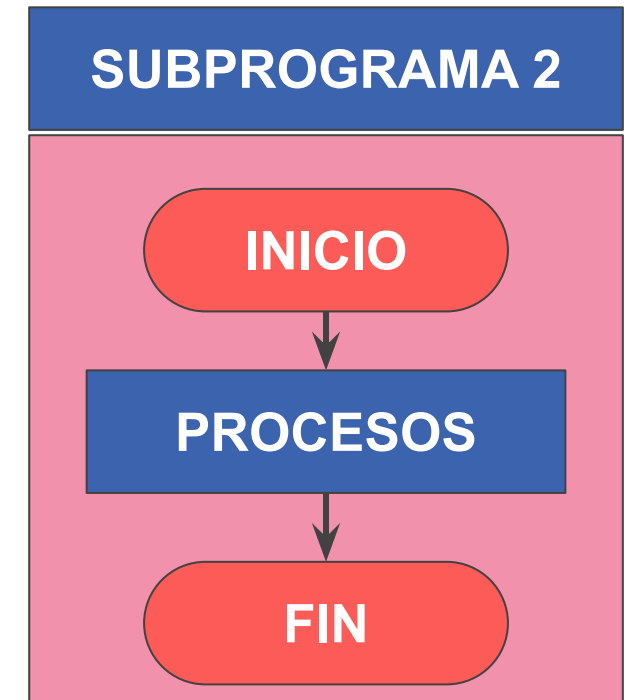
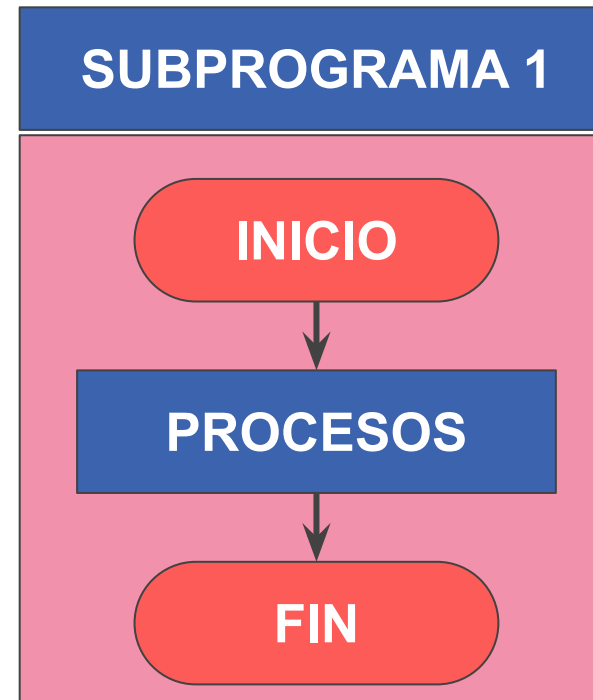
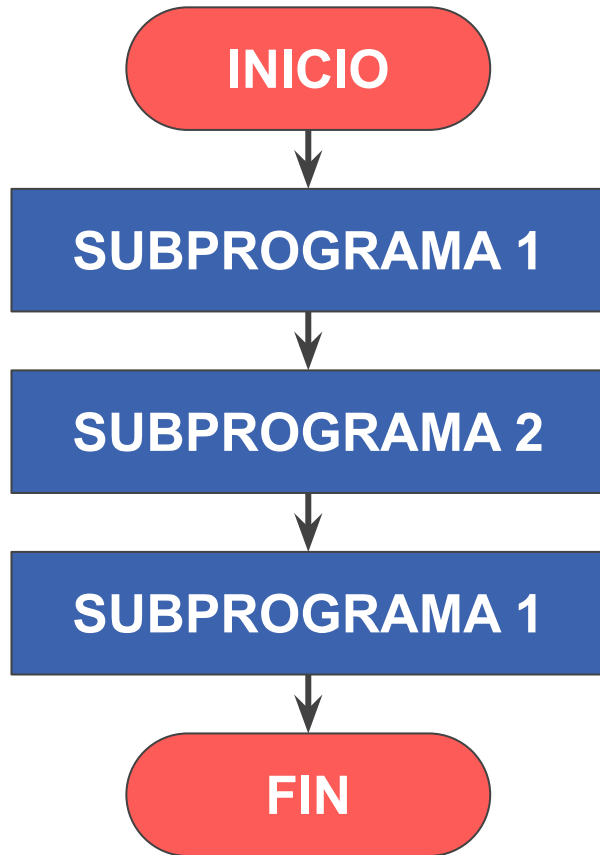


UNIVERSIDAD  
DE ANTIOQUIA  
Facultad de Ingeniería

# Resumen sesión anterior

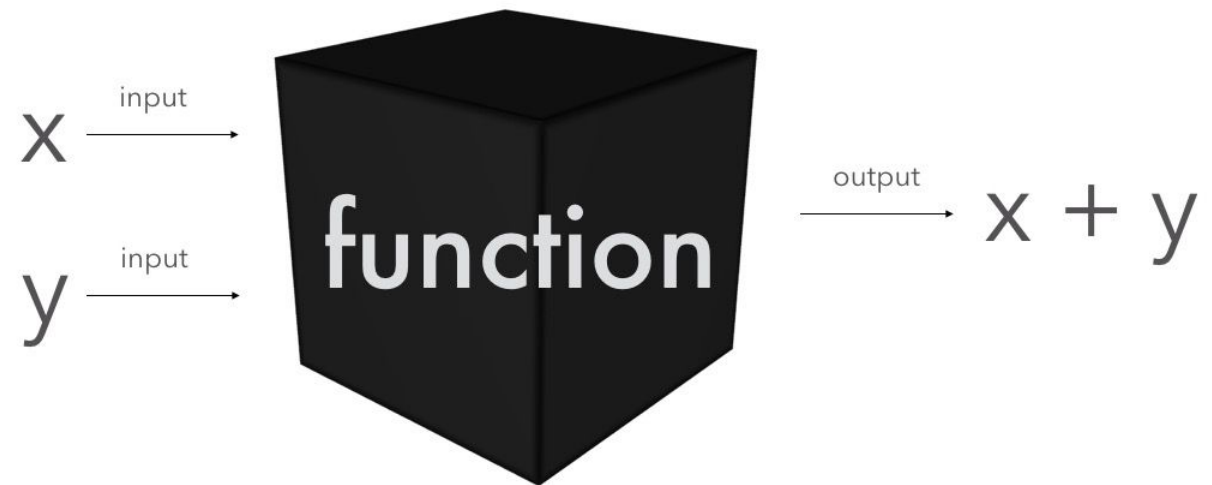


# SUBPROGRAMAS



# ENTRADAS Y SALIDAS

- Las funciones pueden ser alimentadas con parámetros
- Las funciones después de realizar los procesos retorna un conjunto de valores



# Vectores

**JUAN FERNANDO GONZÁLEZ**  
**GRUPOS 79,80,81**  
**Semana 3 Sesión 1**





# VECTORES

“En programación, se le denomina vector, formación, matriz (en inglés array, del cual surge la mala traducción arreglo), a una zona de almacenamiento contiguo que contiene una serie de elementos del mismo tipo”



## ACTIVIDAD 4

Crear un función que tome dos vectores y retorne la suma vectorial de ambos vectores

Ejemplo:

`mi_funcion([0,1,0], [2,1,0])` retorna **[2,2,0]**

# Programación Orientada a Objetos (POO)

**JUAN FERNANDO GONZÁLEZ**  
**GRUPOS 79,80,81**  
**Semana 3 Sesión 2**







# PROGRAMACIÓN ORIENTADA A OBJETOS

Es un paradigma de programación cuyos principales objetivos son:

- Simulación del mundo real
- Programación coherente
- Reutilización de código



## ELEMENTOS DE POO

- **Clases:** Definen la estructura que van a tener los objetos que se creen a partir de ella, indicando que propiedades y métodos tendrán los objetos
- **Atributos:** Definen las características del objeto
- **Métodos:** Definen el comportamiento del objeto

# CLASES

## SUPERCLASE



## CLASE



## CLASE





### Características

- Marca
- Color
- Tipo
- Precio
- No. de Puertas
- Tipo Combustible
- Cilindros
- Transmisión

### Acciones

- Encender
- Avanzar
- Retroceder
- Detener
- Apagar

### Características

- Nombre
- Especie
- Color
- Edad

### Acciones

- Comer
- Dormir
- Correr

# OBJETOS

Un objeto es la **instancia** de una clase, es un elemento particular de la clase definida.

## Clase Taxi

### Atributos:

- **Placa:** n45
- **Color:** Amarillo
- **Marca:** Mercedes

### Métodos:

- **Manejar**
- **Cobrar**





# CLASES EN PYTHON

- **Constructor:** Inicializa cada objeto, en python es el método `__init__(self)`
- **Parámetro self:** Una referencia al objeto en sí mismo
- **Atributos:** Son variables acompañadas del argumento `self` y se definen en el constructor
- **Métodos:** Son “Funciones” que tienen como mínimo el parámetro `self`

# CLASES EN PYTHON

```
class Aves(object):
    def __init__(self, especie, pais):#Constructor
        self.especie=especie
        self.pais=pais

    def volar(self):
        return True

    def poner_huevos(self, cantidad):
        print(f"l@s {self.especie} ponen {cantidad} huevos a la vez")
```



## **Reto 1**

**Crear una clase que permita ejecutar las operaciones de suma y resta de vectores**