

Introducción a la programación

Introducción a la programación	1
¿Qué aprenderás?	2
Introducción	2
Algoritmos	3
Ejemplo	3
Existen tres formas de escribir un algoritmo	4
Diagramas de flujo	4
Símbolos de un diagrama de flujo	4
Ejemplo (1)	5
Ejemplo (2)	6
Ejercicio guiado: Cálculo de IVA	7
Pseudocódigo	8
Ventajas del pseudocódigo	8
Ejemplos de pseudocódigo	8
De pseudocódigo a diagrama de flujos	10
Ejercicio propuesto (1)	11



¡Comencemos!

¿Qué aprenderás?

- Emplear algoritmos básicos mediante técnicas de pseudocódigos y diagramas de flujo para resolver un problema informático.
- Comprender la sintaxis del pseudocódigo para construir nuestro primer sistema informático.

Introducción

Si vamos a la etimología de la palabra *Programación*, esta nos define un plan ordenado y organizado de una serie de pasos para realizar una acción. Por ejemplo, un *programa* de televisión está compuesto por una serie de pasos que se deben seguir para cumplir el objetivo de dicho programa. Apertura, presentación, comerciales, hablar de política, presentar al siguiente invitado, despedida y cierre.

Pues, hoy en día las tecnologías informáticas o de la información, nos permiten resolver casi cualquier problema lógico (que se puede resolver por una serie de pasos) y así lograr optimizar en dar solución a ciertos problemas. Por ejemplo, antes de que existieran las calculadoras, realizar cálculos aritméticos complejos resultaba muy costoso.

En resumen, mediante la programación de software, podemos dar solución a esos problemas que resultan costosos de hacer en la vida diaria. Solo necesitamos conocer el flujo (la serie de pasos para resolver un problema de inicio a fin) y estos llevarlos a un lenguaje que nuestra máquina (computador, celular) pueda conocer.

¡Vamos con todo!



Algoritmos

Se llaman *algoritmos* a la forma o técnica que tiene un programador de escribir o representar a una serie de pasos estructurados de manera lógica y ordenada para resolver un problema determinado.

Ejemplo

Problema:

Hacer un huevo frito.

Pasos para resolver el problema:

1. Encender el fuego en la cocina.
2. Poner la paila en el fuego.
3. Colocar aceite en la paila.
4. Romper el huevo.
5. Colocar el huevo en la paila.
6. Colocar sal al huevo.
7. Si se desea huevo revuelto.
 - a. Revolver.
8. Si se desea huevo entero.
 - a. Tapar la paila.
9. Esperar dos minutos.
10. Apagar el fuego.
11. Servir.

Lo anterior son los pasos que debemos pensar antes de escribir el algoritmo que representará de forma gráfica o escrita los pasos de la solución descrita.

Existen tres formas de escribir un algoritmo

1. Diagramas de flujo

Un diagrama de flujo es una forma de representar de forma gráfica los pasos a seguir para solucionar un problema. Al ser una forma gráfica, se debe tener en consideración la simbología a utilizar para representar cada paso que formará el algoritmo.

Símbolos de un diagrama de flujo

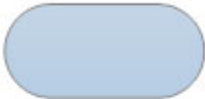

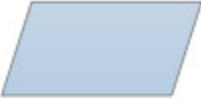


Símbolo	Nombre	Función
	Inicio / Final	Representa el inicio y el final de un proceso
	Línea de Flujo	Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción.
	Entrada / Salida	Representa la lectura de datos en la entrada y la impresión de datos en la salida
	Proceso	Representa cualquier tipo de operación
	Decisión	Nos permite analizar una situación, con base en los valores verdadero y falso

Imagen 1. Símbolos de un diagrama de flujo

Fuente: [Smartdraw](https://www.smartdraw.com).

Ejemplo (1)

Sumar dos números y mostrarlos en la salida.

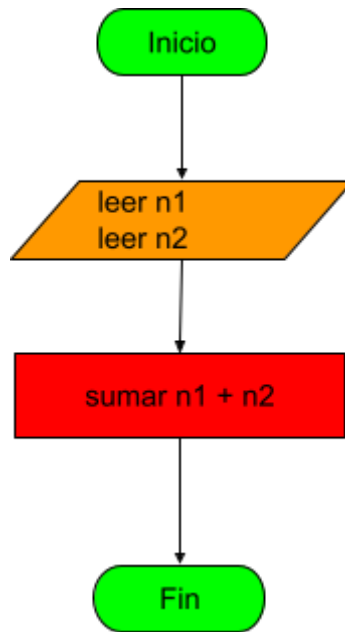


Imagen 2. Diagrama de flujo que suma dos números
Fuente: Desafío Latam.

Diagrama de flujo de la imagen que especifica los pasos a seguir para sumar dos números. El diagrama representa dos entradas de datos mediante las variables n1 y n2 (símbolo naranja), las cuales guardarán el valor de cada número ingresado y el proceso de sumar (símbolo rojo), quien se encarga de representar el proceso de sumar dos números en cuestión.

Ejemplo (2)

Ingresar dos números y mostrar el mayor de ellos.

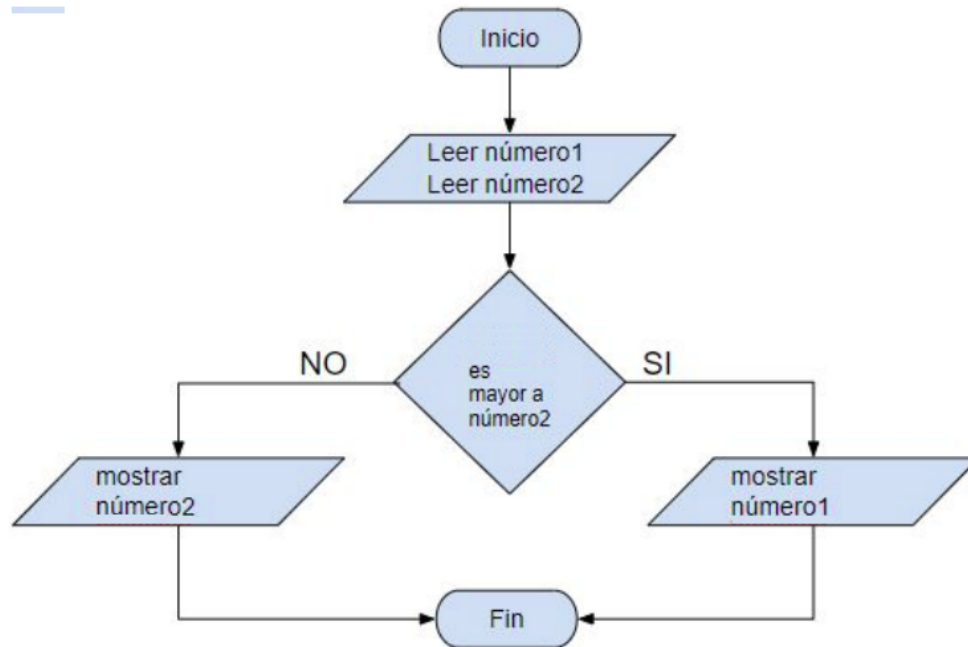


Imagen 3. Diagrama de Flujo que evalúa el mayor de dos números.

Fuente: Desafío Latam.

El diagrama de la imagen representa el algoritmo que da solución al problema de identificar cuál es el número mayor y mostrarlo por pantalla. En este flujo se implementa el rombo de decisión.

Ejercicio guiado: Cálculo de IVA

Crear diagrama de flujos de algoritmo que permita calcular el IVA en base al monto total de una factura.

- **Paso 1:** Se debe ingresar o leer el monto bruto de la factura.
- **Paso 2:** Calcular el IVA de la factura en base a la siguiente función:
$$\text{iva} = \text{montoBruto} * 0,19.$$
- **Paso 3:** Calcular el monto total o monto líquido de la factura en base a la siguiente función:
$$\text{montoLiquido} = \text{montoBruto} + \text{iva}.$$
- **Paso 4:** Se debe mostrar monto bruto, iva y montoLíquido.

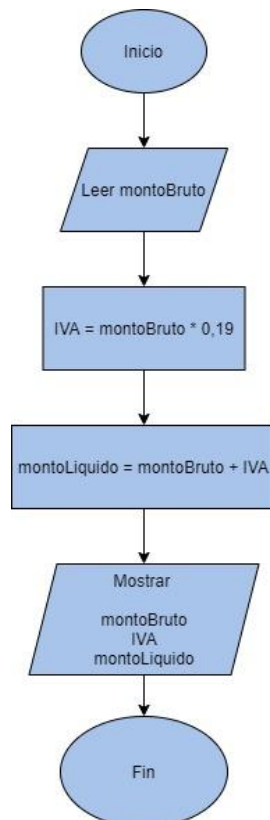


Imagen 4. Cálculo de IVA.
Fuente: Desafío Latam.

2. Pseudocódigo

Es otra forma de representar la solución de un algoritmo, de la manera más detallada posible, diseñada para la lectura, el cual puede ser parecida al lenguaje que posteriormente se utilizará para la codificación del mismo.

Ejemplo:

```
Algoritmo Suma
  Leer valor1
  Leer valor2
  Mostrar valor1 + valor2
FinAlgoritmo
```

En pseudocódigo se utiliza la instrucción **leer** para especificar que el usuario tiene que ingresar un valor y **mostrar** para imprimir el valor en pantalla.

Ventajas del pseudocódigo

El pseudocódigo, al igual que el diagrama de flujo, nos permite pensar en términos independientes al lenguaje de programación y concentrarnos en describir lo que estamos tratando de hacer y los pasos necesarios, en lugar de cómo lograrlo.

Ejemplos de pseudocódigo

Ejemplo 1

Crear un programa que permita calcular el área de un cuadrilátero, sabiendo que esta se calcula con la siguiente fórmula: $\text{área} = \text{base} \times \text{altura}$. Se debe leer por pantalla los valores para realizar el cálculo.

```
Algoritmo AreaCuadrilatero
  Leer base
  Leer altura
  Mostrar base * altura
FinAlgoritmo
```


En el ejemplo anterior, el signo `*` nos permite representar una multiplicación. Así mismo, si queremos realizar una división, podemos utilizar el signo `/`.

Ejemplo 2

Problema: Se necesita leer dos fechas de nacimiento de dos personas y mostrar por pantalla cuál de las dos personas es mayor.

```
Algoritmo QuienEsMayor
  Leer nombrePersona1
  Leer fechaNacimiento1

  Leer nombrePersona2
  Leer fechaNacimiento2

  si fechaNacimiento1 es mayor a fechaNacimiento2 entonces
    Mostrar nombrePersona1 + " es mayor a " + nombrePersona2
  si no
    si fechaNacimiento2 es mayor a fechaNacimiento1 entonces
      Mostrar nombrePersona2 + " es mayor a " + nombrePersona1
    si no
      Mostrar "Las personas tienen la misma edad"
FinQuienEsMayor
```

En este ejemplo, se ha utilizado el operador `+`, el cual nos permite, aparte de hacer una suma aritmética entre dos números, realizar una concatenación de dos cadenas de caracteres (palabras). Por ejemplo:

- El usuario ingresa en `nombrePersona1` (a esto llamamos variable) el nombre de "Pepito". Entonces si queremos escribir por pantalla Pepito clavó un clavito, esto sería: `nombrePersona1 + " clavó un clavito"`.

3. De pseudocódigo a diagrama de flujos

Como nos podemos dar cuenta, un algoritmo es una serie de pasos o instrucciones secuenciales que se escriben de inicio a fin. Pues, al ser el pseudocódigo y los diagramas de flujo dos formas de representar un algoritmo, un mismo problema podemos escribirlo de ambas formas. Por ejemplo, el problema de saber quién de dos personas es mayor en base a su fecha de nacimiento:

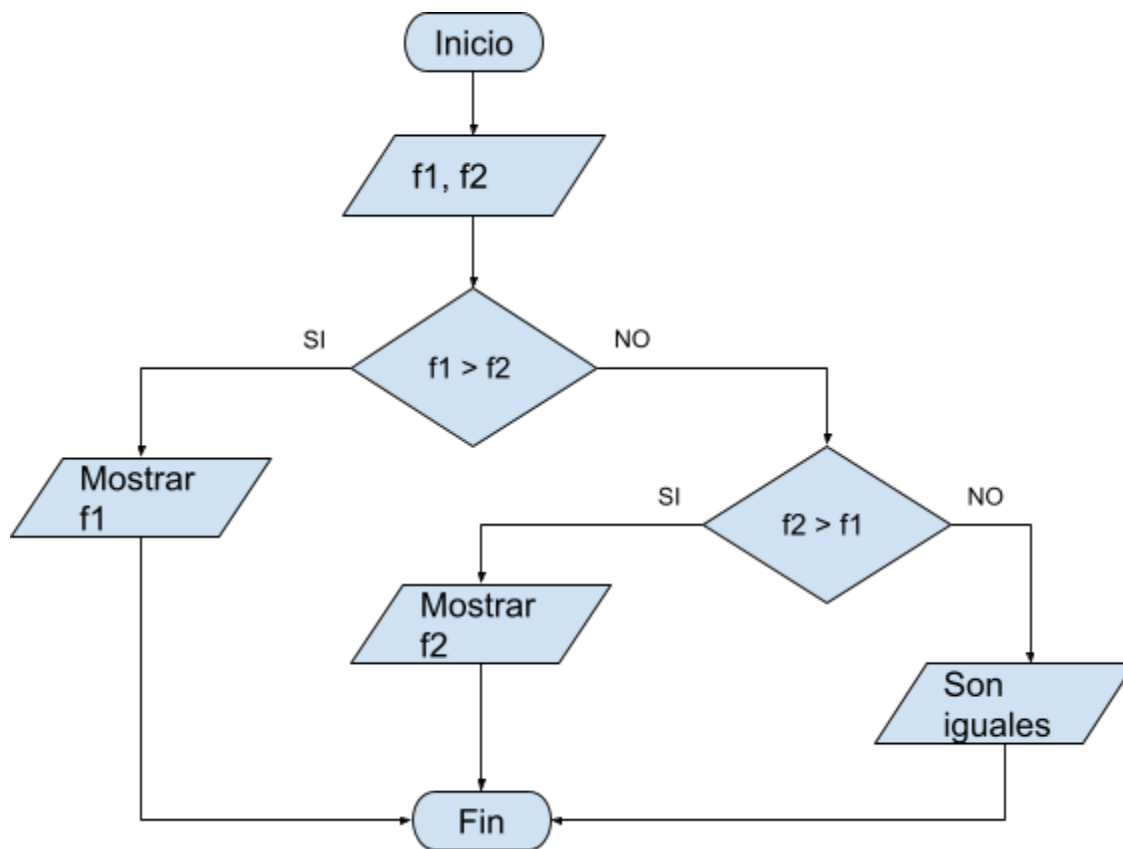


Imagen 5. Diagrama de flujo quién es mayor.

Fuente: Desafío Latam.

El diagrama de flujo de la imagen, evalúa dos fechas de nacimiento ingresadas e identifica cuál de ellas es mayor o bien si son iguales. Nótese que las palabras del flujo en este caso al ser un diagrama gráfico, no es tan literal. Es decir, se utilizan signos $>$ o $<$ para decir si es mayor o menor.