

Instituto tecnológico superior de Martínez de la Torre

Fundamentos de programación

José lameiras pleites

Tema 1: Diseño algorítmico

Actividades de aprendizaje

- Glosario
- Tabla
- Resumen
- Antología

SeleneHerrera Hernandez
Angel Gabriel Gonzáles Vázquez
Alondra yoselin lino perez

Glosario

***Algoritmo:** se llama algoritmo a una secuencia de instrucciones u operaciones específicas que permiten controlar determinados procesos. Se trata de conjuntos finitos y ordenados de pasos, que nos conducen a resolver un problema o tomar una decisión.

***Programa:** Un programa es un conjunto de pasos lógicos escritos en un lenguaje de programación que nos permite realizar una tarea específica. El programa suele contar con una interfaz de usuario, es decir, un medio visual mediante el cual interactuamos con la aplicación.

***Programación:** es el proceso de dar instrucciones a una computadora para que pueda resolver un problema o realizar una tarea específica.

***Paradigmas de programación:** son los métodos o estilos de programación que se utilizan para desarrollar software y resolver problemas. Se trata de un conjunto de principios que definen cómo se implementan los programas y que generan códigos estructurados de manera diferente

***Tipo de dato:** atributo que indica al sistema informático cómo interpretar un dato y cómo almacenar los valores que puede contener:
Indica la clase de datos que se va a manejar
Impone restricciones en los datos, como qué valores pueden tomar y qué operaciones se pueden realizar
Determina cuánta memoria se necesita para almacenar los valores

***Constante:** es un número, un carácter o una cadena de caracteres que se puede utilizar como valor en un programa.

***Variable:** Una variable representa un contenedor o un espacio en la memoria física o virtual de una computadora, donde se almacenan distintos tipos de datos durante la ejecución de un programa. A cada variable se le asigna un nombre descriptivo o un identificador que se refiere al valor guardado.

***Operadores:** representa un símbolo que permite realizar operaciones aritméticas, relacionar elementos o hacer preguntas donde se involucra más de una condición.

***Diagrama de flujo:** es una representación visual de las tareas y actividades que se deben realizar para lograr un objetivo, y que se usa para describir un proceso, sistema o algoritmo informático

***Pseudocódigo:** pseudocódigo es una descripción informal y detallada de un algoritmo o programa, que se escribe en un lenguaje natural estructurado.

Tabla

tipo de dato	Descripción	ejemplo
Int	Enteros, números, sin parte decimal	2,10,-5,0
Float	Número de punto flotante que contiene una parte de decimal	3.14,-0.008,2.0
Str	Cadena de carácter o texto	""Hola" "puthon"
bool	valores, booleanos, representan verdadero o falso	"true" "False"
List	Lista, ordenada, immutable de elementos	[1,2,3][]"a" "b"]
Tuple	Dupla similar a la lista, pero no immutable, no se puede modificar después de su creación	(1, 2 ,3)(""a""b")
Set	Conjunto no ordenado de elementos únicos	{1,2,3}{"a""b"}
dict	diccionario, colección de pares, clave-valor	{ "nombre": ""Ana", " edad": "34"} }

NoneType	representa la ausencia de valor o un valor nulo	None
Complex	Números complejos, comparte real e imaginaria	1+2j,3-5j
Range	Secuencia inmutable de números utilizadas, generalmente para enterrar en bucle	Range (0,10)
Bytes	Secuencia de bytes inmutable	b "hola"
bytearray	secuencia de bytes mutable	bytearray(b"hola")

Resumen

Simbología de las Herramientas para Representar un Algoritmo

1. Diagrama de Flujo

Es una representación gráfica de un proceso o algoritmo, donde cada paso está representado por símbolos conectados entre sí para mostrar el flujo de control. A continuación, se presentan los principales símbolos utilizados:

Símbolo	Descripción	Ejemplo de uso
---------	-------------	----------------

Óvalo	Inicio o fin del diagrama.	Inicio o Fin
-------	----------------------------	--------------

Rectángulo	Proceso o instrucción a ejecutar.	Calcular la suma de dos números
------------	-----------------------------------	---------------------------------

Rombo	Decisión o condición que determina el flujo.	¿Es el número mayor que 10?
-------	--	-----------------------------

Paralelogramo	Entrada o salida de datos.	Leer el valor de A o Imprimir resultado
---------------	----------------------------	---

Flechas	Indican la dirección del flujo del proceso.	Conectar símbolos para mostrar el orden de las operaciones
---------	---	--

Círculo pequeño	Conector utilizado para enlazar secciones del diagrama.	Usado cuando el diagrama continúa en otra parte de la hoja
-----------------	---	--

Ejemplo de un diagrama de flujo:

Vamos a representar un algoritmo simple que determina si un número es mayor que 10:

1. Inicio

2. Leer un número

3. ¿El número es mayor que 10?

- Si sí, imprimir "El número es mayor que 10".
- Si no, imprimir "El número es menor o igual a 10".

4. Fin

2. Pseudocódigo

El pseudocódigo es una forma escrita para representar un algoritmo, que usa una mezcla de lenguaje natural y términos de programación, pero sin la sintaxis estricta de un lenguaje de programación real.

Reglas comunes:

- Usar un lenguaje claro y sencillo.
- Incluir instrucciones condicionales (si, entonces) y bucles (mientras, para).
- Representar entradas/ salidas de datos.

Ejemplo de pseudocódigo para el mismo algoritmo: Inicio

```
Inicio
  Leer número
  Si número > 10 Entonces
    Imprimir "El número es mayor que 10"
  Sino
    Imprimir "El número es menor o igual a 10"
Fin
```

- **Diagrama de flujo:** Usa símbolos gráficos para representar visualmente los pasos de un algoritmo, ideal para visualizar la estructura general y el flujo de control.
- **Pseudocódigo:** Es una representación textual del algoritmo que permite describir los pasos de manera detallada, pero sin las restricciones de un lenguaje de programación específico.

Ambas herramientas son útiles para planificar, diseñar y comunicar la lógica antes de escribir código.

seudocódigo

Problema:1 preparar cafe

```
Inicio
  Llenar la cafetera con agua
  Colocar el café en el filtro
  Encender la cafetera
  Si el café está listo Entonces
    Verter el café en una taza
    Añadir azúcar o leche (si se desea)
    Tomar el café
  Fin Si
Fin
```

Problema 2: encender la computadora

```
Inicio
  Presionar el botón de encendido
  Si la computadora no enciende Entonces
    Verificar conexión a la corriente
    Si está bien conectada Entonces
      Presionar el botón nuevamente
    Fin Si
  Sino
    Esperar a que el sistema operativo cargue
    Introducir usuario y contraseña
  Fin Si
Fin
```

Problema 3: Ir al súper mercado

```
Inicio
  Hacer una lista de compras
  Ir al supermercado
  Para cada artículo en la lista Hacer
    Encontrar el artículo
    Poner el artículo en el carrito
  Fin Para
  Pagar en la caja
  Volver a casa
Fin
```

Problema 4: lavar la ropa

```
Inicio
  Separar ropa por colores
  Poner la ropa en la lavadora
  Añadir detergente
  Encender la lavadora
  Si la lavadora termina el ciclo Entonces
    Sacar la ropa
    Tender o meter en la secadora
  Fin Si
Fin
```

Problema 5: hacer una llamada telefónica

```
Inicio
  Desbloquear el teléfono
  Buscar el contacto o marcar el número
  Presionar el botón de llamada
  Si la llamada es contestada Entonces
    Hablar con la persona
  Sino
    Intentar más tarde
  Fin Si
Fin
```


Conclusión

Esta antología cubre problemas cotidianos utilizando pseudocódigo para estructurar la solución. Estos ejemplos sirven para practicar la lógica detrás de la programación y la estructuración de tareas en problemas de la vida diaria. son problemas tan sencillos q los haces y aveces ni cuenta te das