



UNIVERSIDAD
DE GRANADA



Escuela Técnica Superior de Ingenierías
Informática y de Telecomunicación



Necesitamos tu talento



LA INGENIERÍA EN LA MÚSICA, JUEGOS Y DRONES

Autores: [Elvira Castillo](#) y [Samuel Romero](#)

twitter: [@layoel](#)

Instagram: [eca_ingeniera](#)

QUIERO
SER Ingeniera

CC creative
commons





UNIVERSIDAD
DE GRANADA

¿QUÉ HAREMOS HOY?



1. ARDUINO
2. PROGRAMACIÓN EN C++
3. PRÁCTICA: THEREMIN
4. PRÁCTICA: SEMAFORO
5. PRÁCTICA: JUEGO ATRAPA AL TOPO
6. VISITA AL LABORATORIO DE MECATRÓNICA Y DRONES.



UNIVERSIDAD
DE GRANADA



ARDUINO

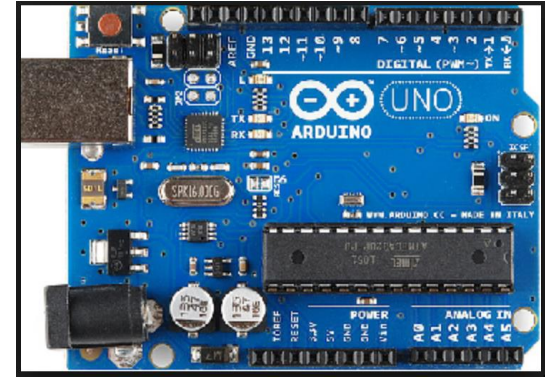


QUIERO
SER *Ingeniera*

¿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)





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.

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/salidas analógicas

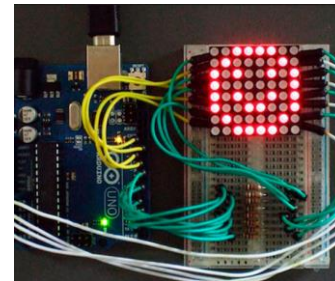
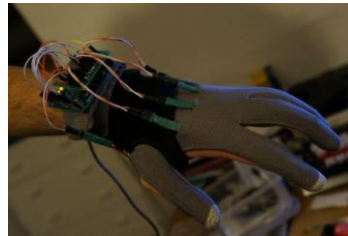
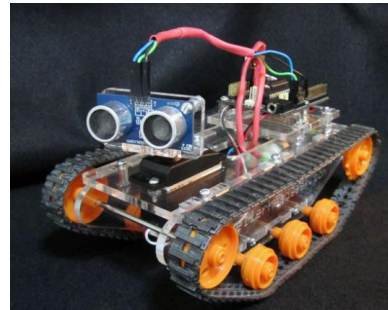
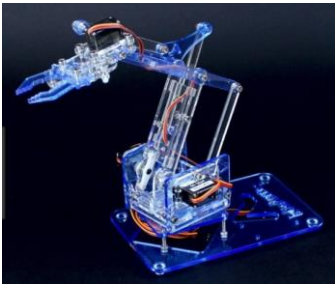


- El arduino no tiene capacidad para trabajar con señales analógicas, tiene que convertirlas a digitales.
- Pines analógicos: A0,A1,A2,A3,A4,A5
- Para tratar Entradas/salidas analógicas usamos las funciones:
 - `analogRead(pin);` // que lee del pin analógico especificado
 - `analogWrite(pin);` // que 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 puedo hacer con Arduino?



¡Todo lo que se te ocurra!





UNIVERSIDAD
DE GRANADA



PROGRAMACIÓN C++



**QUIERO
SER** *Ingeniera*



•¿Qué es un algoritmo?

- Secuencia ordenada de instrucciones que resuelve un problema concreto

•Lenguajes de programación:

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

•¿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.

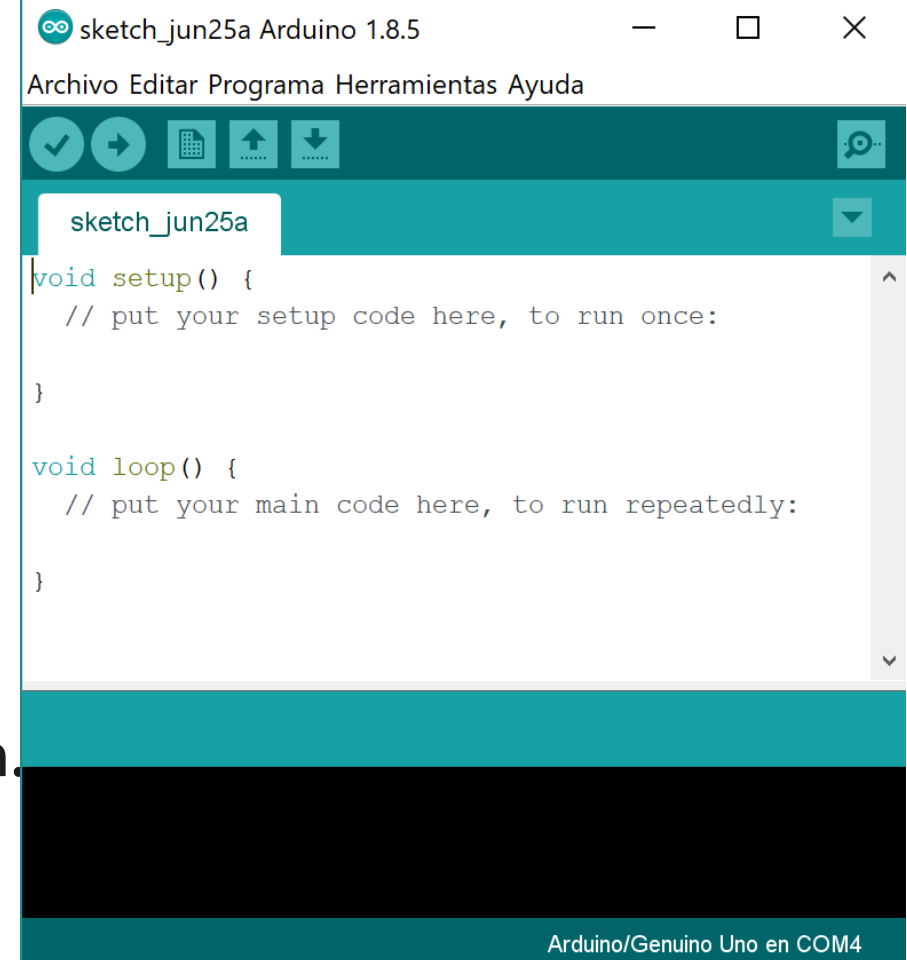
•Compilación

- Traducir un programa a código ejecutable por la máquina

¿Cómo se programa el 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.



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;

- Texto:

- `Char` caracter;
- `String` cadena;

- caracter = 'c';
- cadena = 'hola mundo';

- Lógicos

- `Bool` midato; (sólo toma valor verdadero o falso)

- midato = true;

- Complejos:

- `int` vector[2];

- Vector[0] = 3;
Vector[1] = 2;



• Tipos

• Matemáticos:

- $+$, $-$, $*$, $/$, $\%$.

• Relacionales:

- $==$, $!=$, $<$, $>$, $>=$, $<=$

• Lógicos

- $\&\&$, $\|\|$, $!$.

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

Estructuras Condicionales

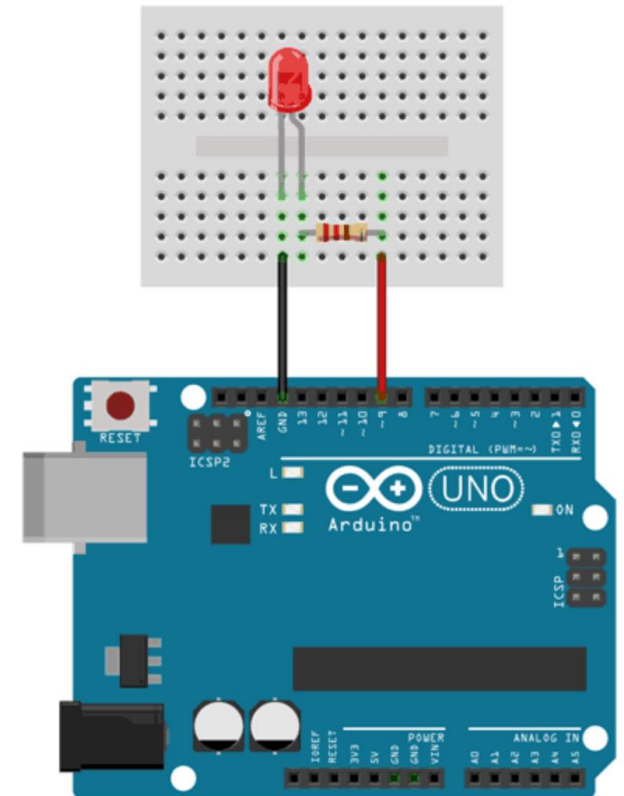


- Conditional simple

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

Ejemplo:

Comprobar si un número es par.
Si lo es encender el led conectado al pin 9.



Estructuras Condicionales



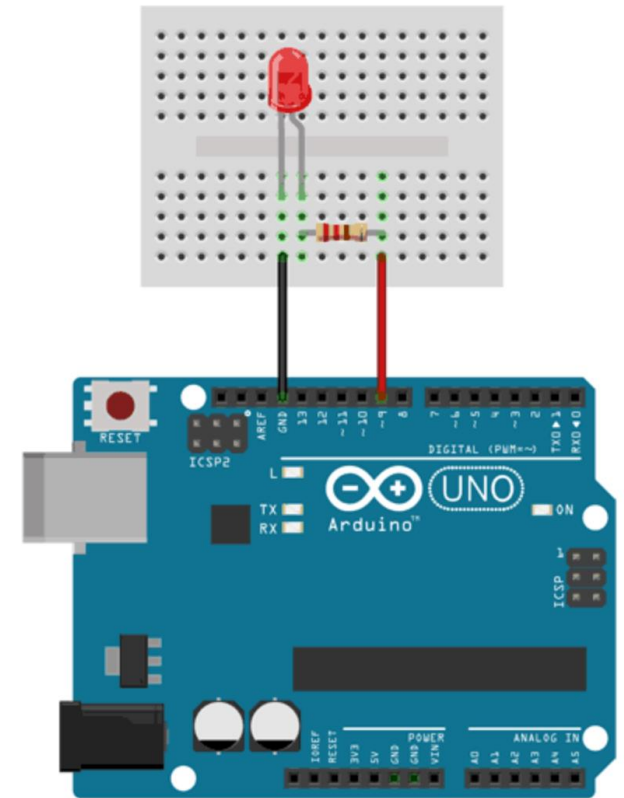
- Condicional compuesto

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

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

Ejemplo:

Comprobar si un número es par y menor que 6.
Si lo es encender el led conectado al pin 9.





- Condicional doble

```

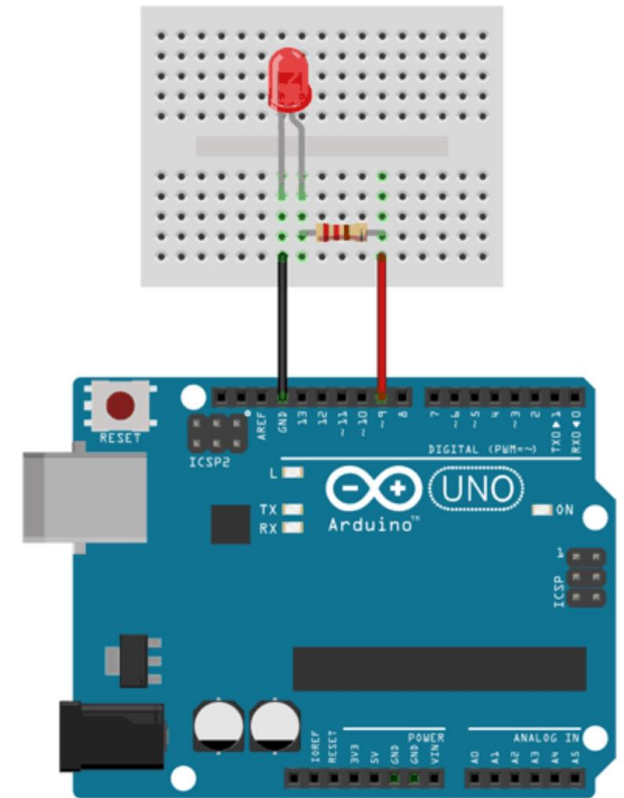
If (condición ) {
    sentencias;
}else{
    sentencias;
}
  
```

Ejemplo:

Comprobar si un número es par.

Si lo es encender el led conectado al pin 9.

Si no lo es encender el led conectado al pin 10.



Estructuras Condicionales



- Anidando condicionales

```
If (condición1 ){  
    If (condición2 ){  
        sentencias;  
    }else{  
        sentencias;  
    }  
}else{  
    sentencias;  
}
```

Ejemplo:

- Si un numero es par y menor que 6 encender el led. Si no es menor que 6 que suene el buzzer
- Si no es par encender el led conectado al pin 10.



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

}

Pensar y Esperar en arduino



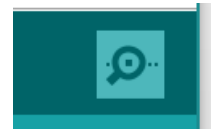
En Arduino:

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);`



¡Vamos a practicar!

Hacer un Theremin



Materiales:

.Microcontrolador:

- Arduino

.Actuador:

- Buzzer

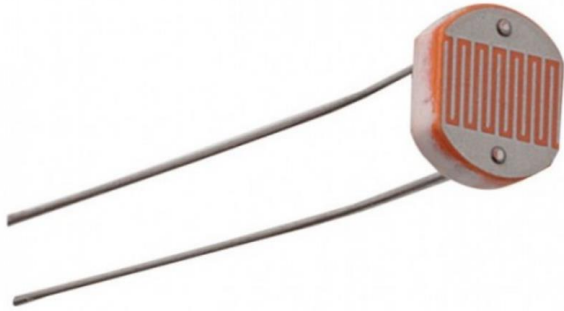
.Sensor:

- Fotorresistencia

.Otros

- Resistencia y cables.

¿Cómo lo conectamos?



<https://www.digikey.es/es/resources/conversion-calculators/conversion-calculator-resistor-color-code-5-band>

¿Cómo se programa?



- Recuerda en el `setup()` ponemos los pines digitales como entrada o salida.
- Busca información sobre la función `tone()` de arduino.
- ¿Para que sirve la función `map()`?
- Los pines se pueden asignar a variables y usar un nombre de variable en lugar del numero de pin.

Semaforo cambia pulsando botón



Materiales:

•Microcontrolador:

- Arduino.

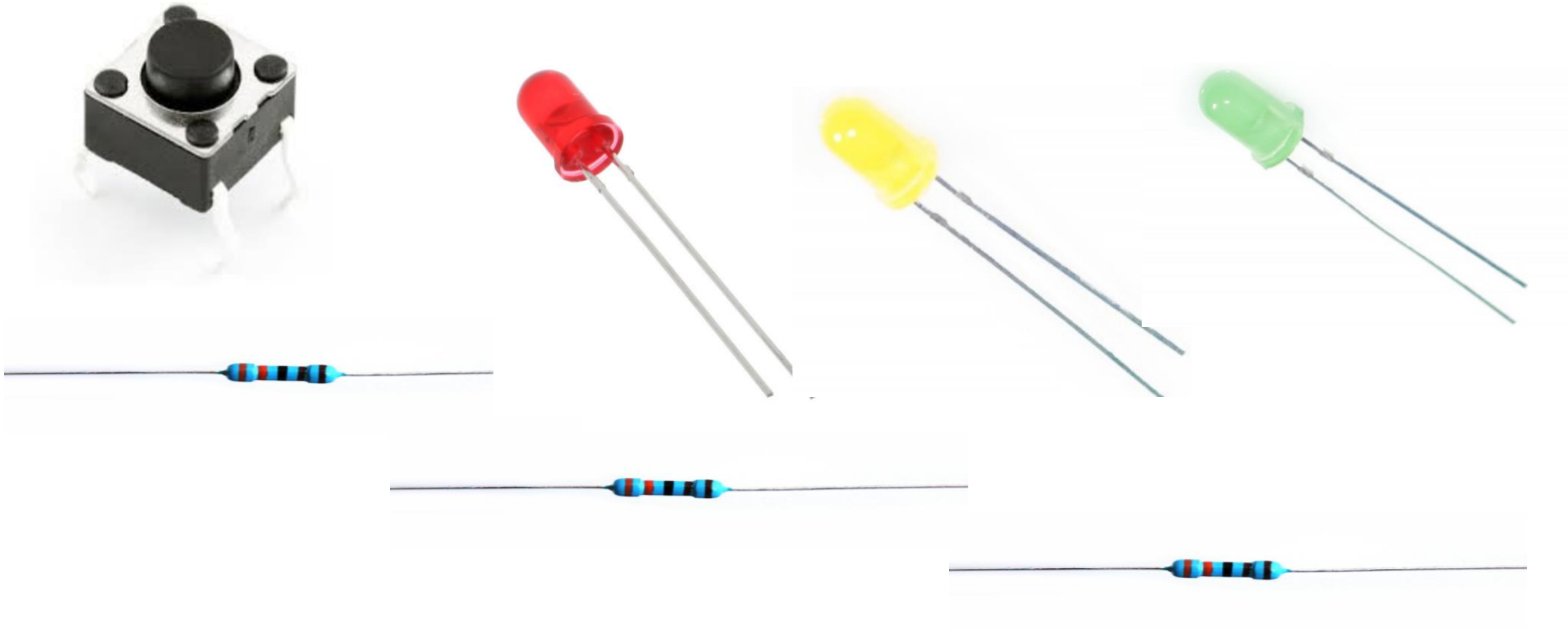
•Actuadores:

- Botón.
- Leds.

•Otros

- Resistencias y cables.

Como los conectamos





Materiales:

.Microcontrolador:

- Arduino.

.Actuadores:

- Botones.
- Leds.
- Buzzer.

.Otros

- Resistencias y cables.

¿Por dónde empiezo?



- Recuerda en el `setup()` ponemos los pines digitales como entrada o salida.
- Busca información sobre la función `random(min,max)` de arduino.
- Piensa primero el algoritmo antes de programarlo.
- Aquí tienes un vídeo de ayuda del funcionamiento del juego <https://www.youtube.com/watch?v=NYZ5i9UxcUk>