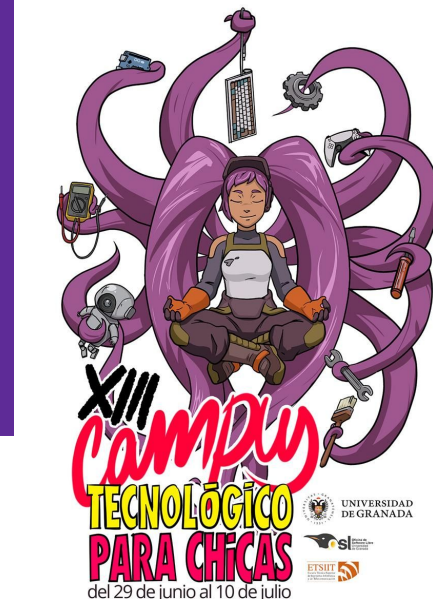


Introducción a la programación

ARDUINO Y C++



amazon cívica UNIT4 CGI GPU Solutions rti

cívica
PEOPLE BEYOND TECH

GPU Solutions

rti

UNIT4



ZEN ZORRITO
TECNOLOGÍAS SL

amazon

MONTA
JE^S
LECTRICOS
JIMENEZ

Si2
soluciones

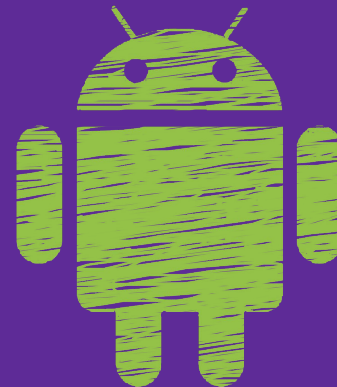
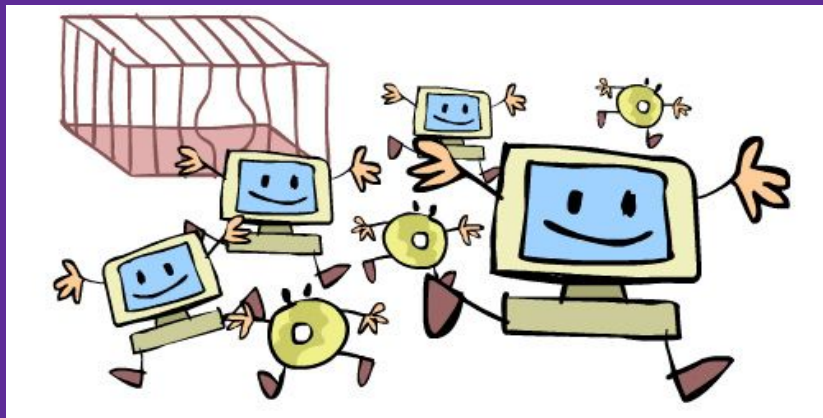
ETSIT
Escuela Técnica Superior
de Ingeniería en Telecomunicación
y de Teletecnología



¿QUÉ HAREMOS HOY?

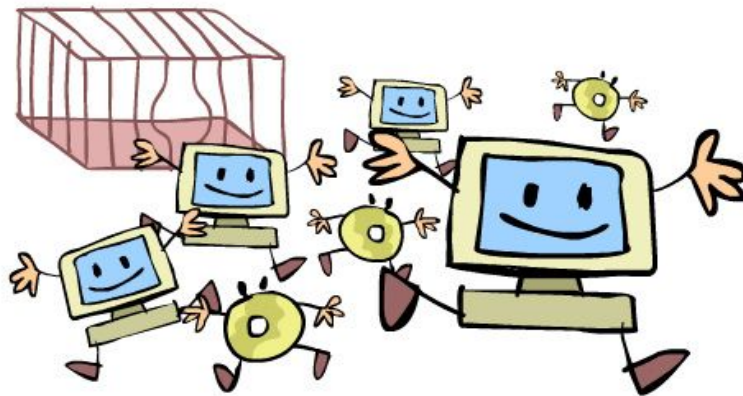
- ❑ ¿Qué es software libre?
- ❑ ARDUINO
- ❑ PRÁCTICA 1: BLINK
- ❑ PRACTICA 2: BLINK CON LED EN PIN 13
- ❑ PRÁCTICA 3: Led+botón
- ❑ PRÁCTICA 4: Led+botón+Puerto Serie
- ❑ PRACTICA 5: LED+ BOTON + SERVO

¿SABÉIS QUE ES EL SOFTWARE LIBRE?



¿Qué es el software libre?

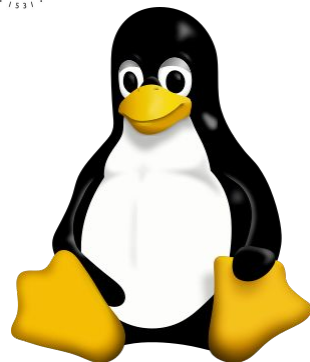
- ❑ Software que le da al usuario la libertad de:
 - ❑ Compartirlo.
 - ❑ Estudiarlo.
 - ❑ Modificarlo.
- ❑ Se llama software libre porque el usuario es libre.



¿QUÉ SOFTWARE LIBRES EXISTEN?



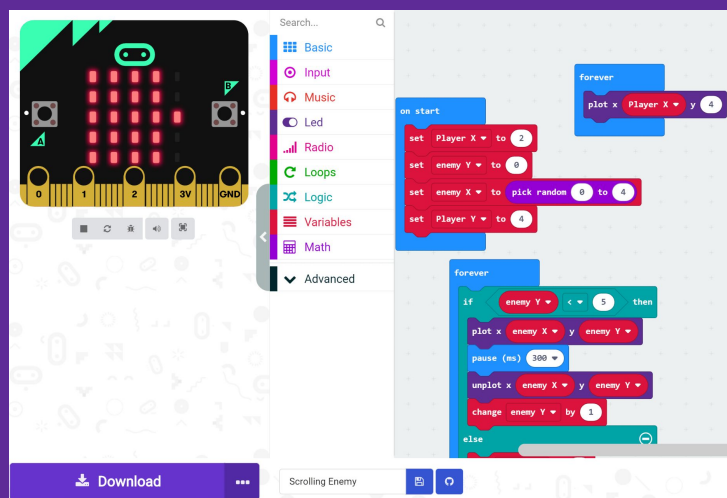
LibreOffice
The Document Foundation



¡AQUÍ OS DEJO UNOS EJEMPLOS!



¿QUIÉN HA PROGRAMADO ANTES?



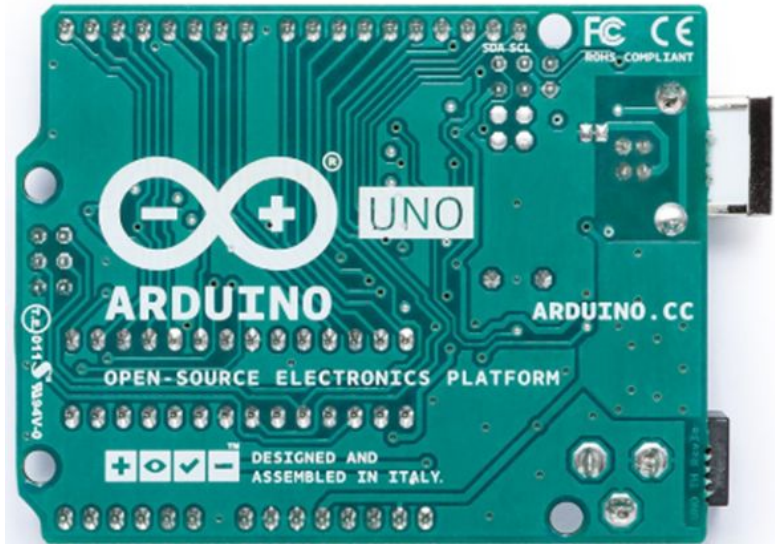
```
a = int(input("enter first number: "))
b = int(input("enter second number: "))
sum = a + b
print("sum:", sum)
```

¿QUÉ HAREMOS HOY?

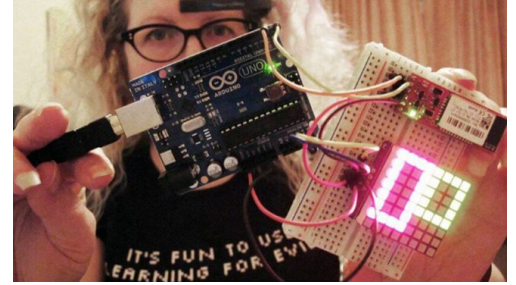
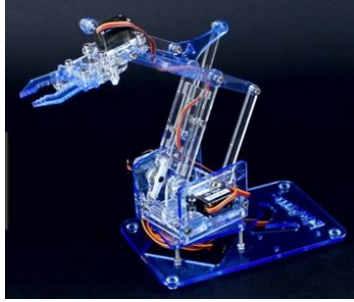
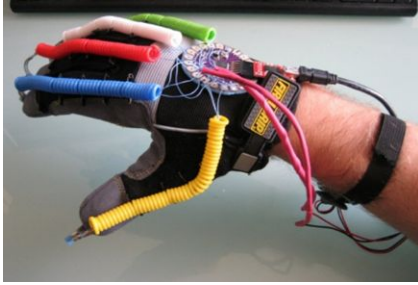
- ❑ ¿Qué es software libre?
- ❑ ARDUINO
- ❑ PRÁCTICA 1: BLINK
- ❑ PRACTICA 2: BLINK CON LED EN PIN 13
- ❑ PRÁCTICA 3: Led+botón
- ❑ PRÁCTICA 4: Led+botón+Puerto Serie
- ❑ PRACTICA 5: LED+ BOTON + SERVO

¿Qué es Arduino?

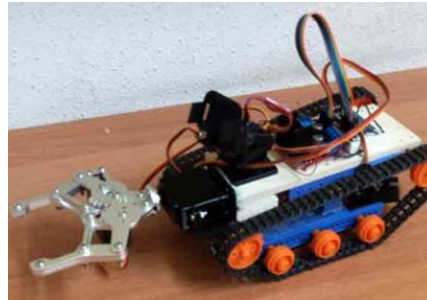
- ☐ Arduino es un micro controlador.
- ☐ Es libre (no es único).
- ☐ Flexible y fácil de manejar.
- ☐ Puertos de entrada y salida analógicos y digitales.
- ☐ Puertos de comunicación.
- ☐ Modular y ampliable.
- ☐ Software: C++, programación por bloques (scratch for arduino)



¿QUÉ PUEDO HACER CON ARDUINO?



¡TODO LO QUE SE TE OCURRA!



TIPOS DE ARDUINO



Arduino Uno



Arduino Nano



LilyPad



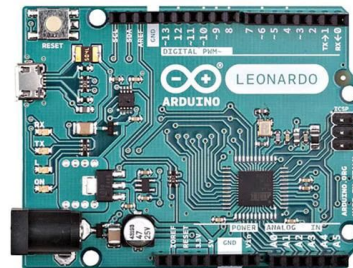
Arduino Micro



Node MCU



Arduino Mega



Arduino Leonardo

... ¡y muchos más!

ARDUINO UNO



Pines Digitales

D0	D5	D10
D1	D6	D11
D2	D7	D12
D3	D8	D13
D4	D9	

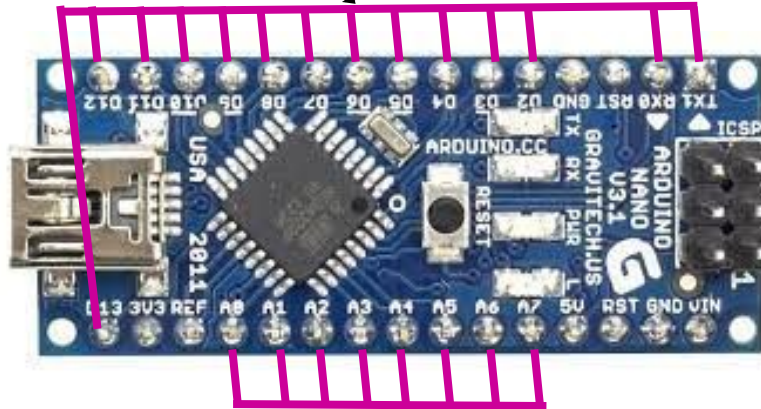
Pines Analógicos

A0, A1, A2, A3, A4, A5

ARDUINO NANO

Pines Digitales

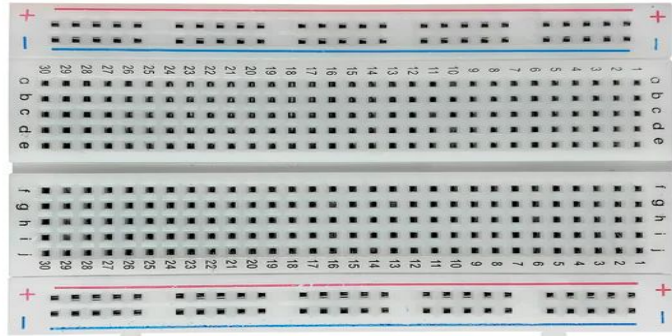
D0	D5	D10
D1	D6	D11
D2	D7	D12
D3	D8	D13
D4	D9	



Pines Analógicos

A0, A1, A2, A3, A4, A5, A6, A7

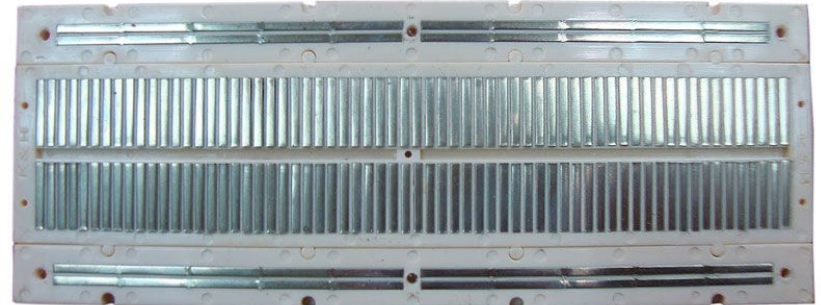
PROTOBOARD



Placa Protoboard



Placa Protoboard por dentro



Tipos de señales

Analógicas:

-  Pueden tomar infinitos valores entre su valor mínimo y su valor máximo.

Digitales:

-  Sólo puede tomar dos valores (0 o 1) donde:
-  **0** es el mínimo, off, falso, tierra.
-  **1** es el máximo, on, verdadero, fuente.

¿CÓMO FUNCIONA ARDUINO?

Hola mundo!

void setup() {

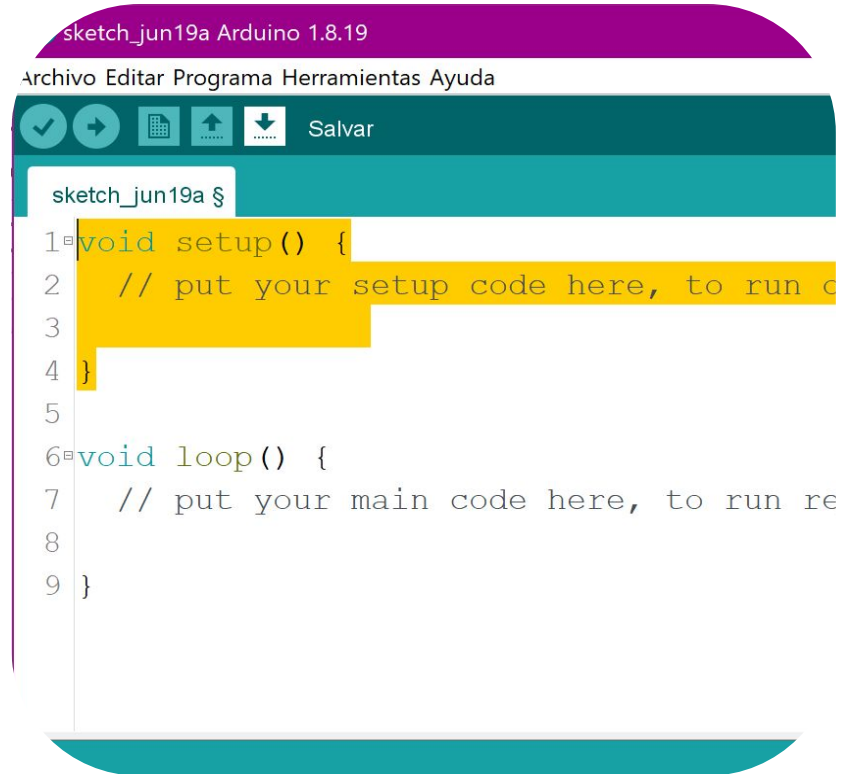
**En setup tenemos un listado de las cosas
conectadas, como una lista de la compra.**

}

void loop() {

Donde se realizarán las acciones

}



Entradas/salidas digitales

- ❑ Para declarar un pin como salida en arduino, hay que hacerlo en:

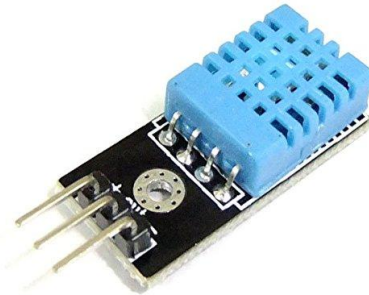
```
setup(){AQUI DENTRO}
```

- ❑ Los pines que usaremos como entrada:

```
pinMode(numero_de_pin, INPUT);
```

- ❑ Los que usaremos como salida:

```
pinMode(numero_de_pin, OUTPUT);
```



- ❑ El arduino no tiene capacidad para trabajar con señales analógicas, tiene que convertirlas a digitales.

- ❑ Pines analógicos: A0,A1,A2,A3,A4,A5

- ❑ Para tratar Entradas/salidas analógicas usamos las **funciones:**

analogRead(pin); // que lee del pin analógico especificado

analogWrite(pin); // que recibe un valor analógico en el pin especificado

map(valor, de bajo, de alto, a bajo, a alto); // para convertir un rango de valores en otro.

- ❑ Entradas analógicas tienen ~ (PWM)

analogWrite(pin);

Programación

C++



¿Qué es un algoritmo?

- ❑ Secuencia ordenada de instrucciones que resuelve un problema concreto.

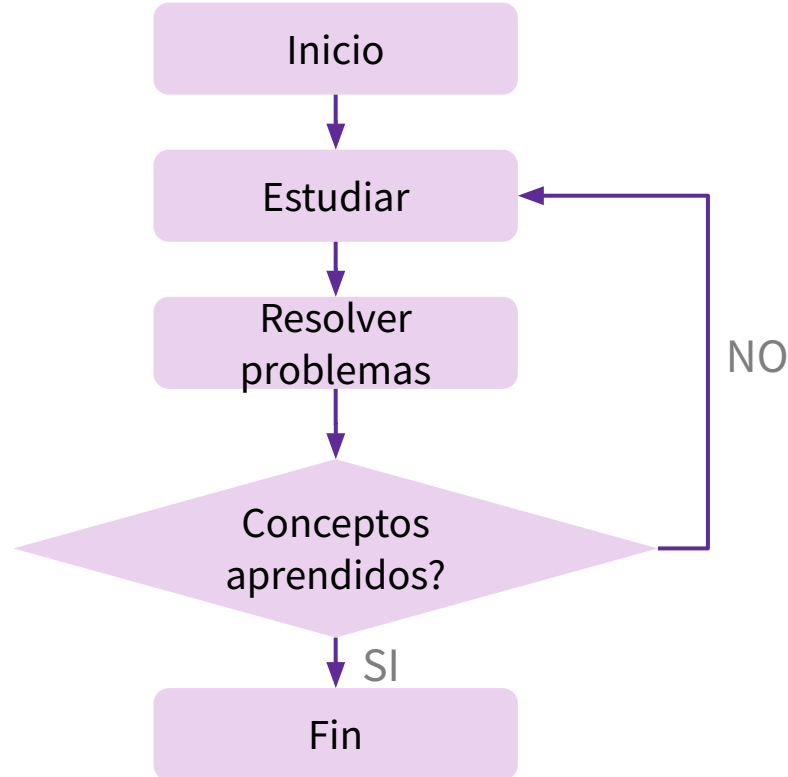
Lenguajes de programación:

- ❑ Bajo nivel: Ensamblador.
- ❑ Alto nivel: C++, C, Java, python ...

¿Qué es un programa?

- ❑ Es un conjunto de instrucciones, especificadas en un lenguaje de programación concreto que pueden ejecutarse en un ordenador o en un microcontrolador.
- ❑ Compilación.
 - ❑ Traducir un programa a código ejecutable por la máquina.

¿Qué es un algoritmo?



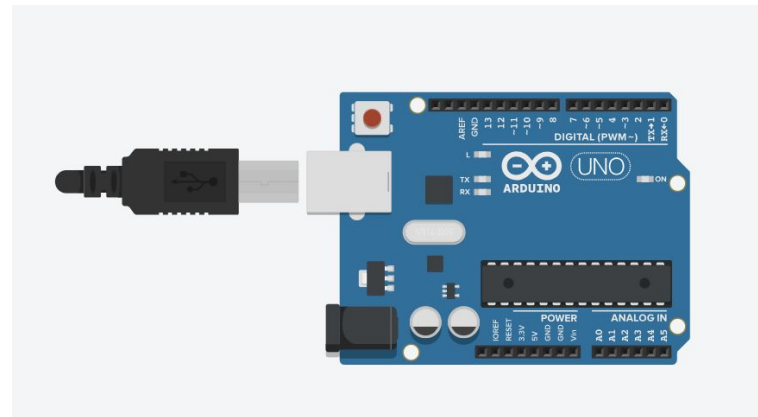
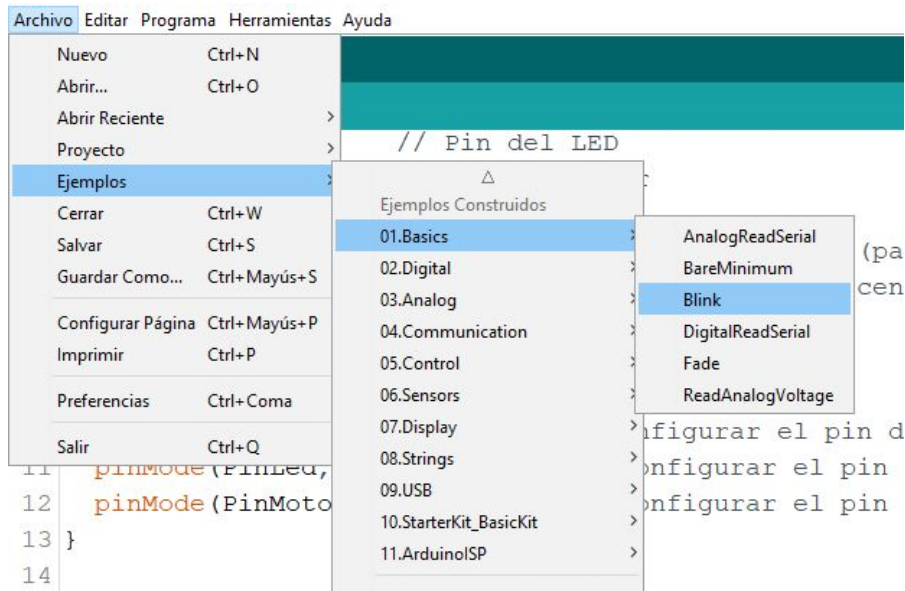
Los algoritmos son la base de la resolución de problemas y los utilizamos todos los días

¿QUÉ ES EL IDE DE ARDUINO?

- ❑ Usamos el IDE de arduino.
- ❑ Vamos a programar en C++.
- ❑ Funciones básicas.
 - ❑ **void setup();**
 - ❑ **void loop();**
- ❑ Ejemplo Blink.
- ❑ Compilar y subir el programa a la placa.



Usar ejemplo Blink y le añadimos un delay (espera de tiempo).



```
void setup() {  
    // initialize digital pin LED_BUILTIN as an output.  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)  
    delay(1000);                        // wait for a second  
    digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW  
    delay(1000);                        // wait for a second  
}
```

MI PRIMER PROGRAMA

Hola mundo!

void setup() {

Aquí vendrán las configuraciones de nuestros pines como entradas o salidas y la configuración del puerto serie si es necesaria.

Todo incluido entre llaves.

}

void loop() {

Aquí pondremos las instrucciones que
ejecutará nuestro programa.

}

Sentencias y Expresiones

Sentencia:

-  Es la parte del código fuente que el compilador traduce en una instrucción que entiende el hardware.

-  Siempre debe terminar en

;

-  Se ejecutan de forma secuencial



Tipos de datos

☐ Numéricos:

☐ Int lado1;

☐ lado1 = 3;

☐ Double lado2;

☐ lado2 = 3,5;

☐ Lógicos

☐ Bool midato;

☐ midato = true;

(sólo toma valor verdadero o falso)

☐ Complejos:

☐ int vector[2];

☐ Vector[0] = 3;

☐ Vector[1] = 2;

Tipos

- ❏ Matemáticos:
 - ❏ $+$, $-$, $*$, $/$, $\%$.
- ❏ Relacionales:
 - ❏ $==$, $!=$, $<$, $>$, $>=$, $<=$
- ❏ Lógicos
 - ❏ $\&\&$, and , $\|\|$, or , $!$

P	!P
True	False
False	True

P	Q	P&&Q	P Q
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

¿Qué son los condicionales en español?

En programación, los condicionales son como las decisiones.
por ejemplo:

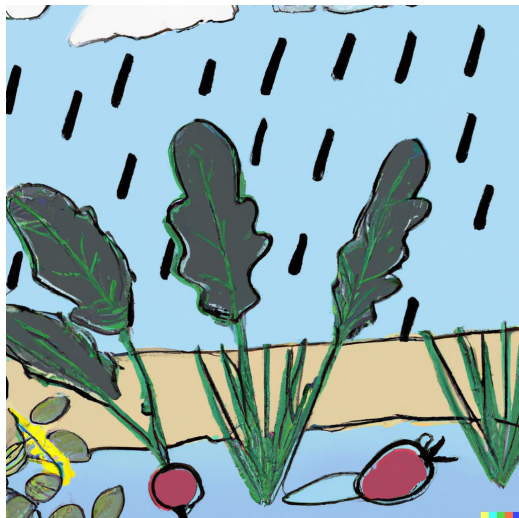
```
si (tienes hambre) {  
    come una fruta;  
}  
sino si (tienes sed) {  
    bebe agua;  
}  
sino {  
    vete a jugar y diviértete;  
}
```



- ❑ Pero en programación no podemos usar “si”, “si no” ... hay que hacerlo en inglés...

¿Cómo se dice “**Si** tienes hambre, come algo” en inglés?

If you are hungry, eat something



- ❑ Pues esto es exactamente igual:

```
if (está lloviendo){  
    muestra por pantalla que está  
    lloviendo  
}
```

CONDICIONALES

Y ahora nos quedarían los **símbolos de comparación**.

- ❑ > Mayor que
 - ❑ < Menor que
 - ❑ == Igual a
 - ❑ != Distinto de
- 3 es ... que 6
6 es ... que 6

```
int edad = 15;  
if (edad > 18) {  
    // Acción si la edad es mayor a 18  
    imprimir "Eres mayor de edad"  
} else {  
    // Acción si la edad no es mayor a 18  
    imprimir "Eres menor de edad"  
}
```

Condicional simple - Ejemplo - Comprobar el estado de un pin digital

```
EstadoPin=digitalRead(2);
```

```
If (EstadoPin == HIGH){
```

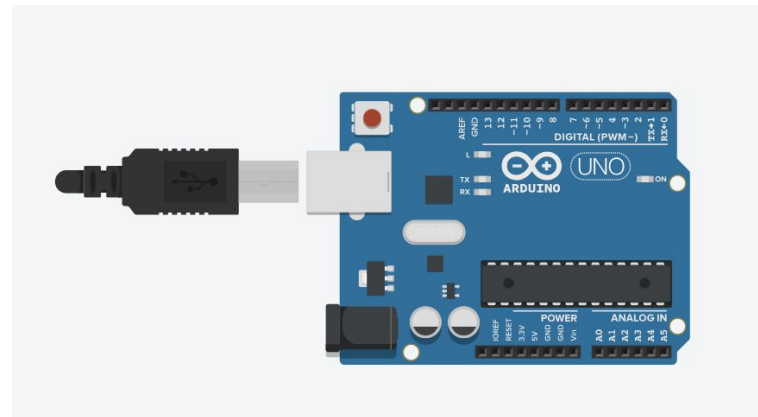
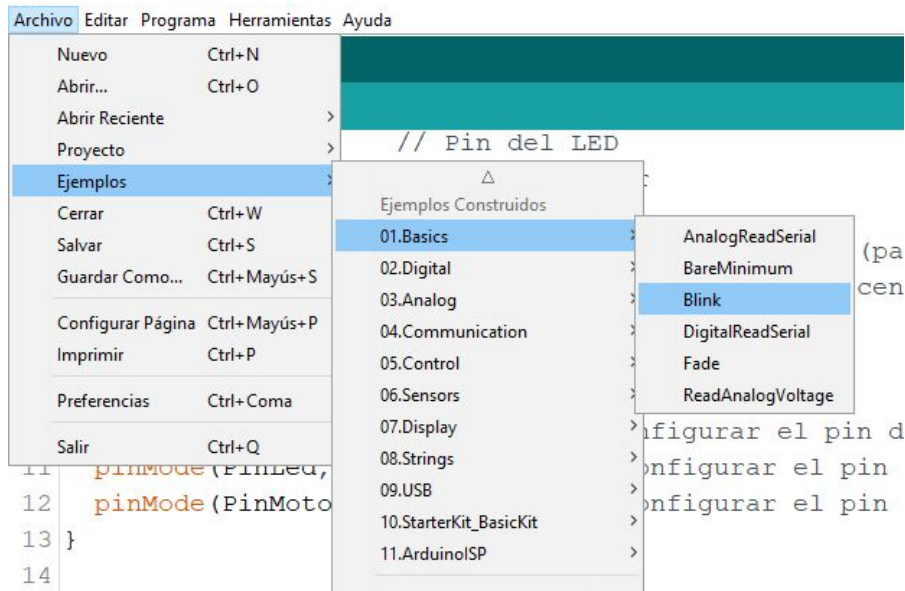
```
    digitalWrite(13,LOW);
```

```
}
```

¿QUÉ HAREMOS HOY?

- ❑ ¿Qué es software libre?
- ❑ ARDUINO
- ❑ PRÁCTICA 1: BLINK
- ❑ PRACTICA 2: BLINK CON LED EN PIN 13
- ❑ PRÁCTICA 3: Led+botón
- ❑ PRÁCTICA 4: Led+botón+Puerto Serie
- ❑ PRACTICA 5: LED+ BOTON + SERVO

Usar ejemplo Blink y le añadimos un delay (espera de tiempo).



```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);                     // wait for a second  
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);                     // wait for a second  
}
```

¿QUÉ HAREMOS HOY?

- ❑ ¿Qué es software libre?
- ❑ ARDUINO
- ❑ PRÁCTICA 1: BLINK
- ❑ PRACTICA 2: BLINK CON LED EN PIN 13
- ❑ PRÁCTICA 3: Led+botón
- ❑ PRÁCTICA 4: Led+botón+Puerto Serie
- ❑ PRACTICA 5: LED+ BOTON + SERVO

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}
```

**EJERCICIO: Conectar un led al pin 13.
¿Qué pasa cuando conectamos el led al
Pin 13?**

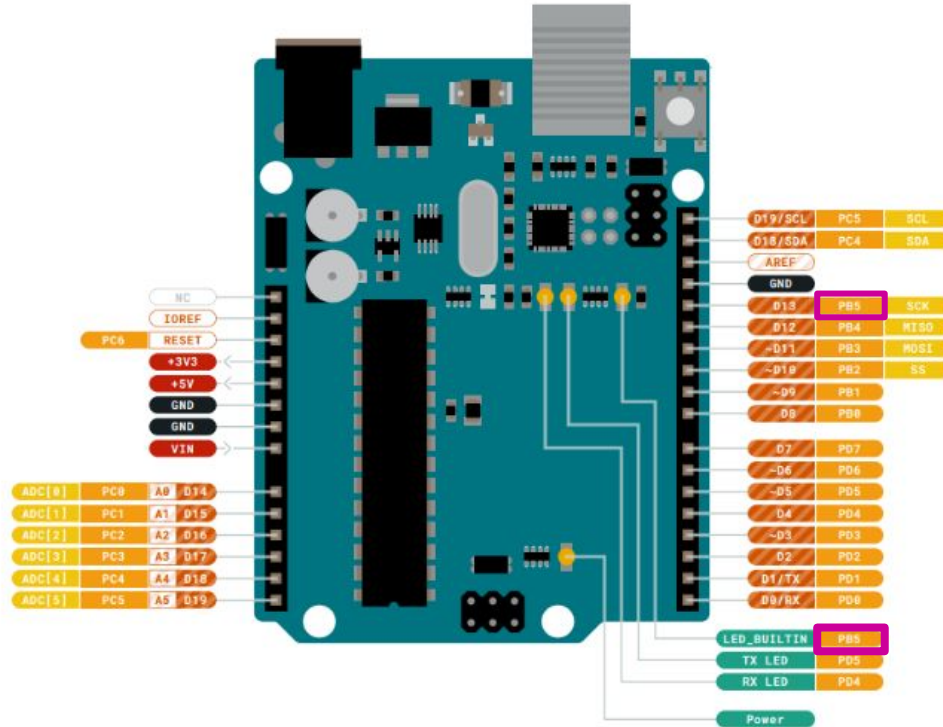
```
// the loop function runs over and over again  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

SOLUCIÓN

```

void setup() {
  // initialize d
  pinMode(LED_BUI
}

// the loop funct
void loop() {
  digitalWrite(LED
  delay(1000);
  digitalWrite(LED
  delay(1000);
}
  
```



ie voltage level)

the voltage LOW

¿QUÉ HAREMOS HOY?

- ❑ ¿Qué es software libre?
- ❑ ARDUINO
- ❑ PRÁCTICA 1: BLINK
- ❑ PRACTICA 2: BLINK CON LED EN PIN 13
- ❑ PRÁCTICA 3: Led+botón
- ❑ PRÁCTICA 4: Led+botón+Puerto Serie
- ❑ PRACTICA 5: LED+ BOTON + SERVO

Condicional simple

```
If (condición){  
    sentencias;
```

```
} EJERCICIO: Conectar un botón y un led.
```

Crear variable encendido = false

Crear variable pulsado = false

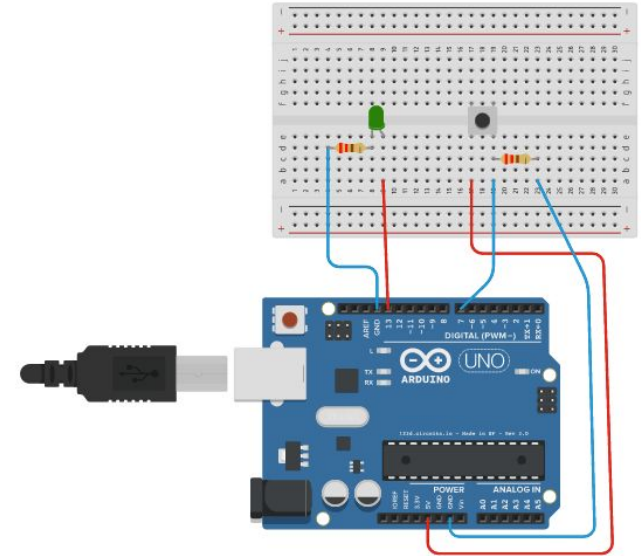
pulsado = Leer estado del botón

Si se ha pulsado el botón,

Encender el led.

encendido = true

pulsado = false



```
1
2 int PinBoton = 2; // Pin del botón
3 int PinLed = 13; // Pin del LED
4
5 boolean encendido = false;
6 boolean pulsado = false;
7
8 void setup() {
9   pinMode(PinBoton, INPUT); // Configurar el pin del botón como entrada
10  pinMode(PinLed, OUTPUT); // Configurar el pin del LED como salida
11 }
12
13 void loop() {
14   pulsado = digitalRead(PinBoton); // Leer el estado del botón
15
16   if (pulsado) {
17     digitalWrite(PinLed, HIGH); // Encender el LED
18     encendido = true; //cambiamos el estado a encendido
19     pulsado = false;
20   } else {
21     digitalWrite(PinLed, LOW); // Apagar el LED
22     encendido = false;
23   }
24 }
25
```

EJERCICIO: Conectar un botón y un led.

Crear variable encendido = false
Crear variable pulsado = false

pulsado = Leer estado del botón

Si se ha pulsado el botón,
Encender el led.
encendido = true
pulsado = false

Condicional compuesto

```
If (condición && condición){  
    sentencias;  
}
```

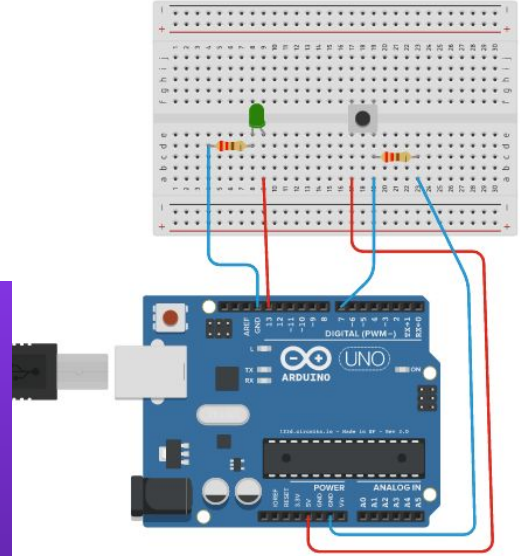
EJERCICIO: Conectar un botón y un led.

Crear variable encendido = false

Crear variable pulsado = false

pulsado = Leer estado del botón

**Si se ha pulsado el botón y el led está encendido,
Apagar el led.
encendido = false
pulsado = false**



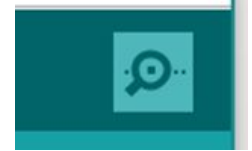
Si pulsas una vez, se enciende el LED, si pulsas otra vez, se apaga

```
1 int PinBoton = 2;  // Pin del botón
2 int PinLed = 13;   // Pin del LED
3 boolean encendido = false;
4 boolean pulsado = false;
5 boolean ultimoEstadoPulsado = false;
6 void setup() {
7     pinMode(PinBoton, INPUT);  // Configurar el pin del botón como entrada
8     pinMode(PinLed, OUTPUT);   // Configurar el pin del LED como salida
9 }
10 void loop() {
11     pulsado = digitalRead(PinBoton); // Leer el estado actual del botón
12
13     if (pulsado && !ultimoEstadoPulsado) {
14         encendido = !encendido; // Invertir el estado del LED
15         if (encendido) {
16             digitalWrite(PinLed, HIGH); // Encender el LED
17         } else {
18             digitalWrite(PinLed, LOW);  // Apagar el LED
19         }
20     }
21     ultimoEstadoPulsado = pulsado; // Actualizar el estado anterior del botón
22 }
```

¿QUÉ HAREMOS HOY?

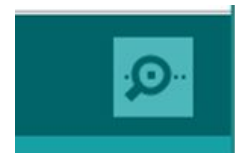
- ❑ ¿Qué es software libre?
- ❑ ARDUINO
- ❑ PRÁCTICA 1: BLINK
- ❑ PRACTICA 2: BLINK CON LED EN PIN 13
- ❑ PRÁCTICA 3: Led+botón
- ❑ PRÁCTICA 4: Led+botón+Puerto Serie
- ❑ PRACTICA 5: LED+ BOTON + SERVO

Puerto Serie Arduino



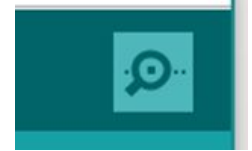
```
void setup() {  
    // Iniciar la comunicación serie a 9600 baudios  
    Serial.begin(9600);  
}  
  
void loop() {  
    Muestra el mensaje 'Hola Mundo' por el puerto serie cada  
        segundo  
  
    // Esperar un segundo  
    delay(1000);  
}
```

Puerto Serie Arduino



```
void setup() {  
    // Iniciar la comunicación serie a 9600 baudios  
    Serial.begin(9600);  
}  
  
void loop() {  
    // Enviar el mensaje "Hola Mundo" al monitor serie  
    Serial.println("Hola Mundo");  
    Serial.print("Hola Mundo");  
  
    // Esperar un segundo  
    delay(1000);  
}
```

Puerto Serie Arduino



```
void setup() {  
  // Iniciar la comunicación serie a 9600 baudios  
  Serial.begin(9600);  
}
```

vo Ejercicio: Muestra “led apagado” o “led encendido”
dependiendo si el led está apagado o encendido

```
  // Esperar un segundo  
  delay(1000);  
}
```

```
1 int PinBoton = 2; // Pin del botón
2 int PinLed = 13; // Pin del LED
3 boolean encendido = false;
4 boolean pulsado = false;
5 boolean ultimoEstadoPulsado = false;
6 void setup() {
7   pinMode(PinBoton, INPUT); // Configurar el pin del botón como entrada
8   pinMode(PinLed, OUTPUT); // Configurar el pin del LED como salida
9 }
10 void loop() {
11   pulsado = digitalRead(PinBoton); // Leer el estado actual del botón
12
13   if (pulsado && !ultimoEstadoPulsado) {
14     encendido = !encendido; // Invertir el estado del LED
15     if (encendido) {
16       digitalWrite(PinLed, HIGH); // Encender el LED
17     } else {
18       digitalWrite(PinLed, LOW); // Apagar el LED
19     }
20   }
21   ultimoEstadoPulsado = pulsado; // Actualizar el estado anterior del botón
22 }
```

Serial.println("Led encendido")

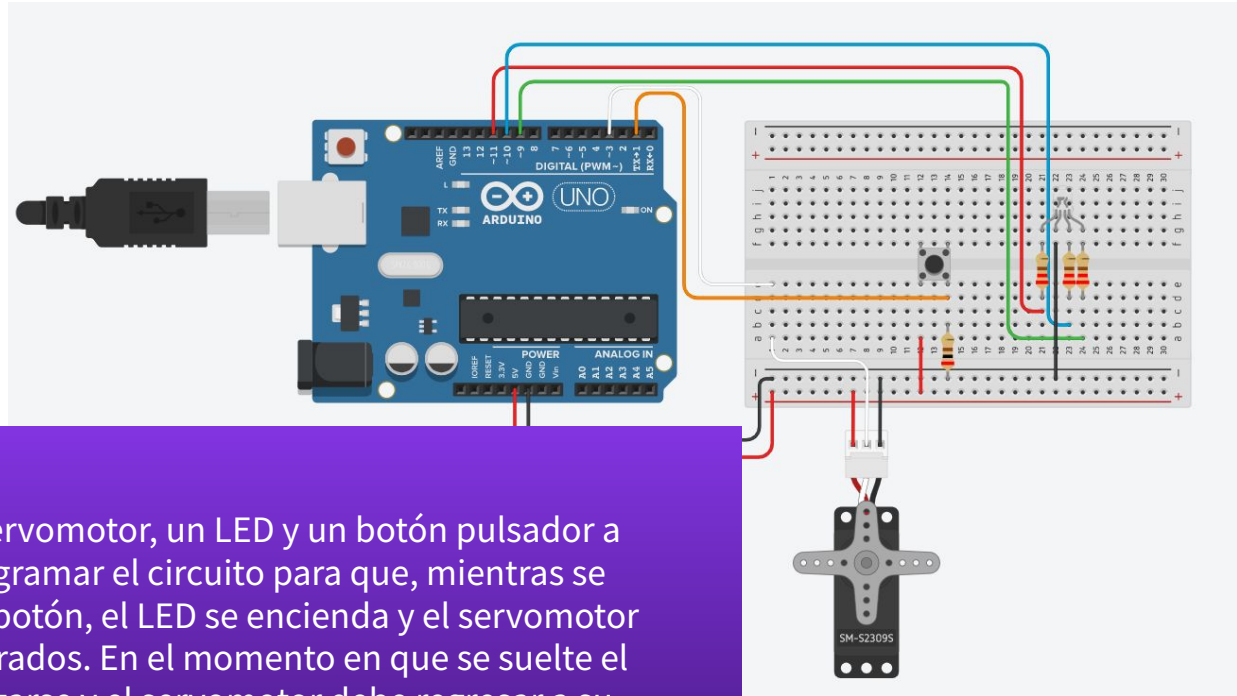
Serial.println("Led apagado")

¿QUÉ HAREMOS HOY?

- ❑ ¿Qué es software libre?
- ❑ ARDUINO
- ❑ PRÁCTICA 1: BLINK
- ❑ PRACTICA 2: BLINK CON LED EN PIN 13
- ❑ PRÁCTICA 3: Led+botón
- ❑ PRÁCTICA 4: Led+botón+Puerto Serie
- ❑ PRACTICA 5: LED+ BOTON + SERVO

SERVO+LED+BOTON

Servo



Ejercicio: Conectar un servomotor, un LED y un botón pulsador a una placa Arduino. Programar el circuito para que, mientras se mantenga presionado el botón, el LED se encienda y el servomotor gire a la posición de 90 grados. En el momento en que se suelte el botón, el LED debe apagarse y el servomotor debe regresar a su posición inicial de 0 grados.

```
#include <Servo.h>

const int pinBoton = 2;
const int pinLed = 13;
const int pinServo = 9;

Servo miServo;

void setup() {
    miServo.attach(pinServo);
    pinMode(pinBoton, INPUT);
    pinMode(pinLed, OUTPUT);
}

void loop() {
    int estadoBoton = digitalRead(pinBoton);

    if (estadoBoton == HIGH) { // Si se presiona el botón
        digitalWrite(pinLed, HIGH); // Enciende el LED
        miServo.write(90);           // Mueve el servo a 90 grados
    } else {
        digitalWrite(pinLed, LOW);  // Apaga el LED
        miServo.write(0);           // Mueve el servo a 0 grados
    }
}
```