

# Curso de Programación

---

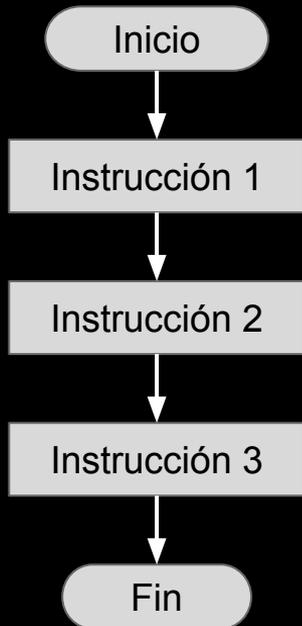
Orientado a videojuegos

Empezamos en minutos...

Maestro Jedi: Daniel Delgado  
Duración: al menos 10 horas.  
El resto depende de ti...

# Estructuras Algorítmicas o de Programación

Como ya hemos visto, en informática los algoritmos, la secuencia de pasos lógicos, bien definidos y finitos, se plantean de forma estructurada y estas estructuras básicas son tres: lineales, condición o decisión, y las cíclicas, de bucle o repetitivas.



## Lineal o secuencial:

Es aquella donde las instrucciones se ejecutan linealmente, en el orden que fueron escritas desde el inicio del programa hasta el final.

Todos los ejemplos y ejercicios que hemos hecho son algoritmos lineales o secuenciales.

Ya hemos visto todas las herramientas básicas que disponemos para hacer algoritmos lineales:

- Definición y Asignación de Variables: Let Number: A = 0;
- Entrada de Datos: Input A;
- Salida de Datos: Print A \* 2;

# Lógica mental, nuestra mejor aliada.

En los ejemplos y ejercicios que ya hemos realizado hemos usado nuestra lógica en silencio para resolver problemas como promedio de números, operaciones matemáticas, hemos jugado con cadenas de caracteres.

El proceso mental que hemos hecho es el que nos enseñaron en matemáticas muy básicas para resolver problemas.

- ° Definir datos que necesitamos.
- ° Usar fórmulas y procesos para resolver el problema. Verificación de resultados con cualquier dato.
- ° Presentar el resultado obtenido.

Nuestra mente ya tiene herramientas lógicas para resolver estos problemas. Así ya podemos crear algoritmos y programas en PseudoCódigo rápidamente para problemas básicos. Veamos cómo hemos pensado hasta ahora.

# Pasos lógicos mentales para Análisis y Solución de Problemas con Algoritmos Lineales

**Problema:** Se necesita un algoritmo que pida al usuario dos números cualquiera y los sume.

Leemos el problema hasta entenderlo bien, y entonces pasa la magia:

Paso 1: apenas leemos la palabra suma, nuestra mente nos dibuja una suma básica.  $2 + 2 = 4$

Paso 2: cuando leemos “que pide al usuario” sabemos que necesitamos leer datos dados por el usuario.

Paso 3: Al leer “Dos números” sabemos qué datos vamos a pedir al usuario y que necesitamos dos variables para esos números.

Paso 4: Necesitamos entonces una fórmula (proceso) para resolver el problema:  $\text{solución} = \text{numero1} + \text{numero2}$ . Verificamos que este proceso nos da la solución siempre a cualquiera dos números.

Paso 5: Aplicamos todo lo enseñado hasta ahora, para crear un PseudoCódigo

Begin Suma

```
Var Number: number1 = 0;
```

```
Var Number: number2 = 0;
```

```
Var Number: suma = 0;
```

```
Print “Escriba un número”;
```

```
Input number1;
```

```
Print “Escriba otro número”;
```

```
Input number2;
```

```
suma = number1 + number2;
```

```
Print “El resultado de la suma de los
```

```
dos números es ” & suma;
```

End Suma

## La práctica y repetición hacen al maestro...

La gracia de un algoritmo y PseudoCódigo (futuro programa) es que resuelva siempre el problema, sin importar los datos de entrada o variables y constantes.

Y estos problemas aún para un algoritmo lineal pueden ser bastantes complejos, con lo cual debemos pensar siempre siguiendo la misma manera de pensar y teniendo en mente que no será nuestro cerebro que haga la operación siempre, sino una computadora. Entonces necesitamos:

Saber qué datos de entrada tenemos, ¿son pedidos al usuario o ya vienen con el programa? Variables y Constantes

Cuál fórmula o qué proceso (herramientas) debemos aplicar a esos datos, para resolver el problema. ¿Se cumple y soluciona siempre o solo en algunos casos? Verificar.

Presentar la solución siempre o almacenarla. (información)

## Ejemplo:

Algoritmo para el cuadrado de un número pedido al usuario.

Datos de entrada: un número cualquier pedido al usuario.

Proceso: El cuadrado de un número es el número multiplicado por si mismo.

Ejemplo:  $6^2 = 6 \times 6 = 36$

Datos de salida: el cuadrado del número.

Begin cuadrado

Var Number: number = 0;

Print “Introduzca un número”;

Input number;

Print “El cuadrado del número ” &  
number & “ es ” & number\*number;

End cuadrado

## Otro más...

Calcular el área y el perímetro de un círculo.

Para esto necesitamos fórmulas matemáticas del área y el perímetro de un círculo.

Estas son:

$$A = \text{Pi} \times \text{Radio}^2$$

$$P = 2 \times \text{Pi} \times \text{Radio};$$

Sabiendo que Pi es una constante igual a 3.1415 aprox., lo único que le debemos pedir al usuario es... el radio del círculo.

Datos de entrada: Pi y Radio del círculo

Procesos:  $A = \text{Pi} \times \text{Radio}^2$  y  $P = 2 \times \text{Pi} \times \text{Radio}$ ;

Datos de salida: Área y Perímetro del círculo.

```
Begin AreaPerimetroCirculo
    Var Number: pi = 3.1415;
    Var Number: radio = 0;
    Print "Introduzca el radio del círculo";
    Input radio;
    Print "El área calculada de su círculo es
" & pi * (radio * radio);
    Print "El perímetro de su círculo es " &
2 * pi * radio;
End AreaPerimetroCirculo
```

Tarea hacer un algoritmo en pseudocódigo que calcule el área y el perímetro de un rectángulo, y de un triángulo equilátero.

# ... Y la complejidad hace genios! (Solo se necesitan horas de vuelo)

Se tiene que la empresa 043 Imagine necesita saber cuántas unidades de sus juegos debe vender para tener un 30% de ganancia, basado en el monto que costó el desarrollo total del juego y un precio de \$20 por juego.

Datos de entrada:

30% de ganancia

Costo de desarrollo total. Debe ser pedido al usuario.

Precio 20\$

Datos de salida: Unidades que se deben vender.

Creemos un ejemplo:

Si el costo del desarrollo del juego en total fueron, por ejemplo, \$180.000.

Entonces multiplicamos  $180.000 \times 0.3$  (30% de ganancia)

El resultado lo sumamos a 180.000 para obtener el dinero total la empresa deberá recibir para obtener su ganancia del 30% con su costo de desarrollo.

$180.000 \times 0.3 + 180.000 = \$234.000$

Para saber cuántas unidades debemos vender dividimos el resultado anterior, entre 20 dólares que será el precio de cada unidad.

$\$234.000 / \$20 = 11.700$  unidades que deberán ser vendidas para obtener una ganancia del 30% sobre el monto gastado en el desarrollo.

El proceso completo verificado es

$\text{unidadesNecesariasVender} = (\text{costoTotalDesarrollo} \times 0.3 + \text{costoTotalDesarrollo}) / 20$

Begin VenderUnidades

Var Number: ganancia = 0.3;

Var Number: precio = 20;

Var Number: costoDesarrollo = 0;

Print "Introduzca el costo del desarrollo";

Input costoDesarrollo;

Print "Ganancia del 30% sobre su costo: " & costoDesarrollo \* 0.3;

Print "Se debe recibir en total: " & costoDesarrollo \* 0.3 + costoDesarrollo;

Print "El número de unidades a vender sería: " &  $(\text{costoTotalDesarrollo} \times 0.3 + \text{costoTotalDesarrollo}) / 20$ ;

End VenderUnidades

Tarea: cambiar este programa para que calcule el precio por unidad. El usuario deberá introducir el costo total de desarrollo, y el número de unidades a vender.

Luis ejercicio Cálculo Precio de Venta

Datos de Entrada:

30% ganancia.

costo de desarrollo total: Debe ser pedido al usuario.

Unidades a Vender: Debe ser pedido al usuario.

datos de salida: precio de venta al público.

Begin PrecioVenta

Var Number: Ganancia = 0.3;

Var Number: costoDesarrollo = 0;

Var Number: unidadesVender = 0;

Input "Introduzca el costo del desarrollo", costoDesarrollo;

Input "Introduzca el número de unidades a vender",

unidadesVender;

Print "Ganancia del 30% sobre su costo es" &

costoDesarrollo \* Ganancia;

Print "Se debe recibir un total de" & costoDesarrollo \*

Ganacia + costoDesarrollo;

Print "El precio de venta al publico es" & (costoDesarrollo

\* Ganacia + costoDesarrollo) / unidadesVender;

End PrecioVenta

Mario ejercicio Cálculo Precio de Venta

Begin PrecioDeVenta

Var Number: ganancia = 0.3;

Var Number: costo\_Desarrollo = 0;

Var Number: unidades\_Vender = 0;

Input "Introduzca el costo del desarrollo", costo\_Desarrollo;

Input "Introduzca el número de unidades a vender",  
unidades\_Vender;

Input "Ganacia del 30% sobre su costo es", costo\_Desarrollo \*  
ganancia;

Input "Se debe recibir un total de", costo\_Desarrollo \* ganacia  
+ costo\_Desarrollo;

Input "El precio de venta para el publico es", (costo\_Desarrollo  
\* ganacia + costo\_Desarrollo) / unidades\_Vender;

End PrecioDeVenta

# Realicemos un algoritmo lineal complejo...

De un número de 4 cifras sacar la cifra que representa la unidad del número, la decena, la centena y la unidad de mil, o unidad de millar.

Dato de entrada: un número de 4 cifras.

Procesos: calcular unidad de mil, centenas, decenas y la unidad.

Datos de salida: unidad de mil, centenas, decenas y la unidad del número dado.

Tenemos el número 1.043. Fácilmente, con nuestros ojos podemos decir que la unidad de mil está representada en el 1. Luego pasamos la vista a las centenas que son el 0. Las decenas están representadas en el 4 y por último la unidad es el 3.

Ahora con lógica de aritmética de programación hagamos el algoritmo que “vea” al número y lo divida en esas partes, sin error.

Lo que el 1 quiere decir es que está multiplicando al 1000

Lo que el 0 quiere decir es que está multiplicando al 100

Lo que el 4 quiere decir es que está multiplicando al 10

Y lo que el 3 quiere decir es que está multiplicando al 1

**El número 1043 se puede escribir como  $1043 = 1 \times 1000 + 0 \times 100 + 4 \times 10 + 3 \times 1$**

**El número 6258 se puede escribir como  $6258 = 6 \times 1000 + 2 \times 100 + 5 \times 10 + 8$  y así todos los números de 4 cifras.**

Saquemos primero la unidad de mil del número. Debería dar 1. **UnidadMillar =  $1043 \setminus 1000$ . ( \ nos da el entero de una división) = 1**

Para sacar las centenas debemos deshacernos de la unidad de mil. Este número puede ser un número entre 1 y 9. Por lo cual debemos entonces multiplicar el resultado anterior por 1000. En nuestro ejemplo la unidad de mil es  $1 \times 1000$  es 1000. Y luego restar de nuestro número completo, estos 1000. Y para tener las centenas simplemente dividir este número de la resta anterior entre 100. **Centenas =  $(1043 - (1 * 1000)) \setminus 100$**

Hagamos el ejemplo:  $(1043 - 1000) \setminus 100 = 43 \setminus 100 = 0.43$  (el \ toma solo el 0)

Hagamos las decenas siguiendo el mismo pensamiento lógico anterior.

Necesitamos deshacernos de la unidad de mil y de las centenas. Para esto igual multiplicamos nuestra unidad de mil por 1000. Las centenas que nos dio en el paso anterior multiplicadas por 100. Y lo sumamos. El resultado debe ser restado de nuestro número. **Decenas =  $(1043 - (1 * 1000 + 0 * 100)) \setminus 10 = 4.3$  (El \ toma solo el 4)**

Por último la unidad. Necesitamos como ya hemos visto, deshacernos de la unidad de mil, de las centenas, y de las decenas. Para esto igual multiplicamos nuestra unidad de mil por 1000. Las centenas que nos dio en el paso anterior multiplicadas por 100. Las decenas del paso anterior por 10. Y sumamos esos 3 resultados. Luego lo restamos de nuestro número y tendremos las unidades. **Unidades =  $(1043 - (1 * 1000 + 0 * 100 + 4 * 10)) = 3$**