

Introducción a la Programación y Computación 1 Sección E

Ing. MSc. Neftalí Calderón

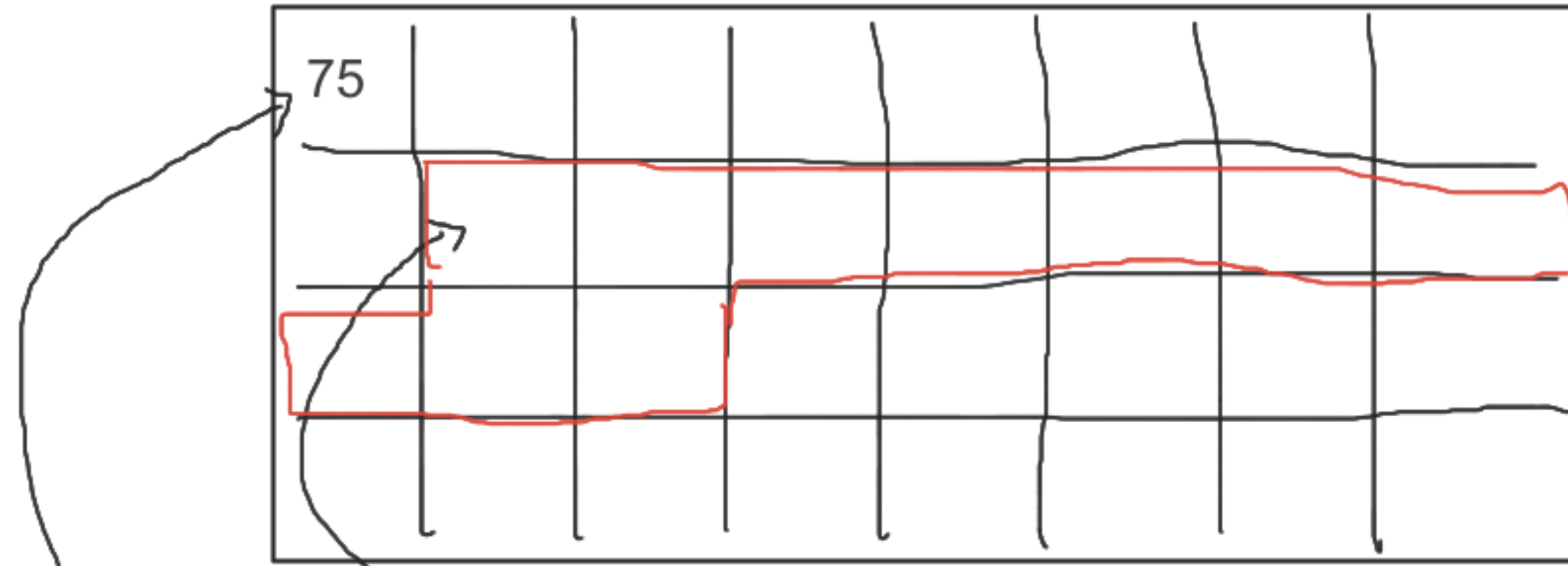
Arreglos

Un arreglo puede definirse como un grupo o una colección finita, homogénea y ordenada de elementos. Los arreglos se pueden definir de la siguiente forma:

- De una dimensión.
- De dos dimensiones.
- De tres o más dimensiones

Tipo de dato, nombre de la variable, tamaño

```
int notaDeExamen[50]
```



int notaDeExamen

int notaDeExamen[10]

char letra = 'm';
string nombre = 'Juan Pérez'

Arreglos unidimensionales

Es un tipo de datos estructurado que está formado de una colección finita y ordenada de datos del mismo tipo.

Están formados por un conjunto de elementos de un mismo tipo de datos que se almacenan bajo un mismo nombre, y se diferencian por la posición que tiene cada elemento dentro del arreglo de datos.

```
int notaDeExamen[50];
```

```
escribir(notaDeExamen[10]);
```

```
notaDeExamen[0] = 60;  
notaDeExamen[1] = 75;  
notaDeExamen[25] = 80;
```

Vectores

```
int notaDeExamen[10];
```

```
notaDeExamen[0] = 50;  
notaDeExamen[1];  
notaDeExamen[2];  
notaDeExamen[3];  
notaDeExamen[4];  
notaDeExamen[5];  
notaDeExamen[6];  
notaDeExamen[7] = 65;  
notaDeExamen[8];  
notaDeExamen[9] = 70;
```

```
notaDeExamen[1];  
notaDeExamen[2];  
notaDeExamen[3];  
notaDeExamen[4];  
notaDeExamen[5];
```

notaDeExamen =



```
string mes[13];
```

```
mes[0] = "";
```

```
mes[1] = 'Enero';
```

```
mes[2] = 'Febrero';
```

```
mes[3] = 'Marzo';
```

```
mes[4] = 'Abril';
```

```
.
```

```
.
```

```
.
```

```
.
```

```
.
```

```
mes[12] = 'Diciembre';
```

```
int numeroMes;
```

```
leer(numeroMes);
```

```
escribir(mes[numeroMes]);
```

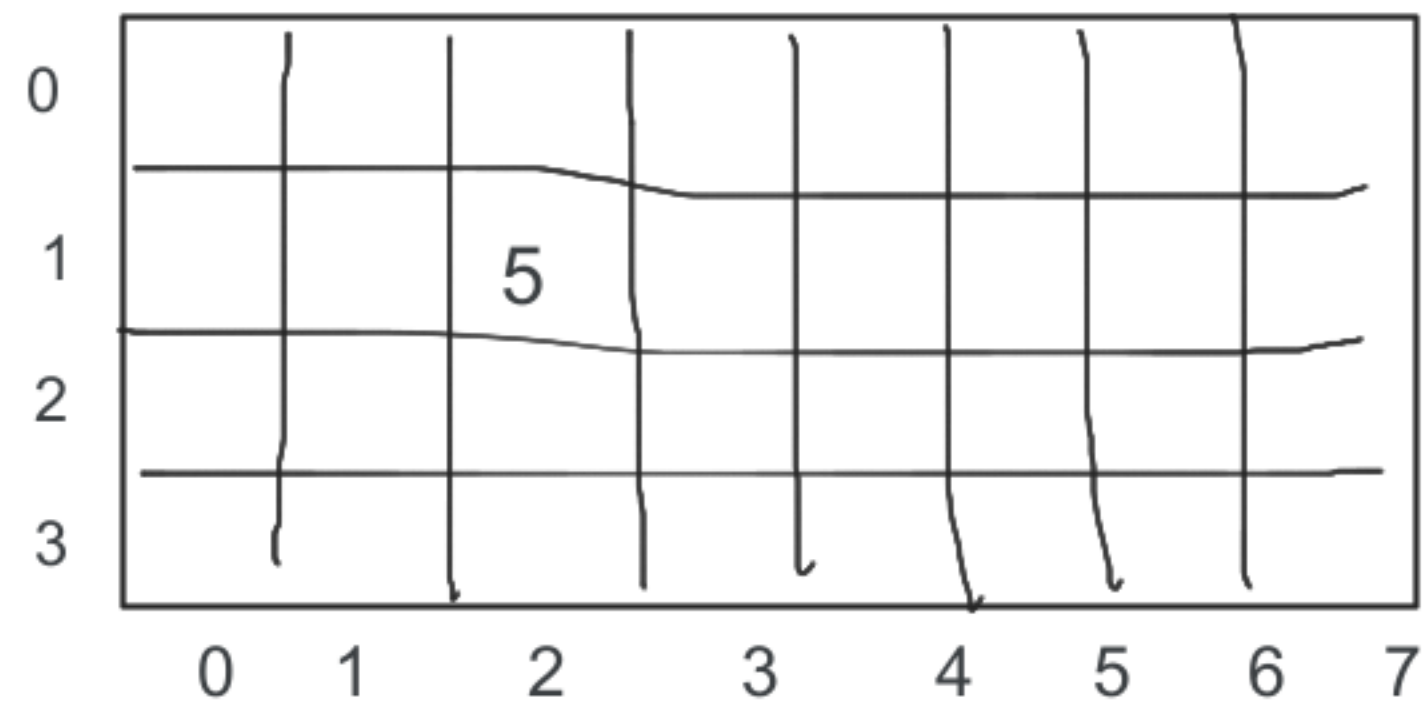
Arreglos multidimensionales

Es un tipo de dato estructurado, que está compuesto por dimensiones.

Para hacer referencia a cada componente del arreglo es necesario utilizar n índices, uno para cada dimensión.

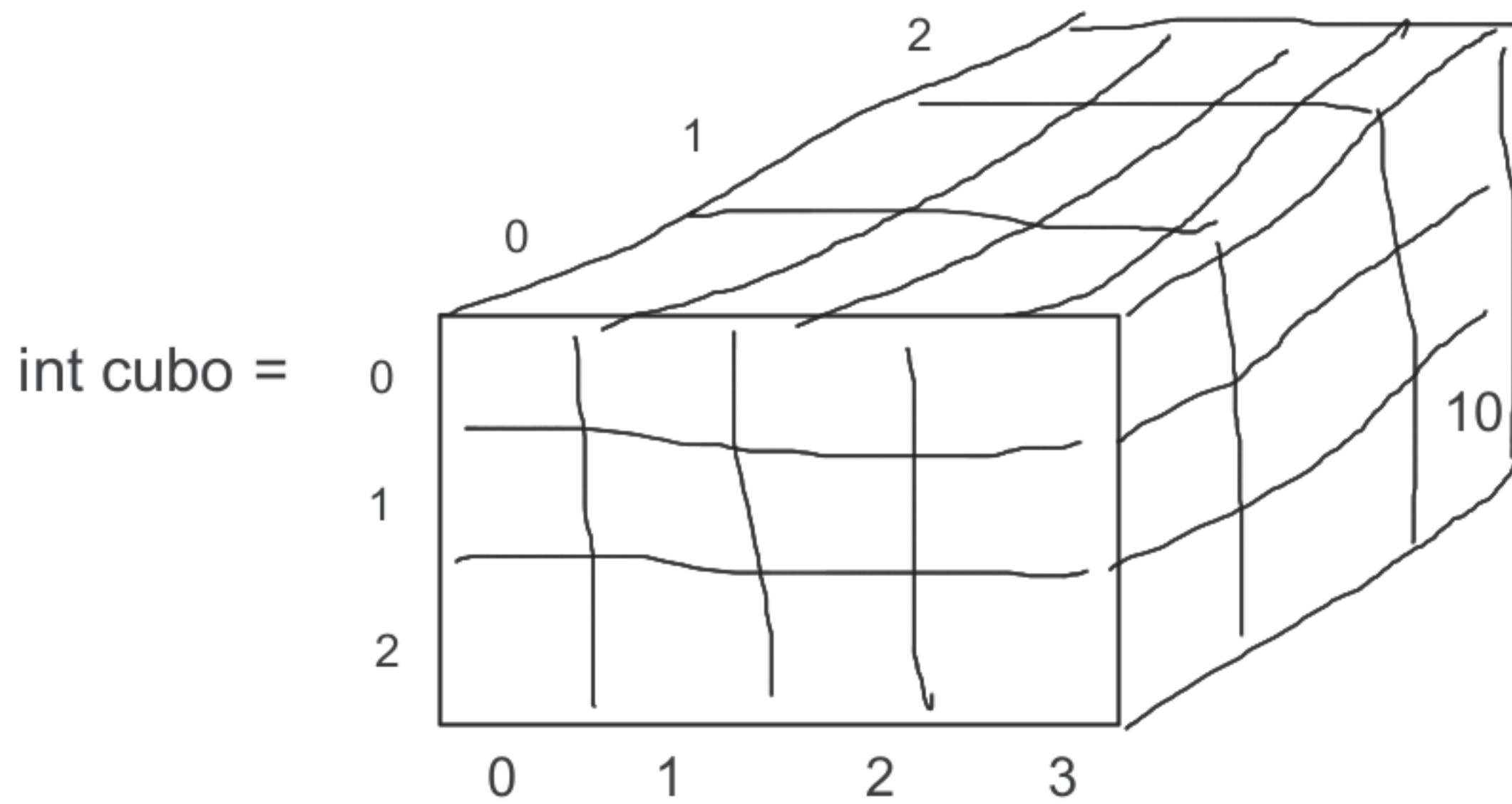
El término dimensión representa el número de índices utilizados para referirse a un elemento particular en el arreglo.

int tablero =



```
int tablero[4][8];
```

```
tablero[1][2] = 5;
```

```
int cubo[3][4][3];
```

```
cubo[2][3][2] = 10;
```

Subproblemas

Uno de los métodos fundamentales para resolver un problema es dividirlo en problemas más pequeños, llamados subproblemas.

Estos problemas pueden a su vez dividirse repetidamente en problemas más pequeños hasta que los problemas sean de fácil solución.

Divide y vencerás .. Cada subproblema es deseable que sea independiente de los demás y se denomina módulo.

Funciones

Tienen su origen ligado al concepto matemático de función de una o más variables. Ejemplo de este tipo de subalgoritmos son las llamadas funciones internas (sin, cos, abs). Las cuales pueden usarse en expresiones algorítmicas como si se tratara de variables.

Otros ejemplos de funciones matemáticas:

$$f(x) = x^2 + 2x - 1$$

$$g(x, y) = x^3 + 5y$$

$$h(x, y, z) = 3x + 2y - z$$

Una función es un objeto del ambiente, con nombre, tipo y valor único. El tipo se asocia al valor que retorna la función cuando es evaluada para un conjunto dado de valores de sus argumentos.

```
int areaTriangulo(int base, altura) {  
    int resultado = 0;  
  
    resultado = (base * altura)/2;  
    return(resultado);  
}
```

```
main() {  
    int dato1;  
    int dato2;  
  
    escribir("ingrese la base del triangulo");  
    leer(dato1);  
    escribir("ingrese la altura del triangulo");  
    leer(dato2);  
  
    escribir("el área del triangulo es: "+areaTriangulo(dato1,dato2));  
}
```

Procedimientos

Un procedimiento o subrutina es un subalgoritmo que recibiendo o no datos no devuelve ningún valor.

Un procedimiento está compuesto por un grupo de sentencias a las que asigna un nombre (identificador o simplemente nombre del procedimiento) y constituye una unidad de programa.

La tarea asignada al procedimiento se ejecutará siempre que se encuentre el identificador (nombre del procedimiento) en el conjunto de sentencias que definen el programa.

```
areaTriangulo(int base, altura) {  
    int resultado = 0;  
  
    resultado = (base * altura)/2;  
    escribir("El area del triangulo es : "+resultado);  
}
```

```
main() {  
    int dato1;  
    int dato2;  
  
    escribir("ingrese la base del triangulo");  
    leer(dato1);  
    escribir("ingrese la altura del triangulo");  
    leer(dato2);  
  
    areaTriangulo(dato1,dato2);  
}
```

```

void menu() {
    int boolean flag = true;
    int opcion;
    int dato1, dato2;

    escribir("ingrese una opción");
    escribir("1:sumas  2:multiplicación  3:salir");

    while(flag) {
        leer(opcion);

        if (opcion ==1) {
            leer(dato1);
            leer(dato2);
            escribir(suma(dato1, dato2));
        }
        elseif (opcion == 2) {
            leer(dato1);
            leer(dato2);
            escribir(multiplicacion(dato1, dato2));
        }
        else {
            flag = false;
        }
    }
}

```

```

int suma(int dato1, dato2) {
    int resultado;

    resultado = dato1 + dato2;

    return(resultado);
}

int multiplicacion(int dato1, dato2) {
    int resultado;

    resultado = dato1 * dato2;

    return(resultado);
}

main() {
    menu();
}

```


Hardware

Toda computadora realiza 4 funciones básicas: recibir, procesar, almacenar y entregar datos. Esto es posible gracias a los dispositivos de entrada, salida, almacenamiento y procesamiento conocidos como hardware.

La palabra Hardware se refiere a las partes físicas y tangibles de un sistema, sus componentes eléctricos, electrónicos, mecánicos y electromecánicos o cualquier otro elemento físico involucrado.

El Hardware se puede clasificar de la siguiente manera:

- De procesamiento
- De almacenamiento
- De comunicación
- De entrada y salida

Unidad de Control: La función principal de la unidad de control es dirigir la secuencia de pasos de modo que la computadora lleve a cabo un ciclo completo de ejecución de una instrucción, y hace esto con todas las instrucciones del programa en ejecución.

Los pasos que sigue la unidad de control son los siguientes:

- Ir a la memoria y extraer el código de la instrucción (ciclo fetch)
- Decodificar la instrucción
- Ejecuta la instrucción
- Se prepara para leer la siguiente instrucción

Unidad aritmética lógica (ALU): es un circuito digital que permite realizar operaciones aritméticas y lógicas entre dos números. La ALU se encuentra en todos los dispositivos electrónicos.

Arquitectura de una computadora

Diagrama de bloques

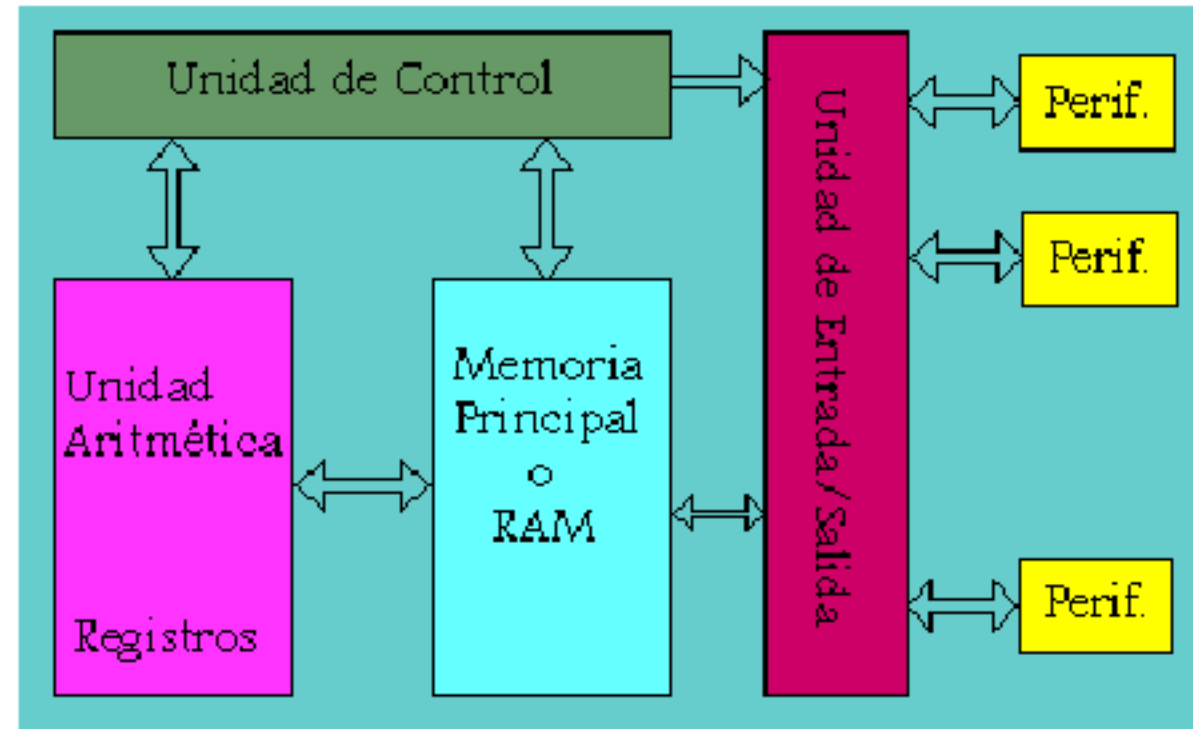


Diagrama de capas

