

# INTRODUCCIÓN A ARDUINO

## CONTROLANDO ARDUINO DESDE DISPOSITIVOS MÓVILES

Jose Luis Núñez  
José Pujol

CEP Sevilla 2019



# ÍNDICE



1. ¿Qué es Arduino?
2. Conceptos básicos sobre Arduino
3. Practicando
4. Hola Mundo



# ¿QUÉ ES ARDUINO?



- Plataforma de prototipado de electrónica abierta
- Se basa en software y hardware libre
- Es flexible y de fácil uso



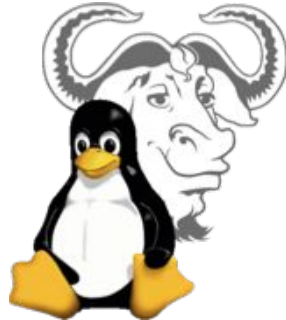
# Software libre

---

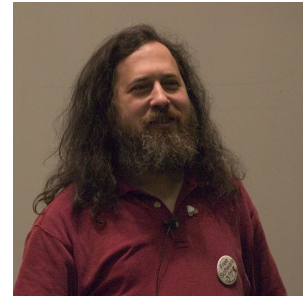
Aquel software que nos da libertad para:

1. Ejecutarlo con cualquier propósito
2. Estudiar cómo está hecho y modificarlo
3. Redistribuir copias
4. Distribuir copias modificadas à Licencia GPL

GNU Linux

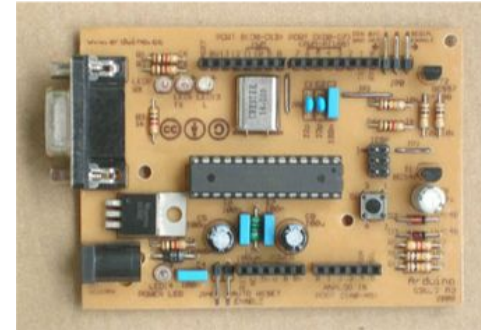
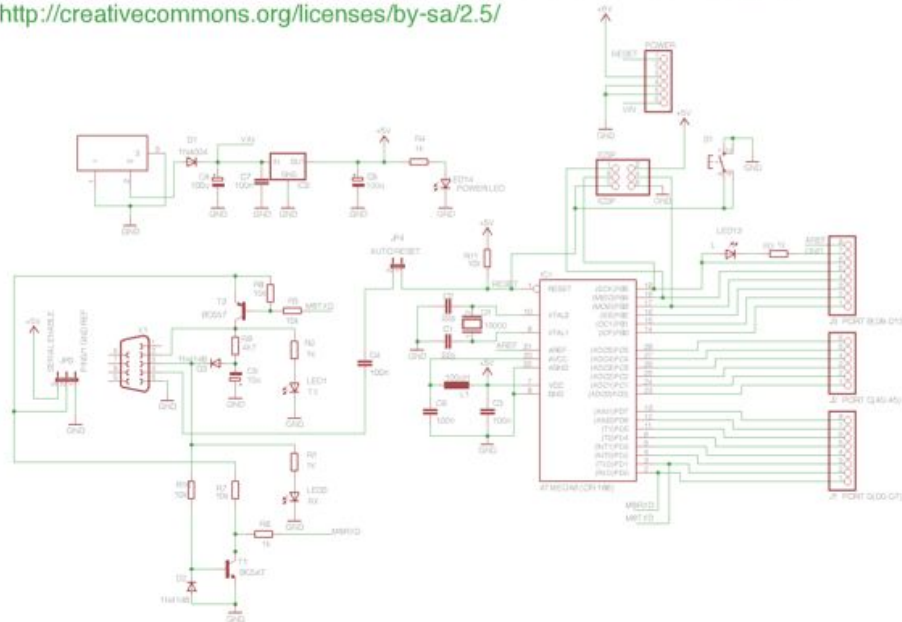


Richard Stallman



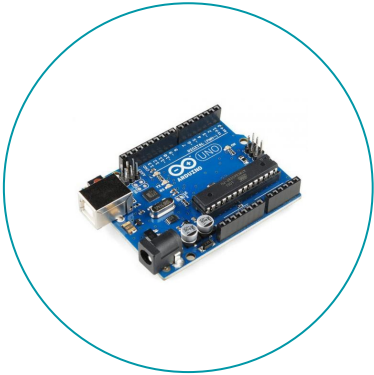
# Hardware libre

Released under the Creative Commons Attribution Share-Alike 2.5 License  
<http://creativecommons.org/licenses/by-sa/2.5/>

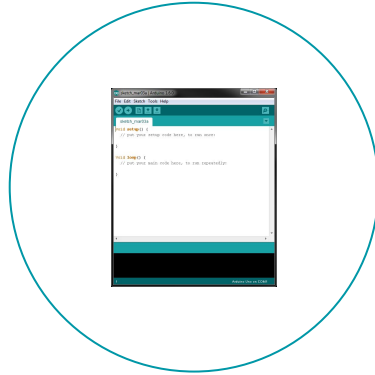


# Ecosistema de Arduino

Hardware



Entorno de desarrollo



Lenguaje Programación

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

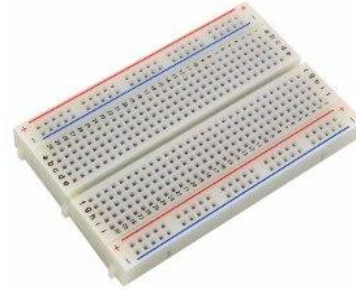
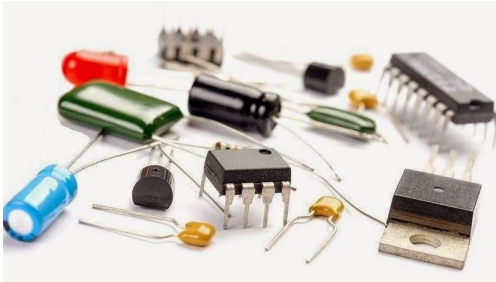
Soporte online



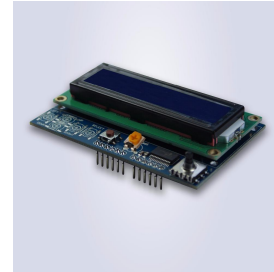
# Hardware



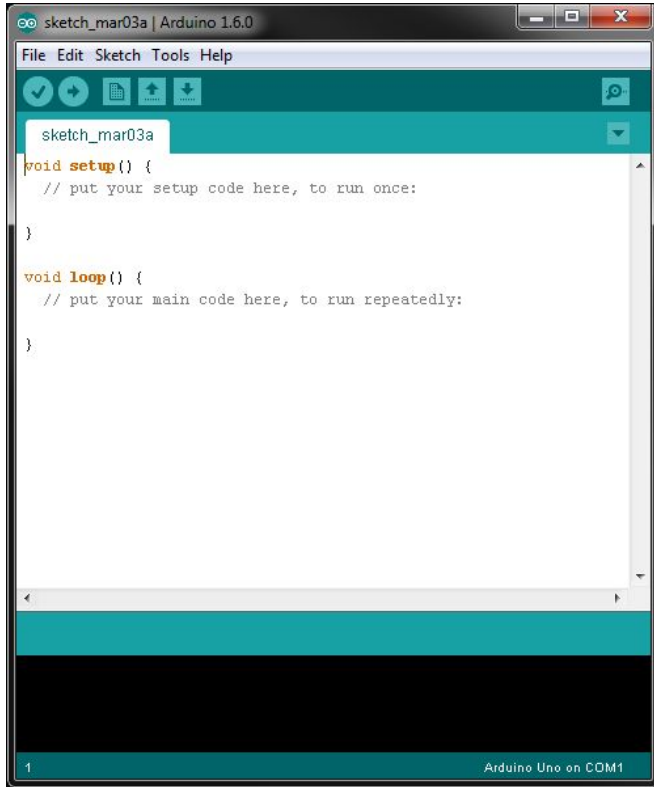
Componentes



Módulos



# IDE



- Es software libre
- Proviene del entorno de Processing

## Funcionalidades:

- Escribir código
- Depurarlo
- Compilarlo y cargarlo en la placa
- Comunicarnos con la placa
- Configurar placa y puerto



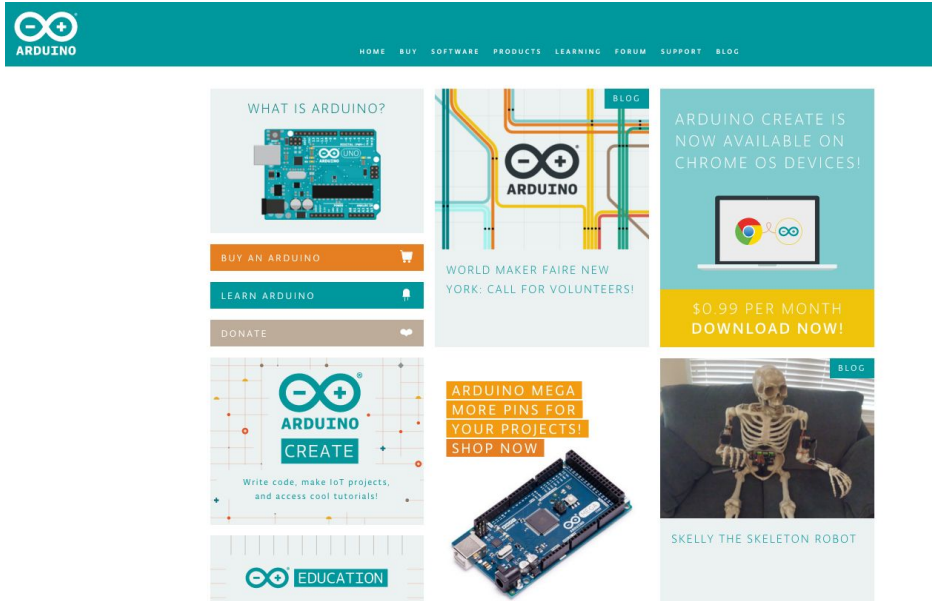
# Software

- Lenguaje con el que se programan las placas de entradas y salidas.
- Basado en C++, programación orientada a objetos con dos estructuras fundamentales: setup y loop.
- Gran cantidad de librerías que aumentan las funcionalidades y permiten controlar los dispositivos de forma sencilla.

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}
```

# Soporte en Red



[arduino.cc](https://arduino.cc)

- Software: IDE
- Información Hardware
- Reference: Lenguaje de programación
- Tutoriales
- Foro
- Proyectos

# CONCEPTOS BÁSICOS SOBRE ARDUINO

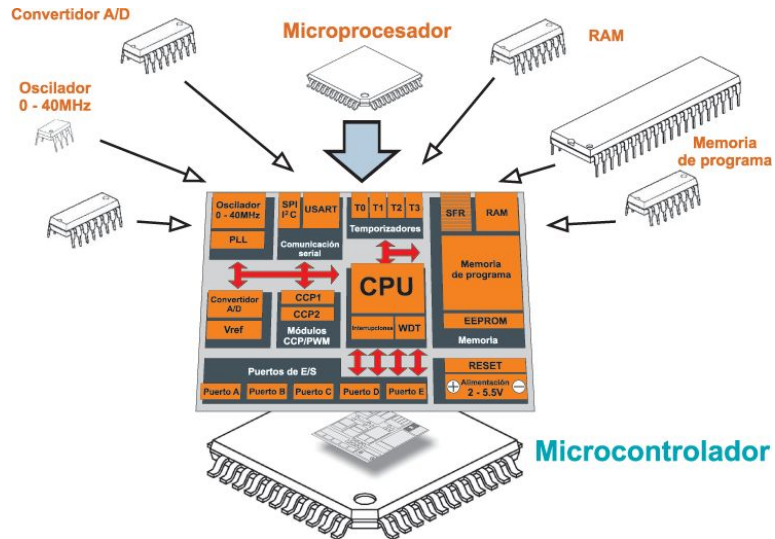


- Microcontrolador
- Entradas y Salidas
- Señales digitales y analógicas
- Conexiones Arduino UNO



# Microcontrolador

- Circuito integrado y programable
- Realiza operaciones básicas a gran velocidad



# Microcontrolador

---



Lectura de entradas → procesamiento de la información y decisión de la actuación → Actuación sobre las salidas

# Entradas y Salidas

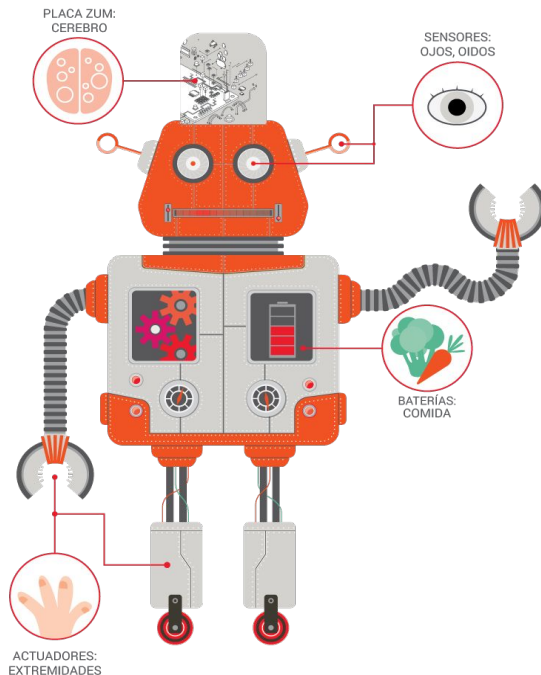
## Entradas: información

### Cuerpo humano

- Sentidos

### Arduino

- Pulsador
- Sensores



## Salidas: actuación

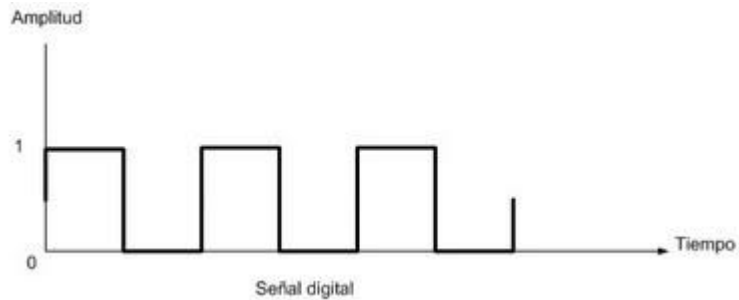
### Cuerpo humano

- Brazos
- Piernas
- Voz

### Arduino

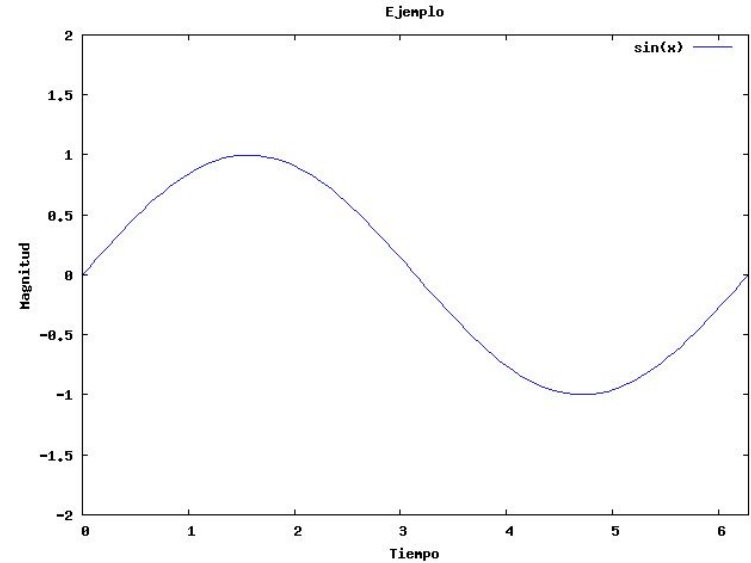
- Motores
- Leds
- Altavoz

# Señales Digitales y Analógicas

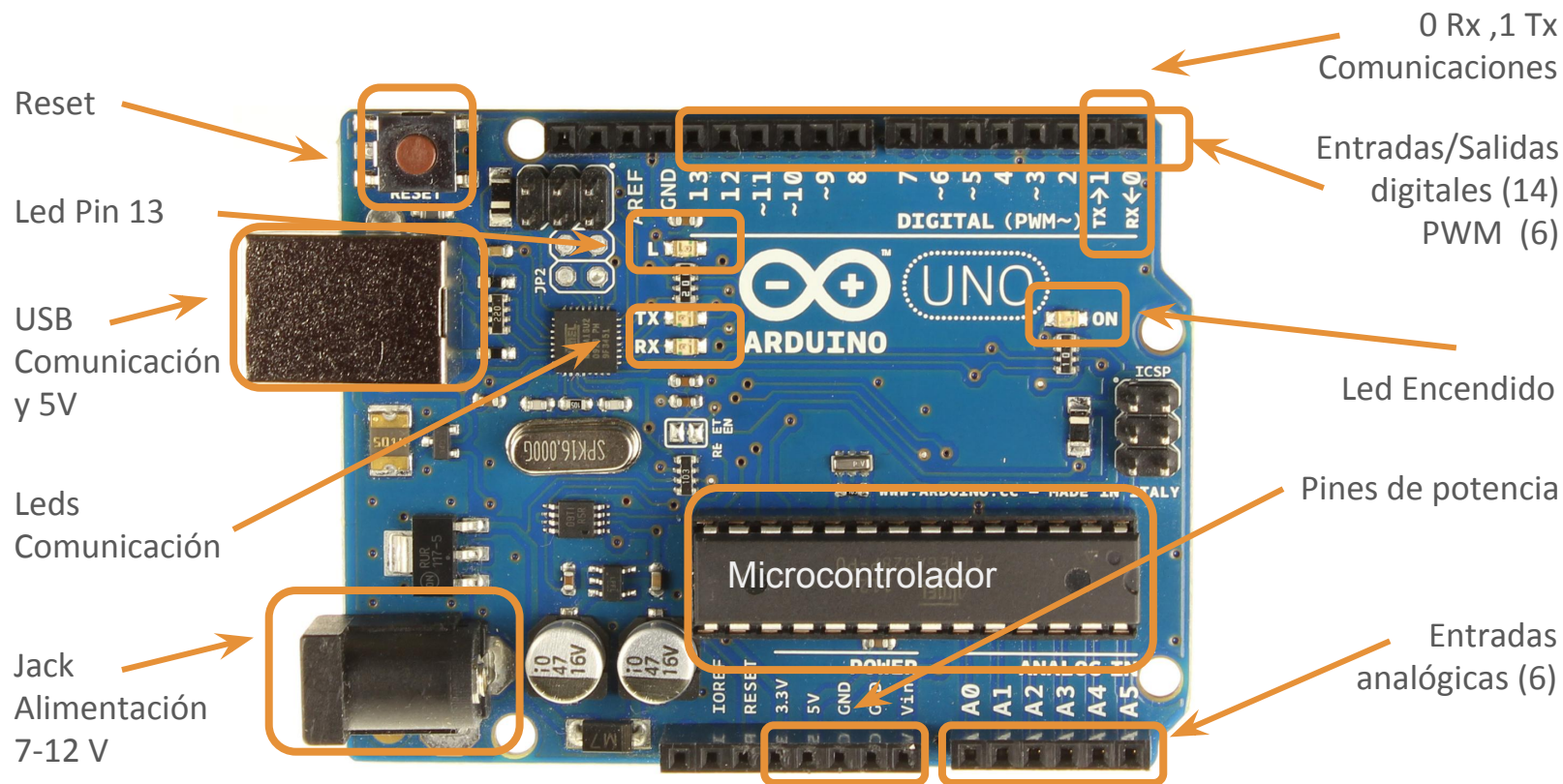


0: 0v: Falso

1: 5v: Verdadero



# Arduino UNO: Conexiones





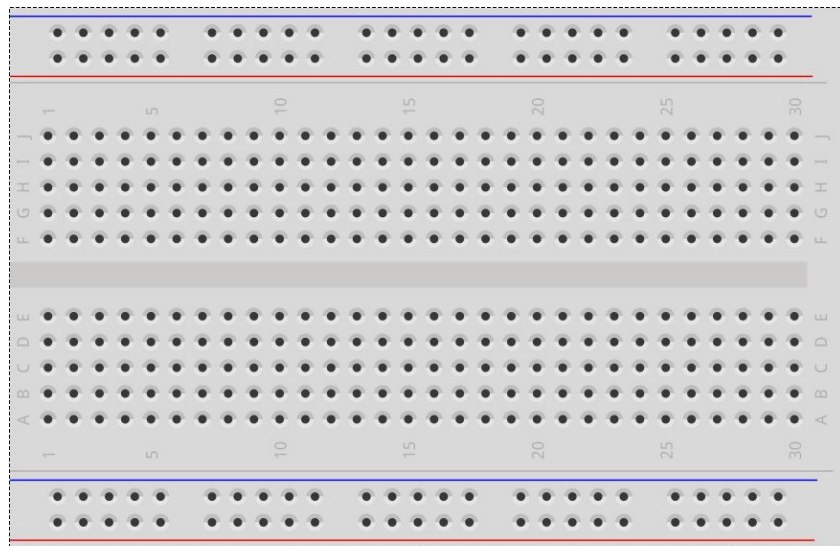
# PRACTICANDO



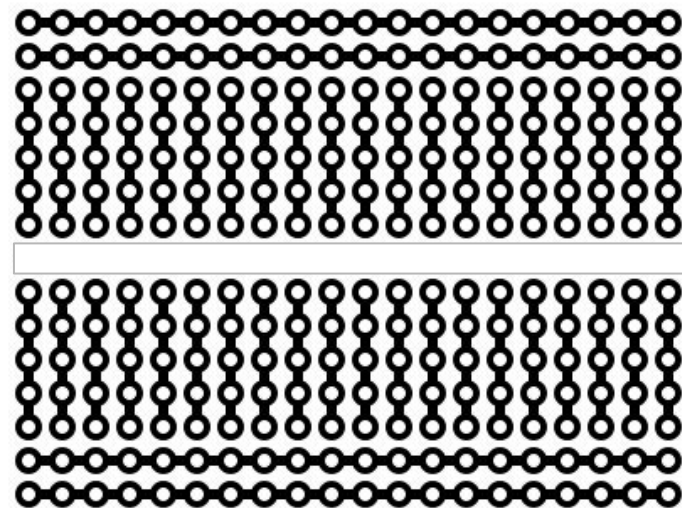
- Placa de prototipos
- Cables de conexión
- Resistencias



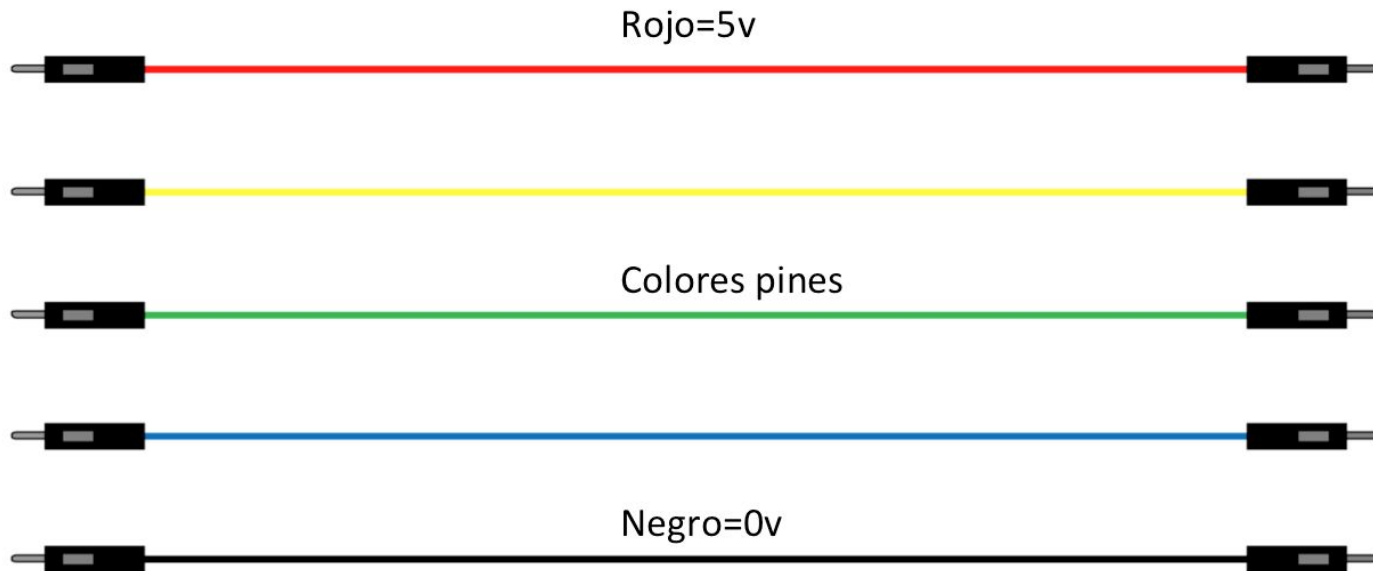
# Placa de prototipos



fritzing



# Cables dupont



# Resistencias

---



220  $\Omega$



1 K $\Omega$



10 K $\Omega$

# HOLA MUNDO



- Finalidad
- Hardware
- Software
- IDE
- Propuesta de actividad



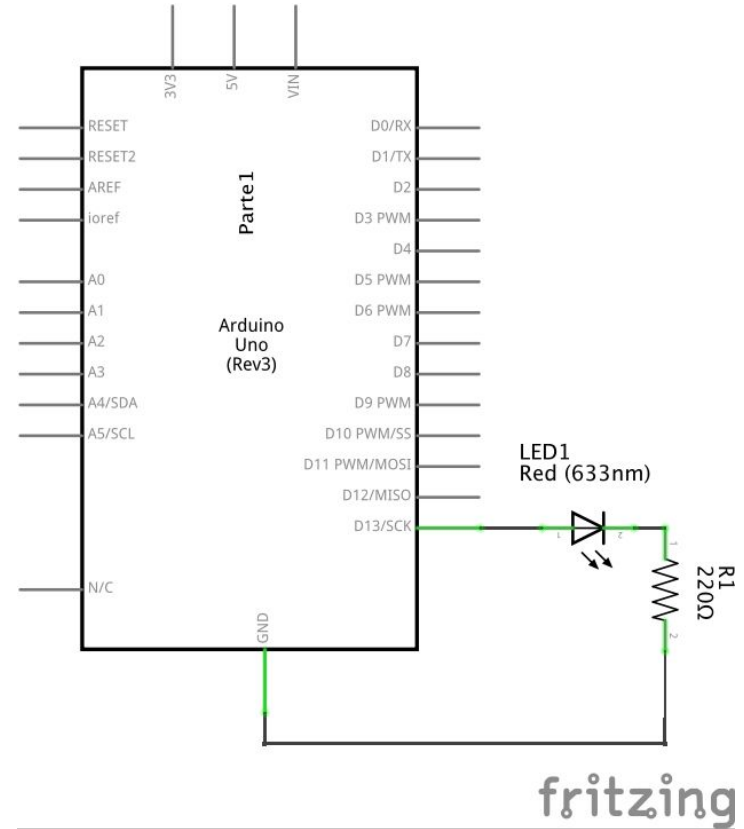
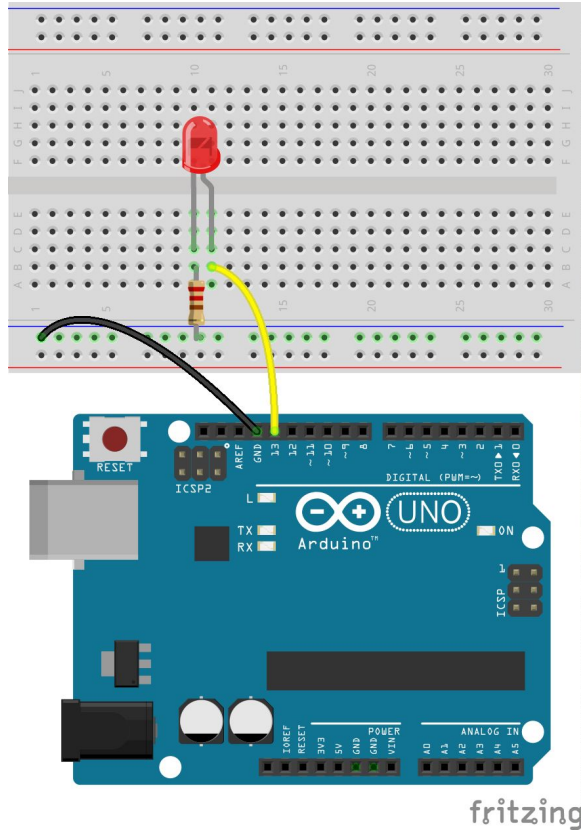
# Finalidad

---

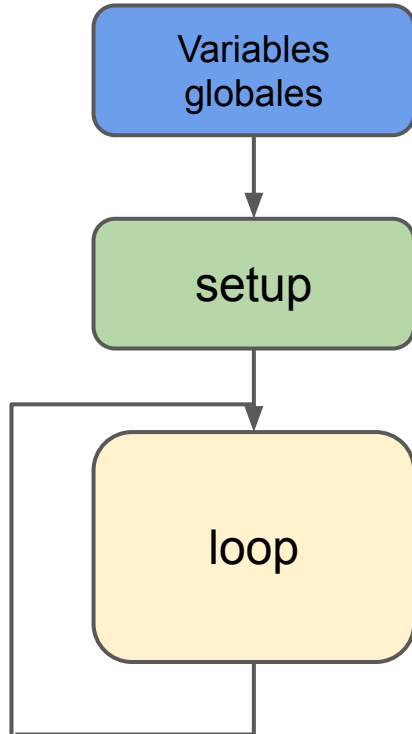
Encender y apagar un led de forma repetitiva



# Hardware



# Software

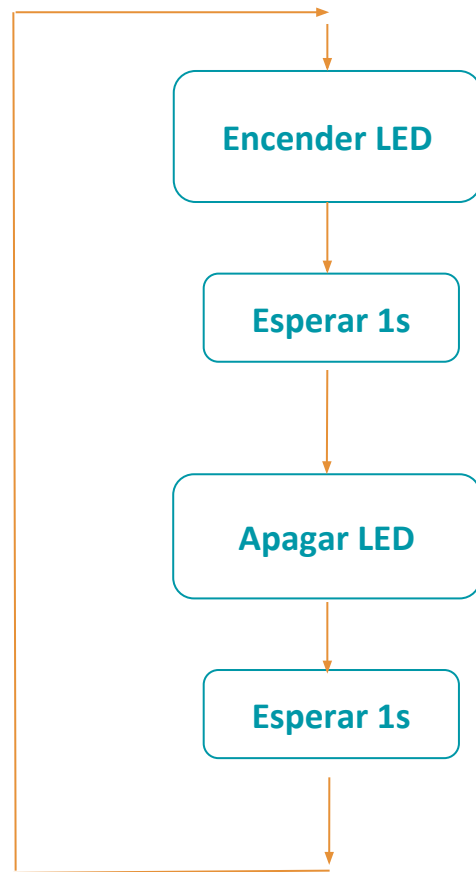


## El código de Arduino se divide en tres partes:

- La primera parte se encarga de las Variables Globales.
- La segunda, el setup, posee la característica de que se ejecuta una sola vez. Se utilizará para realizar la configuración del sistema.
- La tercera parte es el loop, que se ejecuta de forma cíclica y donde colocaremos el cuerpo de nuestro programa.



# Diagrama de flujo



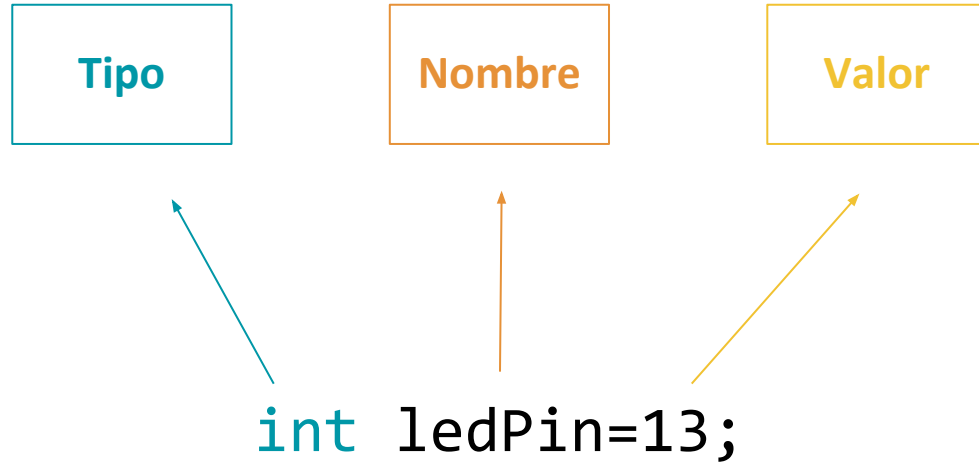
# Código de Arduino

```
8 // declaracion de variables
9 int ledPin=13; // LED asociado al pin 13
10
11 // la funcion setup se ejecuta una sola vez
12 void setup() {
13     pinMode(ledPin, OUTPUT); // inicializa el pin 13 como una salida digital
14 }
15
16 // la funcion loop se ejecuta repetidamente de forma infinita
17 void loop() {
18     digitalWrite(ledPin, HIGH); // establece 5v en el pin del LED
19     delay(1000); // espera un segundo
20     digitalWrite(ledPin, LOW); // establece 0v en el pin del LED
21     delay(1000); // espera un segundo
22 }
```

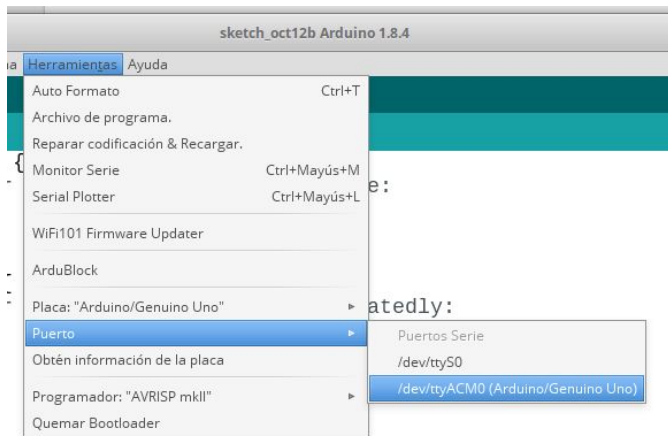
[Código](#)

# Variables

Es un espacio de memoria en el que almacenamos un valor.



# Configuración y carga de programas



**Subir el programa a la placa**

- Seleccionamos el modelo de placa: Herramientas→ Placa→ Arduino UNO
- Seleccionamos el puerto de conexión: Herramientas → Puerto /dev/ttyACM0
- La placa debe estar conectada
- El nombre del puerto varía según SO y modelo de placa

# Propuesta de actividades

---

1. Prueba a cambiar los tiempos de parpadeo
2. Haz que parpadee el LED al ritmo del corazón 90ppm
3. ¿A qué velocidad deja de verse el parpadeo del LED?
4. ¿Cómo podemos hacer que sólo parpadee una vez?
5. Crear variables para el tiempo de parpadeo
6. Decrementar el tiempo de parpadeo en cada iteración
7. Usar TinkerCAD para simular el circuito

# LICENCIA



Esta guía se distribuye bajo licencia Reconocimiento-CompartirIgual Creative commons 4.0

Las diapositivas son obra de Jose Pujol y Jose Luis Núñez creadas para el curso “Controlando Arduino desde el teléfono móvil” para el CEP de Sevilla y han sido creadas a partir del material elaborado para el curso “Tech Project: Arduino en el aula” que fue realizado por Jose Antonio Vacas y Jose Pujol en colaboración con Avante s.l.

