

# Electrónica Digital II

Santiago Rúa Pérez, PhD.

2 de agosto de 2022

# CONCEPTOS BÁSICOS DE PROGRAMACIÓN EN C

# Introducción a programación en C

## Objetivos

- Entender los objetivos Básicos de los computadores.
- Entender los diferentes tipos de lenguaje de programación.
- Ambiente típico de c.
- Estructuras condicionadas.

# Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Our World  
in Data

## Transistor count

50,000,000,000

10,000,000,000

5,000,000,000

1,000,000,000

500,000,000

100,000,000

50,000,000

10,000,000

5,000,000

1,000,000

500,000

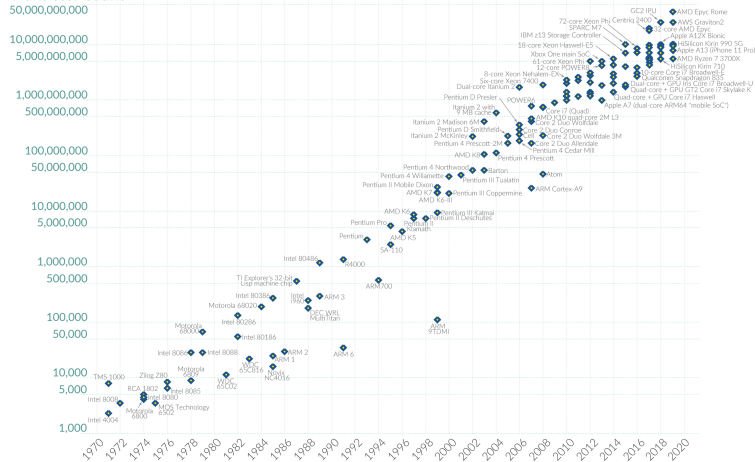
100,000

50,000

10,000

5,000

1,000

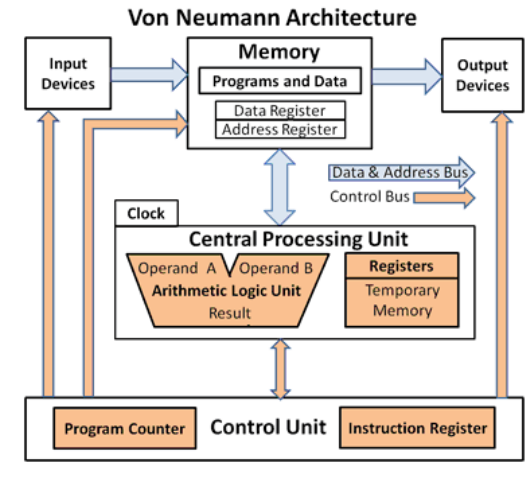


Data source: Wikipedia ([wikipedia.org/wiki/Transistor\\_count](https://en.wikipedia.org/wiki/Transistor_count))

OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

# Arquitectura de computadores



# Lenguajes de programación

## Lenguaje de Máquina

- Depende de la máquina.
- Cadena de números.
- 010011110

## Ensamblador

- Abreviaciones en inglés.
- Base de lenguaje.
- Mas facil de entender por humanos.
- load, add, store

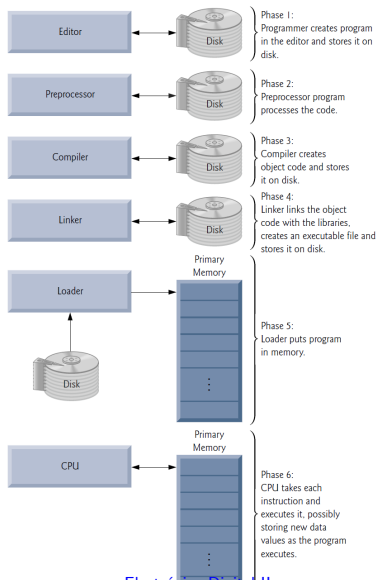
## Alto nivel

- Mas facil de programar.
- Exportabilidad.
- `grossPay = basePay + overTimePay`

# Pseudocódigo

Como seria el pseudocodigo para calcular el salario a pagar a un empleador?

# Lenguaje de Programación C





# Primer Programa en C

Imprimir en consola la frase: 'Hola Mundo'

```
1 // Primer programa en C
2
3 // Incluir libreria de salidas y entradas estandar
4 #include <stdio.h>
5
6 // Funcion donde comienza la ejecucion del programa
7 int main(void)
8 {
9     printf("Hello World!");
10 } // Fin de la funcion
11
```

# Segundo Programa en C

Solicitar dos números enteros y devolver la suma de los mismos.

```
1 // Segundo programa en C
2
3 // Incluir libreria de salidas y entradas estandar
4 #include <stdio.h>
5
6 // Funcion donde comienza la ejecucion del programa
7 int main(void)
8 {
9     // Declaracion de variables
10    int integer1;
11    int integer2;
12
13    // Almacenamiento de los numero dados por el usuario
14    printf("Ingrese el primer entero\n");
15    scanf("%d",&integer1);
16
17    printf("Ingrese el segundo entero\n");
18    scanf("%d",&integer2);
19
20    int sum;
21    sum = integer1 + integer2;
22
23    printf("La suma es %d\n",sum);
24 } // Fin de la funcion
25
```

# Aritmética en C

C operation	Arithmetic operator	Algebraic expression	C expression
Addition	+	$f + 7$	<code>f + 7</code>
Subtraction	-	$p - c$	<code>p - c</code>
Multiplication	*	$bm$	<code>b * m</code>
Division	/	$x / y$ or $\frac{x}{y}$ or $x \div y$	<code>x / y</code>
Remainder	%	$r \bmod s$	<code>r % s</code>

Operator(s)	Operation(s)	Order of evaluation (precedence)
( )	Parentheses	Evaluated first. If the parentheses are nested, the expression in the <i>innermost</i> pair is evaluated first. If there are several pairs of parentheses “on the same level” (i.e., not nested), they’re evaluated left to right.
* / %	Multiplication Division Remainder	Evaluated second. If there are several, they’re evaluated left to right.
+ -	Addition Subtraction	Evaluated third. If there are several, they’re evaluated left to right.
=	Assignment	Evaluated last.

# Operadores de relación en C

Algebraic equality or relational operator	C equality or relational operator	Example of C condition	Meaning of C condition
<i>Relational operators</i>			
>	>	x > y	x is greater than y
<	<	x < y	x is less than y
≥	>=	x >= y	x is greater than or equal to y
≤	<=	x <= y	x is less than or equal to y
<i>Equality operators</i>			
=	==	x == y	x is equal to y
≠	!=	x != y	x is not equal to y

# Palabras reservadas en C

## Keywords

auto	do	goto	signed	unsigned
break	double	if	sizeof	void
case	else	int	static	volatile
char	enum	long	struct	while
const	extern	register	switch	
continue	float	return	typedef	
default	for	short	union	

### *Keywords added in C99 standard*

\_Bool \_Complex \_Imaginary inline restrict

### *Keywords added in C11 standard*

\_Alignas \_Alignof \_Atomic \_Generic \_Noreturn \_Static\_assert \_Thread\_local

## Ejemplos en C

- Hacer un programa que le solicite al usuario tres enteros y obtenga la multiplicación.
- Valide si dos números  $a$  y  $b$  son divisibles entre ellos.
- Encuentre los errores en la imagen.

```
a) printf( "The value is %d\n", &number );
b) scanf( "%d%d", &number1, number2 );
c) if ( c < 7 );{
    printf( "C is less than 7\n" );
}
d) if ( c == 7 ) {
    printf( "C is greater than or equal to 7\n" );
}
```

# Operadores de asignación

El operador de asignación de valor es `=`. Las operaciones típicas son:

*variable = variable operador expresion*

Assignment operator	Sample expression	Explanation	Assigns
<i>Assume: <code>int c = 3, d = 5, e = 4, f = 6, g = 12;</code></i>			
<code>+=</code>	<code>c += 7</code>	<code>c = c + 7</code>	10 to c
<code>-=</code>	<code>d -= 4</code>	<code>d = d - 4</code>	1 to d
<code>*=</code>	<code>e *= 5</code>	<code>e = e * 5</code>	20 to e
<code>/=</code>	<code>f /= 3</code>	<code>f = f / 3</code>	2 to f
<code>%=</code>	<code>g %= 9</code>	<code>g = g % 9</code>	3 to g

Operator	Sample expression	Explanation
<code>++</code>	<code>++a</code>	Increment a by 1, then use the new value of a in the expression in which a resides.
<code>++</code>	<code>a++</code>	Use the current value of a in the expression in which a resides, then increment a by 1.
<code>--</code>	<code>--b</code>	Decrement b by 1, then use the new value of b in the expression in which b resides.
<code>--</code>	<code>b--</code>	Use the current value of b in the expression in which b resides, then decrement b by 1.

# Estructuras de control

## Estructura If

Este tipo de instrucción es utilizado para seleccionar entre dos opciones

### If ...

Si la nota de un estudiante es mayor a 3, entonces ganó el curso.

### If ... else

Si la nota de un estudiante es mayor a 3, entonces ganó el curso, sino perdió.

### ?:

Operador condicional es ternario. `puts( grade >= 60 ? "Passed" : "Failed" )`



# Ejemplo operadores

Encuentre la relación entre dos números enteros.

```
1 // Using if statements, relational
2 // operators, and equality operators.
3 #include <stdio.h>
4
5 // function main begins program execution
6 int main( void )
7 {
8     printf( "Enter two integers, and I will tell you\n" );
9     printf( "the relationships they satisfy: " );
10
11     int num1; // first number to be read from user
12     int num2; // second number to be read from user
13
14     scanf( "%d %d", &num1, &num2 ); // read two integers
15     if ( num1 == num2 ) {
16         printf( "%d is equal to %d\n", num1, num2 );
17     }
18     if ( num1 != num2 ) {
19         printf( "%d is not equal to %d\n", num1, num2 );
20     } // end if
21     if ( num1 < num2 ) {
22         printf( "%d is less than %d\n", num1, num2 );
23     } //
24     if ( num1 > num2 ) {
25         printf( "%d is greater than %d\n", num1, num2 );
26     } // end if
27 }
28
```

# Estructuras condicionales

## If anidados

```
If student's grade is greater than or equal to 90
    Print "A"
else
    If student's grade is greater than or equal to 80
        Print "B"
    else
        If student's grade is greater than or equal to 70
            Print "C"
        else
            If student's grade is greater than or equal to 60
                Print "D"
            else
                Print "F"
```

# If anidados en C - ejemplo

## Código en C.

```
1  if ( grade >= 90 ) {  
2      puts( "A" );  
3  } // end if  
4  else if ( grade >= 80 ) {  
5      puts( "B" );  
6  } // end else if  
7  else if ( grade >= 70 ) {  
8      puts( "C" );  
9  } // end else if  
10 else if ( grade >= 60 ) {  
11     puts( "D" );  
12 } // end else if  
13 else {  
14     puts( "F" );  
15 } // end else  
16
```

# Estructuras condicionales

## Switch... case..

Es una estructura utilizada para la selección multiple. Su estructura es

```
switch( expression )
{
    case value-1:
        Block-1;
        Break;
    case value-2:
        Block-2;
        Break;
    case value-n:
        Block-n;
        Break;
    default:
        Block-1;
        Break;
}
```

# Estructuras condicionales

## Ejemplos

- Hacer un programa en C que solicite dos numeros flotantes. Una vez ingresado los numeros flotantes, imprimir al usuario cuatro opciones de operaciones dadas asi: A. Suma, B. resta, C. Multiplicacion, D. Division. Una vez seleccionada la operación el programa retorna el valor de dicha operación.
- Una empresa tiene que llenar un formato de los impuestos a recolectar de acuerdo al mes de ventas. Hacer un programa que solicite el mes que va a generar los impuestos, las ventas de dicho mes, y calcule los impuestos del municipio (4 %) y los de la nación (5 %).
- Haga un programa en C que calcule cuanto se le consigna a una persona dado el salario acordado. Recuerde que a las persona se le debe deducir salud y pension, 4 % cada uno.

# CONCEPTOS BÁSICOS DE PROGRAMACIÓN EN C

## GRACIAS