

# CLASE 1. Concepto de programación. Especificación de algoritmos

## 1. Introducción a la Programación

La programación es el proceso mediante el cual se crean y diseñan programas de software que permiten la automatización de tareas específicas, solucionan problemas y facilitan la interacción con sistemas informáticos. Este proceso implica la creación de instrucciones claras y precisas, capaces de ser interpretadas por una máquina para llevar a cabo acciones definidas.

Un concepto esencial dentro de la programación es la **abstracción**, donde se simplifican los problemas complejos en tareas más manejables, facilitando su resolución mediante algoritmos y estructuras de datos.

## 2. Lenguajes de Programación y su Clasificación

Los lenguajes de programación son herramientas que permiten a los desarrolladores comunicarse con las computadoras. Existen diversas clasificaciones que se basan en el nivel de abstracción, el propósito del lenguaje o su paradigma. Comprender esta clasificación es crucial para seleccionar el lenguaje adecuado para una tarea específica. A continuación, se detallan las principales clasificaciones:

- **Lenguajes de bajo nivel:** Cercanos al lenguaje máquina, como el ensamblador. Estos lenguajes ofrecen control detallado del hardware pero requieren mayor esfuerzo para programar.
- **Lenguajes de alto nivel:** Proveen abstracciones que facilitan la programación. Ejemplos incluyen Java, Python y C++. Estos

lenguajes son más comprensibles y manejables por los humanos.

- **Lenguajes de propósito general:** Son flexibles y aplicables en diversas áreas. Ejemplos incluyen Java, C++ y Python.
- **Lenguajes de propósito específico:** Diseñados para un tipo particular de tarea, como SQL para la gestión de bases de datos.

## Utilidades de un Lenguaje de Programación

Los lenguajes de programación permiten la creación de sistemas operativos, aplicaciones, sitios web, inteligencia artificial, entre otros. Su capacidad de transformar ideas abstractas en soluciones funcionales es su principal utilidad. Además, proporcionan las herramientas para gestionar recursos de hardware, optimizar procesos y permitir la interacción entre diferentes sistemas.

## 3. El Proceso de Desarrollo de Software

El desarrollo de software es un proceso estructurado que sigue una serie de etapas, cada una con su propósito específico. A continuación se detallan las principales fases:

- **Análisis de Requisitos:** Identificación de las necesidades del usuario.
- **Diseño:** Definición de la arquitectura y componentes del software.
- **Codificación:** Escritura del código fuente utilizando un lenguaje de programación.
- **Pruebas:** Validación del software para asegurar su correcto funcionamiento.
- **Mantenimiento:** Actualización y corrección de errores en el software a lo largo del tiempo.

Cada una de estas etapas requiere una planificación meticulosa y una ejecución coordinada para garantizar que el producto final cumpla con los requisitos establecidos.

#### 4. Algoritmos y su Importancia en la Programación

Un **algoritmo** es una secuencia bien definida de pasos o instrucciones que resuelven un problema específico. En programación, los algoritmos son fundamentales ya que definen la lógica y los procesos que el programa debe seguir para alcanzar los resultados deseados. Su correcta implementación asegura que el programa sea eficiente y efectivo.

- **Ejemplo de Algoritmo:**

Supongamos que queremos desarrollar un algoritmo que determine si un número es par o impar:

1. Tomar un número como entrada.
2. Dividir el número entre 2.
3. Si el residuo es 0, entonces el número es par.
4. Si el residuo es diferente de 0, entonces el número es impar.

La eficiencia de un algoritmo se puede medir en términos de tiempo de ejecución y uso de memoria, aspectos críticos en el desarrollo de software.

#### 5. Pseudocódigo y Diagramas de Flujo

El **pseudocódigo** es una representación textual informal de un algoritmo que facilita la comprensión y el diseño previo de la solución, antes de ser implementada en un lenguaje de programación. Su principal ventaja es la simplicidad con la que se puede representar el flujo lógico del programa, sin preocuparse por la sintaxis exacta de un lenguaje de programación.

- **Estructura de un Algoritmo en Pseudocódigo:**

```
INICIO
  Leer número
  Si número mod 2 = 0 entonces
    Imprimir "El número es par"
  Si no
    Imprimir "El número es impar"
FIN
```

Este ejemplo muestra cómo estructurar un algoritmo simple que determine si un número es par o impar utilizando pseudocódigo. Se observa que la lógica del programa es clara y fácilmente comprensible, lo que facilita su posterior traducción a un lenguaje de programación.

- **Ventajas del Pseudocódigo:**

- Claridad: El pseudocódigo es fácil de leer y entender.
- Flexibilidad: No depende de un lenguaje de programación específico.
- Ideal para planificación: Permite definir la lógica del programa antes de implementar el código real.

- **Desventajas del Pseudocódigo:**

- No ejecutable: No se puede compilar ni ejecutar como un programa.
- Ambigüedad: A veces, su falta de estructura formal puede llevar a interpretaciones erróneas.

- **Palabras reservadas comunes en Pseudocódigo:**

- INICIO , FIN : Marcan el comienzo y el final del algoritmo.
- SI , ENTONCES , SI NO : Condiciones.
- MIENTRAS , PARA : Bucles.

## 6. Diagramas de Flujo

Los **diagramas de flujo** son una representación gráfica de un proceso o algoritmo. Utilizan diferentes símbolos para representar las operaciones que se llevan a cabo y el flujo del programa. Los diagramas de flujo son útiles para visualizar la lógica de un programa, especialmente en fases de diseño.

- **Símbolos Comunes de Diagramas de Flujo:**
  - **Óvalo:** Representa el inicio y el fin del diagrama.
  - **Rectángulo:** Indica una operación o proceso.
  - **Rombo:** Representa una decisión o bifurcación.
  - **Flechas:** Muestran el flujo de control.
- **Ejemplo de Diagrama de Flujo:**

Un diagrama que represente el algoritmo para determinar si un número es par o impar tendría las siguientes etapas:

1. Inicio → 2. Leer número → 3. ¿Número divisible por 2? →  
4. Sí: Imprimir "Número par" → 5. No: Imprimir "Número  
impar" → 6. Fin

Existen diversas herramientas en línea para la creación de diagramas de flujo, como Lucidchart, Draw.io y Creately. Estos programas facilitan la creación de diagramas con plantillas predefinidas y opciones de personalización.