



INSTITUTO POLITECNICO NACIONAL
UNIDAD PROFECIONAL INTERDISCIPLINARIA DE INGENIERIA CIENCIAS
SOCIALES Y ADMINISTRATIVAS

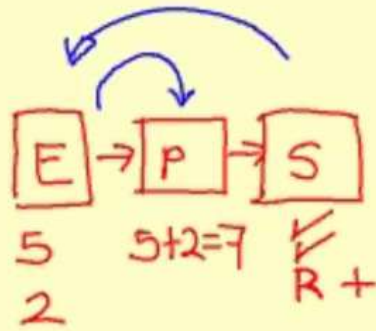
LOGICA DE PROGRAMACION
APUNTES

BRANDON VÁZQUEZ JUÁREZ

SECUENCIA 1NM13

PP21104180





Algoritmo

Secuencia de
pasos

Pseudocódigo

Inicio	→	main()
Var x, y, z	→	{ int x, y, z;
Imprime "Teclea x"	→	printf("Teclea x");
Lee x	→	scanf("%d", &x);
Imprime "Teclea y"	→	printf("Teclea y");
Lee y	→	scanf("%d", &y);
z = x + y	→	z = x + y;
Imprime z	→	printf("%d", z);
Fin	→	}

Programa

Algoritmo
expresado
en un L-Programación.

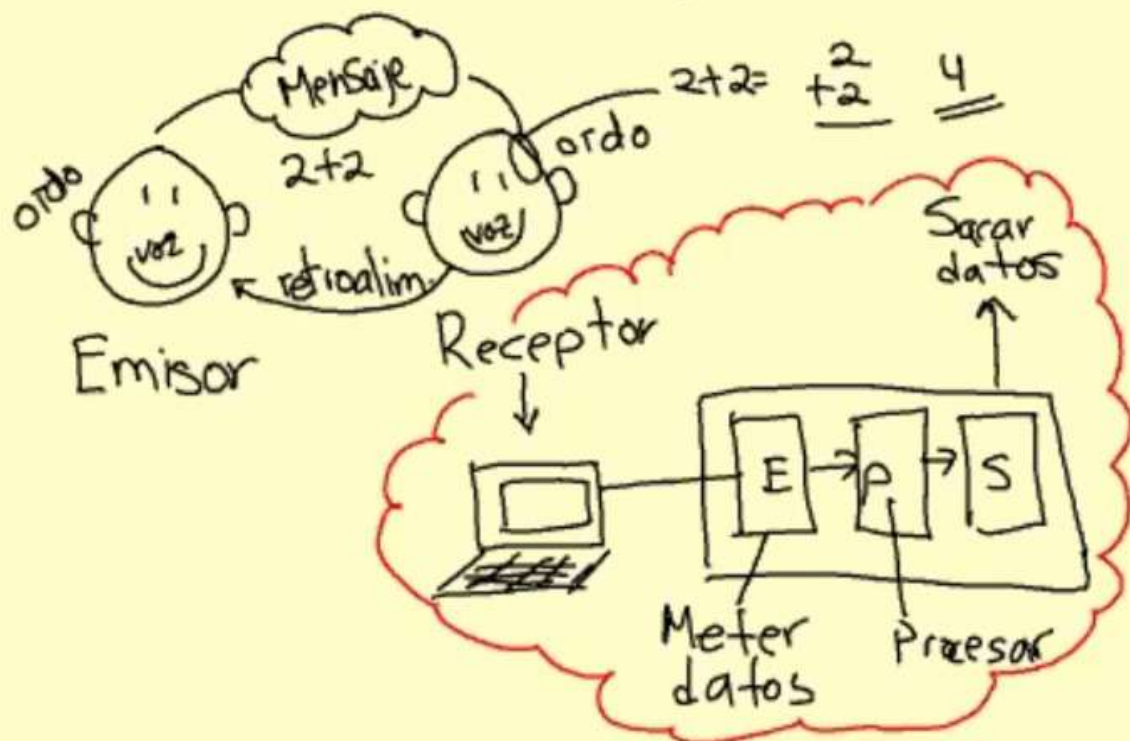
"Lógica de Programación."

Lógica → Razonamiento de solución de problemas.

Problema → Situación que impide alcanzar un objetivo.

Requerimientos → Necesidad ^{cubrir.} → objetivo

Programar = Comunicación Computadora



Estructuras de Control

⑤ if
Evaluación: Si (condición) entonces Acción
Switch()

Repetición
do... while
while
for



```
if (condición)
{
}

if (condición)
{
}
else
{
}
```

(2)

```
if (condición)
{
}
else
{
    if (condición)
    {
    }
}
```

(3)

S: completo

```
{ if (condición) entonces  
                          
                          
  { else  
                            
                            
  { end if
```

```
Inicio
  var edad
  Imprime (Ingresar Edad)
  Lee edad
  if (edad > 18)
    Imprime ("Eres ciudadano")
  end if
Fin
```

```
Inicio
  var num
  Imprime (Ingresar numero)
  Lee num
  if (num > 0) entonces
    Imprime (Es positivo)
  end if
Fin
```

-Inicio

Var num

Imprime ("Ingresa número")

Lee num

if (num > 0)

Imprime ("Es positivo")

else

if (num < 0)

Imprime ("Es negativo")

else

Imprime ("Es cero")

endif

end if

Fin

Estructuras de repetición.

for do...while while

for(inicial, final, incremento)
 documento

$i=0, i \leq 10$
for($i=1, i \leq 10, i++$)
 Imprime ("Hola")
en for i

do... while

do
 =====
 Instrucción
 Control Fin del ciclo
while (condición)
Primero ejecuta
y después pregunta
o evalúa

while

while (condición)
 =====
 Inst. control Fin del ciclo
end while
Primero evalúa y
después ejecuta



$x = 15$
 ~~$x = 1$~~
 $x = 1$
 Imprime "Hola"
 $x = x + 1$
 While ($x \leq 10$)

X	Imprime	$x = x + 1$	$x \leq 10$
1	Hola	$2 = 1 + 1$	$2 \leq 10$ ✓
	Hola	$3 = 2 + 1$	$3 \leq 10$ ✓
	Hola	$4 = 3 + 1$	$4 \leq 10$ ✓
	Hola	$5 = 4 + 1$	$5 \leq 10$ ✓
	Hola	$6 = 5 + 1$	$6 \leq 10$ ✓
	Hola	$7 = 6 + 1$	$7 \leq 10$ ✓
	Hola	$8 = 7 + 1$	$8 \leq 10$ ✓
	Hola	$9 = 8 + 1$	$9 \leq 10$ ✓
	Hola	$10 = 9 + 1$	$10 \leq 10$ ✓
	Hola	$11 = 10 + 1$	$11 \leq 10$ ✗

Presione ESC o haga doble clic para salir del modo de pantalla completa

X	Imprime	$x = x + 1$	$x \leq 10$
15	Hola	$16 = 15 + 1$	$16 \leq 10$ ✗



15

~~X=1~~

while (X <= 10)

Imprime "Hola"

X = X + 1

end while

X	Imprime	X = X + 1	X <= 10
1	Hola	2	1 <= 10 ✓
	Hola	3	2 <= 10 ✓
	Hola	4	3 <= 10 ✓
	:	:	:
	Hola	10	10 <= 10 ✓
X = 15		11	11 <= 10 ✗
			15 <= 10 ✗

do...while
1-n veces

while
0-n veces

Pseudoradigo / D. Detallado

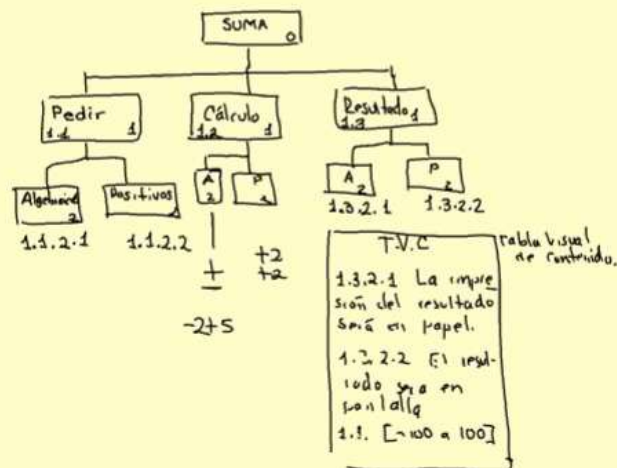
Top Down } D. General.
Wormnier }

Nashishederman D. Detallado

Auxiales:

- Segmentación
- Identación
- Modularidad

TOP DOWN → (Suma) ⇒ Diagrama jerárquico



Activar Windows
Ve a Configuración para activar Windows.

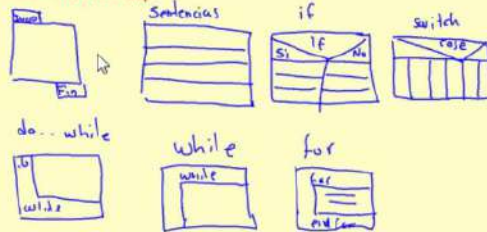
Warnier

$\left\{ \begin{array}{l} \text{Pedir} \left\{ \begin{array}{l} \text{Algebras } [-100 \text{ a } 100] \\ \text{Positivos } [0 \text{ a } 100] \end{array} \right. \\ \\ \text{SUMA} \left\{ \begin{array}{l} \text{Calcular} \left\{ \begin{array}{l} A \\ P \end{array} \right. \\ \\ \text{Resultado} \left\{ \begin{array}{l} A \left\{ \text{Papel} \\ P \left\{ \text{Pantalla.} \end{array} \right. \end{array} \right. \end{array} \right.$

Warnier

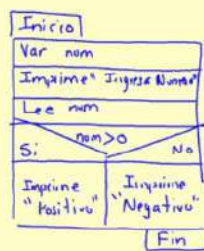
$\left\{ \begin{array}{l} \text{Pedir} \left\{ \begin{array}{l} \text{Algoritmos } [-100 \text{ a } 100] \\ \text{Positivos } [0 \text{ a } 100] \end{array} \right. \\ \\ \text{SUMA} \left\{ \begin{array}{l} \text{Calcular} \left\{ \begin{array}{l} A \\ P \end{array} \right. \\ \\ \text{Resultado} \left\{ \begin{array}{l} A \left\{ \text{Papel} \right. \\ P \left\{ \text{pantalla.} \right. \end{array} \right. \end{array} \right. \end{array} \right.$

Nashiederman

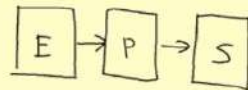


Elaborar un algoritmo que
reciba un numero e indique si es +o-.

Nashishederman



Lenguaje C.



Imprime — `printf("Texto");`
`printf("Tipocito", X);`

Lee — `scanf("%d", &X);`

Tipos de datos

`%d` → Entero

`%f` → flotante (decimal)

`%c` → caracter

`%s` → cadena (Muchos caracteres).

Proceso

Estructuras de Control.

```
for(valor i=0; valor final i<10; incremento i++)  
{  
    decremento
```

```
}
```

```
do  
{
```

```
while (condición)  
{
```

```
} while (condición);
```

```
}
```

Usted está viendo la pantalla de Alejandra Mendez

Ver Opciones

Activar micrófono con (Alt+A). O bien mantenga presionada la tecla ESPACIO para activar temporalmente.

```
switch (opcion)
{
    case 'a':
        break;

    case 'e':
        break;

    :
    case 'n':
        break;

    default:

```

Usted está viendo la pantalla de Alejandra Mendez

Ver Opciones

Vista



```
#include <stdio.h>
main()
{
    printf
    scanf
}
```

Activar mi audio: Alt+A. O bien mantenga presionada la tecla ESPACIO para activar temporalmente.

Cancelar silencio ahora Iniciar video

Participantes

Chat

Compartir pantalla

Grabar

Salir



INSTITUTO POLITECNICO NACIONAL
UNIDAD PROFECIONAL INTERDISCIPLINARIA DE INGENIERIA CIENCIAS
SOCIALES Y ADMINISTRATIVAS

LOGICA DE PROGRAMACION
PROGRAMAS

BRANDON VÁZQUEZ JUÁREZ

SECUENCIA 1NM13

PP21104180

