GUÍA DE ESTUDIO: Eje I - Conceptos Básicos de Programación

1. Lenguajes de Programación

Un lenguaje de programación es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Está compuesto por un conjunto de reglas sintácticas y semánticas que permiten a los programadores expresar instrucciones (código) que la computadora puede entender y ejecutar.

- **Niveles de abstracción:**

  - ✓ **Lenguaje de Bajo Nivel:** Muy cercano al hardware de la máquina (ej. Lenguaje Ensamblador). Permite un control directo sobre los recursos del sistema, pero es complejo de escribir y leer.
  - ✓ **Lenguaje de Alto Nivel:** Más cercano al lenguaje humano y abstracto del hardware (ej. Python, PHP, JavaScript). Facilita la lectura y escritura del código, permitiendo desarrollar aplicaciones complejas con mayor rapidez.

2. Intérpretes y Compiladores

Para que el código fuente escrito en un lenguaje de alto nivel pueda ser ejecutado por el procesador, debe ser traducido a lenguaje de máquina (binario). Esto se realiza a través de dos tipos de traductores principales:

- **Compilador:**

  - ✓ Traduce **todo** el código fuente completo a un archivo ejecutable antes de ejecutarlo.
  - ✓ Si hay un error de sintaxis en cualquier parte del código, la compilación se detiene y no se genera el archivo ejecutable.
  - ✓ Lenguajes típicos: C, C++, C#.

- **Intérprete:**

  - ✓ Lee y traduce el código **línea por línea** (o instrucción por instrucción) en tiempo de ejecución.
  - ✓ Ejecuta el programa de inmediato, pero se detiene en el momento exacto en que encuentra un error.
  - ✓ Lenguajes típicos: PHP, Python, JavaScript.

3. Diseño de Algoritmos y Flujo de Control

Un algoritmo es una secuencia finita, ordenada y precisa de pasos lógicos que permiten resolver un problema o realizar una tarea específica.

Para que un algoritmo tome decisiones y repita procesos, utiliza estructuras de **flujo de control**:

- **Secuenciación:**
    - ✓ Es la ejecución lineal de las instrucciones una tras otra, en el orden exacto en que fueron escritas (de arriba hacia abajo).
- **Selección (Condicionales):**
    - ✓ Permiten tomar decisiones basadas en una condición lógica (Verdadero o Falso). Desvían el flujo del algoritmo hacia caminos diferentes.
    - ✓ *Ejemplos comunes:* `if` (si se cumple), `else` (si no se cumple), `switch` / `case`.
- **Iteración (Bucles o Ciclos):**
    - ✓ Permiten repetir un bloque de código múltiples veces de forma automática hasta que se cumpla o deje de cumplirse una condición específica.
    - ✓ *Ejemplos comunes:* `for` (iteración con un número conocido de repeticiones), `while` y `do-while` (iteración basada en condiciones).