

1 Competencias a desarrollar

1.1 Competencias genéricas

1. Desarrollar habilidades de razonamiento lógico y pensamiento estructurado.

1.2 Competencias específicas

2. Capacidad para diseñar y representar algoritmos utilizando diagramas de flujo, así como comprender los fundamentos de la programación estructurada.
3. Aplicar lógica de programación para resolver problemas simples, reforzando la comprensión y la capacidad de estructurar soluciones efectivas.

2 Conceptos Básicos de Programación

La programación es una habilidad esencial en el mundo moderno, donde la tecnología permea todos los aspectos de nuestra vida. Conocer los conceptos básicos de programación es fundamental para adentrarse en el fascinante universo del desarrollo de software y la resolución de problemas. En este curso, abordaremos aspectos clave que te ayudarán a formar una base sólida.

La programación es el proceso de diseñar y construir un conjunto de instrucciones que permiten a una computadora realizar una tarea específica. En esencia, implica escribir código en un lenguaje que la máquina pueda interpretar y ejecutar. Los lenguajes de programación, como Python, Java y C++, permiten a los programadores expresar sus ideas y soluciones de una forma estructurada y comprensible tanto para humanos como para máquinas (Smith, 2022).

3 El Pensamiento computacional

El **pensamiento computacional (PC)** trasciende la mera codificación; constituye un marco cognitivo para la resolución de problemas mediante la formulación de soluciones que un agente de procesamiento (humano o máquina) pueda ejecutar.

3.1 Dimensiones del PC

Se articula en cuatro dimensiones fundamentales:

1. **Abstracción:** Identificación de los atributos esenciales de un sistema, eliminando el ruido o detalles irrelevantes.
2. **Descomposición:** Aplicación del paradigma *divide y vencerás* para segmentar problemas complejos en módulos manejables.
3. **Reconocimiento de patrones:** Detección de regularidades y analogías entre distintos problemas para optimizar la solución.
4. **Diseño algorítmico:** Construcción de la lógica de control independiente de la sintaxis de un lenguaje específico.

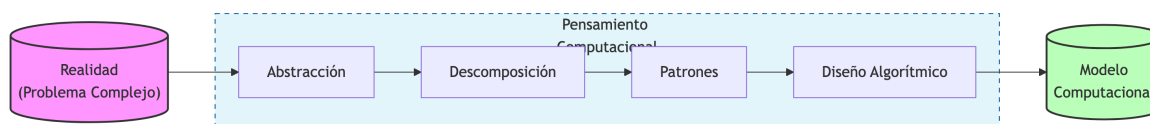


Figure 1: Proceso de abstracción: De la realidad al modelo computacional

3.2 Algoritmo

Un algoritmo es un conjunto de pasos o instrucciones definidos y ordenados que permiten resolver un problema o realizar una tarea específica. En programación, los algoritmos son esenciales porque descomponen problemas complejos en pasos manejables y lógicos, los cuales pueden ser traducidos fácilmente al lenguaje de la computadora (Jones, 2021).

3.2.1 Propiedades

Un algoritmo debe cumplir estrictamente con las siguientes propiedades:

1. **Finitud:** Todo algoritmo debe tener un punto de finalización tras ejecutar una cantidad limitada de pasos. No puede entrar en un bucle infinito; debe existir una condición de parada clara. En términos computacionales, implica que el consumo de recursos (tiempo y memoria) es limitado.
2. **Precisión:** Esta propiedad exige que cada paso sea unívoco. En un lenguaje más sencillo, esto significa que cada instrucción debe tener una única interpretación posible. No debe existir espacio para la duda o la confusión.
 - Ejemplo de ambigüedad: “Camine unos pasos hacia adelante”.
 - Ejemplo de precisión (Unívoco): “Camine 5 pasos hacia adelante”.
3. **Entrada (Input):** Representa el conjunto de datos iniciales necesarios para iniciar el proceso. Estos datos deben pertenecer a un dominio definido (un conjunto de valores válidos).
 - Si un algoritmo calcula raíces cuadradas, su dominio de entrada son los números reales; un carácter alfabético estaría fuera de su dominio.
4. **Salida (Output):** Es el resultado o respuesta que produce el algoritmo al finalizar su ejecución. La salida debe tener una relación funcional con la entrada. Esto significa que, si el algoritmo es correcto, para una entrada específica siempre se debe obtener una salida coherente que resuelva el problema planteado.
5. **Efectividad:** Cada operación dentro del algoritmo debe ser realizable y básica. Esto significa que las instrucciones deben poder ejecutarse en un tiempo razonable utilizando los recursos disponibles (hardware o incluso lápiz y papel). Un paso que requiera “adivinar un número secreto” no es efectivo porque no es una operación lógica o físicamente realizable de forma determinista.

3.3 Programa

Un programa es la implementación material de uno o varios algoritmos mediante un lenguaje de programación.

Mientras que el algoritmo es el modelo lógico (la idea), el programa es el archivo ejecutable que el hardware de una computadora puede procesar para cumplir una función.

Ejemplo de Software

Un **navegador web** (como Chrome o Firefox) es un programa.

- **Instrucciones:** Código que interpreta HTML, CSS y JS.
- **Tarea:** Visualizar páginas web.
- **Resultado:** Interacción del usuario con internet.