

# Desarrollo de Programas

Prof. Lisbeth C. Pérez Rivas  
lisbethpe@ula.ve

# Desarrollo de Programas

- **Ejercicio:**

Calcular el sueldo neto de un trabajador conociendo el número de horas trabajadas, la tarifa horaria y la tasa de impuestos.

# Sueldo Neto de un Trabajador

## Entrada

Información necesaria para la solución del problema.

¿ Qué datos son de entrada ?

Número de horas trabajadas(NH), tasa horaria(TH), tasa de impuestos (TI)

## Proceso:

Operaciones o cálculos necesarios para encontrar la solución del problema.

¿ Qué tipo de ecuaciones ?

$$SB = NH * TH$$

$$Imp = SB * TI$$

$$SN = SB - Imp$$

## Salida

Resultados finales de los cálculos.

- ¿ Cuáles son los datos de salida

**Sueldo Neto (SN)**

- ¿ Cuántos datos de salida se producirán

**Uno**

# Sueldo Neto de un trabajador

Descripción del argumento	Identificador	Tipo	Restricciones
Número de horas	NH	Número	Positiva
Tasa horaria	TH	Número	Positiva
Tasa de impuestos	TI	Número	Positiva
Sueldo Base	SB	Número	
Impuestos	Imp	Número	
Sueldo Neto	SN	Número	

# Sueldo Neto de un trabajador

## Algoritmo:

1. Inicio
2. Escribir("Inserte Numero de horas trabajadas")
3. Leer(NH)
4. Escribir("Inserte la tasa horaria")
5. Leer(TH)
6. Escribir("Inserte la tasa de impuestos")
7. Leer(TI)
8.  $SB = NH * TH$
9.  $Imp = SB * TI$
10.  $SN = SB - Imp$
11. Escribir("El Sueldo Neto es: ")
12. Escribir(SN)
13. Fin

# Diagramas de Flujo

Representación gráfica de un algoritmo

Símbolos:

Terminal



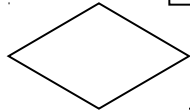
Lectura/Escritura de Datos



Proceso de Datos



Decisión



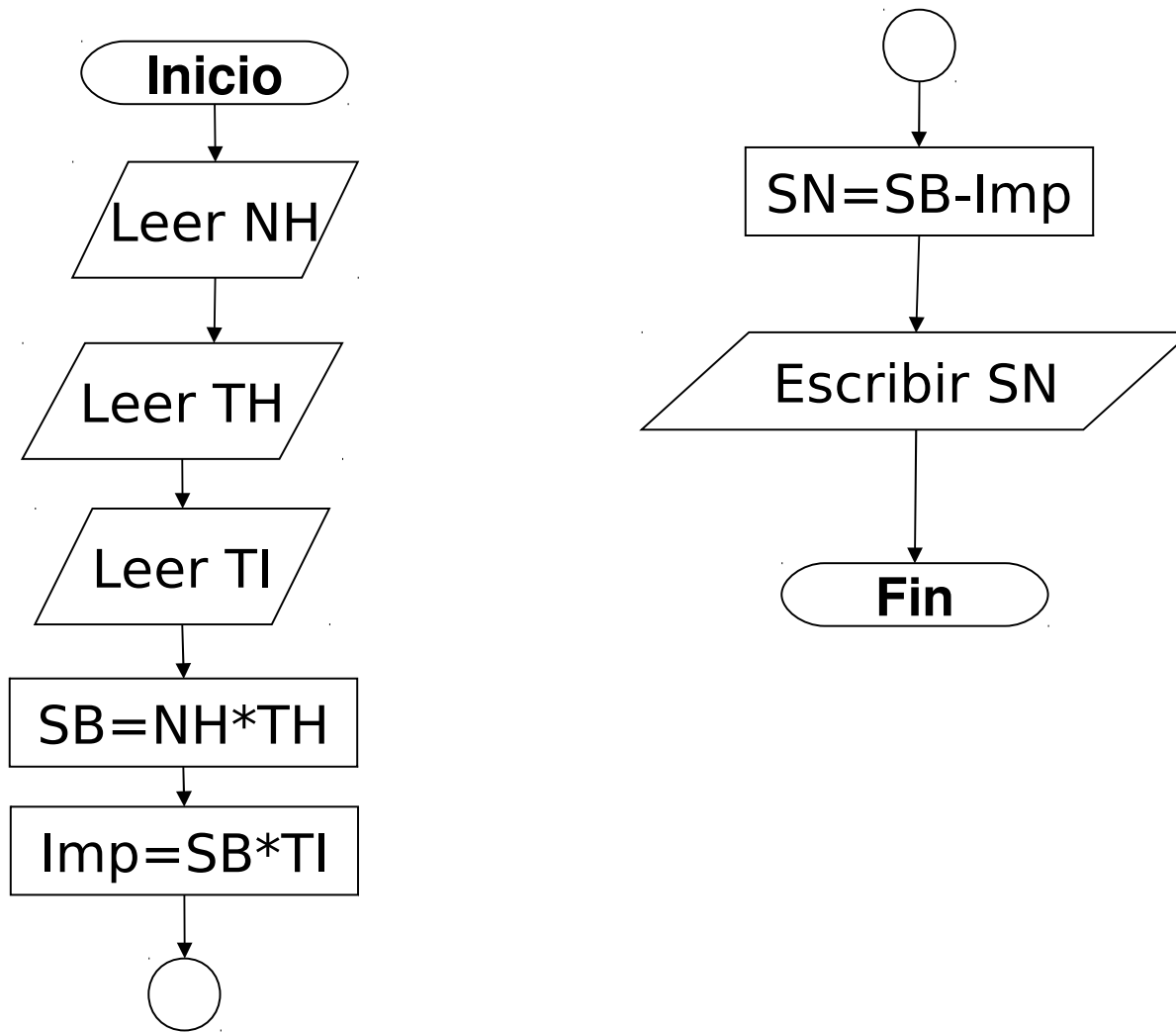
Conector



Impresión



# Sueldo Neto de un trabajador



# Ejercicio

- Hacer el diagrama de flujo del problema del cálculo del área del triángulo.

# Ejercicios

Escribir un algoritmo y diagrama de flujo para:

- a) Sumar dos números enteros.
- b) Restar dos números enteros.
- c) Multiplicar dos números enteros.
- d) Dividir un número entero por otro
- e) Calcular el perímetro ( $2 * \pi * R$ ) y área de una circunferencia ( $\pi * R^2$ ).

# Estructura básica de un programa en C++

```
#include<iostream>
// librerias           →      Esto es un comentario
using namespace std;  de una línea
//variables
//funciones
int main(){
    //variables
    ...
    ...
    ...
    return 0;
}
```

# Hacer un programa que escriba por pantalla HOLA MUNDO

**Entrada**

**Proceso:**

**Salida**

No existe

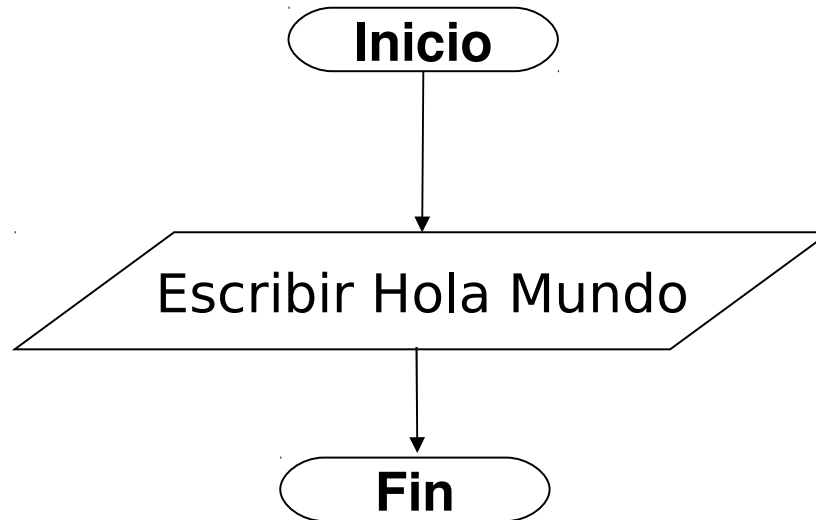
Escritura por  
pantalla

Mensaje  
"HOLA MUNDO"

## Algoritmo:

1. Inicio
2. Escribir("HOLA MUNDO")
3. Fin

# Hola Mundo



# Codificación Hola Mundo

Escribir	→	cout
Leer	→	cin

```
#include<iostream>  
using namespace std;  
int main(){  
    cout<<"HOLA MUNDO";  
    return 0;  
}
```

# ¿Qué sucede si queremos codificar el programa del perímetro del cuadrado?

¿Qué son L y P?

1. Inicio
2. Escribir('Introduzca la longitud del lado del cuadrado')
3. Leer (L)
4.  $P=4*L$
5. Escribir('El perímetro es: ',P)
6. Fin

# Conceptos Básicos

**Dato:** Representación simbólica (numérica, alfabética, etc.), atributo o característica de una Entidad.

**Ejemplo:**

Entidad: Cuadrado

Dato: lado del cuadrado, perímetro del cuadrado.

# Conceptos Básicos

## **Tipos de Datos:**

- **Números:** Entero, punto flotante de precisión variable, dirección de memoria.
- **Carácter:** Básicamente se utilizan para formar palabras.

# Conceptos Básicos

## Valor:

La definición más intuitiva de **valor** es similar a la de elemento perteneciente a un conjunto, el cual representa un **tipo de dato**. Todos los valores del conjunto deben cumplir la misma propiedad, y el conjunto define el tipo de operaciones que se pueden aplicar sobre sus valores.

**Ejemplo:** Sea el conjunto de los enteros positivos  $A = \{0, 1, 2, 3, 4, \dots\}$

el valor 4 pertenece al conjunto A.

Tipo de dato	es equivalente a conjunto
valor	es equivalente a elemento.

# Conceptos Básicos

- **Constante:** Es un dato cuyo valor no puede cambiar durante la ejecución del programa.
- **Variable:** Es un elemento asociado a un campo de memoria al que se le puede cambiar su contenido cuantas veces sea necesario. Se encuentra referenciada por un nombre y pertenece a un tipo de dato.

# Conceptos Básicos

Las variables,

1. Se definen (Declaración)
2. Se crean
3. Se actualiza con un valor inicial (Datos de Entrada, Expresiones)
4. Su valor se modifica (Instrucciones del programa)
5. Llegan a su valor final (Resultado de Salida)
6. Se destruyen (Fin del programa)

# Conceptos Básicos

Criterios a seguir con las variables:

- Utilice solo las necesarias y de los tipos adecuados.
- Cada una con un significado muy claro e inmutable.
- Inicializarlas.
- Controlar que toman valores sensatos.

# Conceptos Básicos

¿Qué son L y P?

El contenido de L y P son valores.

L y P son variables.

# Como se cargan las variables?

- Operación de Asignación:

El signo igual (=) significa que el computador va a realizar lo que está a la derecha del igual y lo va a almacenar en la variable que se encuentre a la izquierda del igual.

Ejm:     a = 8  
          b = a + 10  
          a = 10

# Como se cargan las variables?

A considerar,

- a. Al lado izquierdo del igual solo puede haber una variable.
- b. Al lado derecho del igual puede haber una constante, una variable o una expresión.
- c. El computador siempre resuelve lo de la derecha del igual y su resultado lo almacena en la variable que esté a la izquierda del igual.
- d. Cada vez que se le entra un nuevo valor a una variable, el valor anterior se pierde

# Ejercicio

$$a = 10$$

$$b = 20$$

$$c = 5$$

$$a = a + 3$$

$$b = b + 4 - a$$

$$c = a + b + c$$

$$a = a + c$$

$$b = 4$$

$$c = c + 3 - b + 2$$

Qué valores quedan almacenados en las variables a, b y c ?

# Nombres

Identificador que consta de varios caracteres alfanuméricos, de los cuales el primero normalmente es una letra.

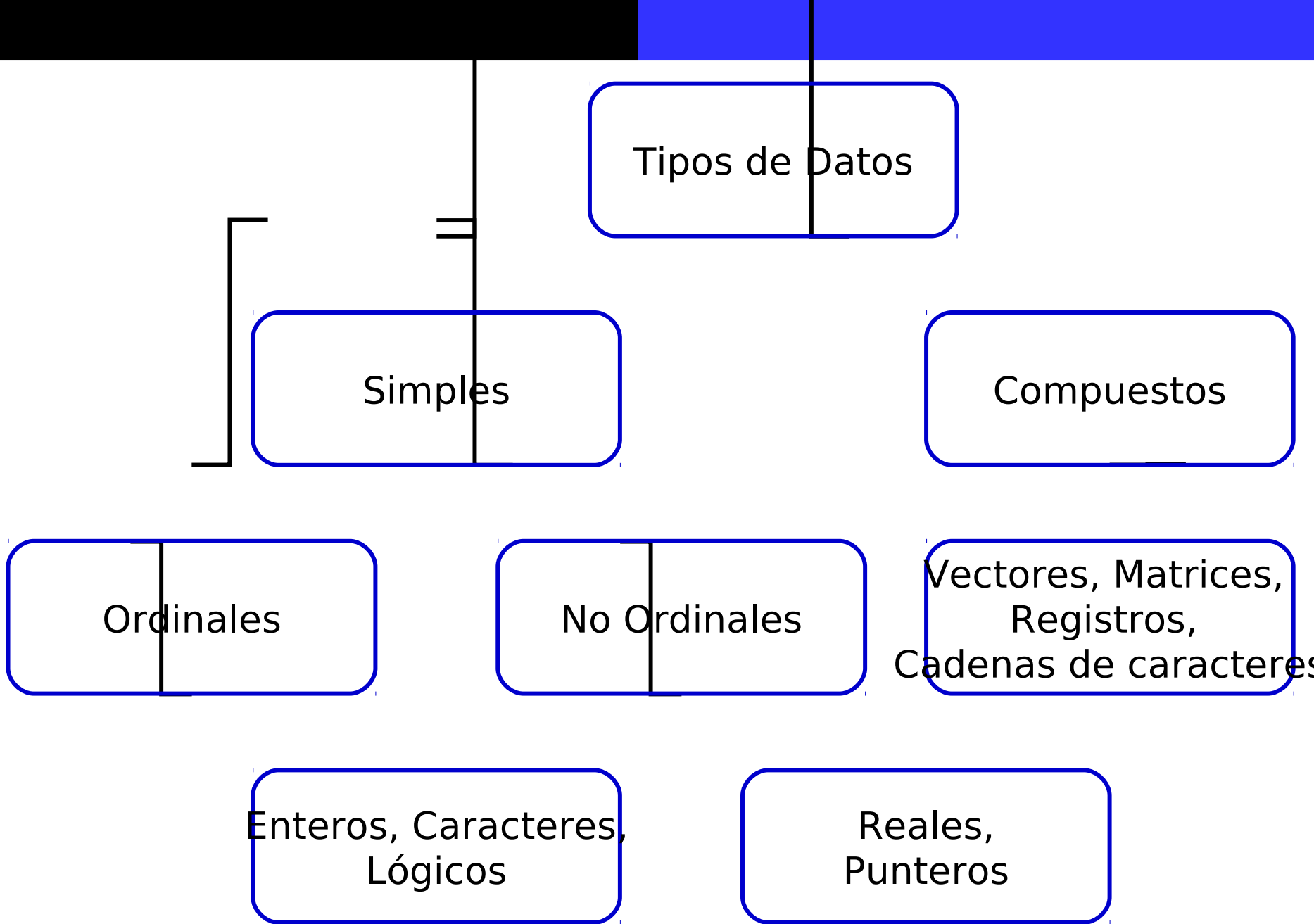
Ejemplo: a,b,c,casa,c1,Fecha\_inscripcion,lado1.

Erróneo: 1\_a

Correcto: a\_1

# Tipos de Datos

Toda variable debe estar asociada a un *tipo de dato* con el que se puede determinar unívocamente el dominio de valores.



# Tipo Entero (int)

Un dato de tipo entero es un número que no tiene punto decimal, por lo tanto en sus operaciones jamás va a generar decimales.

Ejemplo: 5 6 -15 4 0 -1340

# Tipo Real (float)

Un dato de tipo real es un número que tiene punto decimal, por lo tanto en sus operaciones puede generar decimales.

Se conocen como números de punto flotante cuya representación consta de una mantisa (parte fraccional), de una base y de un exponente (potencia a la cual se eleva la base).

# Tipo Carácter (char)

Conjunto finito y ordenado de los caracteres que la computadora reconoce (letra, dígito, signo de puntuación, etc.)

Un carácter es almacenado en un byte usando el código de 8 bits ASCII (American Standard Code for Information Interchange), lo que da la posibilidad de representar  $2^8 = 256$  caracteres diferentes.

Ejm: 'a', 'b', ',', '#'

# Tipo lógico o booleano

Pueden tomar solo dos valores verdadero o falso

- True → Verdadero
- False → Falso

Ejemplo:

```
bandera= true;  
logica=false;
```

# ¿Qué sucede si queremos codificar el programa del perímetro del cuadrado?

1. Inicio
2. Leer (L)
3.  $P=4*L$
4. Escribir(P)
5. Fin

¿Qué son L y P?

Variables de tipo ...

¿Entero?

¿Real?

¿Carácter?

# ¿Qué sucede si queremos codificar el programa del perímetro del cuadrado?

```
#include<iostream>
using namespace std;
int main(){
    //definición de
    variables
    ...
    return 0;
}
```

Lista de Variables = L y P

# Declarar Variables

Los objetivos de la declaración de variables son:

- Asociar un tipo de dato y un identificador único a la variable. Esto también permite que el compilador pueda verificar la correctitud de las operaciones en donde interviene la variable.
- Permitir que el compilador sepa cuánto espacio de memoria se necesita para almacenar el valor de la variable, y asignar la dirección de memoria donde este valor se va a almacenar.

# Declarar Variables

```
tipo_de_dato Nombre_variable;
```

Ejemplo:

```
float var1;
```

```
int numero1,numero2;
```

```
char letra;
```

¿Qué sucede si queremos codificar el programa del perímetro del cuadrado?

```
#include<iostream>
using namespace std;
int main(){
    float L,P;
    ...
    return 0;
}
```

Lista de Variables = L y P son reales

```
float L,P;
```

# ¿Qué sucede si queremos codificar el programa del perímetro del cuadrado?

1. Inicio
2. Escribir('Introduzca la longitud del lado del cuadrado')
3. Leer (L)
4.  $P=4*L$
5. Escribir('El perímetro es: ',P)
6. Fin



```
#include<iostream>
using namespace std;
int main(){
    float L,P;
    cout<<"Introduzca la longitud del
        lado del cuadrado";
    cin>>L;
    P=4*L;
    cout<<"El perímetro es: "<<P;
    return 0;
}
```

¿Qué sucede si queremos codificar  
el programa del perímetro del  
circulo?

Aparece la constante  $\pi$ .

¿Como hacemos?

Declarar una constante

# Declaración de Constantes

`#define NOMBRE valor`

Ejemplos:

```
#define PI 3.141592      Constante real
#define CUENTA 632      Constante entera
#define SUMA (2.5+40)/(3.5-4)  Constante real
#define COLOR "AZUL"    Constante cadena      (string)
#define SEC 'X';        Constante caracter
```

# Declaración de Constantes

```
#include<iostream>
// definición de constantes
#define PI (2.5+40)/(3.5-4)
#define ANCHO "ALTO"
#define letra 'A'

using namespace std;
int main(){
    //definición de variables
    ....

    return 0;
}
```

# Ejercicios

- Diga que tipo de dato son:

10

20.5

20.50

$21 \times 10^{-10}$

Casa

A

B

True

-50000