

Diagramas de flujo y pseudocódigo

Contenido de la clase

- ¿Qué es un algoritmo?
- Pseudocódigo
- Algoritmos recursivos
- Analizando herramientas para la construcción de pseudocódigo (PSEINT)
- Creando mis primeros algoritmos a través de pseudocódigo
- Diagramas de flujo y herramientas para su construcción

¿Qué es un algoritmo?

Un algoritmo se define como el conjunto de instrucciones que son seguidas y realizadas por una tarea particular.
Un algoritmo está compuesto por un conjunto de pasos finitos, los cuales requieren una o más operaciones.

Los algoritmos deben satisfacer los siguientes criterios:

- Entrada: Puede ser una cadena de texto, un numero, un símbolo.
- Salida: Resultado de los procesos realizados por el algoritmo, es lo esperado por el usuario.
- Definición: Las instrucciones deben ser claras, ordenadas y no ambiguas.
- Finitud: Todos los algoritmos deben tener un número determinado de pasos para realizarse. Las instrucciones son limitadas.
- Efectividad: Cada instrucción debe ser básica para que pueda ser llevada a cabo, en principio por una persona usando solo lápiz y papel.

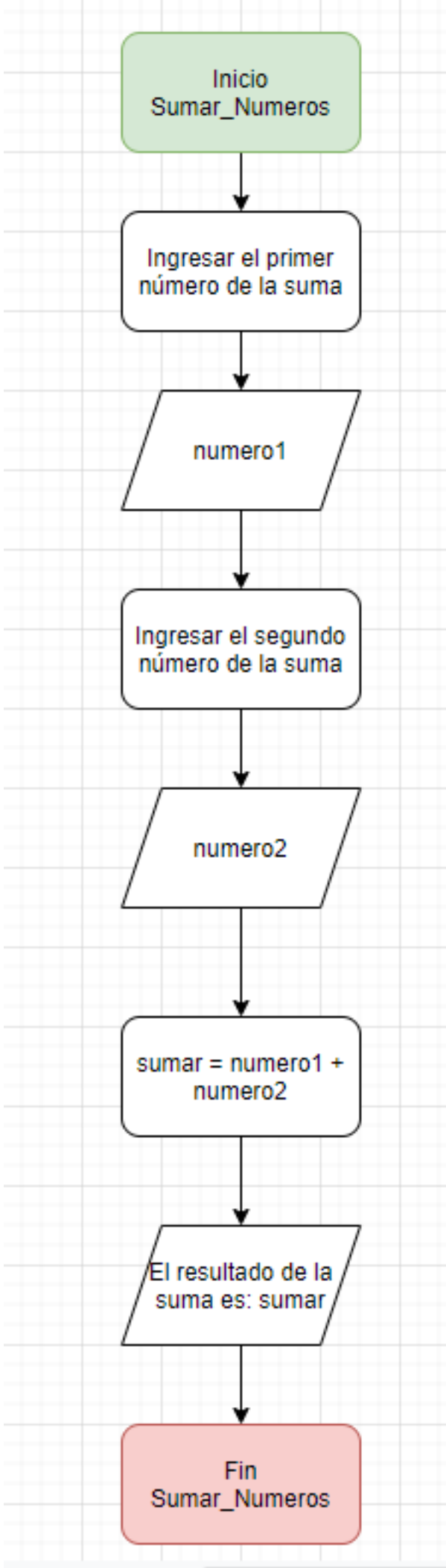
Pseudocódigo

Un pseudocódigo es la manera de representar un algoritmo ya sea en lenguaje natural o en representaciones gráficas a través de diagramas de flujo. Un pseudocódigo también puede escribirse bajo la sintaxis propia de un lenguaje de programación. La construcción de un algoritmo por medio de pseudocódigo facilita el paso a un lenguaje de programación, para la creación de un programa.

Ejemplo de un pseudocódigo

```
Algoritmo Sumar_Numeros
  Escribir "Ingresar el primer número de la suma"
  Leer numero1
  Escribir "Ingresa el segundo número de la suma"
  Leer numero2
  sumar = numero1 + numero2
  Escribir "El resultado de la suma es: " sumar
FinAlgoritmo
```

Representación del pseudocódigo en diagramas de flujo



Representación del pseudocódigo en lenguaje de programación python

```
In [1]: print("Ingresa el primer número de la suma")
numero1 = int(input("numero1: "))
print("Ingresa el segundo número de la suma")
numero2 = int(input("numero2: "))

sumar = numero1 + numero2
print("el resultado de la suma es: ")
print(sumar)
```

Ingresa el primer número de la suma
numero1: 4
Ingresa el segundo número de la suma
numero2: 5
el resultado de la suma es:
9

En pseudocódigo podemos utilizar todos los operadores matemáticos como

- Suma (+)
- Resta (-)
- Multiplicación (*)
- División (/)
- Potencia (^)
- Modulo (%)

También es posible utilizar los comparadores matemáticos

- Mayor que (>)
- Menor que (<)
- Igual (=)
- Menor o igual (<=)
- mayor o igual (>=)

Recursividad

En algoritmos, recursividad es el llamado que un algoritmo o funcionalidad realiza a sí mismo un número de veces finito. Todos los algoritmos tienen un punto de parada o de fin

Recursividad ≠ Iteración

Iteración corresponde a repetir un conjunto de sentencias de un algoritmo para lograr una acción o proceso

```
Funcion facto*factorial(number)
  si (number=1) Entonces
    facto*1
  SiNo
    facto*number*factorial(number-1)
  FinSi

Fin Funcion

Algoritmo Ejecutar_Factorial
  resultado_factorial*factorial(3)
  Escribir resultado_factorial
FinAlgoritmo
```

Ejemplo de algoritmo con iteraciones

```
Algoritmo Validar_Numero
  Repetir
    Escribir "Escribir numero"
    Leer number
    Si number%2 ≠ 0 Entonces
      Escribir "Ese número no es divisible por 2"
    SiNo
      Escribir "Numero correcto"
    Fin Si
  Hasta Que number%2 = 0
  Escribir "Fin del algoritmo"
FinAlgoritmo
```

Creando mis primeros algoritmos a través de pseudocódigo

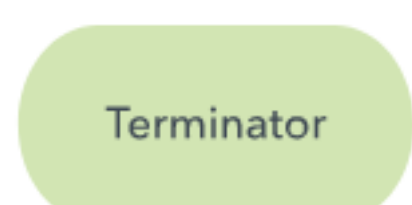
- Calcular el iva de un producto (19%)
- Comprobar que una contraseña es válida (coincidencia y tamaño) (Repetir)
- Mostrar en pantalla "-." al ingresar la letra A, "...." al ingresar la letra H, "---." al ingresar la letra Q de lo contrario "No es posible mostrar un valor" (Según)
- Ejecutar programa hasta que se haya completado el pago del salario a los empleados (Mientras)
- Validar que una persona visito un número de ciudades esperadas en sus vacaciones. Si el número de la lista es par, la ciudad fue visitada, de lo contrario no fue visitada (Para / Si - Entonces)

Diagramas de flujo y herramientas para su construcción

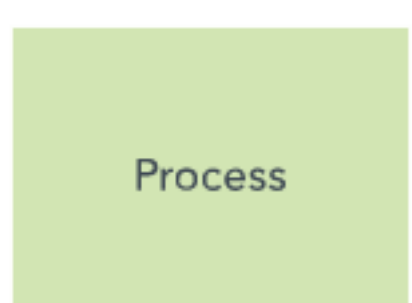
Un diagrama de flujo describe el proceso de un sistema o programa. Permite comprender como se comporta un sistema, como es la comunicación con cada componente o funcionalidad y a su vez como es la comunicación con los usuarios.

Símbolos o componentes de un diagrama de flujo

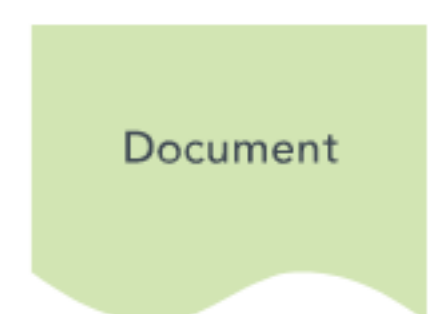
Inicio o fin de un diagrama:



Proceso o funcionalidad que realizará



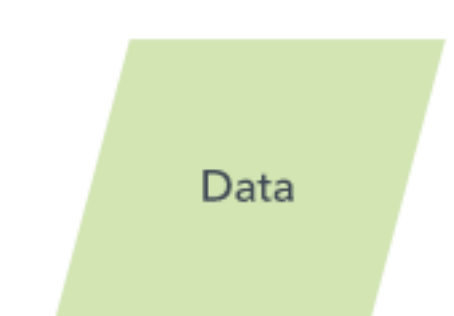
Resultado impreso y listo para mostrar al usuario



Tomar decisiones dentro del flujo



Permite la entrada de datos para diferentes operaciones en el flujo y a su vez entrega el resultado de una operación



Comunicación entre componentes del flujo



Referencias y links de apoyo

Libro: Fundamentals of computer algorithms, autor: Ellis Horowitz

PSEINT online: <https://www.rollapp.com/app/pseint>

PSEINT: <http://pseint.sourceforge.net>

Diagramas de flujo: <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo>

Herramientas para diseñar diagramas de flujo: <https://app.diagrams.net>