

U1.1 - Mi primer programa



Índice 1

- U1.1 - Mi primer programa
- 1. La Programación
 - Definición
 - Ciclo de Vida de un Programa
- 2. El Ordenador
 - ¿Qué es un ordenador?

Índice 2

- 2. El Ordenador
 - Estructura de Von Neumann
- 3. ¿Qué es un Programa o Software?
 - Definición de Software
 - Tipos de Software
- 4. Relación Hardware - Software

Índice 3

- 4. Relación Hardware - Software
 - Hardware y Software
 - Elementos principales
- 5. Algoritmos
 - Definición de Algoritmo
 - Ejemplo de Algoritmo (Pseudocódigo)

Índice 4

- 5. Algoritmos
 - Diagramas de Flujo
- 6. Lenguajes de Programación
 - Definición
 - Clasificación y Usos

Índice 5

- Cierre de la Unidad
 - Resumen
 - Preguntas para Reflexionar

1. La Programación

Definición

- Proceso de crear programas para resolver problemas o automatizar tareas.
- Usa lenguajes de programación para dar instrucciones a un dispositivo.
- Requiere un traductor (compilador/intérprete) para convertir a lenguaje máquina.
- Lenguaje máquina: único que entiende la CPU.
- Microprocesadores ejecutan estas instrucciones de forma secuencial.

Ciclo de Vida de un Programa

- Comprender el problema.
- Recopilar requisitos.
- Planificar la solución.
- Diseñar el programa.
- Programar (codificar).
- Probar el programa.
- Desplegarlo para su uso.
- Mantener y mejorar el software.

2. El Ordenador

¿Qué es un ordenador?

- Máquina electrónica que procesa información.
- Puede ser digital o analógica.
- Ejecuta programas para resolver problemas lógicos o matemáticos.
- Posee memoria y métodos de tratamiento de datos.
- Realiza operaciones siguiendo instrucciones.

Estructura de Von Neumann

- Modelo clásico de funcionamiento del ordenador.
- Partes principales: CPU, memoria, dispositivos de entrada/salida.
- CPU ejecuta instrucciones de memoria.
- Memoria RAM almacena datos e instrucciones temporalmente.
- Discos y periféricos aportan almacenamiento y comunicación.

3. ¿Qué es un Programa o Software?

Definición de Software

- Conjunto de programas, datos y documentación.
- Da instrucciones al hardware sobre qué hacer y cómo.
- Permite realizar tareas específicas en dispositivos.
- Incluye desde sistemas operativos hasta aplicaciones.
- IEEE: software = programas + procedimientos + datos asociados.

Tipos de Software

- **De sistema:** sistema operativo, drivers.
- **De aplicación:** navegadores, ofimática, juegos.
- **De desarrollo:** compiladores, editores, depuradores.
- Todos cooperan entre sí.
- Los drivers permiten que el hardware funcione correctamente.

4. Relación Hardware - Software

Hardware y Software

- Hardware: parte física del ordenador.
- Software: conjunto de instrucciones que dirigen al hardware.
- Ambos se necesitan mutuamente.
- Hardware sin software: máquina inerte.
- Software sin hardware: no puede ejecutarse.

Elementos principales

- **Disco duro:** guarda programas y datos de forma permanente.
- **Memoria RAM:** almacén temporal de programas y datos en uso.
- **CPU:** ejecuta instrucciones y coordina el sistema.
- **Entrada/Salida (E/S):** teclado, ratón, pantalla, impresora.
- Disco duro también se considera periférico de E/S.

5. Algoritmos

Definición de Algoritmo

- Secuencia de pasos lógicos para resolver un problema.
- Es independiente del lenguaje de programación.
- Primero se diseña el algoritmo, luego se codifica.
- Todo algoritmo debe ser: preciso, definido y finito.
- Base fundamental de la programación.

Ejemplo de Algoritmo (Pseudocódigo)

```
Si la lámpara funciona entonces
    fin. # programa termina
Si no
    Si NO está enchufada entonces
        Enchufar.
    Si el foco está quemado entonces
        Reemplazar foco.
    Si sigue sin funcionar entonces
        Comprar nueva lámpara.
fin.
```

Diagramas de Flujo

- Representación gráfica de un algoritmo.
- Óvalos: inicio y fin.
- Rectángulos: acciones o procesos.
- Rombos: decisiones (sí/no).
- Flechas: orden de ejecución.
- Claros y visuales para entender procesos.

6. Lenguajes de Programación

Definición

- Lenguajes formales con reglas estrictas.
- Permiten expresar algoritmos en código ejecutable.
- Controlan flujo de ejecución y manipulan datos.
- Ejemplos: Python, Java, C, Kotlin, JavaScript.
- Traducidos a lenguaje máquina mediante compiladores o intérpretes.

Clasificación y Usos

- Multiplataforma: Kotlin, Java, C#.
- Desarrollo web: JavaScript, PHP, Python.
- Sistemas operativos: C, C++.
- Ciencia de datos: Python, R.
- Cada lenguaje tiene fortalezas y contextos ideales.

Cierre de la Unidad

Resumen

- Programar = transformar problemas en soluciones ejecutables.
- Ordenadores siguen el modelo de Von Neumann.
- Software guía al hardware en las tareas.
- Algoritmos = base de la programación.
- Pseudocódigo y diagramas ayudan a planificar.
- Lenguajes formales convierten algoritmos en programas.

Preguntas para Reflexionar

- ¿Qué lenguajes de programación conoces?
- ¿Cuál te resulta más interesante y por qué?
- ¿Qué ventajas tiene escribir pseudocódigo antes de programar?
- ¿Cómo ayuda un diagrama de flujo a entender un problema?
- ¿Crees que hardware y software podrían existir uno sin otro?

Dudas



¡Gracias por su atención!



