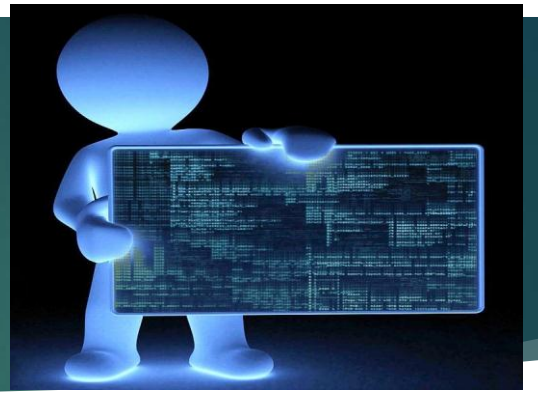
FUNDAMENTOS DE PROGRAMACIÓN

Vargas Suasnavas Jonathan
Club de Robótica EPN

1. PROGRAMACIÓN



- Es un recurso que permite crear secuencias de pasos lógicos con el objetivo de resolver un problema.
- La programación es un arte que requiere desarrollar una habilidad lógica en el programador
- Es el proceso de diseñar, escribir, probar, depurar y mantener el código fuente de programas computacionales
- El objetivo de la programación es crear programas que logren un comportamiento deseado.

Vargas Suasnavas Jonathan
Club de Robótica EPN

1.2. Máquina

- Sistema de código directamente interpretable por un sistema microcontrolado.

- Sola mente tienen dos niveles de voltaje simbolizados por 1 y 0.

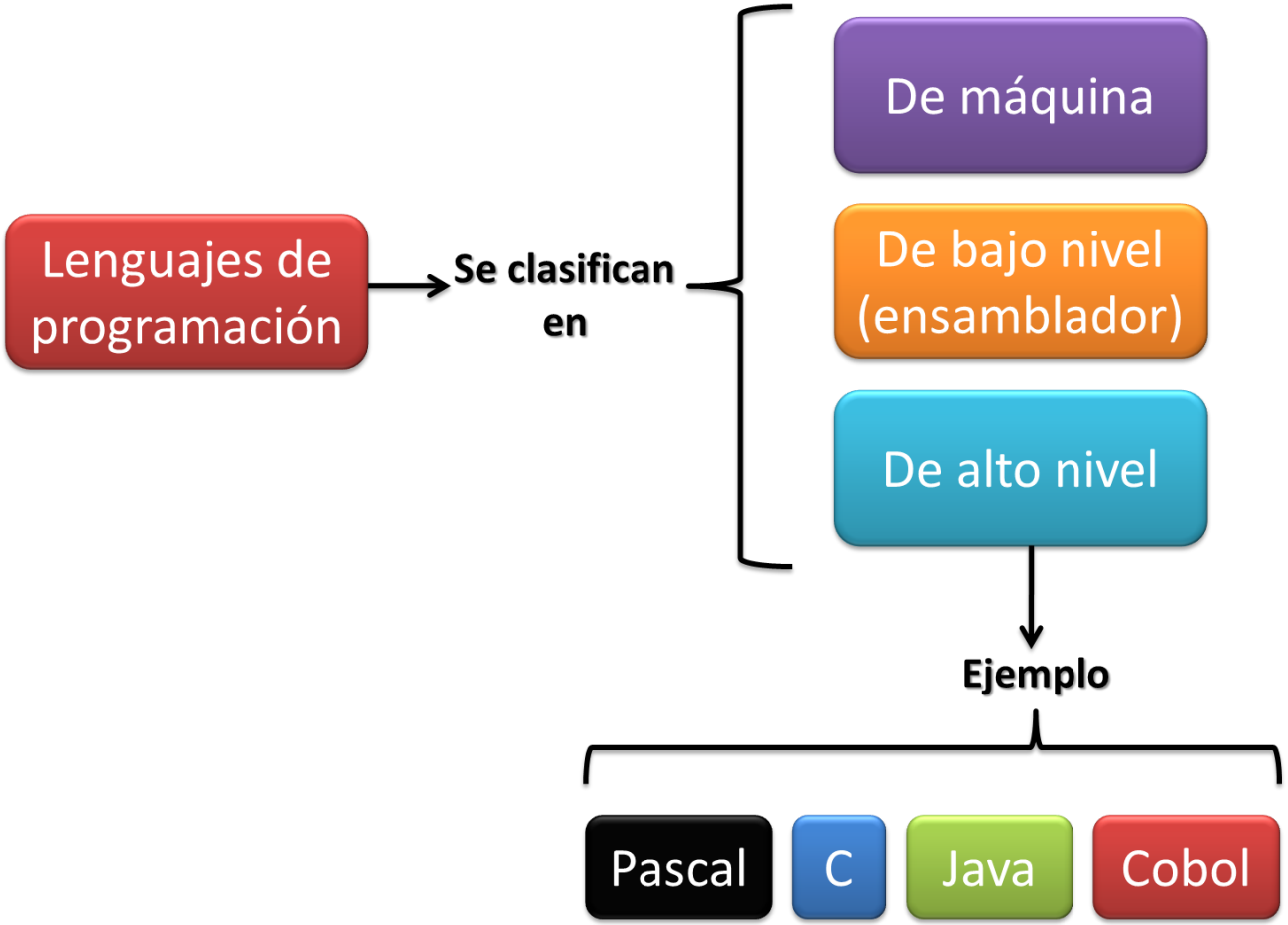
Vargas Suasnavas Jonathan
Club de Robótica EPN

1.1. Lenguajes de Programación

- Es un idioma artificial creado para comunicarse con una máquina como un computador.
- Se los puede usar para crear programas que controlen el comportamiento físico y lógico de una máquina, expresar algoritmos o como modo de comunicación humano máquina.
- Tienen un conjunto de símbolos y reglas sintácticas y semánticas.

Vargas Suasnavas Jonathan
Club de Robótica EPN

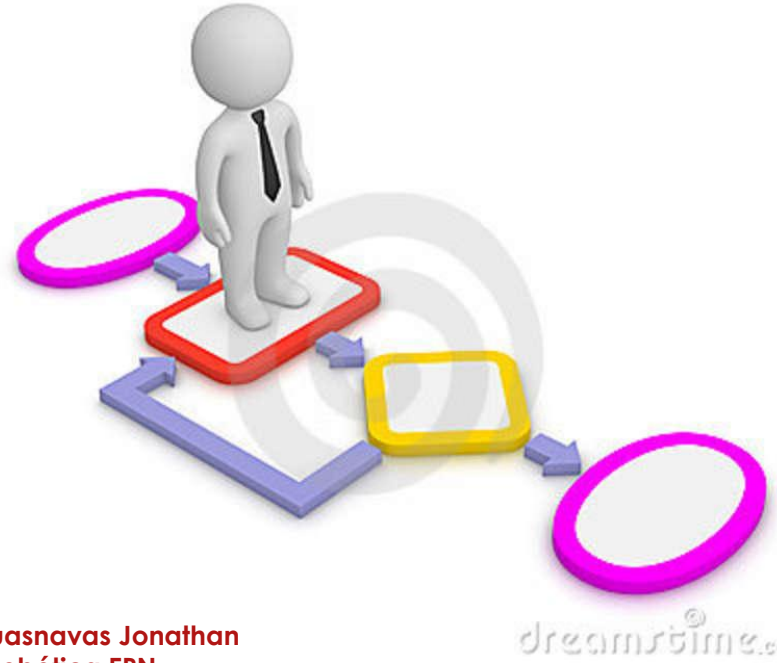
Lenguajes de Programación

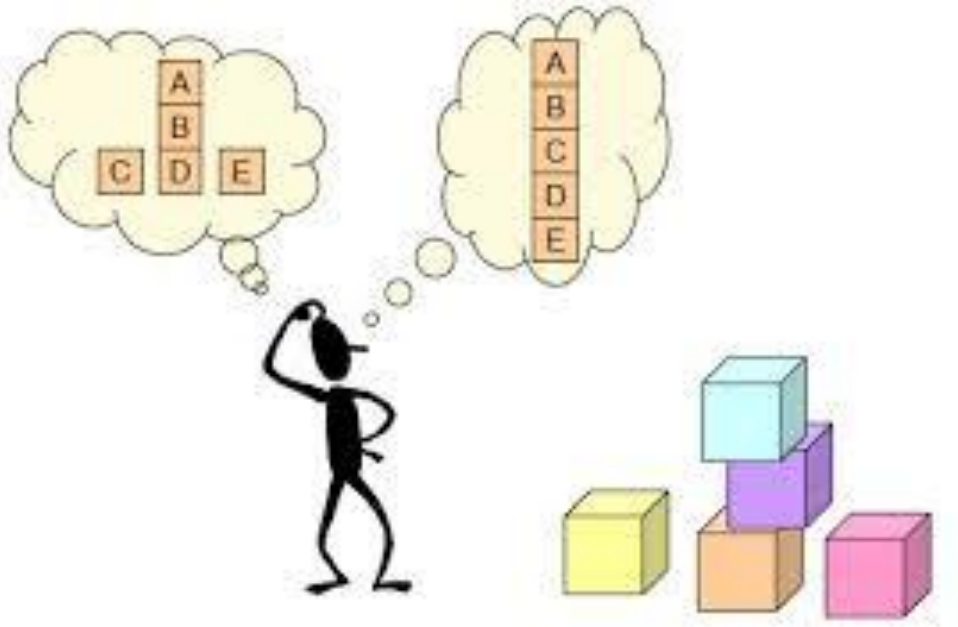


Vargas Suasnavas Jonathan
Club de Robótica EPN

2. ALGORITMO

- Conjunto de pasos que permiten resolver un problema.

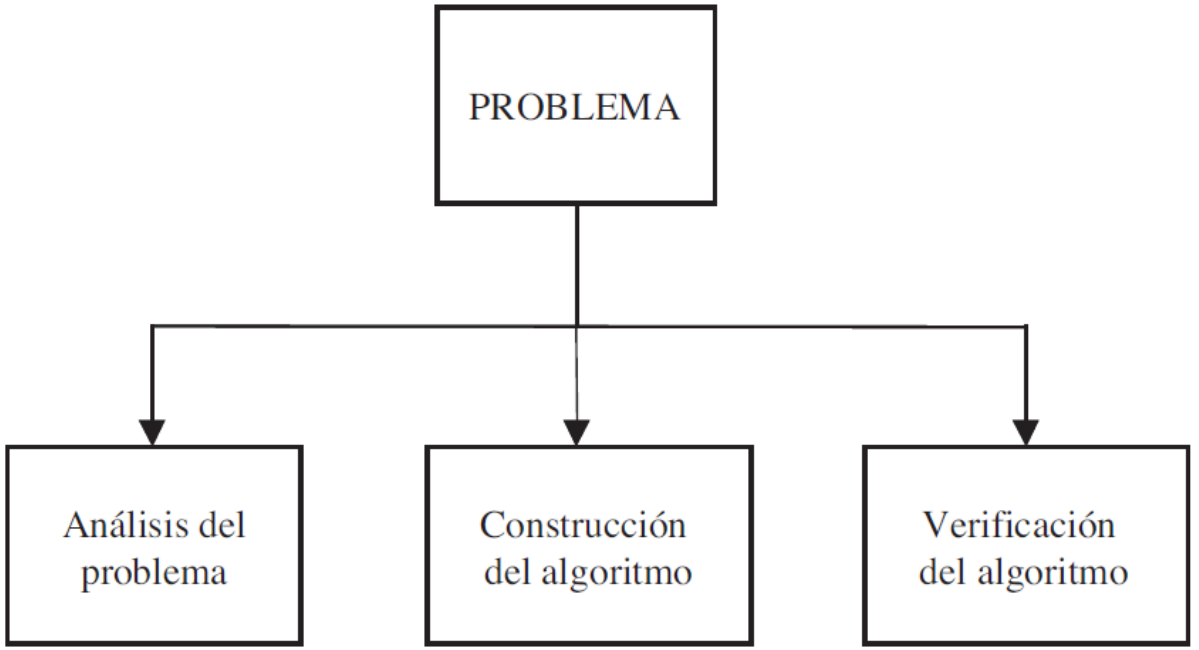




Vargas Suasnavas Jonathan
Club de Robótica EPN

2.1. CARACTERÍSTICAS DE LOS ALGORITMOS

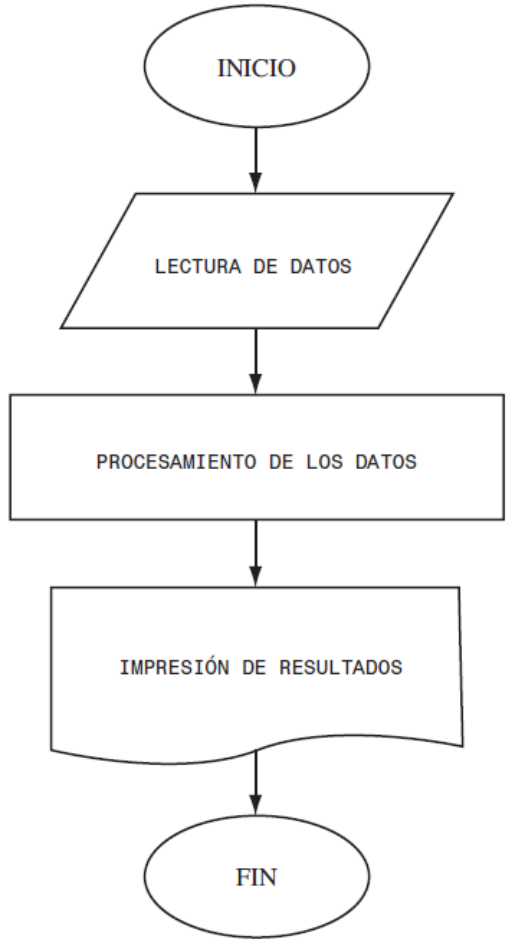
- Precisión (Los pasos deben ser claros)
- Determinismo (a datos idénticos deben arrojar el mismo resultado)
- Finitud (Siempre debe tener una longitud finita)



2.2. PARTES DE UN ALGORITMO

3. DIAGRAMAS DE FLUJO

- Representación Gráfica de un algoritmo
- A partir del mismo se puede escribir el programa en cualquier lenguaje de programación.



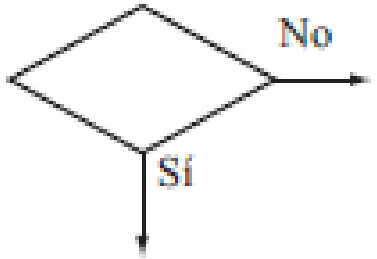
Vargas Suasnavas Jonathan
Club de Robótica EPN

3.1. SÍMBOLOS USADOS EN DIAGRAMAS DE FLUJO

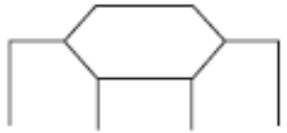
Inicio o fin de un programa

Datos de entrada

Procesos



Decisiones o lazos



Decisiones múltiples

Impresión de un resultado

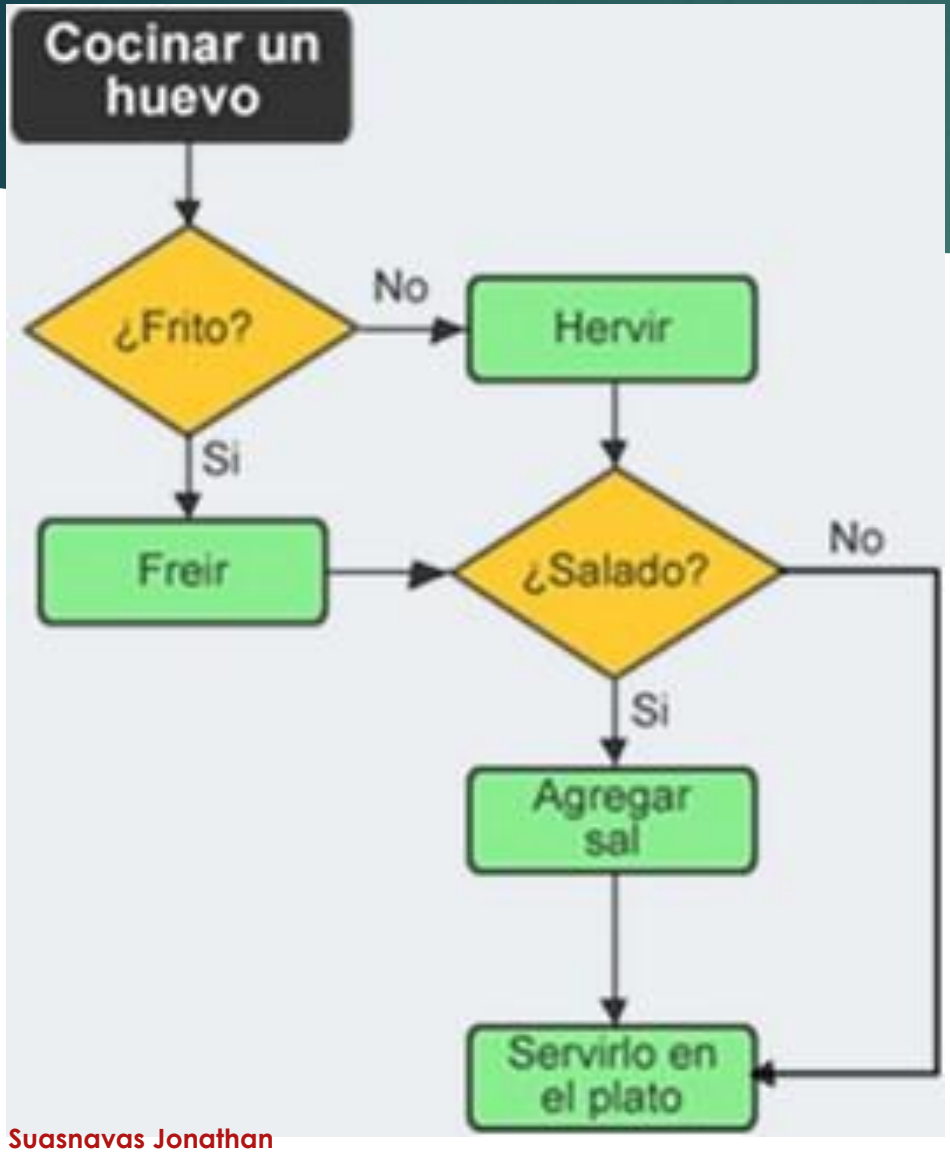
Dirección del flujo

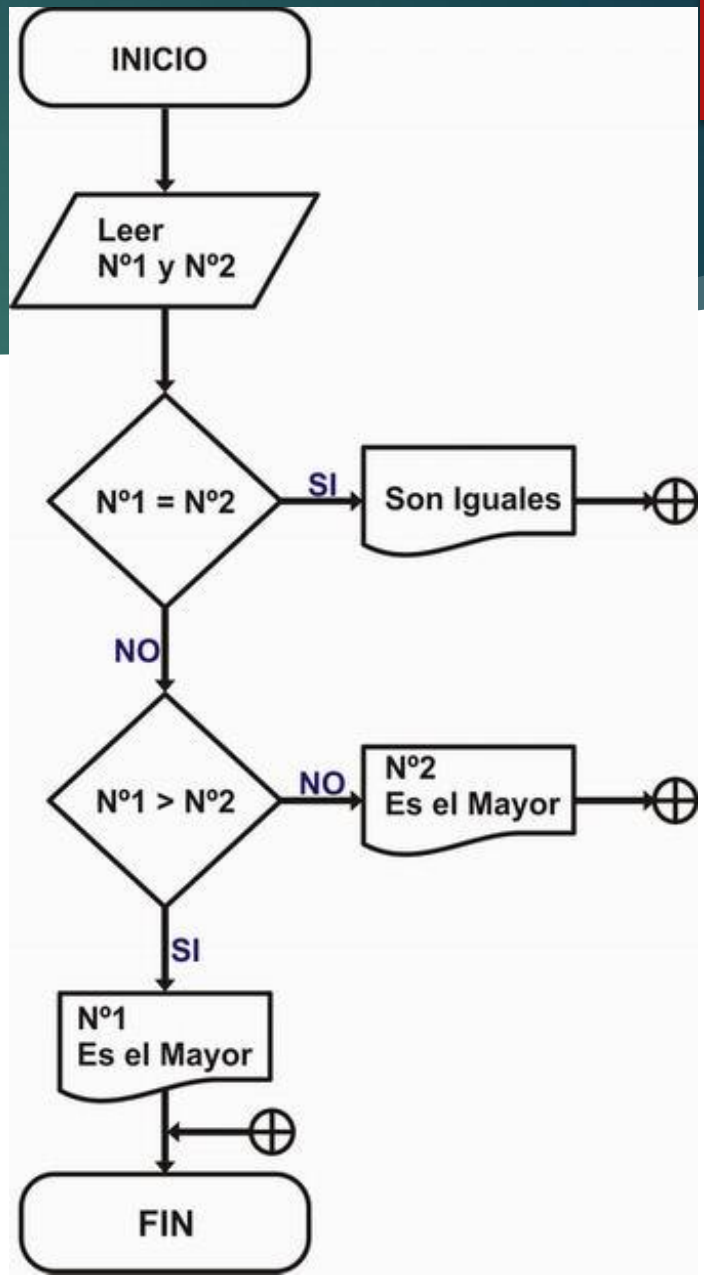
Conexión dentro de la misma página

Conexión de diferentes páginas

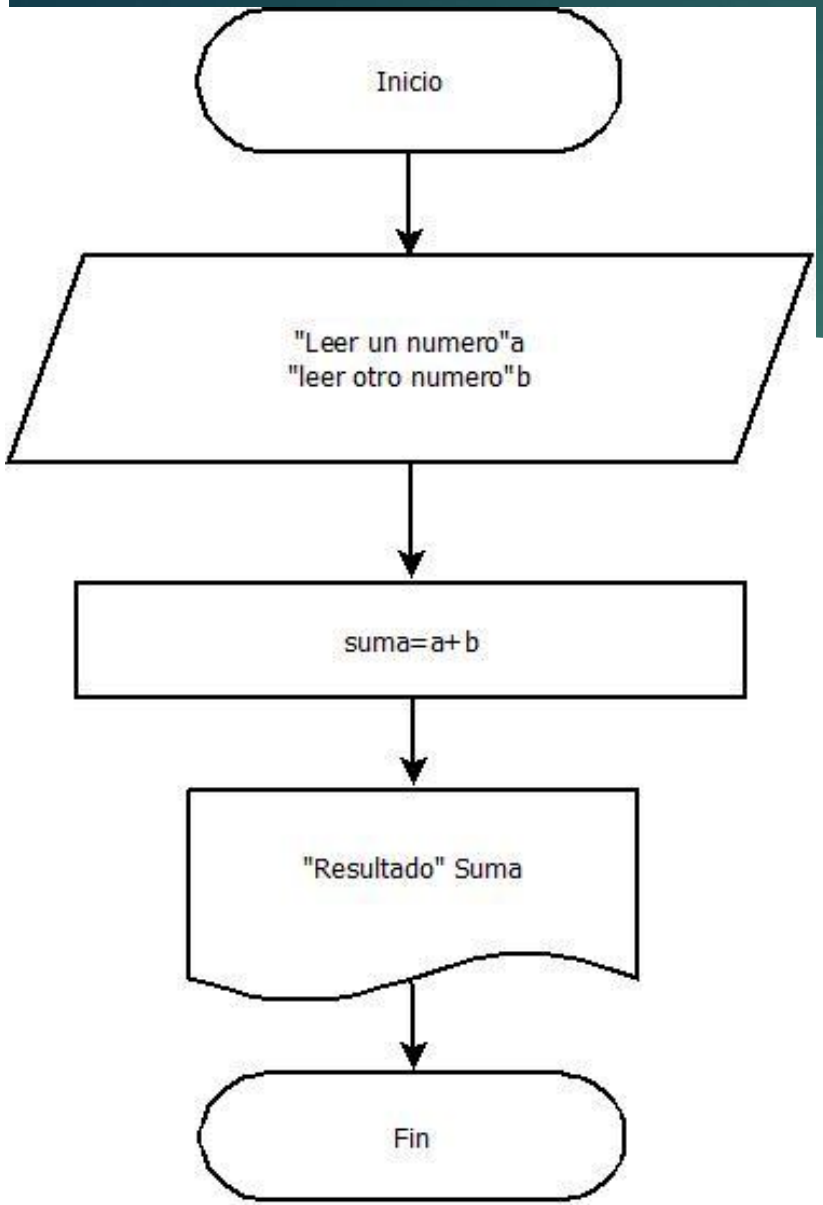
Módulo de subproblemas

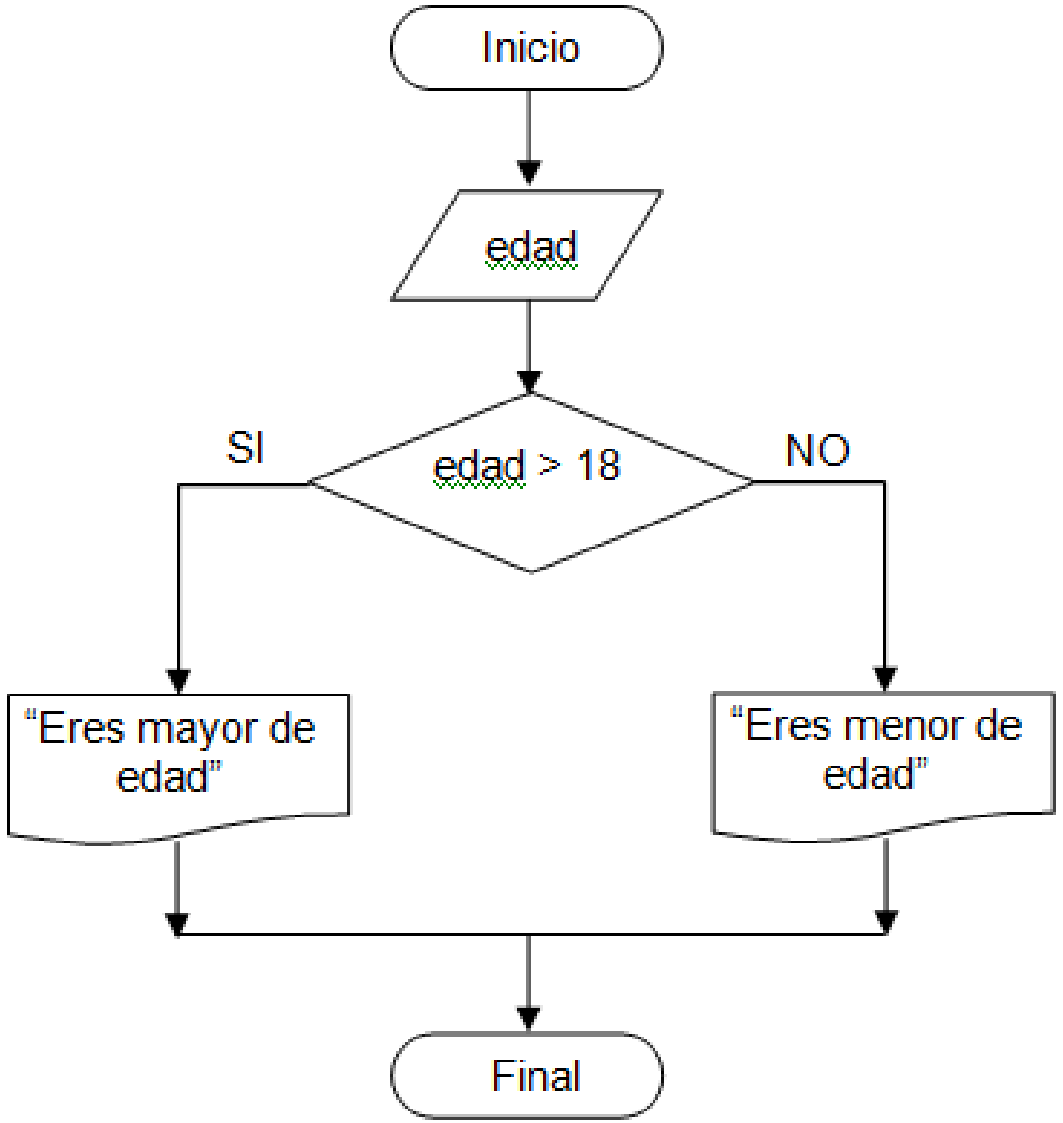
Vargas Suasnavas Jonathan
Club de Robótica EPN





Vargas Suasnavas Jonathan
Club de Robótica EPN

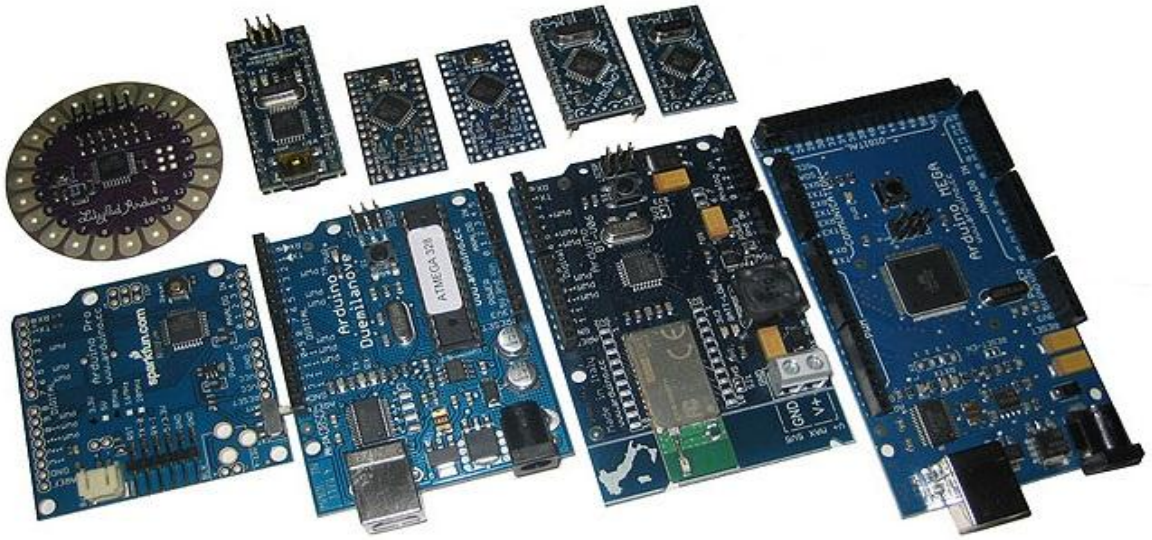




Vargas Suasnavas Jonathan
Club de Robótica EPN

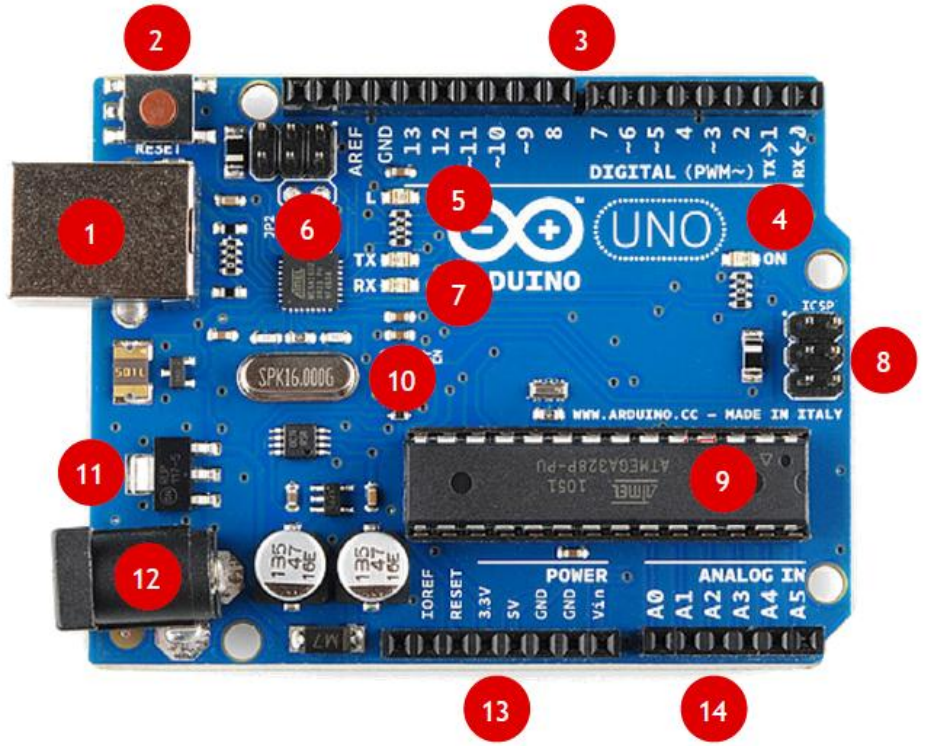
¿QUÉ ES ARDUINO?

- Plataforma de Hardware Libre, basada en una placa con un microcontrolador y un entorno de desarrollo.
- El hardware tiene una placa con microcontrolador Atmel AVR.
- Se programa mediante el lenguaje arduino (basado en Wiring) y el entorno de desarrollo arduino (basado en Processing)



Vargas Suasnavas Jonathan
Club de Robótica EPN

PARTES DE ARDUINO



1. Conector USB para el cable Tipo AB
2. Pulsador de Reset
3. Pines de E/S digitales y PWM
4. LED verde de placa encendida
5. LED naranja conectado al pin13
6. ATmega 16U2 encargado de la comunicación con el PC
7. LED TX (Transmisor) y RX (Receptor) de la comunicación serial
8. Puerto ICSP para programación serial
9. Microcontrolador ATmega 328, cerebro del Arduino
10. Cristal de cuarzo de 16Mhz
11. Regulador de voltaje
12. Conector hembra 2.1mm con centro positivo
13. Pines de voltaje y tierra
14. Entradas análogas

Vargas Suasnavas Jonathan
Club de Robótica EPN

Especificaciones Técnicas de Arduino

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Vargas Suasnavas Jonathan
Club de Robótica EPN

1. ESTRUCTURA DE UN PROGRAMA

```
// LIBRERIAS ADICIONALES
// DECLARACION DE VARIABLES

void setup()
{
  // SE COLOCAN LAS CONFIGURACIONES INICIALES

}

void loop()
{
  //EL PROGRAMA PROPIAMENTE DICHO

}
```

- COMENTARIOS
- FUNCIÓN SETUP
- FUNCIÓN LOOP

Vargas Suasnavas Jonathan
Club de Robótica EPN

2. TIPOS DE VARIABLES

- Datos o conjunto de datos que cambian su valor con la ejecución del programa

- Booleano
- Entero
- Carácter
- Byte
- Word
- Double
- Float

```
boolean encendido=true;

int pinLed=5;

char letra= 'a';

float fraccionario=3.1416;

void setup()
{
  // SE COLOCAN LAS CONFIGURACIONES INICIALES

}

void loop()
{
  double decimal=10.10;
}
```

Vargas Suasnavas Jonathan
Club de Robótica EPN

Tipo	Tam. Bits	Dígitos de precisión	Rango	
			Min	Max
Bool	8	0	0	1
Char	8	2	-128	127
Signed char	8	2	-128	127
unsigned char	8	2	0	255
short int	16	4	-32,768	32,767
unsigned short int	16	4	0	65,535
Int	32	9	-2,147,483,648	2,147,483,647
unsigned int	32	9	0	4,294,967,295
long int	32	9	-2,147,483,648	2,147,483,647
unsigned long int	32	9	0	4,294,967,295
long long int	64	18	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
unsigned long long int	64	18	0	18,446,744,073,709,551,615
Float	32	6	1.17549e-38	3.40282e+38
Double	64	15	2.22507e-308	1.79769e+308

3. OPERADORES

Operadores booleanos

- **&&** (and)
- **||** (or)
- **!** (not)

Operadores aritméticos

- **=** (asignación)
- **+** (suma)
- **-** (resta)
- ***** (multiplicación)
- **/** (división)
- **%** (módulo)

Operadores de comparación

- **==** (igual que)
- **!=** (no igual que)
- **<** (menor que)
- **>** (mayor que)
- **<=** (menor o igual que)
- **>=** (mayor o igual que)

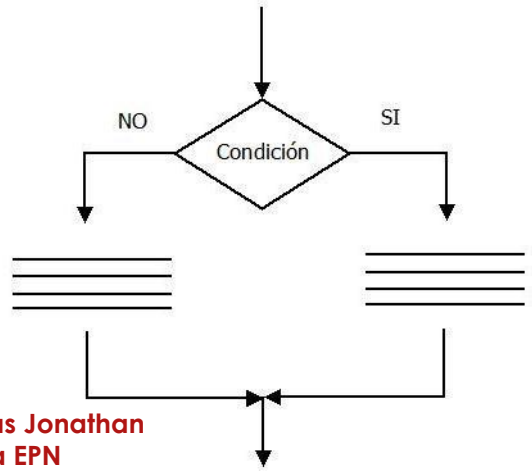
Vargas Suasnavas Jonathan
Club de Robótica EPN

4. ESTRUCTURAS DE CONTROL

if (si)

```
if (unaVariable ?? valor)
{
ejecutaInstrucciones;
}
```

Estructura IF



if… else (si….. sino ..)

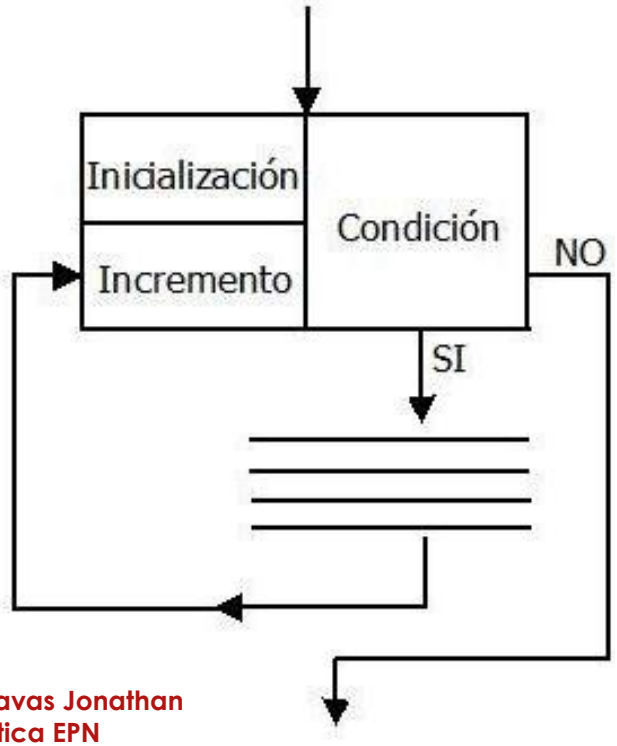
```
if (inputPin == HIGH) // si el valor de la entrada inputPin es alto
      {
            instruccionesA; //ejecuta si se cumple la condición
      }
else
      {
            instruccionesB; //ejecuta si no se cumple la condición
      }
```

```
if (inputPin < 500)
      {
            instruccionesA; // ejecuta las operaciones A
      }
else if (inputPin >= 1000)
      {
            instruccionesB; // ejecuta las operacione B
      }
else
      {
            instruccionesC; // ejecuta las operaciones C
      }
```

Vargas Suasnavas Jonathan
Club de Robótica EPN

ESTRUCTURAS DE CONTROL

Estructura FOR



for

```
for (inicialización; condición; expresión)
{
      ejecutaInstrucciones;
}
```

```
for (int i=0; i<20; i++)          // declara i, prueba que es menor que
                                  20, incrementa i en 1
{
      digitalWrite(13, HIGH);     // envía un 1 al pin 13
      delay(250);                 // espera ¼ seg.
      digitalWrite(13, LOW);      // envía un 0 al pin 13
      delay(250);                 // espera ¼ de seg.
}
```

Vargas Suasnavas Jonathan
Club de Robótica EPN

ESTRUCTURAS DE CONTROL

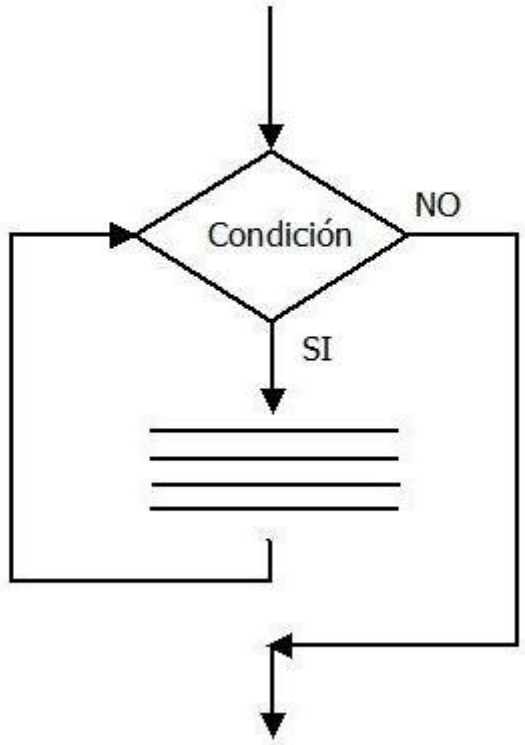
while

```
while (unaVariable ?? valor)
{

      ejecutarSentencias;
}
```

```
While (unaVariable < 200)      // testea si es menor que 200
{
      instrucciones;           // ejecuta las instrucciones entre llaves
      unaVariable++;           // incrementa la variable en 1
}
```

Estructura WHILE

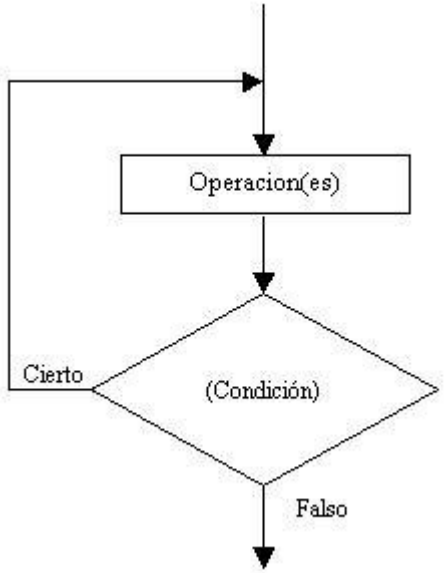


Vargas Suasnavas Jonathan
Club de Robótica EPN

ESTRUCTURAS DE CONTROL

do… while

```
do
{
      Instrucciones;
} while (unaVariable ?? valor);
```




```
do
{
      x = leeSensor();
      delay(50);
} while (x < 100);
```

Vargas Suasnavas Jonathan
Club de Robótica EPN

ESTRUCTURAS DE CONTROL

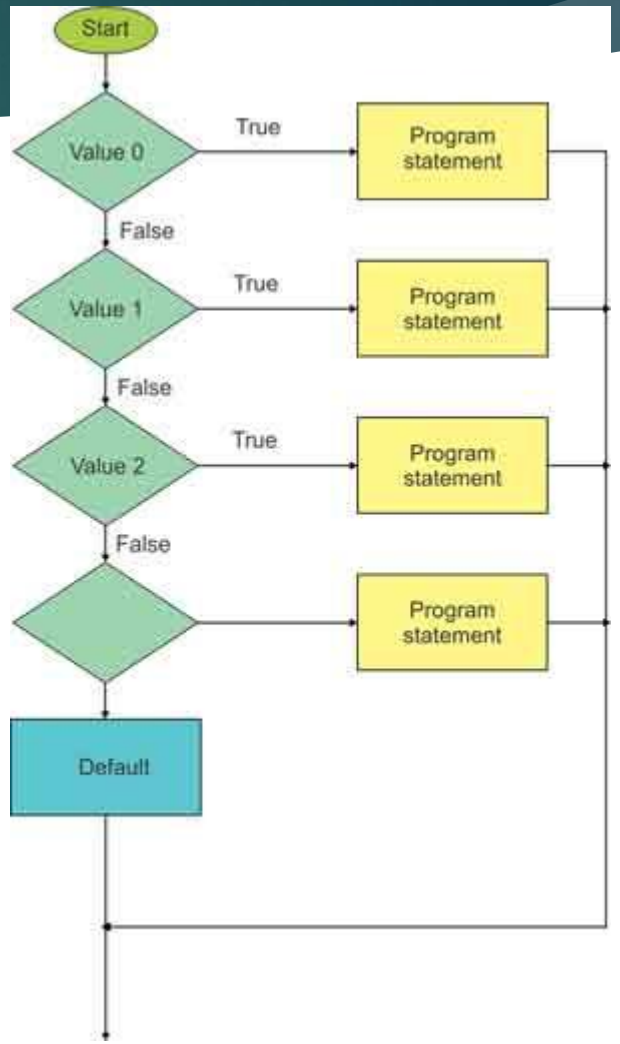
Switch/ Case

Syntax

```
switch (var) {
  case label:
    // statements
    break;
  case label:
    // statements
    break;
  default:
    // statements
}
```

Example

```
switch (var) {
  case 1:
    //do something when var equals 1
    break;
  case 2:
    //do something when var equals 2
    break;
  default:
    // if nothing else matches, do the default
    // default is optional
}
```



Vargas Suasnavas Jonathan
Club de Robótica EPN

ESTRUCTURA DE CONTROL

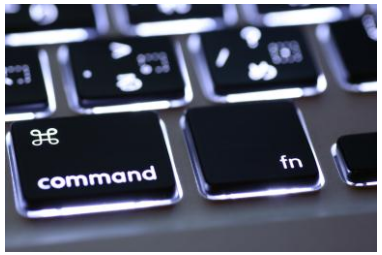
goto

```
for(byte r = 0; r < 255; r++)
{
   for(byte g = 255; g > -1; g--)
   {
      for(byte b = 0; b < 255; b++)
      {
        if (analogRead(0) > 250)
       {
        goto salir;
       }
        // more statements ...
      }
   }
}
salir:
//continuación del programa
```

Vargas Suasnavas Jonathan
Club de Robótica EPN

5. FUNCIONES

- Conjunto de líneas de código que realizan una tarea específica, puede retornar un valor.
- Pueden tomar parámetros que modifiquen su funcionamiento
- Permiten descomponer grandes problemas de una manera simple



Vargas Suasnavas Jonathan
Club de Robótica EPN

5.1. FUNCIONES DIGITALES

pinMode()

Permite configurar un pin

pinMode(pin,modo)

```
pinMode (13,OUTPUT);
pinMode (a,INPUT);
```

digitalRead()

Leer un pin digital (0 ó 1)

digitalRead(pin)

```
int a = digitalRead (13);
```

digitalWrite()

Escribir un pin digital con 1 ó 0

digitalWrite(pin,estado)

```
digitalWrite (13,HIGH);
digitalWrite (13,LOW);
```

Vargas Suasnavas Jonathan
Club de Robótica EPN