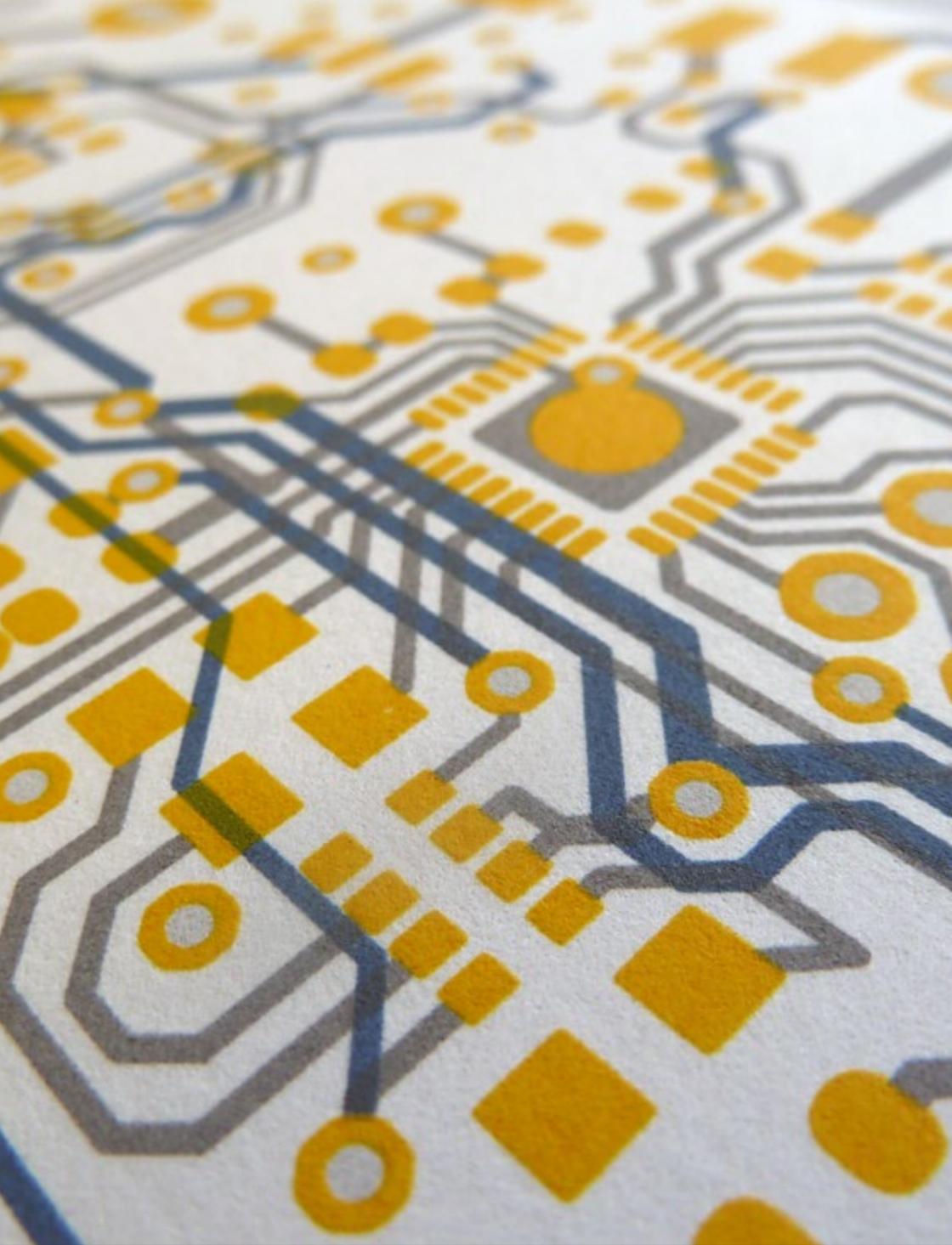


**MESH**

**ELECTRÓNICA Y PROGRAMACIÓN**

Núria Alonso



## ÍNDICE

1. Introducción – Conceptos clave .....	1
a. De qué hablamos cuando hablamos de electrónica	
b. De qué hablamos cuando hablamos de programación	
c. ¿Qué es la robótica?	
2. Software y hardware libre. ¿Por qué importa? .....	2
3. ¿Qué es esto que estoy viendo? La placa Arduino UNO	
4. ¡Manos a la obra! .....	3
5. Observando el mundo: sensores .....	6
6. Transformando el mundo: actuadores	
7. Artista de referencia y links de interés .....	8
8. Biografía de la Mentora .....	9

Fuente: [Sandy Noble](#)

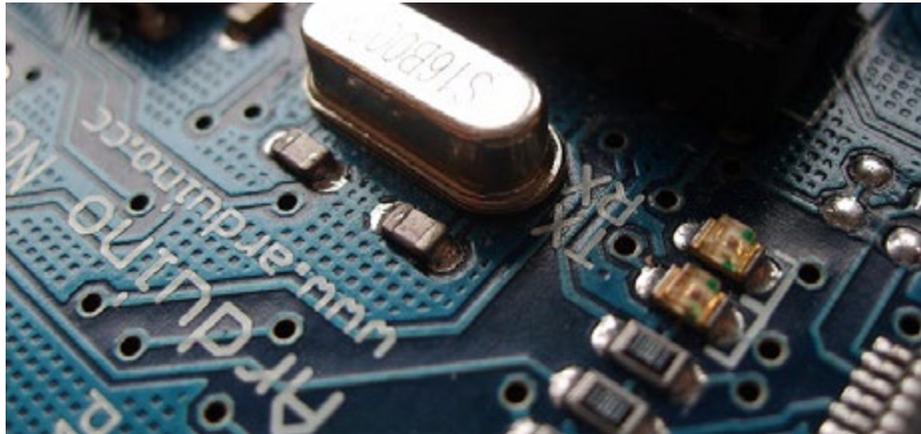
# 1. INTRODUCCIÓN – CONCEPTOS CLAVE

## a) De qué hablamos cuando hablamos de electrónica

Mira tu entorno: seguramente, cada metro cuadrado que rodea tiene algo de eléctrico. Desde las luces de la habitación, pasando por la nevera de tu cocina o tu teléfono móvil. Todos estos objetos tienen un mismo elemento común: son circuitos por los que pasa la electricidad, y gracias a su mayor o menor complejidad, activamos o desactivamos cosas.

## b) De qué hablamos cuando hablamos de programación

Podemos manipular estos impulsos eléctricos que pasan por los circuitos, manipulando como funcionan los mismos circuitos. Para ello, debemos aprender un lenguaje que entiendan los propios circuitos: un lenguaje de programación. Y aquí empieza la diversión: riéte de la Torre de Babel.



Con toda esta diversidad, ¿por dónde empezar a aprender? Una buena idea puede ser empezar por lenguajes sencillos que nos permitan adquirir las competencias necesarias para ser una buena programadora. Los lenguajes visuales pueden ser una buena opción para pequeños y grandes aprendices.

<https://code.org/>

<https://scratch.mit.edu>

<https://hourofcode.com>

Pregúntate si sabes qué son o por qué importan los siguientes conceptos:

- Un algoritmo
- Secuenciación
- Un bucle
- Una variable

Al final de este módulo habrás usado estos conceptos para crear tu propia solución tecnológica.

## c) ¿Qué es la robótica?

**Electrónica + programación = ¡Robótica!**

Tenemos estos pequeños circuitos electrónicos y hemos aprendido a programar. Ahora llega la hora de empezar a hacer que trabajen para nosotras: **vamos a automatizar ciertas tareas para crear nuestro primer robot.** Estamos rodeadas de tecnología, ¿no sería maravilloso no sólo entenderla sino imaginar, hacer, crear nuestros propios complementos tecnológicos?

Pero, ¿cómo funciona un robot? Te proponemos que realices esta dinámica con tus compañeros o con tu grupo:

*Nos ponemos en círculo y buscamos a nuestra voluntaria. Esta persona hará de nuestro primer robot. Como robot, solo entenderá y seguirá las órdenes que reciba en su propio lenguaje. Entre todas acordamos **cómo será este lenguaje: una palmada significa un paso hacia delante, una patada, girar hacia la derecha, un chasquido de nuestros dedos, girar hacia la izquierda...** como acordemos. El robot debe cerrar los ojos y solo moverse siguiendo estas instrucciones.*

*Ahora escogemos a una segunda voluntaria. Esta segunda persona, será un virus, y nuestro robot debe atraparlo. Le pedimos al virus que se sitúe en un punto dentro de nuestro círculo, y **deberemos guiar al robot (que tiene los ojos cerrados) hacia el virus.** ¡Usando su lenguaje de programación, claro!*

*Una vez hemos visto cómo funciona, nos dividimos en pequeños grupos de tres: uno hará de robot, otro de virus y la tercera de programadora. Cuando el primer robot haya atrapado el virus, intercambiamos los papeles y seguimos jugando.*

*Al cabo de un rato, nos volvemos a reunir todas e intercambiamos nuestras impresiones. ¿Cómo te has sentido siendo un robot? ¿Y como programadora? ¿El lenguaje que hemos hecho, es útil? ¿Cómo lo mejorarías?*

Esta dinámica (encontrarás más en [codinc.fun](http://codinc.fun)) nos puede servir para comprender en nuestro propio cuerpo conceptos como secuencia, repetición, bucle, etc., y para entender que **un robot solo va a hacer aquello que nosotras le digamos.** ¡Debemos estar atentas a todos los pasos que demos!

## 2. SOFTWARE Y HARDWARE LIBRE. ¿POR QUÉ IMPORTA?

Basaremos este módulo en aprender herramientas libres. No solo el software, todos los programas que usaremos están hechos en software libre y abierto. También los circuitos electrónicos, los pequeños procesadores y todos y cada uno de sus componentes, son también de código abierto.

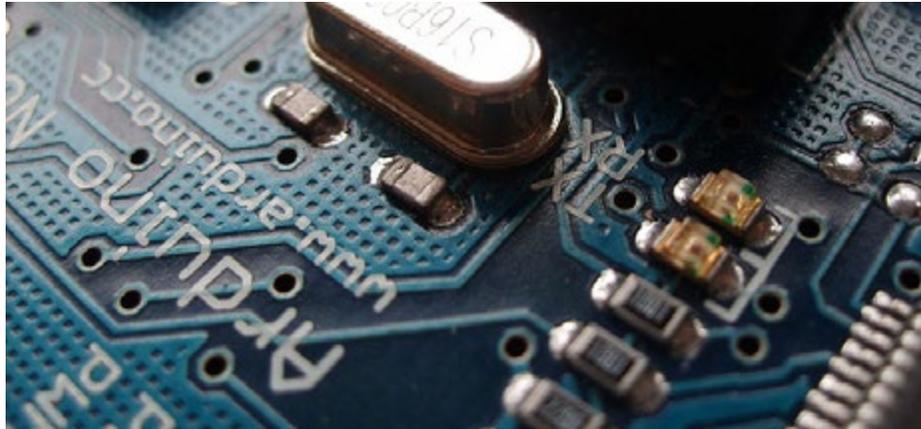
Arduino es uno de los proyectos de hardware libre más populares del mundo. Nos permite trabajar en proyectos de robótica con bajo coste y, además, cuenta con una gran documentación y soporte realizada por una gran comunidad interconectada.

Una vez iniciado el aprendizaje, es importante reforzar la capacidad de búsqueda de soluciones de manera autónoma por parte de los participantes.

Nuestro rol como formadores aquí, se centrará en ofrecer unos conocimientos y competencias básicas y facilitar el desarrollo de inquietudes por parte de las personas participantes.

## 3. ¿QUÉ ES ESTO QUE ESTOY VIENDO? LA PLACA ARDUINO UNO

Te presentamos a Arduino UNO, una placa de aprendizaje muy potente que nos acompañará durante mucho tiempo. Para que entendamos esta placa, debemos imaginarla como un circuito eléctrico al que hemos añadido componentes que nos permiten más interacción.



Fuente: [wikipedia](https://es.wikipedia.org/wiki/Arduino_Uno)

Estos elementos pueden ser microcontroladores, es decir, pequeños circuitos integrados que procesan información a gran velocidad, permitiendo programar funciones y tareas. Al final de este módulo, habrás aprendido cómo programar tu primer sistema robotizado. Para esto necesitarás:

### Un ordenador

PC, con linux o windows, o un Mac. Trabajamos con herramientas libres y multiplataforma. Nuestra apuesta es empoderar tecnológicamente a nuestras participantes, ¡no atarlas a un modelo tecnológico cerrado!

### Instalar Arduino IDE... ¡no! Trabajamos online.

Tenemos dos opciones para trabajar con Arduino. Podemos instalar el programa que usaremos como interfaz de programación

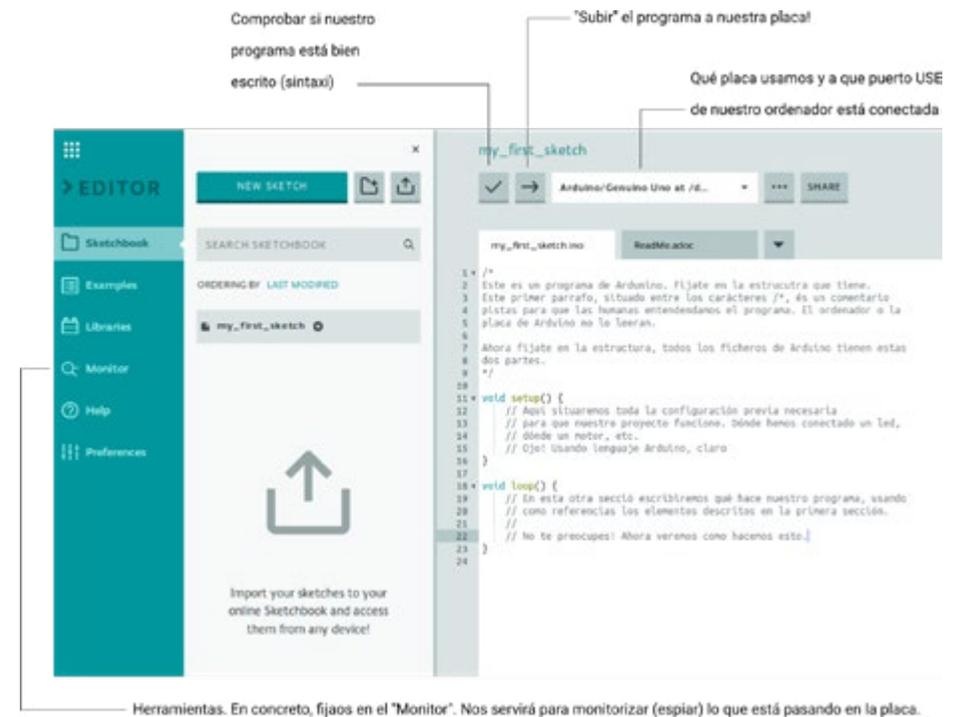
y comunicación con nuestros proyectos de robótica con Arduino. Esta alternativa es fundamental si no disponemos o tenemos problemas con nuestra conexión a Internet.

<https://www.arduino.cc/en/Main/Software>

Pero os proponemos trabajar con la versión online. ¡Ojo! También requiere que hagamos una primera instalación del plugin que nos permitirá establecer una comunicación entre el navegador web donde programaremos y la placa de Arduino que queremos programar.

<https://create.arduino.cc/>

<https://create.arduino.cc/getting-started/plugin>

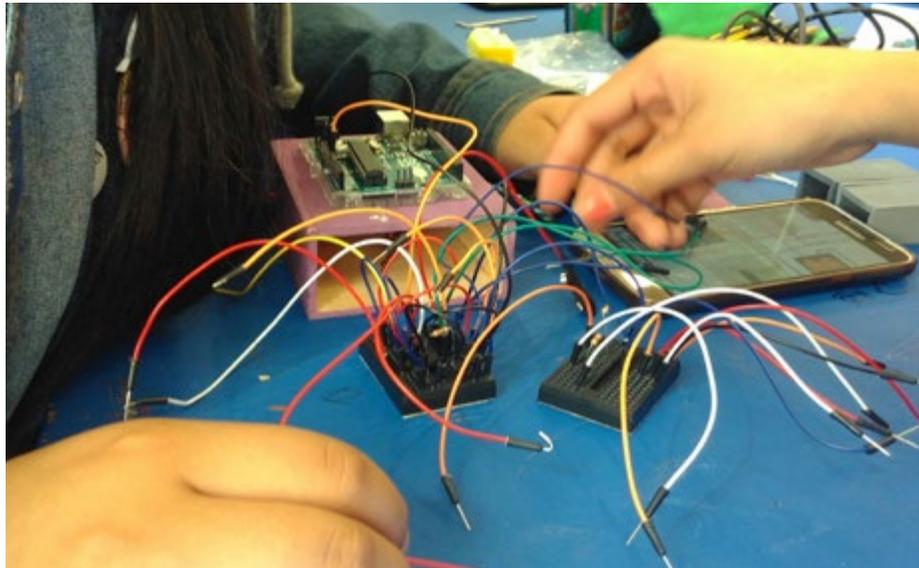


Dentro de las ventajas de trabajar online está que accederemos a mayor información, siempre actualizada y más fácil para compartir con nuestras compañeras.

## 4. ¡MANOS A LA OBRA!

Fíjate en el siguiente esquema. Verás las partes más importantes que debes conocer para empezar a programar usando Arduino. Conecta en tu ordenador la placa Arduino Uno usando el cable USB.

Si todo está bien configurado verás cómo se iluminan los leds integrados en la placa. Si no, revisa las conexiones y la configuración de tu ordenador.



### >> Hello world

El primer paso que vamos a dar es el de controlar estos leds. Abre el proyecto que encontrarás en **Examples / 01.BASICS / Blink**. En la **sección setup** definimos con qué vamos a trabajar.

Fíjate que de momento usamos el led integrado en la placa.

En la **sección loop** describimos lo que va a hacer nuestro programa. Ojo que Arduino va a ejecutar este programa en loop infinito, a menos que le digamos lo contrario.

En este caso, se enciende (HIGH) el led durante un segundo (el delay de 1000 milisegundos) y se apagará (LOW) durante el mismo tiempo. En loop. Todo el rato.

Sube el programa usando el botón Upload. Si todo va bien, los leds de la placa parpadearán rápidamente y empezará a ejecutarse el programa. Ahora comprueba que pasa si haces variar los valores de los delays.

Hagámoslo un poco más difícil. Siguiendo [este enlace](#) verás cómo podemos controlar también un led externo a la placa.

NOTA: Estamos construyendo un **circuito eléctrico**. Toda la energía que sale (en este caso, por el pin 9), debe volver por algún sitio: he ahí el significado del pin GND (del inglés "ground", tierra).

Imagina siempre como circula la energía eléctrica en los circuitos que construyes, y pregúntate cómo se activan los distintos componentes según vaya pasando la energía por ellos.

## >> ¿Analógico o digital?

Si te fijas, hemos empezado con un proyecto muy simple: la luz estaba o encendida o apagada. Pero el mundo rara vez es tan aburrido. La realidad casi nunca es blanco o negro, no podemos reducirla a un esquema tan binario. Es más, ¡nos encantan los colores!

Pero el mundo de la electrónica se basa en este esquema binario: o pasa electricidad o no, el led está encendido o apagado... ¿Cómo podemos cambiar esto? Arduino nos permite trabajar con la Modulación por ancho de pulsos (PWM, *pulse-width modulation*), una técnica usada para simular el comportamiento analógico (el del mundo real) a partir de señales digitales (las que nos proporciona nuestro circuito electrónico con Arduino).

<https://www.arduino.cc/en/Tutorial/PWM> (Tutorial en inglés)

Sobre señales analógicas, digitales y su relación con PWM

<https://aprendiendoarduino.wordpress.com/2017/10/22/entradas-y-salidas-analogicas-arduino-pwm/>

Para ver cómo usamos el PWM en un ejemplo real, abrimos nuestro editor de Arduino y buscamos el siguiente proyecto de ejemplo: **Examples / 01.BASICS / Fade**

Monta el circuito. Fíjate en dos cosas:

1. Para conectar el led, usamos como pin de salida el pin 9. En la placa está marcado con una “~”. Eso significa que este pin nos permite trabajar con PWM y simular las señales analógicas sobre nuestra placa digital.

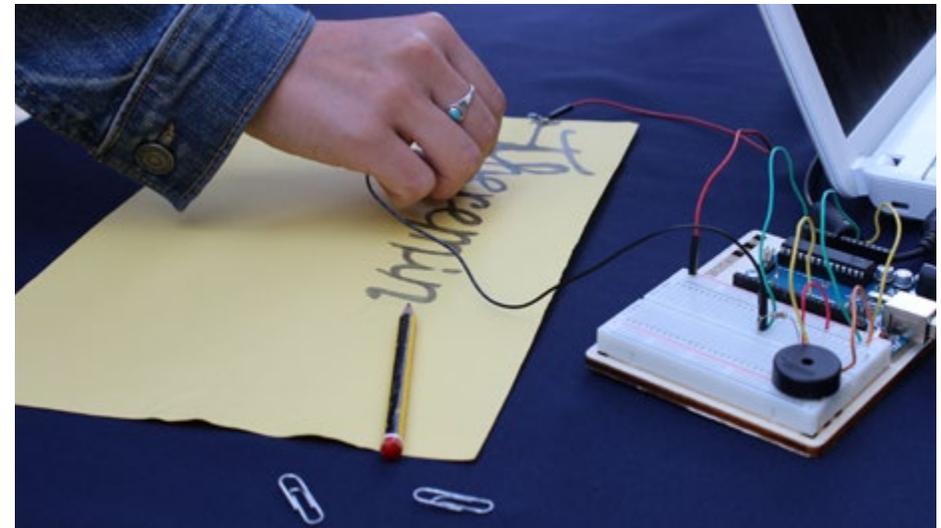
2. Conectamos una pequeña pieza delante del led. Esto es una **resistencia**, que nos gradúa la intensidad de la corriente para que no quememos el led con los 5v con los que trabaja la placa de Arduino.

Video explicación sobre por qué usamos resistencias.

<https://bit.ly/2SQTE70>

Con el circuito montado, conecta la placa de Arduino con el USB de tu ordenador y sube el código del ejemplo. Verás como el led se ilumina y se apaga gradualmente. Ahora fijémonos un poco en el código que acabamos de subir a nuestra placa:

1. Antes que nada, este código define las variables que vamos a usar en el programa. Estas variables son numéricas, van desde el



pin que usaremos (el 9), el brillo del led (0 será apagado, 255 la luminosidad máxima que tendrá) o los pasos con los que se irá iluminando u oscureciendo el led.

2. Una vez declaradas las variables, encontramos la **sección setup**. Como en el ejemplo anterior, en el setup definimos la configuración inicial: el led 9 será un dispositivo de salida.

3. En la sección loop podemos ver como este código tiene algo más de chicha.

En `brightness = brightness + fadeAmount`; definimos que el brillo irá aumentando con los pasos definidos en la variable `fadeAmount`.

En las líneas:

```
if (brightness <= 0 || brightness >= 255) {  
  fadeAmount = -fadeAmount;  
}
```

Usamos un condicional. Cuando los pasos llegan a su valor máximo (255) cambiamos el valor de estos a negativo, de manera que empezamos a decrecer el valor del brillo. Al llegar a su valor mínimo (0) volvemos a cambiar el valor de los pasos a positivo.

¡Todo esto, en loop!

## 5. Observando el mundo: sensores

El siguiente paso que vamos a realizar es aprender cómo recoger información de nuestro entorno usando la placa de Arduino.

Para hacer esto, deberemos añadir un elemento más a nuestro circuito: los sensores que son los ojos, las orejas, nariz de nuestro robot! Usaremos otro ejemplo de nuestro editor de Arduino. En este caso abre **Examples / 06.SENSORS / Ping** . Realiza el circuito, sube el código a tu placa y...¿no pasa nada?

Estamos usando un sensor de ultrasonidos. Este sensor funciona como un pequeño sonar. Emite una señal de ultrasonido que rebota contra los objetos que tiene delante, retornando al sensor.

Podemos leer cuánto tarda esta señal en volver, y a partir de este dato, calcular a qué distancia está. Pero, ¿dónde están estos datos?

Deberás abrir el **Monitor Serial** para poder leerlos. Acuérdate que al inicio de este módulo hemos hablado de las herramientas que incluye el IDE de Arduino, y una de ellas es el Monitor Serie (Serial monitor). Lo encontrarás en la barra lateral izquierda, en el icono de la lupa.

Para poder trabajar con el monitor y leer los datos con los que trabaja nuestra placa es indispensable que se lo digamos a la placa. Para ello, debemos declararlo en el **loop setup** tal y como ves en este ejemplo:

```
void setup() {  
  Serial.begin(9600);  
}
```

A partir de aquí, empezaremos a dotar a nuestro robot de algo más de interactividad.



## 6. TRANSFORMANDO EL MUNDO: ACTUADORES

Porque la tecnología que nos gusta es la tecnología que ayuda a las personas y nos ayuda a construir un mundo mejor. Nos permite diseñar soluciones tecnológicas adaptadas a nuestro entorno, por lo tanto necesitaremos:

1. **Sensores:** para recoger información de nuestro alrededor
2. **Actuadores:** para automatizar respuestas y obrar en consecuencia.

Te proponemos que sigas este proyecto de Arduino:  
<https://create.arduino.cc/projecthub/einsteinunicorn/aid-for-visually-impaired-haptic-feedback-ed1192>

Como hemos hecho en el ejercicio anterior, usamos el sensor de ultrasonidos pero esta vez **vamos a programar una respuesta a los objetos que detecte**. Esta respuesta será mover un motor, indicando la posición del objeto que ha “visto”.

Construiremos un pequeño dispositivo que ayudará a personas con problemas de visión a detectar posibles obstáculos para que se puedan mover con más confianza y agilidad.

Además, este proyecto requerirá que pienses este dispositivo tecnológico de manera integral. Deberás imprimir en 3D una parte, y hacer el montaje también de las piezas mecánicas.

## 7. ARTISTA DE REFERENCIA Y LINKS DE INTERÉS

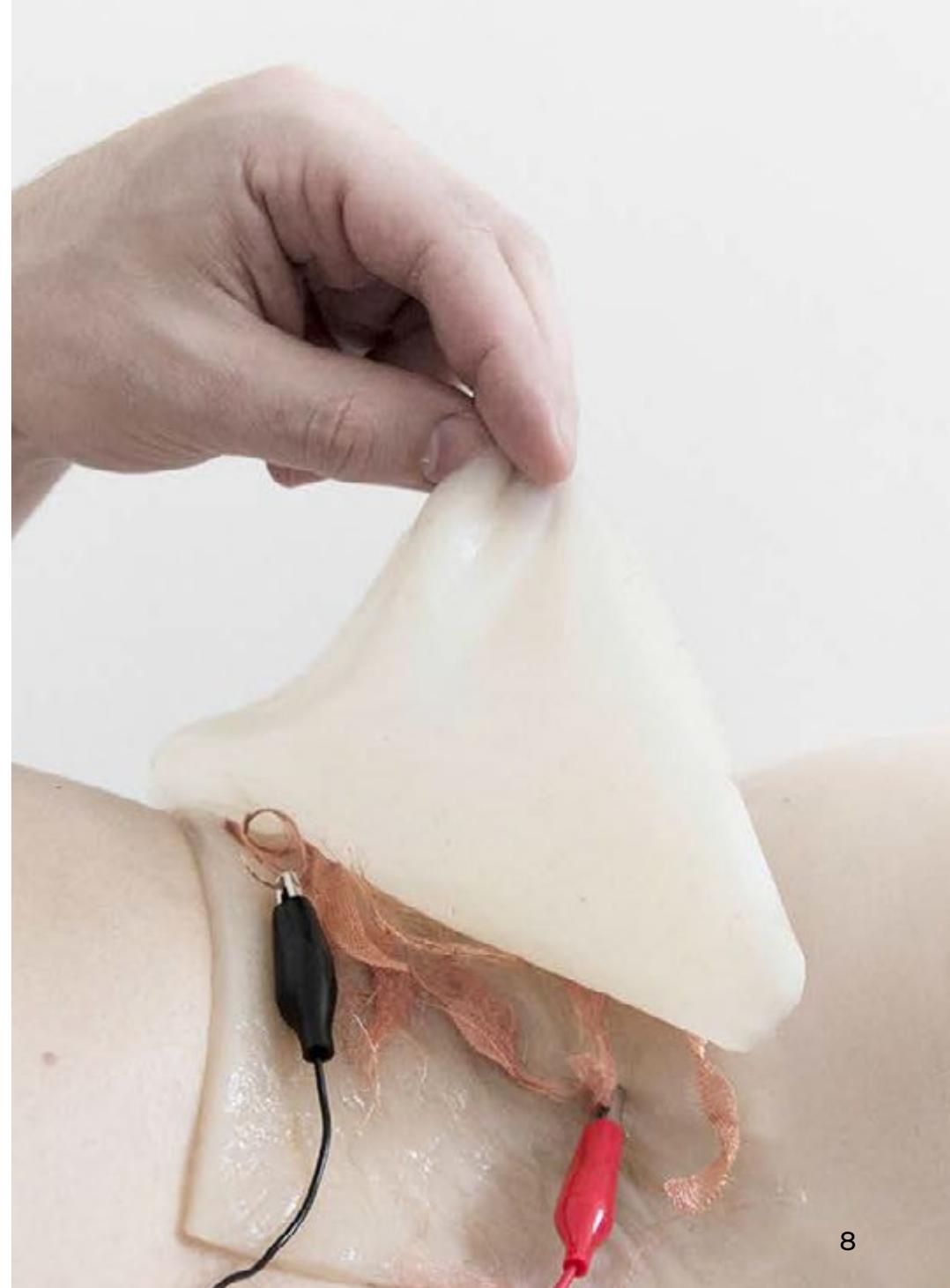
Si has llegado hasta aquí, ¡felicidades! Hemos repasado los conceptos claves de la robótica educativa, con los que podrás continuar aprendiendo mientras piensas, diseñas y desarrollas soluciones tecnológicas para mejorar tu entorno. Te recomendamos que continúes explorando otros proyectos, y pienses cómo pueden ser útiles para los grupos con los que trabajas.

Explora esta galería de proyectos <https://create.arduino.cc/projecthub> y piensa que Internet es un sitio maravilloso que está lleno de otros proyectos que te pueden inspirar y de los que puedes aprender.

Como el trabajo de **Giulia Tomasello**. A través de proyectos como *Rethinking the bra*, propone realizar una mirada crítica a una pieza de ropa interior mediante la aplicación de innovaciones tecnológicas. La tecnología aquí nos sirve para cuestionarnos lo que damos por hecho (¿por que los sujetadores son tan incómodos?, ¿quizá es una pieza de ropa que la diseñó alguien que no tenía que llevarla?) y empezar a pensar soluciones mejores usando todas las tecnologías que tenemos en nuestras manos.

<https://gitomasello.com/>

Fuente: [Giulia Tomasello - Conductive Bio Skin](#)



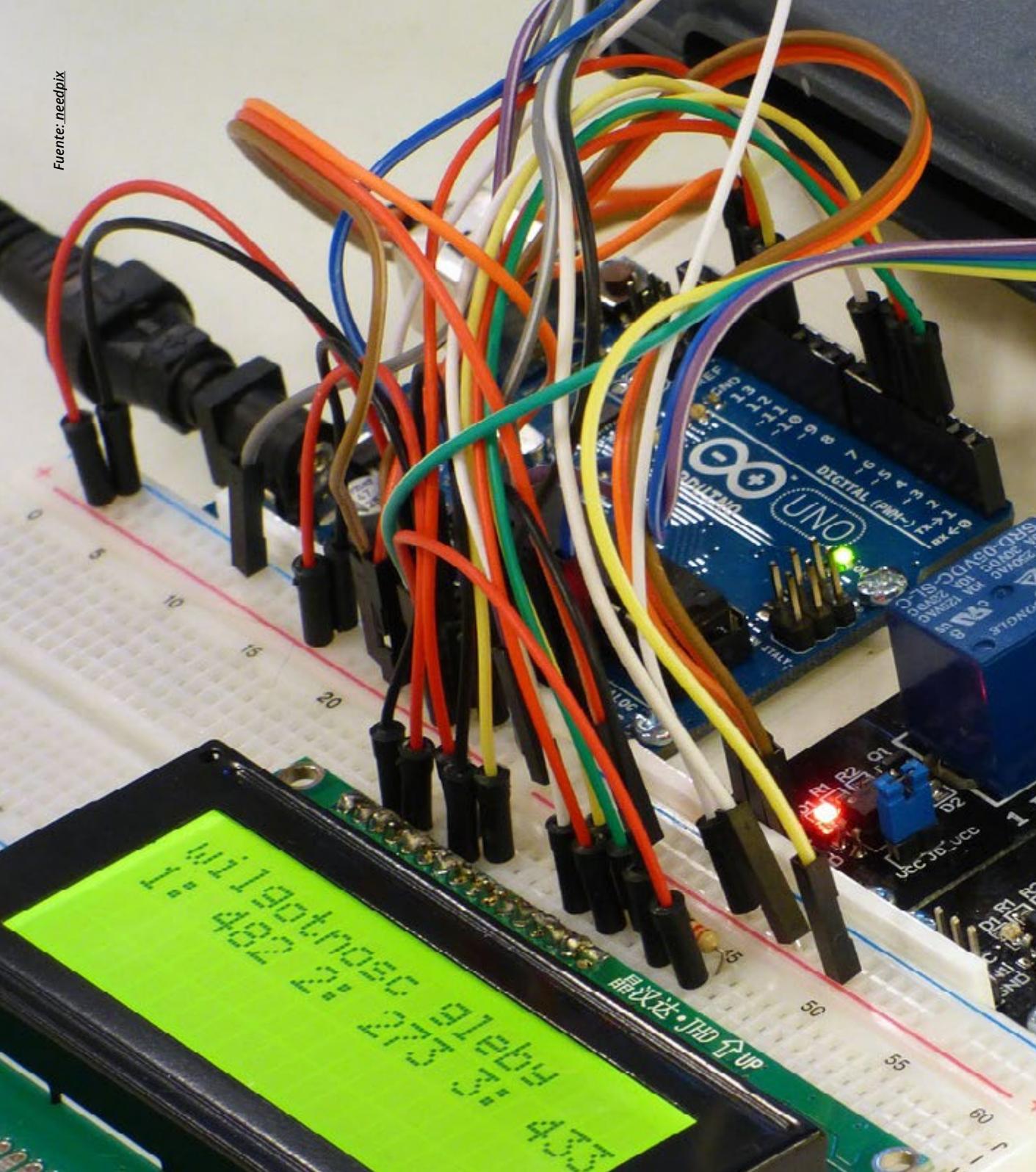


## **NÚRIA ALONSO**

Licenciada en Bellas Artes reconvertida a técnica informática. Apasionada de GNU/Linux y todo lo que lleve licencias abiertas, donde siempre se puede aprender más y más.

Desde hace años colabora en proyectos que piensan como incorporar la tecnología para mejorar nuestra sociedad. Socia de Colectic, cooperativa sin ánimo de lucro que impulsa proyectos tecnológicos que incluyan valores de responsabilidad social.

Trabajando por una tecnología abierta, libre, feminista y reciclable.



# MESH

## ELECTRÓNICA Y PROGRAMACIÓN

Núria Alonso  
2019

Technology of love by

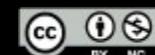


**SOKO**  
TECH

Amb el suport de l'Ajuntament de Barcelona



Ajuntament de  
Barcelona



Atribución-NoComercial 4.0 Internacional