Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

Introducción a la programación

Arduino y C++

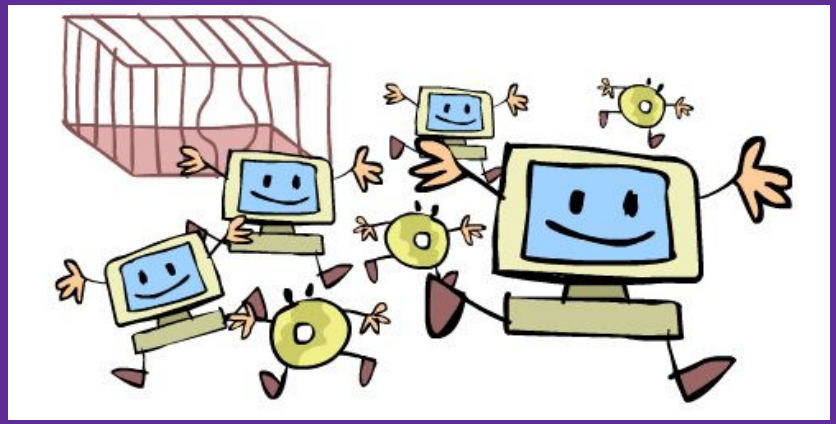
¿QUÉ HAREMOS HOY?

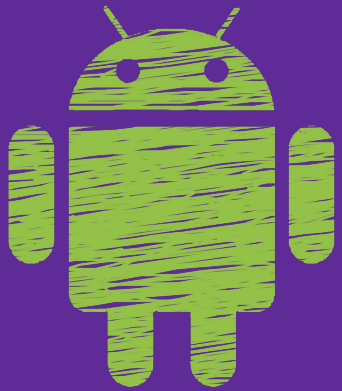
- ¿Qué es software libre?
- ARDUINO
- PRÁCTICA 1: BLINK
- PRÁCTICA 2: Led+botón
- PRÁCTICA 3: Bucles
- PRÁCTICA 4: Puerto Serie

Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

¿SABÉIS QUE ES EL SOFTWARE LIBRE?

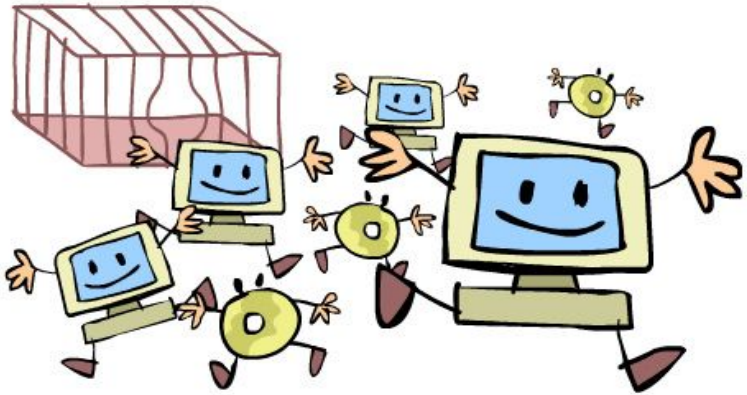




Software libre

¿Qué es el software libre

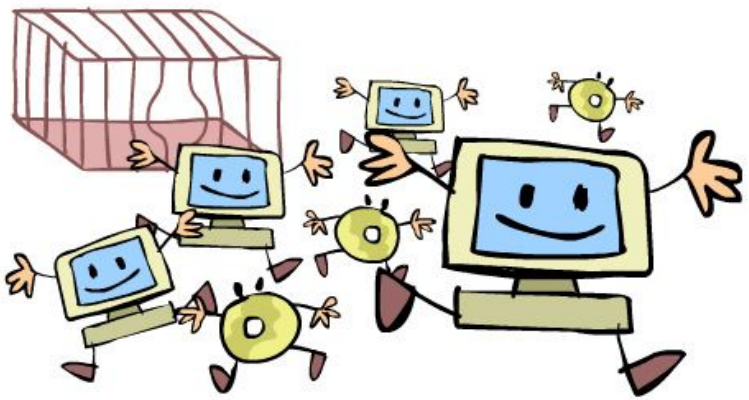
- Software que le da al usuario la libertad de:
  - Compartirlo.
  - Estudiarlo.
  - Modificarlo.
- Se llama software libre porque el usuario es libre.



Software Libre

Las 4 libertades:

- Libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a las propias necesidades.
- Libertad de distribuir copias del programa, con lo cual se puede ayudar a otros usuarios.
- Libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.
- Libertad para usar el software para cualquier propósito.

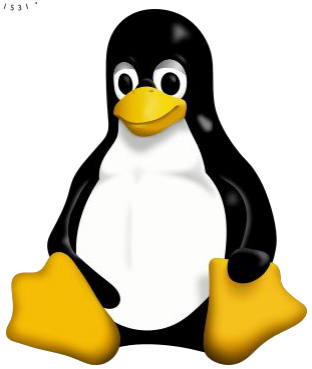


Oficina de Software Libre Universidad de Granada

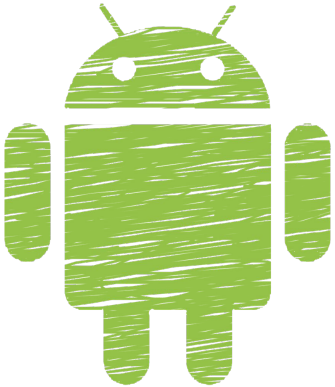
¿QUÉ SOFTWARE LIBRES EXISTEN?

UNIVERSIDAD DE GRANADA

LibreOffice
The Document Foundation



¡AQUÍ OS DEJO UNOS EJEMPLOS!



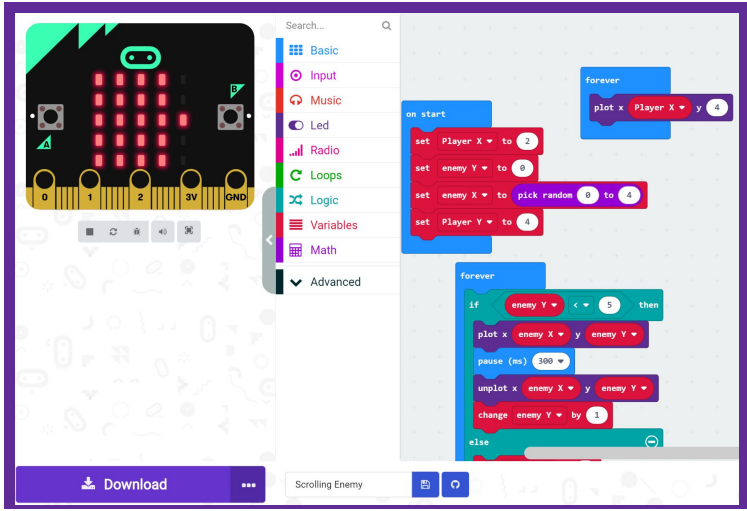
MySQL

blender

Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

¿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
- PRÁCTICA 2: Led+botón
- PRÁCTICA 3: Bucles
- PRÁCTICA 4: Puerto Serie

Conceptos básicos Arduino

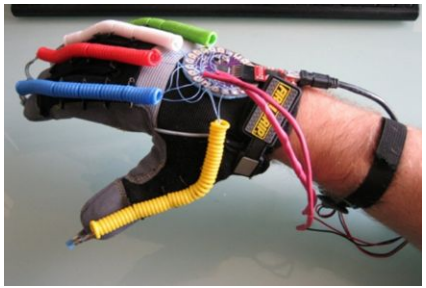
¿Qué es Arduino?

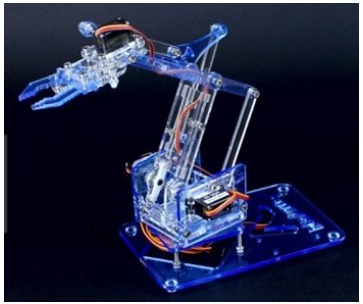
- ❏ Arduino es un micro controlador.
- ❏ Es libre (no es único).
- ❏ Flexible y fácil de manejar.
- ❏ Puertos de entrada y salida analógicos y digitales.
- ❏ Puertos de comunicación.
- ❏ Modular y ampliable.
- ❏ Software: C++, programación por bloques (scratch for arduino)

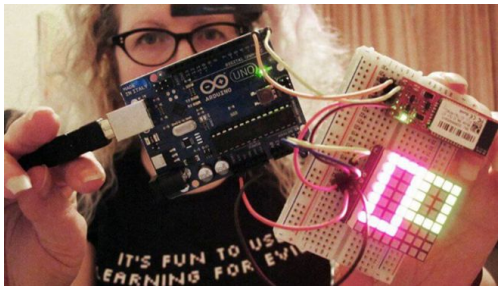
Oficina de Software Libre Universidad de Granada

¿QUÉ PUEDO HACER CON ARDUINO?

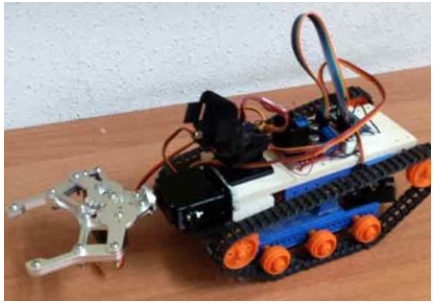
UNIVERSIDAD DE GRANADA







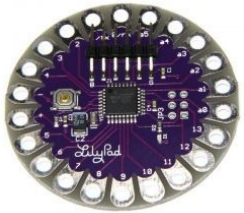
¡TODO LO QUE SE TE OCURRA!



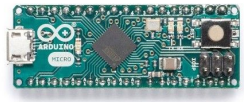
Tipos de Arduino

Arduino Uno

Arduino Nano



Lilypad



Arduino Micro

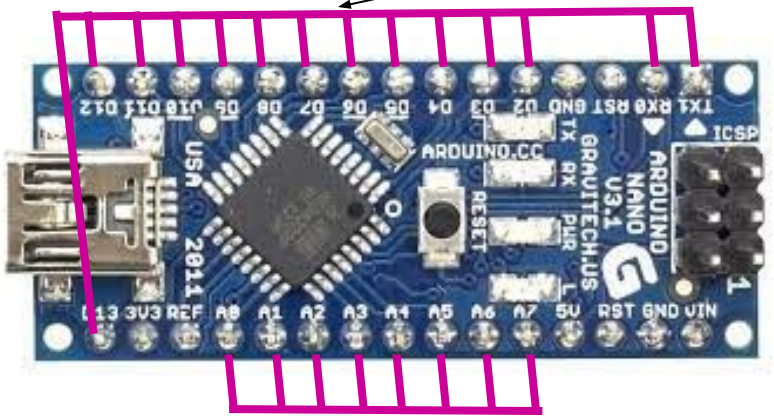
Node MCU

Arduino Mega

Arduino Leonardo

… ¡y muchos más!

Arduino 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

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 Mega

Pines Digitales

D0 D5 D10

D1 D6 …

D2 D7 D52

D3 D8 D53

D4 D9

Pines Analógicos

A0, A1, A2, A3, A4, A5,...,A13

NODEMCU

Pines Digitales

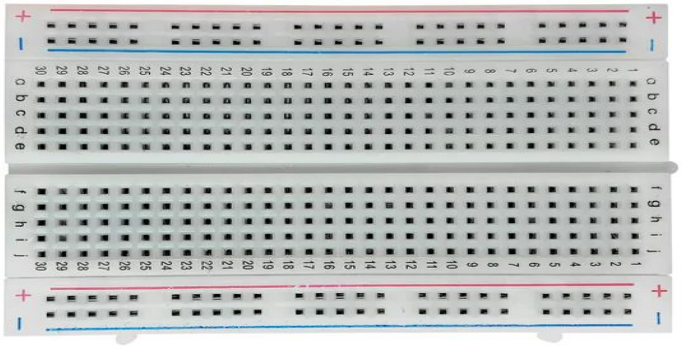
D0
D1
D2
D3
D4
D5
D6
D7
D8

Pin Analógico

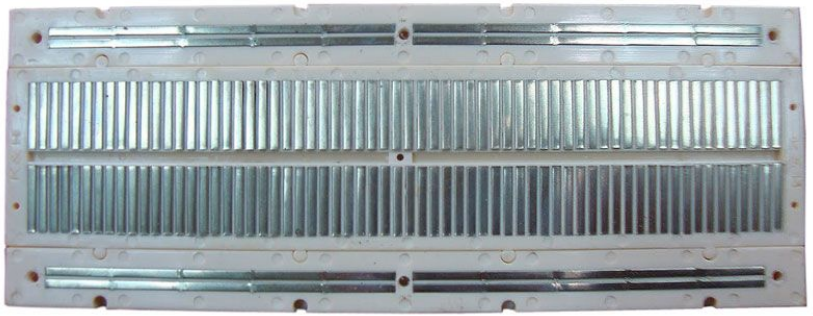
A0

Protoboard

Placa Protoboard



Placa Protoboard por dentro

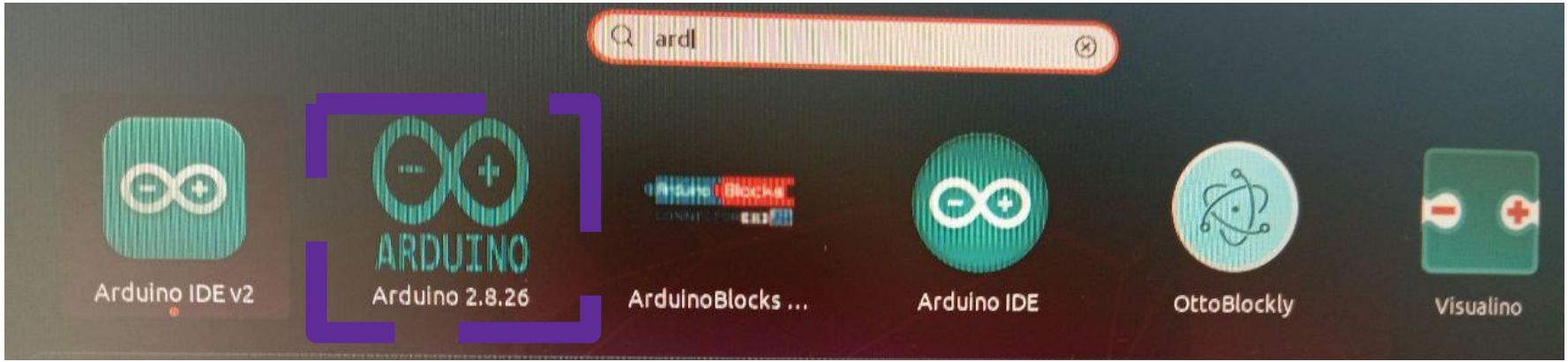


¿Cómo funciona Arduino?

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.

¿QUÉ IDE VAMOS A USAR?



¿Cómo funciona Arduino?

Hola mundo!

```
void setup() {
```

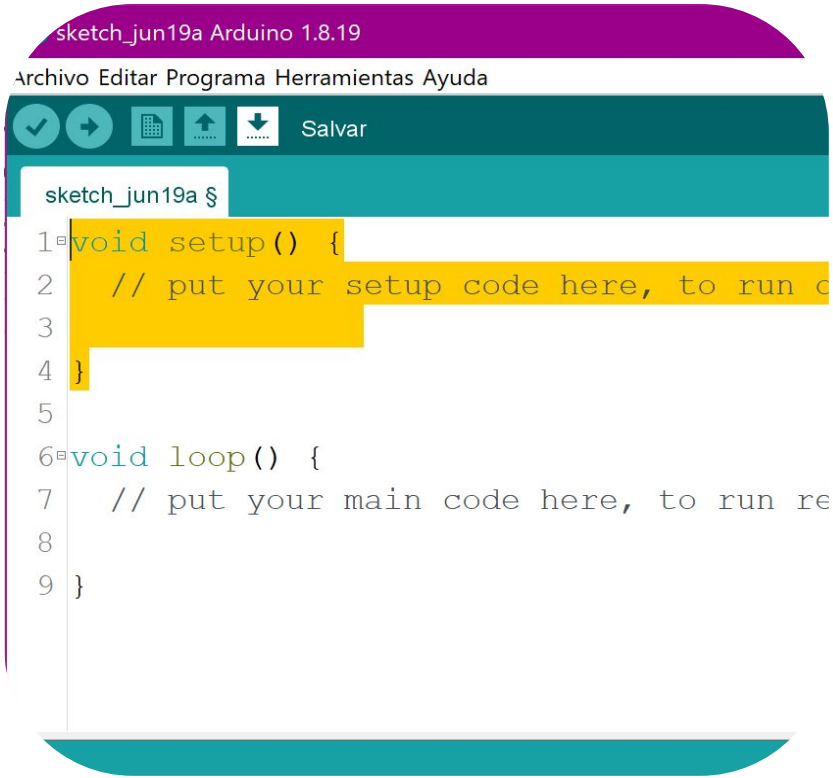
En setup tenemos un listado de las cosas conectadas, como una lista de la compra.

```
}

void loop() {
```

Donde se realizarán las acciones

```
}
```



Entradas y salidas digitales

Entradas/salidas digitales

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

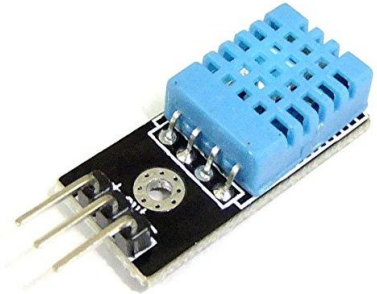
  setup(){AQUI DENTRO}

- Los pines que usaremos como entrada:

  pinMode(numero_de_pin, INPUT);

- Los que usaremos como salida:

  pinMode(numero_de_pin, OUTPUT);



Entradas y salidas analógicas

- **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);

¿QUÉ HAREMOS HOY?

- ¿Qué es software libre?
- ARDUINO
- PRÁCTICA 1: BLINK
- PRÁCTICA 2: Led+botón
- PRÁCTICA 3: Bucles
- PRÁCTICA 4: Puerto Serie

Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

Programación C++

Cuestiones básicas

¿Qué es un algoritmo?

- Secuencia ordenada de instrucciones que resuelve un problema concreto.

Lenguajes de programación:

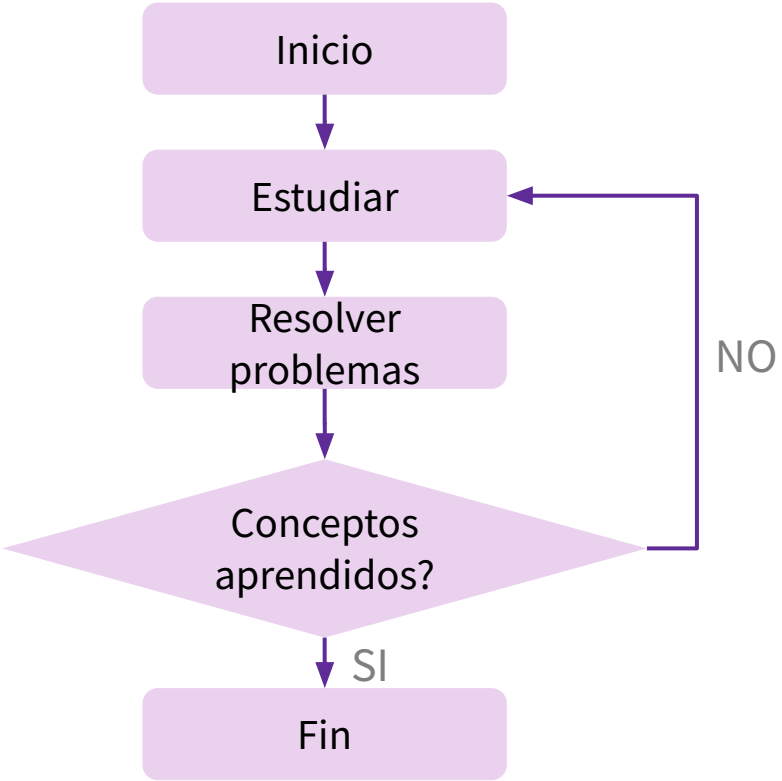
- Bajo nivel: Ensamblador.
- Alto nivel: C++, C, Java, python …

¿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.

Cuestiones básicas

¿Qué es un algoritmo?



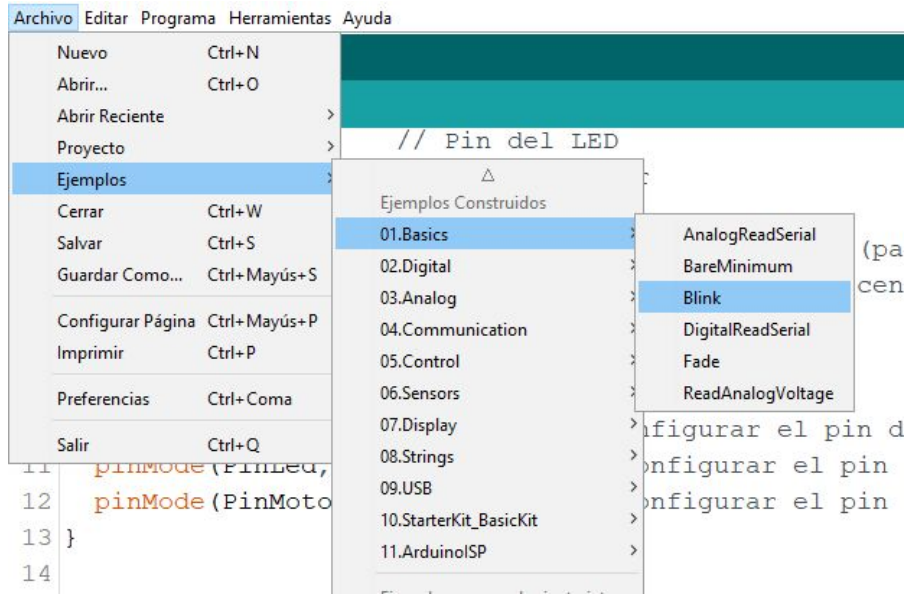
Los algoritmos son la base de la resolución de problemas y los utilizamos todos los días

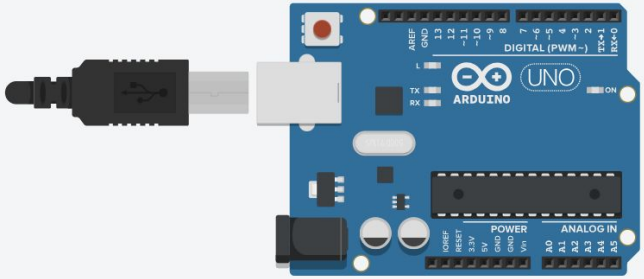
¿Qué es el IDE de arduino?

- 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.

Pensar y Esperar

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





SOLUCIÓN


```
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

Declaración y asignación de variables

Tipos de datos

- Numéricos:
  - Int lado1;
  - Double lado2;
  - lado1 = 3;
  - lado2 = 3,5;
- Lógicos
  - Bool midato;

    (sólo toma valor verdadero o falso)
  - midato = true;
- Complejos:
  - int vector[2];
  - Vector[0] = 3;
  - Vector[1] = 2;

Operadores

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

Condicionales

¿Qué son los condicionales en español?

En programación, los condicionales son como las decisiones.

por ejemplo:

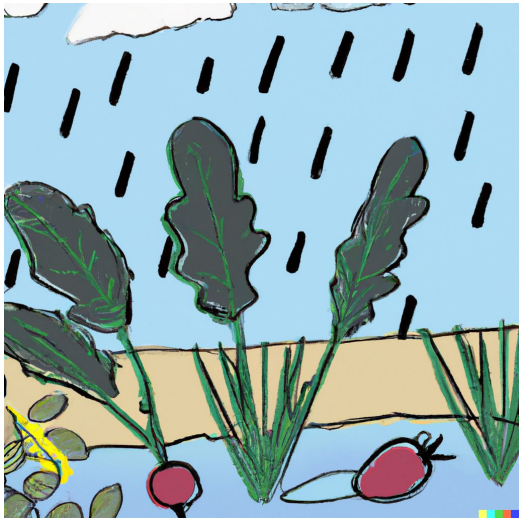
```
si (tienes debe) {
    come una fruta;
} sino si (tienes sed) {
    bebe agua;
} sino {
    vete a jugar y diviértete;
}
```

Condicionales

- 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"
}
```

Estructuras Condicionales

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
- PRÁCTICA 2: Led+botón
- PRÁCTICA 3: Bucles
- PRÁCTICA 4: Puerto Serie

Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

Vamos a...

¡PRACTICAR!

Estructuras Condicionales

Condicional simple

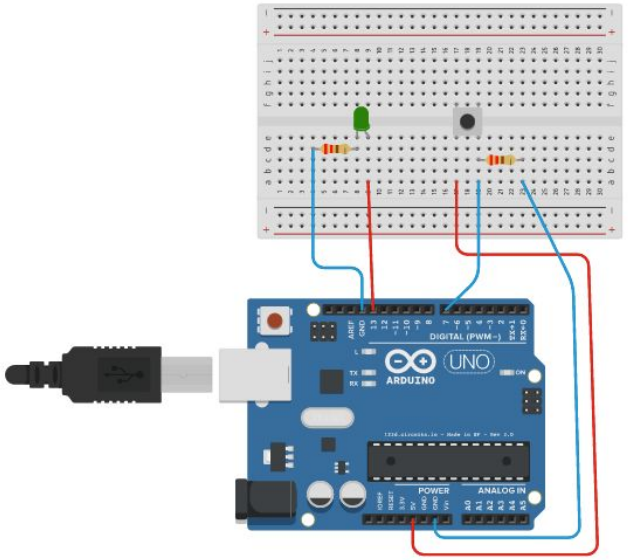
```
If (condición){
    sentencias;
}
```

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



Solución

```
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

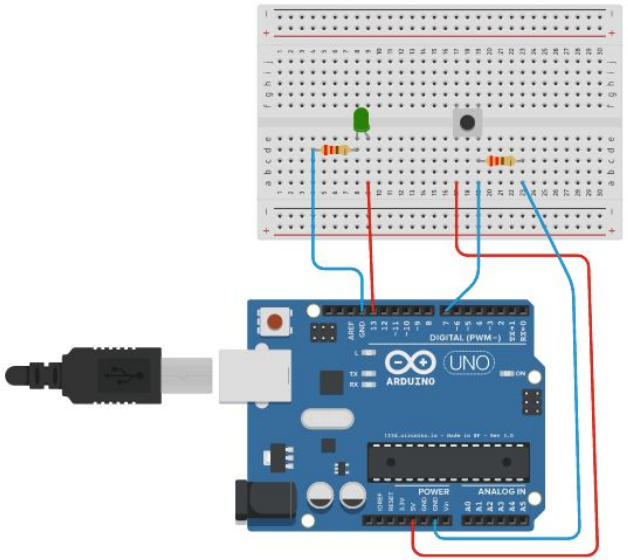
pulsado = Leer estado del botón

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

Estructuras Condicionales

Condicional compuesto

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



Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

Vamos a...

¡PRACTICAR!

Estructuras Condicionales

Condicional compuesto

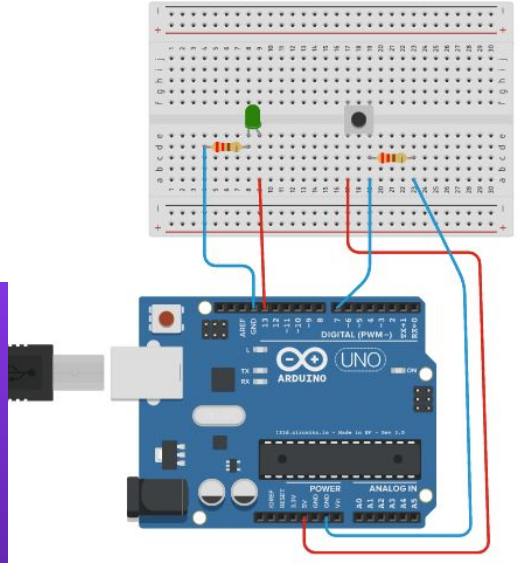
```
If (condición && condición){
    sentencias;
}
```

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



Solución

```
int PinBoton = 2;  // Pin del botón
int PinLed = 13;    // Pin del LED
boolean encendido = false;
boolean pulsado = false;
boolean ultimoEstadoPulsado = false;
void setup() {
  pinMode(PinBoton, INPUT);    // Configurar el pin del botón como entrada
  pinMode(PinLed, OUTPUT);     // Configurar el pin del LED como salida
}
void loop() {
  pulsado = digitalRead(PinBoton);  // Leer el estado actual del botón

  if (pulsado && !ultimoEstadoPulsado) {
    encendido = !encendido;  // Invertir el estado del LED
    if (encendido) {
      digitalWrite(PinLed, HIGH);  // Encender el LED
    } else {
      digitalWrite(PinLed, LOW);   // Apagar el LED
    }
  }
  ultimoEstadoPulsado = pulsado;  // Actualizar el estado anterior del botón
}
```

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

¿QUÉ HAREMOS HOY?

- ¿Qué es software libre?
- ARDUINO
- PRÁCTICA 1: BLINK
- PRÁCTICA 2: Led+botón
- PRÁCTICA 3: Bucles
- PRÁCTICA 4: Puerto Serie

Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

Vamos a...

¡PRACTICAR!

Bucles o ciclos

Bucles

```
Loop(){
    Lo que se va a repetir siempre
}

for (inicio; fin; incremento){
    Lo que se va a repetir desde inicio a fin
}

while(condición){
    Lo que quiero que haga mientras se cumpla la condición
}
```

Bucles

Encender y apagar un LED 10 veces utilizando el bucle FOR, es decir que esté parpadeando sólo 10 veces

```
void loop() {
    for (int i = 0; i < 10; i++) {
     digitalWrite(ledPin, HIGH); // Encender el LED
     delay(500);                 // Esperar 500 ms
     digitalWrite(ledPin, LOW);  // Apagar el LED
     delay(500);                 // Esperar 500 ms
    }
  // Esperar 2 segundos antes de empezar de nuevo
 delay(2000);
 }
```

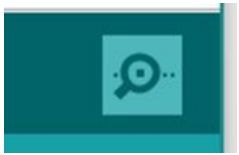
¿QUÉ HAREMOS HOY?

- ¿Qué es software libre?
- ARDUINO
- PRÁCTICA 1: BLINK
- PRÁCTICA 2: Led+botón
- PRÁCTICA 3: Bucles
- PRÁCTICA 4: Puerto Serie

Pensar y Esperar

En Arduino:

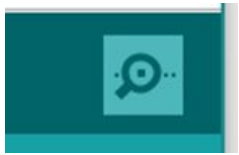
- **Pensar** es escribir en pantalla un mensaje, por lo que tenemos que iniciar en el **setup()** el terminal serie de arduino
  - **Serial.begin(9600);**



- Y en el **loop()** cuando queramos imprimir algún valor por pantalla:
  - **Serial.print("lo que queramos que escriba por pantalla");**
- Para **esperar** usaremos: **delay(tiempo_a_esperar);**

Puerto Serie Arduino

Muestra el mensaje ‘Hola Mundo’ por el puerto serie cada segundo



```
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);
}
```

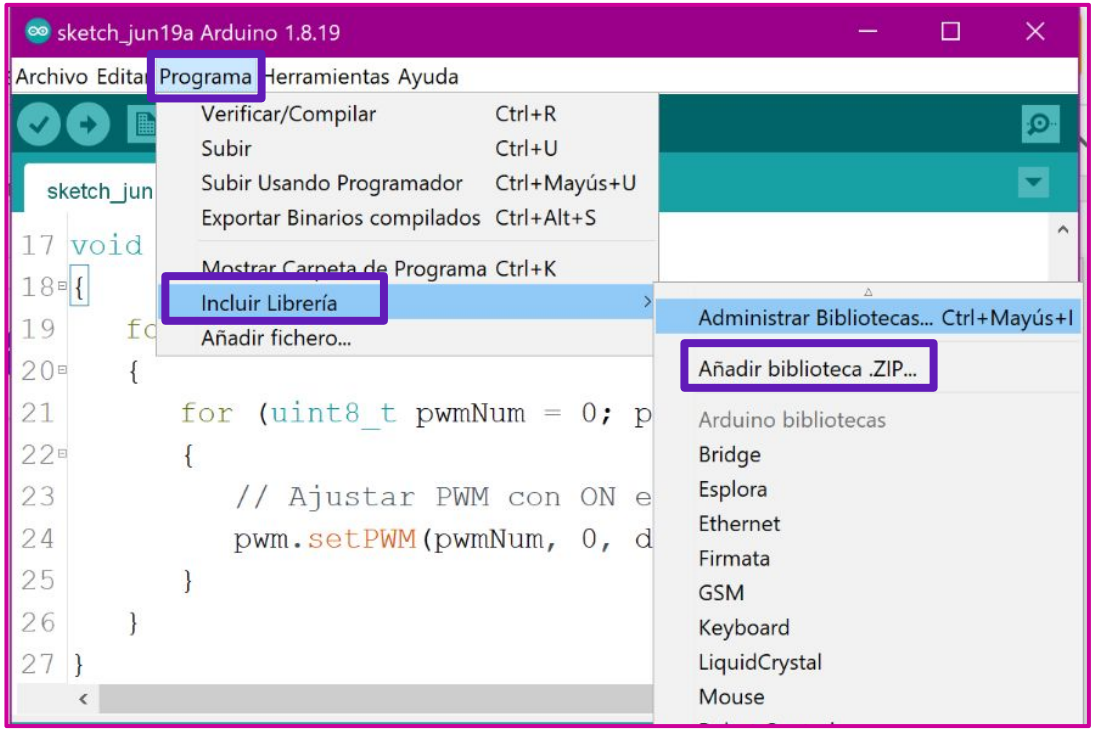
Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

Vamos a...

¡PRACTICAR!

Cargar librería



Repetir este paso para cada una de las librerías.

Añadimos las **librerías** necesarias.

- BatReader.zip
- EnableInterrupt.zip
- LedMatrix.zip
- MaxMatrix.zip
- Oscillator.zip
- Otto.zip
- OttoSerialCommand.zip
- US.zip

Oficina de Software Libre Universidad de Granada

UNIVERSIDAD DE GRANADA

Polímetro

POLÍMETRO

- El polímetro o multímetro es un instrumento que nos permite comprobar el funcionamiento de los circuitos eléctricos.
- Incluye herramientas para medir las tres magnitudes fundamentales de la electricidad: tensión, corriente y resistencia
- Permite descartar problemas de conexión: verificar si las conexiones son correctas.
- Nosotras lo vamos a utilizar para dos cosas:
  - Medir el valor de las resistencias de nuestro circuito
  - Comprobar conexiones

¿SABÍAS QUE...?

Cuando algo no funciona, el primer paso siempre es comprobar que todo esté bien conectado. El polímetro es una herramienta fundamental para comprobar conexiones.

POLÍMETRO

Medir Valor de resistencias

- Ajustar la escala según el valor de la resistencia

POLÍMETRO

Comprobar Conexiones

- Suena cuando hay conexión