

ArgentinaPrograma
YoProgramo

Paradigmas de Programación

por Leonardo Blautzik, Federico Gasior y Lucas Videla

Julio / Diciembre 2021



Ministerio de
Desarrollo Productivo
Argentina

¿Qué es la programación?

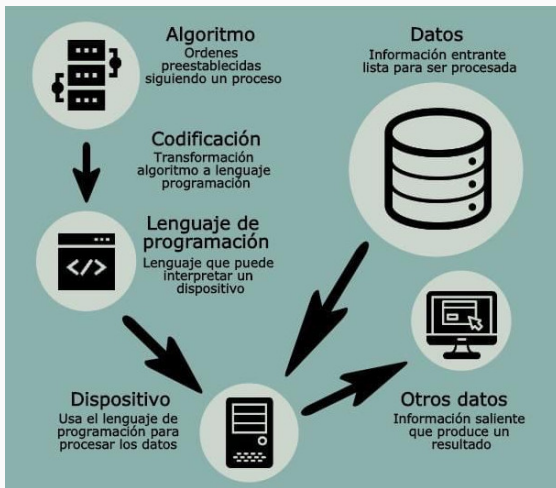


Figure 1: ¿Que es la programación?

Tareas que realiza un programador



Figure 2: ¿Qué hace un programador?

Paradigma: conjunto de creencias, prácticas y conocimientos que guían el desarrollo de una disciplina durante un período de tiempo.

Un **paradigma de programación** es un estilo de desarrollo de programas.

Es decir, un modelo para resolver problemas computacionales.

Los lenguajes de programación, necesariamente, se encuadran en uno o varios paradigmas a la vez a partir del tipo de órdenes que permiten implementar, algo que tiene una relación directa con su sintaxis.

¿Cuáles son los principales paradigmas de programación?

- **Imperativo:** Los programas se componen de un conjunto de sentencias (órdenes), que se ejecutan de manera secuencial. Es el paradigma en el que todos hemos comenzado a programar.
- **Declarativo:** Opuesto al imperativo. Los programas describen los resultados esperados sin listar explícitamente los pasos a llevar a cabo para alcanzarlos. El qué es más importante que el cómo.
 - **Lógico:** El problema se modela con enunciados de lógica de primer orden. Ej. Prolog
 - **Funcional:** Los programas se componen de funciones, es decir, implementaciones de comportamiento que reciben un conjunto de datos de entrada y devuelven un valor de salida. Ej. Haskell
- **Orientado a objetos:** El comportamiento del programa es llevado a cabo por **objetos**. Los **objetos** serán entendidos como entidades que representan elementos del problema a resolver, tienen atributos y comportamiento.

Ventajas de la Programación Orientada a Objetos

- Descomponer un problema: Se divide el problema en partes pequeñas gracias a los conceptos de Abstracción y Modularización.
- Orden y Legibilidad: La POO por lo general da como resultado un código más ordenado y legible. Las clases y objetos son fáciles de identificar gracias a la modularización, y se reutiliza código gracias a la herencia y el polimorfismo.
- Facilidad de mantenimiento: Un código más ordenado y legible significan también, programas más fáciles de mantener. Gracias a la agrupación de código en clases y objetos, resulta más sencilla la tarea modificar o eliminar código.

Desventajas de la Programación Orientada a Objetos

- No es aplicable para tareas simples: Para tareas y programas sencillos, siempre es más fácil acudir a la vieja y querida programación estructurada.
- A veces el concepto que un programador tiene de un objeto puede diferir del que tiene otro, es por eso que es necesario contar con la documentación correspondiente.
- ¿Curva de Aprendizaje? No!!

Consigna:

Sumar los 100 primeros números naturales ($1+2+3+4+\dots+100$), imprimiendo por cada suma el resultado parcial obtenido.

En C usando programación secuencial:

```
int main(){  
    int suma=0;  
    for(int i=1;i<=100;i++){  
        suma+=i;  
        printf("%d\n",suma);  
    }  
}
```


En C usando programación estructurada:

```
int sumados(int);
```

```
int main(){  
    for(int i=1; i<=10;i++)  
        printf("%d\n",sumados(i));  
}
```

```
int sumados(int n){  
    int suma=0;  
    for(int i=1;i<=n;i++)  
        suma+=i;  
    return suma;  
}
```

En Java:

```
public class Sumados{  
    public int sumados(int n){  
        int suma=0;  
        for(int i=1;i<=n;i++)  
            suma+=i;  
        return suma;  
    }  
    public static void main(String[] args){  
        Sumados sumi = new Sumados();  
        for(int i=1;i<=10;i++){  
            System.out.println(sumi.sumados(i));  
        }  
    }  
}
```

En prolog:

```
sumados(1,1):-!. 
```

```
sumados(X,G):-X>1,X1 is X-1,sumados(X1,G1), G is X+G1.
```

¡Muchas Gracias!

continuará...



Ministerio de
Desarrollo Productivo
Argentina