

GUÍA TEÓRICO-PRÁCTICA: INTRODUCCIÓN A LAS CIENCIAS DE LA COMPUTACIÓN

UNIDAD I

SEMANA 1: FUNDAMENTOS DE COMPUTACIÓN Y PENSAMIENTO COMPUTACIONAL

TEMA 1.1: CONCEPTO Y FUNCIONES DE LA COMPUTADORA

Teoría: Una computadora es una máquina electrónica capaz de recibir, procesar y devolver información de manera automática

Funciona bajo el control de un conjunto de instrucciones llamadas programas.

Funciones básicas:

- **Entrada de datos:** Captura información del exterior (teclado, mouse, escáner)
- **Procesamiento:** Transforma los datos mediante operaciones lógicas y aritméticas (CPU)
- **Almacenamiento:** Guarda información temporal o permanentemente (RAM, disco duro)
- **Salida:** Presenta resultados (monitor, impresora, altavoces)

Pregunta de reflexión 1: ¿Cuál consideras que es la función más importante de una computadora y por qué?

Ejemplo

Cuando escribes un documento:

- **Entrada:** Tecleas texto
- **Procesamiento:** La CPU convierte las teclas en caracteres y aplica formato
- **Almacenamiento:** Se guarda en RAM temporalmente y en disco al guardar
- **Salida:** Se muestra en pantalla o se imprime

Ejercicio 1.1

Identifica las funciones de entrada, procesamiento, almacenamiento y salida en las siguientes actividades: a) Escuchar música en Spotify b) Tomar una fotografía con el celular c) Realizar una búsqueda en Google

TEMA 1.2: GENERACIONES DE COMPUTADORAS Y SU IMPACTO

Teoría

Primera Generación (1940-1956)

- Utilizaban **tubos de vacío** (válvulas)
- Grandes dimensiones y alto consumo energético
- Lenguaje de máquina y tarjetas perforadas
- Ejemplo: ENIAC, UNIVAC

Segunda Generación (1956-1963)

- Reemplazo de tubos por **transistores**
- Menor tamaño y consumo
- Lenguajes de programación: COBOL, FORTRAN
- Más rápidas y confiables

Tercera Generación (1964-1971)

- **Circuitos integrados** (chips)
- Multiprogramación
- Sistemas operativos
- Menor costo de producción

Cuarta Generación (1971-1981)

- **Microprocesadores**
- Nacimiento de la PC personal
- Interfaces gráficas
- Internet comienza a desarrollarse

Quinta Generación (1981-Actualidad)

- Inteligencia Artificial
- Computación en la nube
- Dispositivos móviles
- Big Data, IoT

Pregunta de reflexión 2: ¿Qué generación consideras que ha tenido mayor impacto en tu vida diaria? Explica.

Ejemplo

Comparación de tamaño:

- **1ra gen:** ENIAC ocupaba 167 m² y pesaba 27 toneladas
- **4ta gen:** Una laptop cabe en una mochila y pesa 2 kg
- **5ta gen:** Un smartphone en tu bolsillo es más potente que las supercomputadoras de 1990

Ejercicio 1.2

Investiga y completa: a) ¿En qué año fue lanzada la primera PC de IBM? b) ¿Qué dispositivo actual consideras que representa la quinta generación? c) Menciona 3 diferencias entre una computadora de 1950 y una actual

TEMA 1.3: ¿QUÉ ES EL PENSAMIENTO COMPUTACIONAL?

Teoría

El pensamiento computacional es un proceso de resolución de problemas que implica formular problemas y sus soluciones de manera que puedan ser ejecutadas por una computadora

No se trata solo de programar, sino de pensar de manera lógica y estructurada.

Fases del pensamiento computacional:

1. **Descomposición:** Dividir problemas complejos en partes más pequeñas
2. **Reconocimiento de patrones:** Identificar similitudes y tendencias

3. **Abstracción:** Enfocarse en lo importante, ignorando detalles irrelevantes
4. **Algoritmos:** Crear pasos ordenados para resolver el problema
5. **Codificación y depuración:** Implementar y corregir

Pregunta de reflexión 3: ¿Cómo aplicarías el pensamiento computacional para organizar tu día de clases?

Ejemplo

Problema: Organizar una fiesta de cumpleaños

Aplicación del pensamiento computacional:

- **Descomposición:** Dividir en: invitations, comida, decoración, música, juegos
- **Patrones:** Las fiestas anteriores funcionaron mejor con cierta música o comida
- **Abstracción:** Lo importante es que los invitados se diviertan, no el color exacto de los globos
- **Algoritmo:**
 1. Hacer lista de invitados
 2. Enviar invitaciones 2 semanas antes
 3. Comprar comida y decoración
 4. Preparar playlist
 5. El día del evento: decorar, recibir invitados, servir comida

Ejercicio 1.3

Aplica las 4 fases del pensamiento computacional para resolver: **Problema:** "Necesitas estudiar para 3 exámenes que tienes la próxima semana"

Descomposición: _____

Patrones: _____

Abstracción: _____

Algoritmo: _____

TEMA 1.4: DESCOMPOSICIÓN DE PROBLEMAS COTIDIANOS

Teoría

La descomposición consiste en dividir un problema complejo en partes más pequeñas y manejables, para luego enfocarse en resolver cada una de estas partes

Es como desmontar un rompecabezas pieza por pieza

Ventajas de la descomposición:

- Facilita la comprensión del problema
- Permite trabajar en partes simultáneamente
- Reduce la complejidad
- Hace el problema menos abrumador

Pregunta de reflexión 4: ¿Por qué crees que es más fácil resolver problemas pequeños que uno grande?

Ejemplo

Problema complejo: "Quiero crear una aplicación móvil"

Descomposición:

1. **Diseño:**
 - Definir funcionalidades
 - Crear interfaz de usuario
 - Diseñar experiencia de usuario
2. **Programación:**
 - Configurar entorno de desarrollo
 - Programar cada pantalla
 - Conectar con base de datos
3. **Pruebas:**
 - Probar cada función
 - Corregir errores

- Optimizar rendimiento

4. Publicación:

- Crear cuenta de desarrollador
- Subir a tienda de aplicaciones
- Promocionar

Ejercicio 1.4

Descompón los siguientes problemas en al menos 4 partes:

a) Planear unas vacaciones familiares:

1. _____
2. _____
3. _____
4. _____

b) Aprender a tocar guitarra:

1. _____
2. _____
3. _____
4. _____

SEMANA 2: ELEMENTOS DEL PENSAMIENTO COMPUTACIONAL

TEMA 2.1: RECONOCIMIENTO DE PATRONES

Teoría

El reconocimiento de patrones es uno de los cuatro pilares del pensamiento computacional. Implica encontrar similitudes, tendencias o regularidades entre problemas pequeños y descompuestos

Tipos de patrones:

- **Numéricos:** Secuencias matemáticas
- **Visuales:** Formas geométricas repetitivas
- **Temporales:** Eventos que ocurren en intervalos

- **De comportamiento:** Acciones que se repiten

Pregunta de reflexión 5: ¿Qué patrones observas en tu rutina diaria?

Ejemplos

Patrón numérico: Secuencia: 2, 4, 6, 8, 10, ... **Patrón:** Sumar 2 al número anterior

Patrón visual: ●●●●●● **Patrón:** Alternar círculos rojos y azules

Patrón en la naturaleza:

- Las abejas crean panales hexagonales
- Las hojas de los helechos tienen patrones repetitivos
- Las estaciones del año se repiten cíclicamente

Ejercicio 2.1

Identifica el patrón y continúa las secuencias:

a) 3, 6, 9, 12, __, __, __ Patrón: _____

b) 1, 1, 2, 3, 5, 8, __, __ Patrón: _____

c) Lunes, Miércoles, Viernes, __, __ Patrón: _____

d) ▲▼▲▼▲, __, __ Patrón: _____

TEMA 2.2: ELABORACIÓN DE PATRONES CON FIGURAS Y NÚMEROS

Teoría

Crear patrones requiere identificar una regla que se repite sistemáticamente. Los patrones pueden ser:

- **Crecientes:** Aumentan en tamaño o cantidad
- **Decrecientes:** Disminuyen progresivamente
- **Cíclicos:** Se repiten en el mismo orden
- **Simétricos:** Se reflejan alrededor de un eje

Pregunta de reflexión 6: ¿Para qué crees que es útil identificar patrones en la programación?

Ejemplos

Patrón creciente numérico: 1, 3, 6, 10, 15, 21... **Regla:** Sumar números consecutivos (+2, +3, +4, +5, +6...)

Patrón con figuras: □ □□ □□□ □□□□ **Regla:** Cada fila tiene un cuadrado más que la anterior

Patrón mixto: A1, B2, C3, D4, E5... **Regla:** Letra consecutiva + número consecutivo

Ejercicio 2.2

Crea los siguientes patrones:

a) **Patrón numérico decreciente** comenzando en 20: 20, ____, ____, ____, ____

b) **Patrón cíclico con letras** (usa A, B, C): ____, ____, ____, ____, ____, ____

c) **Patrón creciente con símbolos** (usa ★):

d) **Inventa un patrón mixto** (letras y números): ____, ____, ____, ____

TEMA 2.3: ABSTRACCIÓN - REPRESENTACIÓN DE OBJETOS

Teoría

La abstracción es el proceso de enfocarse en la información importante de un objeto o problema, ignorando los detalles irrelevantes

Es "el alma" de la representación computacional

Niveles de abstracción:

- **Alto nivel:** Conceptos generales
- **Bajo nivel:** Detalles específicos

Ejemplo de abstracción: Un mapa del metro es una abstracción: muestra las estaciones y conexiones, pero no la distancia real ni los edificios cercanos.

Pregunta de reflexión 7: ¿Por qué un mapa no muestra todos los detalles de una ciudad?

Ejemplos

Objeto: Automóvil

Abstracción para un niño:

- Tiene ruedas
- Se mueve
- Lleva personas

Abstracción para un mecánico:

- Motor (cilindros, pistones)
- Transmisión
- Sistema de frenos
- Suspensión

Abstracción para un programador de autos autónomos:

- Sensores de distancia
- GPS
- Cámara de reconocimiento
- Sistema de control

Objeto: Biblioteca

Abstracción (lo esencial):

- Libros organizados
- Sistema de préstamo
- Catálogo de búsqueda
- Espacio de lectura

Detalles irrelevantes (para la abstracción):

- Color de las paredes
- Tipo de sillas
- Marca de las computadoras

Ejercicio 2.3

Realiza la abstracción de los siguientes objetos, identificando solo 3 características esenciales:

a) Un teléfono celular:

1. _____
2. _____
3. _____

b) Una escuela:

1. _____
2. _____
3. _____

c) Un restaurante:

1. _____
 2. _____
 3. _____
-

TEMA 2.4: PRÁCTICA - ESQUEMAS Y TEXTOS SIMPLIFICADOS

Teoría

Los esquemas son representaciones visuales que organizan la información de manera jerárquica o secuencial. Los textos simplificados eliminan información redundante y se enfocan en lo esencial.

Tipos de esquemas:

- **Mapas conceptuales:** Conectan conceptos con palabras de enlace

- **Diagramas:** Representación gráfica de procesos
- **Flujogramas:** Muestran secuencia de pasos
- **Cuadros sinópticos:** Organización jerárquica con llaves

Pregunta de reflexión 8: ¿Qué ventajas tiene representar información con esquemas en lugar de solo texto?

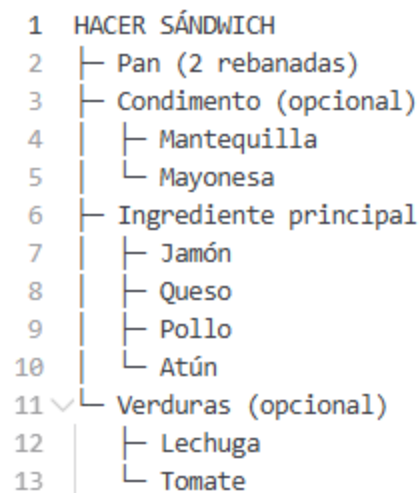
Ejemplo

Texto original: "Para hacer un sándwich, primero debes conseguir dos rebanadas de pan. Luego, puedes agregar mantequilla o mayonesa si lo deseas. Después, coloca el ingrediente principal que puede ser jamón, queso, pollo o atún. Agrega lechuga y tomate si te gustan las verduras. Finalmente, junta las dos rebanadas y tu sándwich estará listo para comer."

Texto simplificado:

1. Conseguir 2 rebanadas de pan
2. Untar condimento (opcional)
3. Agregar ingrediente principal
4. Añadir verduras (opcional)
5. Unir rebanadas

Esquema:



Ejercicio 2.4

a) Simplifica el siguiente texto: "Para cepillarte los dientes, primero debes mojar el cepillo con agua. Después, aplica una cantidad pequeña de pasta dental, más o menos del tamaño de un guisante. Luego, coloca el cepillo en un ángulo de 45 grados contra tus dientes y comienza a cepillar con movimientos circulares suaves. Debes cepillar todas las superficies: las que mastican, las exteriores y las interiores. No olvides cepillar también tu lengua. Finalmente, enjuaga tu boca con agua y lava el cepillo."

Versión simplificada:

1. _____
2. _____
3. _____
4. _____
5. _____

b) Crea un esquema del proceso "Preparar una taza de café":

SEMANA 3: ALGORITMOS

TEMA 3.1: ¿QUÉ SON LOS ALGORITMOS? CARACTERÍSTICAS

Teoría

Un algoritmo es una secuencia ordenada y finita de pasos o instrucciones que permiten resolver un problema específico o realizar una tarea

Los algoritmos están presentes en nuestra vida diaria, aunque no siempre los llamamos así.

Características de los algoritmos:

1. **Precisos:** Cada paso debe estar claramente definido
2. **Finitos:** Deben tener un número limitado de pasos
3. **Definidos:** Siempre producen el mismo resultado con los mismos datos de entrada
4. **Efectivos:** Cada paso debe ser realizable

5. **Secuenciales:** Los pasos siguen un orden lógico
6. **Independientes:** Pueden ejecutarse sin depender de un lenguaje de programación específico

Pregunta de reflexión 9: ¿Puedes identificar un algoritmo que sigas diariamente sin darte cuenta?

Ejemplos

Algoritmo: Calcular el área de un rectángulo

1. Inicio
2. Leer base (b)
3. Leer altura (h)
4. Calcular área: $A = b \times h$
5. Mostrar resultado: A
6. Fin

Algoritmo cotidiano: Cruzar la calle

1. Detenerse en la esquina
2. Mirar a la izquierda
3. Mirar a la derecha
4. Si viene un auto, esperar
5. Si no viene ningún auto, cruzar
6. Llegar al otro lado

Ejercicio 3.1

Identifica si las siguientes secuencias son algoritmos (SÍ/NO) y justifica:

- a) "Para hacer café: hierve agua, agrega café, sirve en taza" ¿Es algoritmo? ___ Justificación: _____
- b) "Sé feliz" ¿Es algoritmo? ___ Justificación: _____
- c) "Receta de pastel con cantidades y tiempos exactos" ¿Es algoritmo? ___ Justificación: _____

Crea un algoritmo para: "Preparar un vaso con cereal y leche"

1. _____
 2. _____
 3. _____
 4. _____
 5. _____
-

TEMA 3.2: ELEMENTOS DE UN ALGORITMO: ENTRADA, PROCESO, SALIDA

Teoría

Todo algoritmo consta de tres elementos fundamentales

1. Entrada (Input):

- Datos que recibe el algoritmo
- Pueden ser números, texto, imágenes, etc.
- Proviene del usuario, archivos o sensores

2. Proceso (Process):

- Operaciones que se realizan con los datos
- Cálculos matemáticos
- Comparaciones lógicas
- Transformaciones de datos

3. Salida (Output):

- Resultado del procesamiento
- Información que se muestra al usuario
- Datos guardados en archivos
- Acciones ejecutadas

Modelo: ENTRADA → PROCESO → SALIDA

Pregunta de reflexión 10: En una calculadora, ¿cuáles serían la entrada, el proceso y la salida?

Ejemplos

Ejemplo 1: Sumar dos números

- **Entrada:** Número A = 5, Número B = 3
- **Proceso:** Sumar $A + B = 8$
- **Salida:** Resultado = 8

Ejemplo 2: Calcular promedio de calificaciones

- **Entrada:** Calificación 1 = 8, Calificación 2 = 9, Calificación 3 = 7
- **Proceso:**
 - Sumar: $8 + 9 + 7 = 24$
 - Dividir entre 3: $24 \div 3 = 8$
- **Salida:** Promedio = 8

Ejemplo 3: Verificar si alguien es mayor de edad

- **Entrada:** Edad = 17
- **Proceso:** Comparar: ¿Edad ≥ 18 ?
- **Salida:** "Menor de edad"

Ejercicio 3.2

Identifica entrada, proceso y salida en los siguientes casos:

a) Calcular el área de un círculo

- Entrada: _____
- Proceso: _____
- Salida: _____

b) Convertir Celsius a Fahrenheit

- Entrada: _____
- Proceso: _____

- Salida: _____

c) Determinar si un número es par o impar

- Entrada: _____
- Proceso: _____
- Salida: _____

d) Crea tu propio ejemplo: Problema: _____

- Entrada: _____
- Proceso: _____
- Salida: _____

TEMA 3.3: REPRESENTACIÓN VISUAL DE ALGORITMOS

Teoría

Los algoritmos pueden representarse visualmente para facilitar su comprensión y comunicación. Las representaciones visuales permiten ver el flujo de ejecución de manera gráfica

Ventajas de la representación visual:

- Facilita la detección de errores
- Mejora la comunicación entre programadores y usuarios
- Permite ver el flujo lógico claramente
- Ayuda en la planificación antes de programar

Tipos de representación:

1. **Diagramas de flujo** (ordinogramas)
2. **Pseudocódigo**
3. **Diagramas Nassi-Schneiderman**

Pregunta de reflexión 11: ¿Por qué crees que es más fácil entender un diagrama que solo texto?

Ejemplo

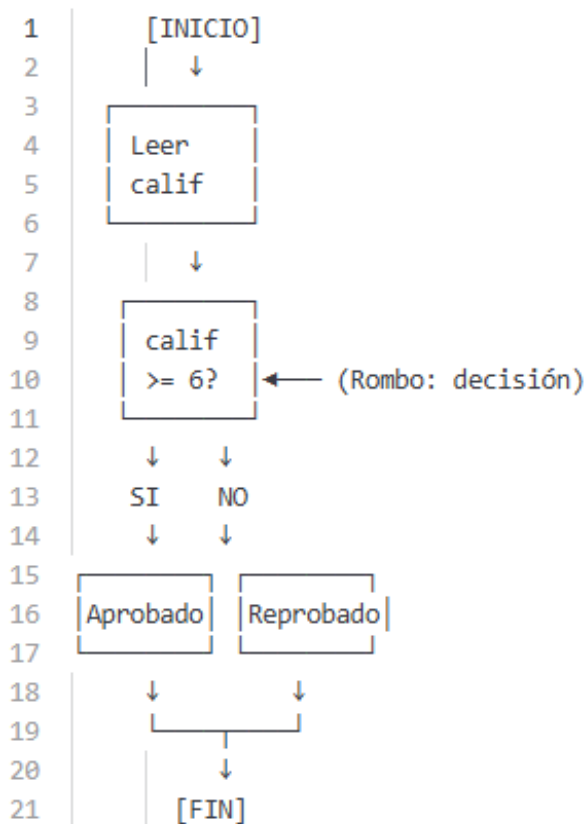
Problema: Determinar si un estudiante aprobó

Representación con pseudocódigo:

```

1  INICIO
2  |  LEER calificacion
3  |  SI calificacion >= 6 ENTONCES
4  |  |  ESCRIBIR "Aprobado"
5  |  SINO
6  |  |  ESCRIBIR "Reprobado"
7  |  FIN_SI
8  FIN
    
```

Representación con diagrama de flujo:



Ejercicio 3.3

a) Convierte a pseudocódigo: "Para calcular el IVA de un producto: leer el precio, multiplicar por 0.16, mostrar el resultado"

b) Describe con palabras el siguiente diagrama:

Proceso: _____

TEMA 3.4: DIAGRAMAS DE FLUJO

Teoría

Un diagrama de flujo es una representación gráfica de un algoritmo que utiliza símbolos estandarizados conectados por flechas que indican el flujo de ejecución

Símbolos básicos:

Símbolo	Nombre	Función
○	Terminal	Inicio/Fin del algoritmo
▭	Entrada/Salida	Leer datos o mostrar resultados
▭	Proceso	Operaciones y cálculos
◇	Decisión	Comparaciones (Sí/No)
→	Flujo	Dirección de ejecución

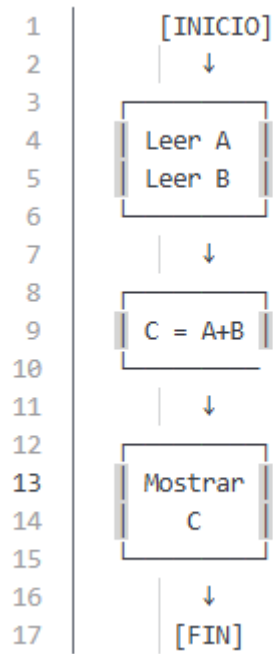
Reglas para elaborar diagramas de flujo:

1. Todo diagrama debe tener un inicio y un fin
2. Las líneas de flujo no deben cruzarse
3. Usar solo líneas horizontales y verticales
4. Los símbolos de decisión tienen 2 o 3 salidas
5. Leer de arriba hacia abajo y de izquierda a derecha

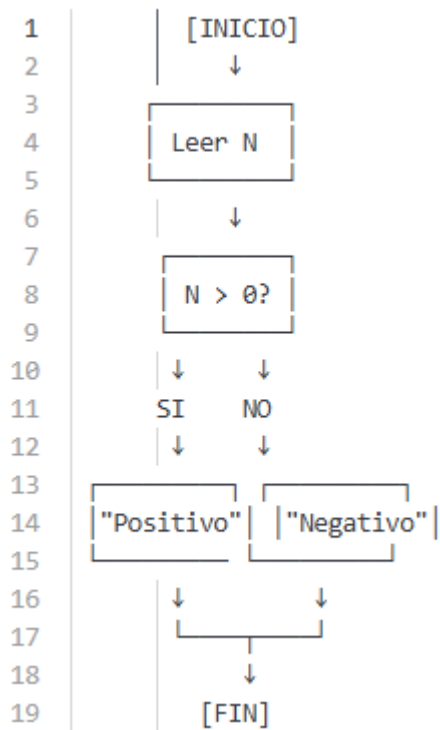
Pregunta de reflexión 12: ¿Qué símbolo usarías para representar "Sumar dos números"?

Ejemplos

Ejemplo 1: Sumar dos números



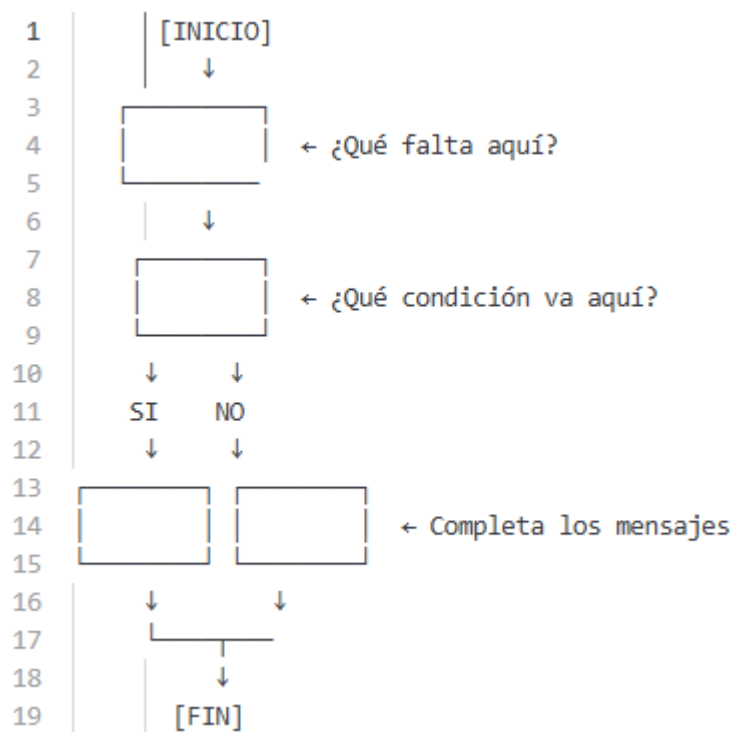
Ejemplo 2: Determinar si un número es positivo o negativo



Ejercicio 3.4

a) **Dibuja el diagrama de flujo para:** "Calcular el área de un triángulo ($A = \text{base} \times \text{altura} \div 2$)"

b) **Completa el diagrama para determinar si alguien puede votar (debe tener 18 años o más):**



SEMANA 4: PSEUDOCÓDIGO, HARDWARE Y SOFTWARE

TEMA 4.1: PSEUDOCÓDIGO - ESTRUCTURA DE ALGORITMOS

Teoría

El pseudocódigo es una descripción de un algoritmo escrita en un lenguaje intermedio entre el lenguaje natural (español, inglés) y un lenguaje de programación

Utiliza palabras clave que se asemejan a los lenguajes de programación, pero son más fáciles de entender.

Ventajas del pseudocódigo:

- Fácil de escribir y leer
- Independiente del lenguaje de programación

- Permite concentrarse en la lógica del algoritmo
- Facilita la transición a código real

Estructura básica:

```
1  INICIO
2  // Declaración de variables
3  <tipo> <nombre_variable>
4
5  // Entrada de datos
6  LEER <variable>
7
8  // Proceso
9  <variable> = <expresión>
10
11 // Salida
12 ESCRIBIR <mensaje o variable>
13
14 // Estructuras de control
15 SI <condición> ENTONCES
16 | <instrucciones>
17 SINO
18 | <instrucciones>
19 FIN_SI
20
21 // Ciclos
22 MIENTRAS <condición> HACER
23 | <instrucciones>
24 FIN_MIENTRAS
25 FIN
```

Pregunta de reflexión 13: ¿Por qué el pseudocódigo es útil antes de escribir el programa real?

Ejemplos

Ejemplo 1: Calcular el promedio de 3 calificaciones

```
1  √ INICIO
2  | NUMERO cal1, cal2, cal3, promedio
3  |
4  | ESCRIBIR "Ingresa la primera calificación:"
5  | LEER cal1
6  |
7  | ESCRIBIR "Ingresa la segunda calificación:"
8  | LEER cal2
9  |
10 | ESCRIBIR "Ingresa la tercera calificación:"
11 | LEER cal3
12 |
13 | promedio = (cal1 + cal2 + cal3) / 3
14 |
15 | ESCRIBIR "El promedio es: ", promedio
16 | FIN
```

Ejemplo 2: Determinar el mayor de dos números

```
1  √ INICIO
2  | NUMERO a, b
3  |
4  | ESCRIBIR "Ingresa el primer número:"
5  | LEER a
6  |
7  | ESCRIBIR "Ingresa el segundo número:"
8  | LEER b
9  |
10 | √ SI a > b ENTONCES
11 | | ESCRIBIR "El mayor es: ", a
12 | √ SINO
13 | | ESCRIBIR "El mayor es: ", b
14 | | FIN_SI
15 | FIN
```

Ejercicio 4.1

a) Completa el pseudocódigo para calcular el doble de un número:

```
1  INICIO
2  |  NUMERO numero, resultado
3  |
4  |  ESCRIBIR "Ingresa un número:"
5  |  LEER _____
6  |
7  |  resultado = _____ 2
8  |
9  |  ESCRIBIR "El doble es: ", _____
10 FIN
```

b) Escribe el pseudocódigo para: "Calcular el área de un rectángulo (área = base × altura)"

c) Crea un algoritmo en pseudocódigo que: "Pida la edad de una persona y diga si es mayor o menor de edad"

TEMA 4.2: HARDWARE - COMPONENTES FÍSICOS

Teoría

El hardware son todos los componentes físicos y tangibles de una computadora

Clasificación del hardware:

1. Dispositivos de Entrada:

- **Teclado:** Introduce texto y comandos
- **Mouse/Ratón:** Apunta y selecciona
- **Escáner:** Digitaliza imágenes y documentos
- **Micrófono:** Captura audio
- **Cámara web:** Captura video

2. Dispositivos de Procesamiento:

- **CPU (Unidad Central de Proceso):** "Cerebro" de la computadora
 - **ALU (Unidad Aritmético-Lógica):** Realiza cálculos
 - **UC (Unidad de Control):** Dirige operaciones
- **GPU (Unidad de Procesamiento Gráfico):** Procesa gráficos

3. Dispositivos de Almacenamiento:

- **RAM (Memoria de Acceso Aleatorio):** Almacenamiento temporal
- **Disco duro (HDD/SSD):** Almacenamiento permanente
- **Memoria USB:** Almacenamiento portátil

4. Dispositivos de Salida:

- **Monitor:** Muestra información visual
- **Impresora:** Imprime en papel
- **Altavoces/Audífonos:** Reproducen audio
- **Proyector:** Proyecta imágenes

Pregunta de reflexión 14: Si la CPU es el cerebro, ¿qué dispositivo sería el corazón de la computadora?

Ejemplos

Ejemplo de flujo de información:

1. **Entrada:** Escribes "2+2" en el teclado
2. **Procesamiento:** La CPU calcula el resultado
3. **Almacenamiento:** Se guarda en RAM temporalmente
4. **Salida:** El monitor muestra "4"

Comparación de almacenamiento:

- **RAM:** 8 GB (se borra al apagar)
- **Disco duro:** 500 GB - 1 TB (permanente)
- **USB:** 16 GB - 128 GB (portátil)

Ejercicio 4.2

a) Clasifica los siguientes dispositivos: Monitor, Teclado, CPU, Impresora, Mouse, Disco duro, Micrófono, Altavoces

Entrada	Procesamiento	Almacenamiento	Salida

b) Responde:

- ¿Qué pasa con la información en RAM cuando apagas la computadora?

- ¿Cuál es la diferencia entre HDD y SSD?

- Nombra 3 componentes internos de la CPU:

1. _____

2. _____
3. _____

TEMA 4.3: SOFTWARE - CLASIFICACIÓN Y EJEMPLOS

Teoría

El software es el conjunto de programas, instrucciones y datos que le dicen al hardware qué hacer

Clasificación del software:

1. Software de Sistema:

- **Sistemas Operativos:** Gestionan hardware y software
 - Windows, macOS, Linux, Android, iOS
- **Controladores (Drivers):** Permiten comunicación con hardware
- **Utilidades:** Antivirus, compresores, limpieza

2. Software de Aplicación:

- **Procesadores de texto:** Word, Google Docs
- **Hojas de cálculo:** Excel, Google Sheets
- **Navegadores:** Chrome, Firefox, Safari
- **Reproductores:** VLC, Windows Media Player
- **Editores de imagen:** Photoshop, GIMP

3. Software de Programación:

- **Editores de código:** VS Code, Sublime Text
- **Compiladores:** Traducen código a lenguaje máquina
- **Depuradores:** Detectan y corrigen errores

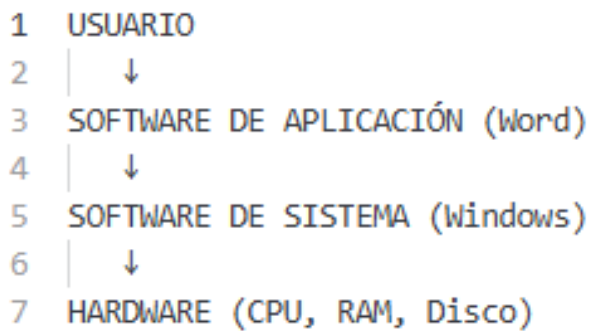
Pregunta de reflexión 15: ¿Sin qué software no podrías vivir? Explica por qué.

Ejemplos

Ejemplo de interacción:

- **Hardware:** Computadora con teclado, mouse, monitor
- **Software de sistema:** Windows 11
- **Software de aplicación:** Microsoft Word
- **Usuario:** Escribe un documento

Jerarquía:



Ejercicio 4.3

a) Clasifica el siguiente software:

Software	Sistema	Aplicación	Programación
Windows 11	✓		
Google Chrome			
Python			
Android			
Photoshop			
Excel			
Linux			
VS Code			

b) Investiga y responde:

- ¿Cuál es la diferencia entre software libre y software propietario?

- Nombra 3 ejemplos de software libre:

1. _____
2. _____
3. _____

TEMA 4.4: SISTEMAS OPERATIVOS - FUNCIONES PRINCIPALES

Teoría

Un sistema operativo (SO) es el software principal que gestiona el hardware de la computadora y proporciona servicios para los programas de aplicación

Funciones principales del SO:

1. Gestión de procesos:

- Crea, programa y elimina procesos
- Asigna tiempo de CPU
- Maneja la multitarea

2. Gestión de memoria:

- Asigna memoria RAM a programas
- Libera memoria cuando no se usa
- Usa memoria virtual

3. Gestión de archivos:

- Crea, modifica y elimina archivos
- Organiza en carpetas/directorios
- Controla permisos de acceso

4. Gestión de dispositivos:

- Controla periféricos (impresoras, discos)
- Usa controladores (drivers)
- Maneja colas de impresión

5. Interfaz de usuario:

- **GUI (Interfaz Gráfica):** Ventanas, iconos, menús
- **CLI (Línea de comandos):** Texto y comandos

6. Seguridad:

- Control de acceso con usuarios y contraseñas
- Permisos de archivos
- Firewall y protección

Sistemas operativos comunes:

- **Escritorio:** Windows, macOS, Linux
- **Móviles:** Android, iOS
- **Servidores:** Linux, Windows Server

Pregunta de reflexión 16: ¿Qué sistema operativo usas y qué es lo que más te gusta de él?

Ejemplos

Ejemplo de gestión de archivos:

```
1 C:/ (Raíz)
2 |─ Usuarios/
3 |   |─ Juan/
4 |   |   |─ Documentos/
5 |   |   |─ Imágenes/
6 |   |   └─ Descargas/
7 |   └─ María/
8 |─ Programas/
9 └─ Windows/
```

Ejemplo de multitarea: Mientras lees esto, el SO:

- Ejecuta el navegador
- Reproduce música de fondo
- Descarga actualizaciones
- Escanea antivirus
- Todo "al mismo tiempo" (en realidad, alterna muy rápido)

Ejercicio 4.4

a) Completa:

- El SO que viene preinstalado en iPhones es: _____
- El SO más usado en computadoras personales es: _____
- El SO de código abierto más popular es: _____
- Android está basado en: _____

b) Describe 3 funciones del SO que uses diariamente:

1. _____
2. _____
3. _____

c) ¿Qué sucede cuando abres un programa? Explica el proceso paso a paso:

1. _____
 2. _____
 3. _____
 4. _____
-

SEMANA 5: PRÁCTICA INTEGRADORA

TEMA 5.1-5.4: VALIDACIÓN Y PRESENTACIÓN DE ALGORITMOS

Teoría

Validación de algoritmos: Validar un algoritmo significa verificar que funcione correctamente para diferentes casos de entrada

Tipos de pruebas:

1. **Prueba de escritorio:** Ejecutar el algoritmo manualmente con datos específicos
2. **Casos límite:** Probar con valores extremos
3. **Casos de error:** Verificar cómo maneja entradas inválidas

Pregunta de reflexión 17: ¿Por qué es importante probar un algoritmo antes de usarlo?

Ejemplo

Algoritmo: Calcular el promedio de 2 números

```

1  INICIO
2  NUMERO n1, n2, promedio
3
4  LEER n1
5  LEER n2
6
7  promedio = (n1 + n2) / 2
8
9  ESCRIBIR "Promedio: ", promedio
10 FIN
    
```

Prueba de escritorio:

Caso	n1	n2	Proceso	Resultado esperado
1	10	8	$(10+8)/2 = 9$	9 ✓
2	0	0	$(0+0)/2 = 0$	0 ✓
3	100	100	$(100+100)/2 = 100$	100 ✓

Ejercicio 5.1

Realiza la prueba de escritorio del siguiente algoritmo:

```

1  INICIO
2  | NUMERO edad
3  |
4  | LEER edad
5  |
6  | SI edad >= 18 ENTONCES
7  | | ESCRIBIR "Mayor de edad"
8  | SINO
9  | | ESCRIBIR "Menor de edad"
10 | FIN_SI
11 FIN
    
```

Caso	edad	Resultado esperado
1	25	_____
2	17	_____
3	18	_____
4	0	_____

RETOS FINALES DE LA UNIDAD 1

RETO 1: "ORGANIZADOR DE EVENTO"

Situación: Eres el encargado de organizar un evento escolar y necesitas crear un sistema completo que ayude a planificarlo.

Requisitos:

Parte 1 - Descomposición y Patrones: a) Descompón el problema "Organizar evento escolar" en al menos 6 subproblemas b) Identifica 3 patrones que se repitan en eventos anteriores

Parte 2 - Algoritmo: c) Crea un algoritmo en pseudocódigo que:

- Pida el tipo de evento (fiesta, conferencia, torneo)
- Pida el número de asistentes
- Calcule:
 - Si es fiesta: $\text{costo} = \text{asistentes} \times \150
 - Si es conferencia: $\text{costo} = \text{asistentes} \times \200
 - Si es torneo: $\text{costo} = \text{asistentes} \times \100
- Muestre el costo total
- Indique si se necesita permiso especial (más de 100 personas)

Parte 3 - Diagrama de flujo: d) Dibuja el diagrama de flujo del algoritmo anterior

Parte 4 - Abstracción: e) Realiza una abstracción del evento, identificando solo 5 elementos esenciales

Entregables:

- Documento con descomposición y patrones
- Pseudocódigo completo
- Diagrama de flujo
- Abstracción del problema

RETO 2 "CALCULADORA INTELIGENTE"***

Situación: Debes diseñar una calculadora que realice diferentes operaciones según las necesidades del usuario.

Requisitos:

Parte 1 - Análisis: a) Identifica los componentes de hardware necesarios para una calculadora b) Describe qué software (sistema operativo y aplicación) necesitaría

Parte 2 - Algoritmo principal: c) Crea un algoritmo en pseudocódigo que:

- Muestre un menú con opciones:
 1. Sumar
 2. Restar
 3. Multiplicar
 4. Dividir
 5. Calcular área de figuras (círculo, cuadrado, triángulo)
 6. Salir
- Según la opción elegida:
 - Pida los datos necesarios
 - Realice la operación
 - Muestre el resultado
- Permita realizar múltiples operaciones hasta elegir "Salir"

Parte 3 - Validación: d) Realiza pruebas de escritorio con al menos 5 casos diferentes:

- Suma de números positivos
- Resta que dé negativo
- Multiplicación por cero
- División (incluye qué pasa si divides entre cero)
- Cálculo de área

Parte 4 - Representación: e) Elige UNA de las operaciones y crea:

- Diagrama de flujo completo
- Identifica claramente: Entrada, Proceso y Salida

Parte 5 - Pensamiento computacional: f) Explica cómo aplicaste:

- Descomposición: _____
- Reconocimiento de patrones: _____
- Abstracción: _____
- Algoritmos: _____

Entregables:

- Análisis de hardware y software
- Pseudocódigo completo con menú
- Tabla de pruebas de escritorio
- Diagrama de flujo de una operación
- Reflexión sobre pensamiento computacional

BIBLIOGRAFÍA

Bibliografía Fundamental:

1. **Joyanes Aguilar, L.** (2008). *Fundamentos de programación: Algoritmos, estructura de datos y objetos*. 4ª edición. McGraw-Hill.
2. **Wing, J. M.** (2006). "Computational Thinking". *Communications of the ACM*, 49(3), 33-35.
3. **Barr, V., & Stephenson, C.** (2011). "Bringing computational thinking to K-12". *ACM Inroads*, 2(1), 48-54.
4. **Wikipedia.** "Generaciones de computadoras".
https://es.wikipedia.org/wiki/Generaciones_de_computadoras
5. **BBC Bitesize.** "Decomposition - GCSE Computer Science".
<https://www.bbc.co.uk/bitesize/guides/zjddqhv/revision/2>
6. **CS Unplugged.** "Pensamiento computacional".
<https://www.csunplugged.org/es/computational-thinking/>

7. **Tuts+.** "Los Fundamentos del Pensamiento Computacional".
<https://webdesign.tutsplus.com/es/the-basics-of-computational-thinking--cms-30172a>
8. **Educative.io.** "Hardware vs Software Components".
<https://www.educative.io/blog/hardware-vs-software-components-computer>

Bibliografía Complementaria:

9. **Ferrer, C.** (2022). *Algoritmos, diagramas de flujo y pseudocódigo: Los fundamentos*. Editorial Grupo ANAYA.
10. **Cassel, L. N., et al.** (2011). "Report from the education board: Computational thinking". *ACM Inroads*, 2(4), 24-27.
11. **Google.** "Exploring Computational Thinking".
<https://www.google.com/edu/ctl>
12. **Code.org.** "What is Computational Thinking?"
<https://code.org/curriculum>
13. **ISTE.** "Computational Thinking Competencies".
<https://iste.org/standards/computational-thinking>
14. **UNAM.** "Componentes básicos de hardware y software".
<https://repositorio-uapa.cuaed.unam.mx>
15. **Universidad de Alcalá.** (2020). *Introducción a la Informática y al Pensamiento Computacional*. Editorial Universidad de Alcalá.

Nota para el estudiante: Esta guía está diseñada para ser completada de manera progresiva. Se recomienda resolver los ejercicios en orden y no avanzar al siguiente tema sin haber comprendido el anterior. Los retos finales integran todos los conocimientos de la unidad.

© 2026 - Material educativo para la Unidad I: Introducción a las Ciencias de la Computación

**Diseñado por: Lic. Carlos M. Galdámez
AVC El Salvador**