

## Modelo de Aprendizaje para Arduino Uno Básico

### Learning Model for Arduino Uno Basic

LEDESMA-URIBE, Norma Alejandra†\*, CARDOSO-FALCÓN, Berenice y TORRES-GARCÍA, Oliva

*Universidad Tecnológica de San Juan del Río*

ID 1<sup>er</sup> Autor: *Norma Alejandra, Ledesma-Uribe* / ORC ID: 0000-0001-8422-2046, Researcher ID Thomson: S-4833-2018, CVU CONACYT ID: 673202

ID 1<sup>er</sup> Coautor: *Berenice, Cardoso-Falcón* / ORC ID: 0000-0002-7785-1095, Researcher ID Thomson: 6490-2018, CVU CONACYT ID: 955172

ID 2<sup>do</sup> Coautor: *Oliva, Orres-García* / ORC ID: 000002-4804-6708, Researcher ID Thomson: 6542-2018, CVU CONACYT ID: 955189

DOI: 10.35429/JCA.2019.10.3.15.22

Recibido Abril 20, 2019; Aceptado Junio 30, 2019

#### Resumen

Cada vez es más común que los grados de ingeniería incluyan la enseñanza de programación en sus planes de estudio. (Rubio, Mañoso, Romero Zaliz, & P. de Madrid, 2014). El presente trabajo se centra en la enseñanza de programación básica de Arduino IDE e implementación de una placa Arduino Uno. La enseñanza de la programación de Arduino IDE es básica y se puede aplicar a carreras relacionadas como Tecnologías de la Información y Mecatrónica. Es un modelo apropiado para que los alumnos de dichas carreras se motiven a visualizar en forma concreta los resultados de los programas desarrollados por ellos mismos. Se introdujo como elemento principal una placa Arduino Uno basada en principios de hardware y software para la realización de prácticas de programación cuyas entradas y salidas son sensores ultrasónicos HC-04 y otros componentes como Buzzer y Micromotor. Este trabajo se basa en la planeación de un prototipo de bastón inteligente que ayuda a personas débiles visuales. Estos aprendizajes adquiridos sobre programación básica en Arduino IDE, se pueden implementar en distintos proyectos donde se utilice Arduino Uno.

**Programación, Enseñanza básica, Arduino Uno**

#### Abstract

It is increasingly common for engineering degrees to include the teaching of programming in their curricula. The present work focuses on the teaching of basic Arduino IDE programming and implementation of an Arduino Uno board. The teaching of Arduino IDE programming is basic and can be applied to related careers such as Information Technology and Mechatronics. It is an appropriate model so that the students of these careers are motivated to visualize in concrete form the results of the programs developed by them. An Arduino Uno board was introduced as the main element, based on hardware and software principles for the implementation of programming practices whose inputs and outputs are HC-04 ultrasonic sensors and other components such as Buzzer and Micromotor. This work is based on the planning of a smart cane prototype that helps visually impaired people. These learnings acquired on basic programming in Arduino IDE, can be implemented in different projects where Arduino Uno is used

**Programming, Basic education, Arduino Uno**

**Citación:** LEDESMA-URIBE, Norma Alejandra, CARDOSO-FALCÓN, Berenice y TORRES-GARCÍA, Oliva. Modelo de Aprendizaje para Arduino uno Básico. Revista de Cómputo Aplicado. 2019, 3-10: 15-22

† Investigador contribuyendo como primer Autor.

## Introducción

Este modelo se compone de teoría y práctica basa en un ejemplo de bastón inteligente que establecen las bases necesarias para entender la plataforma hardware y software Arduino. Dado que el aprendizaje es un factor muy importante en el sector de la industria, se desarrolló un modelo de aprendizaje donde se aprenderán los principales conceptos de la programación Arduino, ya que este es una tarjeta electrónica digital y un IDE (entrono de desarrollo) en un lenguaje de programación basado en C++, con las que cualquier persona pude crear prototipos de todo tipo. Las instrucciones de Arduino son muy fáciles de aprender y usar, incluso para personas con poco conocimiento sobre electrónica y programación.

La idea será que las personas, tanto avanzadas como los novatos, sepan desarrollar unos prototipos respecto a sus necesidades y usarla libremente sin tener que comprar alguna muy costosa. Dentro de las tarjetas Arduino más conocidas se encuentra Arduino Uno, que es la utilizada y referenciada en este trabajo. (Hetpro, 2017).

Arduino es entonces una herramienta de procesamiento digital parecido a una computadora. Contiene puertos de entrada y salida digital a los cuales se les puede conectar: botones, leds, teclados o sensores digitales. Además de estos sensores mencionados, también tiene entradas analógicas que nos permite medir las señales de sensores análogos. Para poder visualizar la información que arrojan estos sensores, la tarjeta Arduino Uno cuenta con un puerto de comunicación serial que, mediante un Puerto USB de una computadora, permite enviar y recibir mensajes digitales. (Hetpro, 2017)

Las principales ventajas de Arduino Uno son:

- Costo
- Facilidad de aprender la herramienta
- Un software de desarrollo sencillo, no es necesario un programador externo a la tarjeta
- Un desarrollo de hardware libre, lo cual hace que se puedan crear proyectos para poderlos comercializar.

También una de las mayores ventajas de aprender Arduino es sencillamente el interés sobre programación, sobre algoritmos y sintaxis creando robots de Arduino, el código que se usa para programar pacas Arduino es bastante sencillo, así que será una manera más sencilla para poder entrenar un poco más el cerebro para entender la lógica usada en lenguajes de programación un poco más avanzados y si el interés trata sobre los electrónicos, este trabajo te introducirá al mundo de los circuitos, transistores, botones, pantallas, luces led y sensores, entre otros.

Arduino es una herramienta que vino a cambiar el modo de aplicar la tecnología embebida, la facilidad de uso de la tarjeta y su costo accesible al público y sobre todo la cantidad de programadores y desarrolladores hacen que la revolución sobre la tecnología sea más grande en el campo de los sistemas embebidos. En Arduino se permite un sin fin de aplicaciones y/o prototipos de forma muy accesible y rápida.

## Objetivos

- Fomentar el aprendizaje en un nuevo hardware utilizando la plataforma Arduino.
- Desarrollar prototipos respecto al lenguaje de programación usado y ejecutarlos sobre la plataforma Arduino.
- Conocer los componentes de hardware básicos para recibir señales externas y controlar elementos que le rodeen para interactuar con el mundo físico.

## Metodología

Las características principales de Arduino Uno son:

Características de Arduino Uno	
Velocidad	16Mhz
14 pines	Entrada/Salida digital
6 canales	Entrada analógica
1 puerto	UART
1 puerto	I2C
1 puerto	SPI
1 procesador	ATmega328p de 8bits
Memoria Flash	32KB
RAM	2KB
EEPROM	1KB
Voltaje	5Volts

**Tabla 1** Características Arduino Uno

– Pines de alimentación



**Figura 1** Tarjeta Arduino Uno Pines de Alimentación  
Fuente: Universidad de Cádiz

Se alimenta al Arduino mediante la conexión USB o mediante una fuente externa (recomendada de 7-12V).

Pines	
VIN	Se trata de la fuente tensión de entrada que contendrá la tensión a la que estamos alimentando al Arduino mediante la fuente externa. (Hetpro, 2017)
5V	Fuente de tensión regulada de 5V, esta tensión puede venir ya sea de pin VIN a través de un regulador interno, o se suministra a través de USB o de otra fuente de 5V regulada. (Hetpro, 2017)
3.3V	Fuente de 3.3 volts generados por el regulador interno con un consumo máximo de corriente de 50mA. (Hetpro, 2017)
GND:	Pines de tierra. (Hetpro, 2017)

**Tabla 2** Pines de Alimentación

– Digital Inputs/Outputs



**Figura 2** Tarjeta Arduino Uno Inputs/Outputs  
Fuente: Universidad de Cádiz

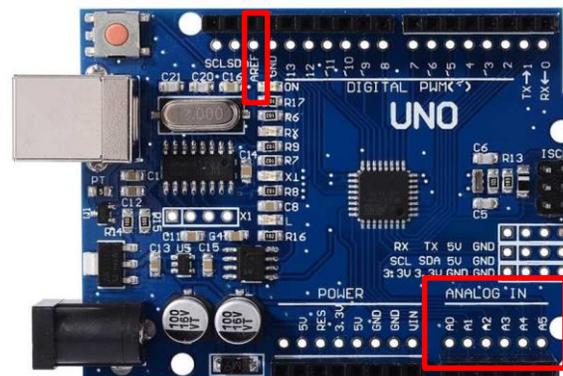
Cada uno de los 14 pines digitales se puede utilizar como una entrada o salida. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectado por defecto) de 20 a 50 kOhm.

Además, algunos pines tienen funciones especializadas como:

Pines inputs/outputs	
Pin 0 (RX) y 1 (TX)	Se utiliza para recibir (RX) y la transmisión (TX) de datos serie TTL. (Hetpro, 2017)
Pin 2 y 3	Interrupciones externas. Se trata de pines encargados de interrumpir el programa secuencial establecido por el usuario. (Hetpro, 2017)
Pin 3, 5, 6, 9, 10 y 11	PWM (modulación por ancho de pulso): Constituyen 8 bits de salida PWM con la función analogWrite ().(Hetpro, 2017)
Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)	Estos pines son de apoyo a la comunicación SPI. (Hetpro, 2017)
Pin 13 LED	Hay un LED conectado al pin digital 13. Cuando el pin es de alto valor, el LED está encendido, cuando el valor está bajo, es apagado. (Hetpro, 2017)

**Tabla 3** Pines Inputs/Outputs

– Analog Inputs



**Figura 3** Tabla de Arduino Uno Analog Inputs. Fuente: Universidad de Cádiz

El Arduino posee 6 entradas analógicas, etiquetadas desde la A0 a A5, cada una de las cuales ofrecen 10 bits de resolución. Por defecto, tenemos una tensión de 5V.

**Instrucciones para Arduino**

- Descargar IDE (software) de Arduino
- Conectar la tarjeta Arduino

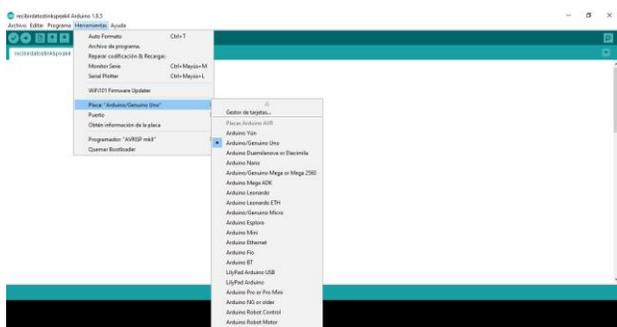
Se conecta la tarjeta Arduino a la computadora con el cable USB, una vez conectada el led de la placa PWR (led de alimentación) deberá permanecer encendido a partir de ahora.

- Instalar los drivers

Al conectarse el Arduino, Windows automáticamente deberá de inicializar la instalación de los drivers.

- Ejecución de placa y puerto serie

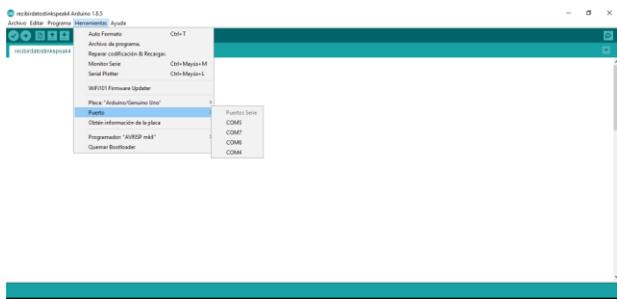
Abierta la aplicación IDE seleccionar Herramientas -> Placa -> Arduino Uno



**Figura 4** Ejecución de placa y puertos

Fuente: Universidad de Cádiz

Una vez seleccionado el modelo de placa, seleccionar el dispositivo el puerto en donde se conectó la placa, por lo regular este puerto se configura por defecto. (Cádiz, Comenzando con Arduino, 2017).



**Figura 5** Puerto Serial

Fuente: Universidad de Cádiz

La estructura de lenguaje de arduino es muy simple y se divide en configuración que incluye la declaración de variables y la inicialización de los pines. El segundo elemento del lenguaje es continuamente leído y ejecutado (loop) de acuerdo a las entradas y salidas que se le configuran.



**Figura 6** Lenguaje de Arduino

Fuente: Universidad de Cádiz

- Cargar el programa a la tarjeta

Para que el programa sea ejecutado, debe cargarse como



**Figura 7** Subir programa desarrollado

Fuente: Universidad de Cádiz

## Funciones Básicas

- E/S Digitales

pinMode. Configura el pin especificado para comportarse como una entrada (INPUT) o una salida (OUTPUT).

digitalWrite. Asigna el valor HIGH (5V) o LOW (0V) a un pin digital. Ejemplo: digitalWrite(Pin13, HIGH);

digitalRead. Lee el valor de un pin digital especificado, HIGH o LOW. Ejemplo: val=digitalRead(Pin13);

- E/S Analógicas

Lee el valor de tensión en el pin analógico especificado. La tarjeta Arduino posee 6 canales conectados a un conversor analógico digital de 10bits. Esto significa que convertirá tensiones entre 0 y 5 volts a un número entero entre 0 y 1023.

Esto proporciona una resolución en la lectura de 5 volts/1024 unidades, es decir: 00.0049 volts (4.9mv) por unidad. El rango de entrada puede ser cambiado usando la función `analogReference()`.

`analogWrite()`. Se escribe un valor analógico en un pin. Puede ser usado para controlar la luminosidad de un LED o la velocidad de un motor. Después de llamar a la función `analogWrite()`, el pin generará una onda cuadrada estable con el ciclo de trabajo especificado hasta que se vuelva a llamar a la función `analogWrite()`. La frecuencia de la señal será de aproximadamente 490 HZ, los valores de `analogRead` van desde 0 a 1023 y los valores de `analogWrite` van desde 0 hasta 255. (Cádiz, Comenzando con Arduino, 2017)

#### – Comunicación Serie

Se utiliza para la comunicación entre la tarjeta Arduino y una computadora u otros dispositivos. Todas las placas Arduino tienen al menos un puerto serie Serial. Se comunica a través de los pines digitales 0 (RX) Y 1 (TX), así como con la computadora por medio del puerto de USB. Selecciona la misma velocidad de baudios utilizada en el renglón `begin()`.

Comunicación serie	
<code>Serial.begin</code>	Selecciona la misma velocidad de baudios utilizada en el renglón <code>begin()</code> . Aquí se establece la velocidad de datos en bits por segundo o baudios, para que se transmitan los datos en serie. Para comunicarse con la computadora, se pueden utilizar varias velocidades: 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 o 115200. Ejemplo: <code>Serial.begin(9600);</code>
<code>Serial.Read()</code>	Lee los datos entrantes del puerto serie. Ejemplo: <code>Byte=Serial.read();</code>
<code>Serial.print(val,[format])</code>	Imprime los datos al puerto serie como texto ASCII. Val: El valor a imprimir de cualquier tipo Format: Especifica la base (formato) a usar; los valores permitidos son BYTE, BIN (binarios), OCT (octales), DEC (decimales), HEX (hexadecimales). Este parámetro especifica el número de posiciones decimales a usar.

<code>Serial.println(val,[format])</code>	Imprime los datos al puerto serie como texto ASCII. Ejemplo: <code>Serial.println(analogValue);</code> imprime como ASCII decimal <code>Serial.println(analogValue, HEX);</code> imprime como ASCII hexadecimal.
<code>Serial.available()</code>	Devuelve el número de bytes disponibles para ser leídos por el puerto serie. <code>if (Serial.available() &gt; 0) {</code> realiza la lectura del puerto serie <code>}</code>

**Tabla 4** Comunicación Serie

Fuente: Universidad de Cádiz

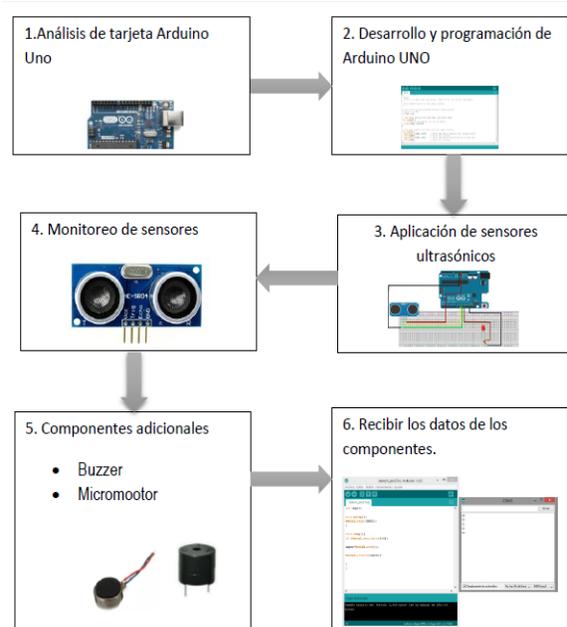
#### Análisis del método de un Bastón inteligente.

Para el desarrollo de este bastón inteligente se utilizó una placa de Arduino uno, software Arduino IDE para desarrollar la programación de la placa, dos sensores ultrasónicos HC04, un micromotor de vibración, un buzzer que emite sonidos, cables macho-hembra, un bastón, una protoboard, y una caja de cartón.

La conexión de los cables macho-hembra de la placa Arduino Uno a la protoboard, el cual tiene como fin más conexiones del propio pin del Arduino a los sensores ultrasónicos, micromotor y buzzer, de la protoboard, conexiones a los sensores ultrasónicos que se localizan en la parte media e inferior del bastón, estos sensores mandan las señales obtenidas de acuerdo a la programación en Arduino IDE, lo cuales detectan los obstáculos de 1.60m de altura y 50cm de distancia, estos mismos sensores mandarán una señal al micromotor localizado en la parte superior del bastón, que de igual manera está conectado en la protoboard, esta señal hace que tenga una vibración cuando los sensores hayan localizado obstáculos, de esta manera el buzzer emitirá un sonido cuando haya obstáculos, teniendo como funcionalidad una mejor advertencia para las personas débiles visuales.

Los materiales mencionados como la placa Arduino Uno, la protoboard, los cables macho-hembra, los sensores ultrasónicos HC04, y el buzzer se encuentran dentro de la caja de cartón que está localizada debajo del micromotor en la parte superior del bastón.

El análisis del método explica claramente el análisis del método de la solución del desarrollo del proyecto bastón inteligente dando a notar los principales puntos para desarrollarlo y dar el buen funcionamiento.



**Figura 8** Secuencia de desarrollo para Bastón Inteligente  
Fuente: *Elaboración Propia*

## Material a utilizar

### Arduino Uno

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteo. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador. (JADIAZ, 2016)

### Protoboard

Es empleada para realizar pruebas de circuitos electrónicos, insertando en ella componentes electrónicos y cables como puente. Es el boceto de un circuito electrónico donde se realizan las pruebas de funcionamiento necesarias antes de trasladarlo sobre un circuito impreso. (TuElectronica, 2016)

### Sensor de proximidad HC-SR04

El sensor HC-SR04 es un módulo que incorpora un par de transductores de ultrasonido que se utilizan de manera conjunta para determinar la distancia del sensor con un objeto colocado enfrente de este. Quizá la característica más destacada del HC-SR04 es que puede ser adquirido por una baja suma de dinero y esto mismo lo ha hecho muy popular. Afortunadamente el módulo HC-SR04 es bastante fácil de utilizar a pesar de su bajo precio y no demanda gran cantidad de trabajo ponerlo a funcionar, mucho menos si utilizamos una librería para sensores ultrasónicos.

### Bastón

Vara que sirve para apoyarse al andar. (Dictionary, 2018)

### Buzzer

Zumbador, buzzer en inglés, es un transductor electroacústico que produce un sonido o zumbido continuo o intermitente de un mismo tono (generalmente agudo). Sirve como mecanismo de señalización o aviso y se utiliza en múltiples sistemas, como en automóviles o en electrodomésticos, incluidos los despertadores. Su construcción consta de dos elementos, un electroimán y una lámina metálica de acero. El zumbador puede ser conectado a circuitos integrados especiales para así lograr distintos tonos. Cuando se acciona, la corriente pasa por la bobina del electroimán y produce un campo magnético variable que hace vibrar la lámina de acero sobre la armadura. (Carrod Electronica, 2018)

### Cable de Arduino

Se usa para conectar Arduino Uno, Arduino Mega 2560, Arduino 101 o cualquier placa con el puerto hembra USB A de su computadora. La longitud del cable es de aproximadamente 178 cm. El color y la forma del cable pueden variar. (Arduino, 2018)

## Resultados

De acuerdo al análisis del método se dan como resultados el bastón inteligente que conectado a una placa Arduino Uno, que obtiene señales de obstáculos localizados por medio de dos sensores ultrasónicos, estos mandan información al micromotor para que mande una vibración de acuerdo a la programación que se realizó en Arduino IDE, y al mismo tiempo el buzzer emita un sonido cuando estén los obstáculos cerca de acuerdo a las medidas que se establecieron en la programación de Arduino IDE.

Dada la programación arduino se expone la siguiente programación, para el funcionamiento del bastón inteligente.

- Se define cada uno de los pines del sensor.
- Se declaran variables para determinar distancia.
- Pines para el buzzer.
- Notas para el sonido
- Pines de entrada y salida de cada sensor.
- Imprime distancia
- Generar el pulso.
- Arroja respuesta el buzzer entre la distancia predeterminada.
- Detecta el vibrador.
- Imprime respuesta de sensor 1

Los pasos anteriores son fundamentales para funcionalidad de la programación y así se tenga el resultado obtenido.

## Conclusión

Al concluir con el modelo de aprendizaje para Arduino Uno, finalizo en cómo se analizó el desarrollo del bastón inteligente ya que se reforzaron habilidades y conocimientos que se obtuvieron durante el periodo de estudios en la Universidad Tecnológica de San Juan del Río.

Se obtuvieron más conocimientos sobre la programación en Arduino IDE, las conexiones que se realizaron de los sensores ultrasónicos y la configuración sobre la altura y distancia máxima que alcanzan estos sensores.

Los materiales Micromotor de vibración y buzzer, son materiales que al ocuparlos en este proyecto tuvimos un mejor desempeño para poder realizar un mejor funcionamiento del bastón inteligente, ya que son necesarios para el desarrollo, y la información obtenida para poderlos programar, de igual forma podemos decir que estos dos materiales ocupados en la conexión con el Arduino son materiales de los cuales no se sabía correctamente su funcionalidad, pero con investigación y pruebas se logró el objetivo.

Para esto fue importante saber para que se utilizó cada componente, cuál es su funcionalidad y como se puede utilizar correctamente cada material.

Para ello la planeación y análisis de investigación de la metodología del buen funcionamiento de Arduino uno fue elaborado correctamente siguiendo paso a paso para llegar a buenos resultados.

## Referencias

(18 de Octubre de 2017). Obtenido de Hetpro: <https://hetpro-store.com/TUTORIALES/que-es-arduino/>

Arduino. (20 de Agosto de 2018). *Cable USB tipo A / B*. Obtenido de Cable USB tipo A / B.: <https://store.arduino.cc/usa/usb-2-0-cable-type-a-b>

Arduino, T. (16 de Mayo de 2016). *Sensor ultrasónico HC-SR04 y Arduino*. Obtenido de Sensor ultrasónico HC-SR04 y Arduino: <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/sensor-ultrasonico-hc-sr04-y-arduino/>

Cádiz, U. d. (Marzo de 2017). *Comenzando con Arduino*. Obtenido de Unidad de Innovación Docente:

[http://www.uca.es/recursos/doc/Unidades/Unidad\\_Innovacion/Innovacion\\_Docente/ANEXOS\\_2011\\_2012/22232441\\_310201212102.pdf](http://www.uca.es/recursos/doc/Unidades/Unidad_Innovacion/Innovacion_Docente/ANEXOS_2011_2012/22232441_310201212102.pdf)

Cádiz, U. d. (s.f.). *Comenzando con Arduino*.

Carrod Electronica. (20 de 09 de 2018). Obtenido de Carrod Electronica: <https://www.carrod.mx/products/buzzer-12-v-12-cm-qsx-1206>

*Dictionary*. (20 de 09 de 2018). Obtenido de Dictionary:

<https://es.thefreedictionary.com/bast%C3%B3n>

*Hetpro*. (18 de Octubre de 2017). Obtenido de <https://hetpro-store.com/TUTORIALES/que-es-arduino/>

*Hetpro*. (18 de Octubre de 2017). Obtenido de ¿Qué es Arduino?: <https://hetpro-store.com/TUTORIALES/que-es-arduino/>

*Hetpro*. (18 de Octubre de 2017). Obtenido de ¿Qué es Arduino?: <https://hetpro-store.com/TUTORIALES/que-es-arduino/>

JADIAZ. (21 de Enero de 2016). *MiArduino*. Obtenido de MiArduino: <http://www.iescamp.es/miarduino/2016/01/21/p-laca-arduino-uno/>

Rubio, M. Á., Mañoso, C., Romero Zaliz, R., & P. de Madrid, Á. (28 de Octubre de 2014). *Universitat Politècnica de Catalunya*. Obtenido de Portal de acceso abierto al conocimiento de la UPC:

<https://upcommons.upc.edu/handle/2099/15503?show=full>

TuElectronica. (4 de Febrero de 2016). *Que es la protoboard (breadboard)*. Obtenido de Que es la protoboard (breadboard): <https://tuelectronica.es/que-es-la-protoboard/>