

## ESTRUCTURAS BÁSICAS DE UN PROGRAMA

Un programa informático (programa) es una secuencia de acciones (instrucciones) que manipulan un conjunto de objetos (datos).

El bloque de instrucciones está compuesto a su vez por tres partes, aunque en ocasiones no están perfectamente delimitadas, y aparecerán entremezcladas en la secuencia del programa, podemos localizarlas según su función.

Partes del bloque de instrucciones		
Entrada	--> Algoritmo -->	Salida
Inicio de programa: datos	Proceso de programa: cálculos	Fin de programa: resultados

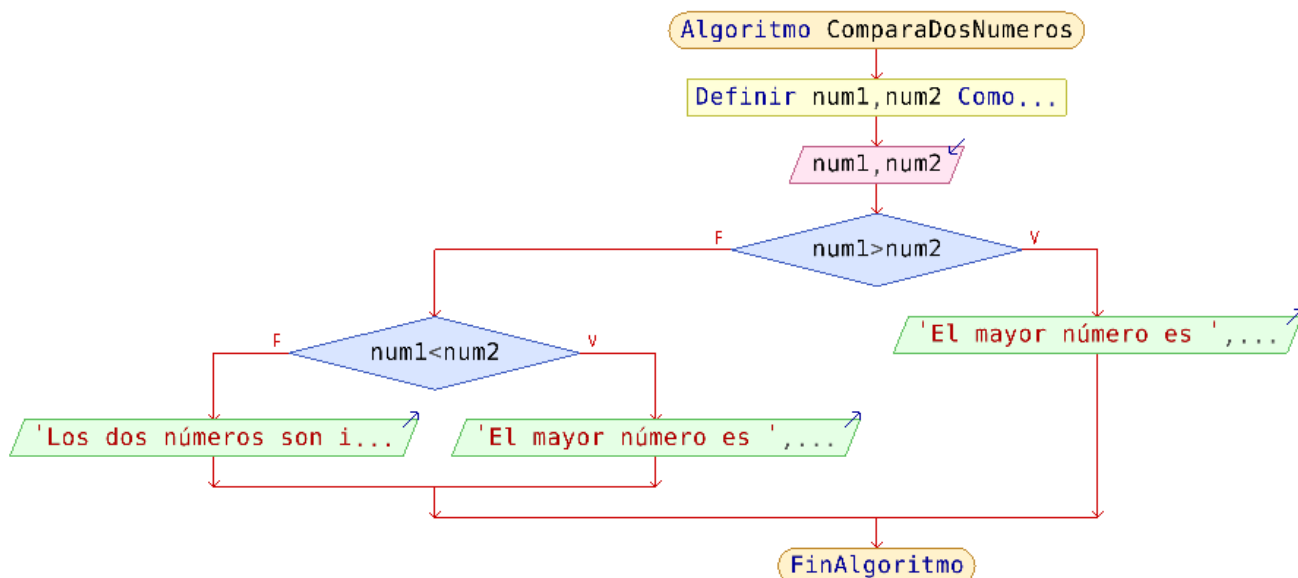
Estructura de un programa informático	
<b>Cabecera</b>	A modo de comentarios se suele especificar: <ul style="list-style-type: none"><li>• Nombre del programa</li><li>• Datos de entrada</li><li>• Datos de salida</li></ul>
<b>Funciones</b>	Definición de funciones propias creadas por el programador para usarlas en varias ocasiones
<b>Declaraciones</b>	Definiciones y tipos de datos: <ul style="list-style-type: none"><li>• variables</li><li>• constantes</li><li>• nuevos tipos de datos</li></ul>
<b>Asignaciones</b>	Valores iniciales de los datos declarados previamente
<b>Entradas</b>	Instrucciones para almacenar en memoria los valores de algunos datos iniciales
<b>Control</b>	Instrucciones de control de flujo del programa. Pueden ser: <ul style="list-style-type: none"><li>• Secuenciales</li><li>• De selección</li><li>• Iterativas</li></ul>
<b>Salidas</b>	Instrucciones para devolver los resultados obtenidos

## EJEMPLO

A continuación, te detallamos el ejemplo de programa informático ya citado para comparar dos números enteros y devolver el mayor.

### Paso 1: Diagrama de flujo

Con dicho gráfico representamos el orden secuencial y posibilidades de ejecución del programa.



### Paso 2: Pseudocódigo (aclaración)

Explicación de cada bloque de pseudocódigo según su función.

```
1 // *****
2 // ComparadosNumeros: Compara dos números y devuelve el mayor
3 // *****
4 // Entrada: num1, num2
5 // Salida: mayor entre num1, num2
6 // *****
7 // Comentarios
8 Algoritmo ComparaDosNumeros
9 Definir num1,num2 Como ENTERO Definición y tipo de variables
10 Leer num1,num2 Entrada de datos
11 Si num1>num2 Entonces
12     Escribir 'El mayor número es ',num1 Salida de resultados
13 Sino
14     Si num1<num2 Entonces
15         Escribir 'El mayor número es ',num2
16     Sino
17         Escribir 'Los dos números son iguales'
18     FinSi
19 FinSi Instrucción alternativa doble
20 FinAlgoritmo
```

### Paso 3: Traducción

El siguiente paso es convertir el pseudocódigo general a un lenguaje de programación concreto, en este caso a lenguaje PHP.

```
<?php
// *****
// Comparadosnumeros: Compara dos números y devuelve el mayor
// *****
// Entrada: num1, num2
// Salida: mayor entre num1, num2
// *****
//
$stdin = fopen('php://stdin','r');
settype($num1,'integer');
settype($num2,'integer');
fscanf($stdin,"%d",$num1);
fscanf($stdin,"%d",$num2);
if ($num1>$num2) {
    echo 'El mayor número es ', $num1, PHP_EOL;
} else {
    if ($num1<$num2) {
        echo 'El mayor número es ', $num2, PHP_EOL;
    } else {
        echo 'Los dos números son iguales', PHP_EOL;
    }
}
?>
```

### CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN POR GENERACIONES

**Primera generación (lenguaje de bajo nivel):** los primeros ordenadores se programaban directamente en **código de máquina** (basado en **sistema binario**), que puede representarse mediante secuencias de 0 y 1.

**Segunda generación:** los **lenguajes simbólicos**, asimismo propios de la máquina, simplifican la escritura de las instrucciones y las hacen más legibles. Se refiere al lenguaje **ensamblador**.

**Tercera generación:** los **lenguajes de alto nivel** sustituyen las instrucciones simbólicas por códigos independientes de la máquina, parecidas al lenguaje humano o al de las **Matemáticas**. Entre ellos se encuentran **C, Fortran, Smalltalk, Ada, Cobol, Delphi**, entre otros.

**Cuarta generación:** para la **programación orientada a objetos** (POO). tienen una estructura muy parecida al idioma inglés. Algunas de sus características son: acceso a **base de datos**, capacidades gráficas, generación de código automáticamente, así como poder programar visualmente (como por ejemplo **SQL, C++, C#, Java, Ruby, Visual Basic, PHP** o Python).

**Quinta generación:** se centran en la resolución de problemas utilizando inteligencia artificial y técnicas de programación declarativa. Programación declarativa consiste en decirle a un programa lo que tiene que hacer en lugar de decirle cómo debería hacerlo. Ejemplos:

- **Mercury:** Un lenguaje de programación funcional basado en lógica que utiliza un enfoque declarativo para la programación.
- **OPS5:** Un lenguaje de programación basado en reglas que se utiliza en sistemas expertos.
- **Prolog:** Un lenguaje de programación lógico que se utiliza para la programación de inteligencia artificial y sistemas expertos.
- **Haskell:** Un lenguaje de programación funcional que se utiliza en la programación de inteligencia artificial y aprendizaje automático.
- **Lisp:** Un lenguaje de programación funcional que se utiliza en la programación de inteligencia artificial y sistemas expertos.