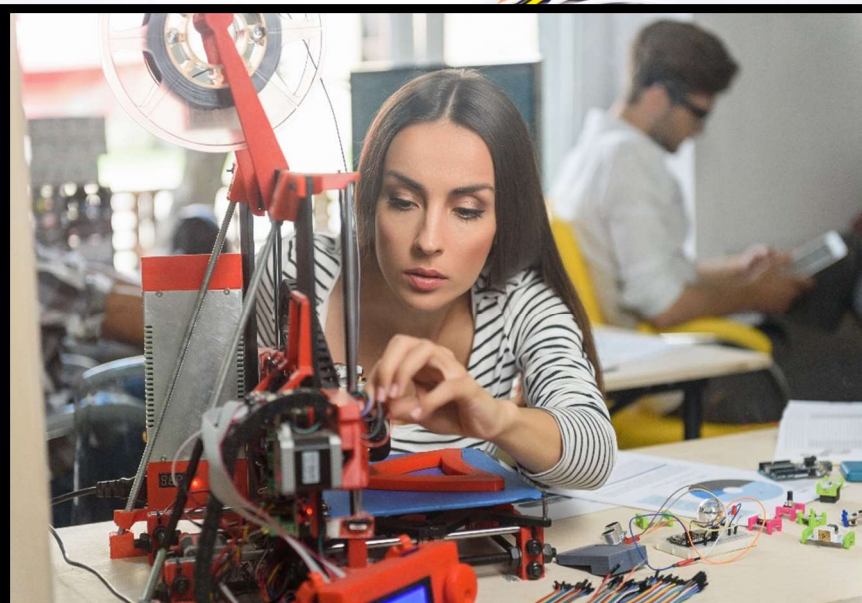
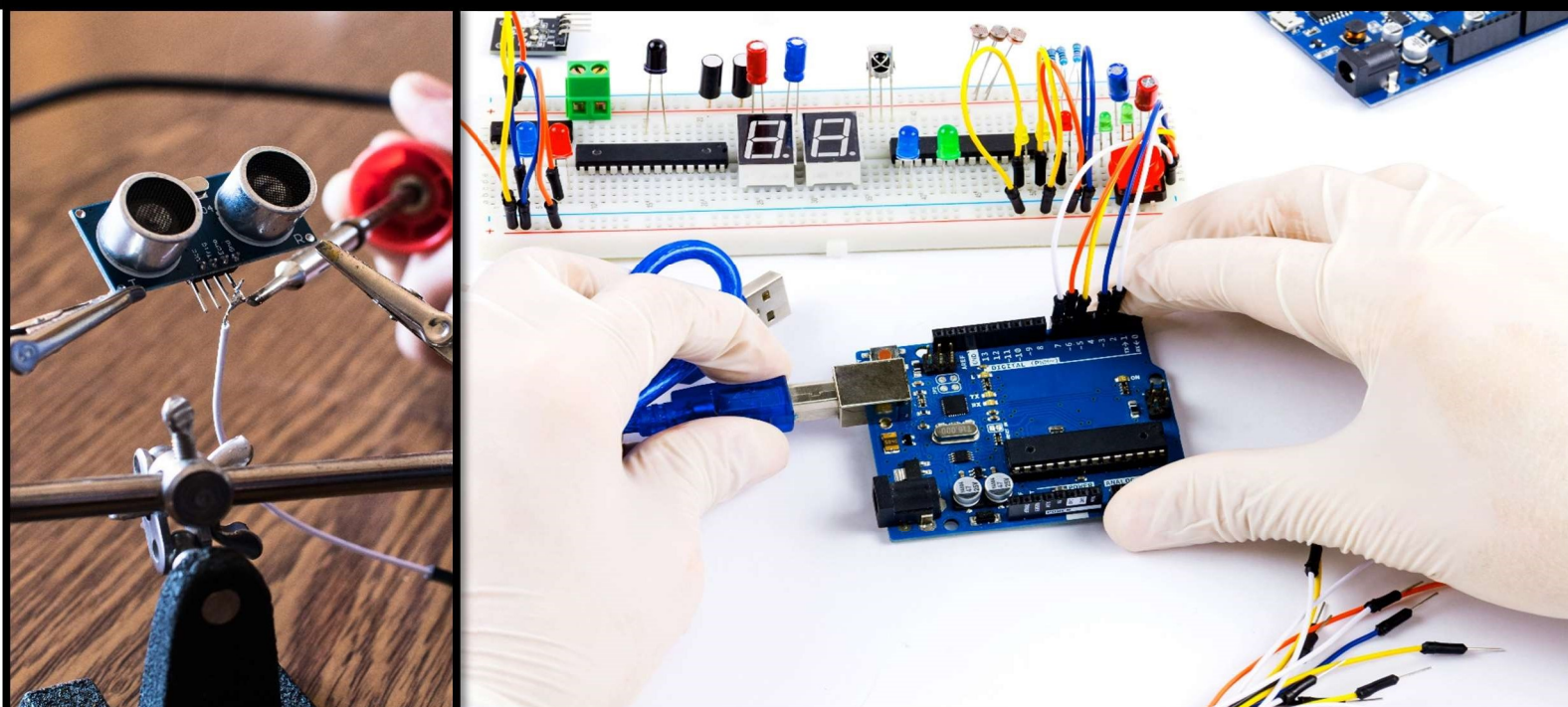


## *Arduino I – Fundamentos*



## Índice

1. Datos generales .....	1
2. Introducción .....	1
2.1. Objetivos .....	1
2.2. Descripción del curso .....	1
3. Fundamentos de Arduino .....	3
3.1. ¿Qué es Arduino? .....	3
3.2. ¿Por qué es tan popular? .....	3
3.3. ¿Qué tipo de proyectos se pueden hacer? .....	4
3.4. Arduino UNO - Características .....	6
3.5. Arduino NANO - Características .....	6
4. Electrónica básica .....	7
4.1. La Ley de Ohm .....	7
4.2. La Protoboard .....	9
4.3. El LED .....	9
4.4. Circuito 1: Mi primer LED .....	11
4.5. Pulsadores .....	11
4.6. Circuito 2: LED con pulsador .....	12
4.7. Señales analógicas y digitales .....	12
5. Software e instalación .....	13
5.1. Descripción de mBlock .....	14
5.2. Instalación de mBlock .....	15
5.3. Descripción de Arduino IDE .....	15
5.4. Instalación Arduino IDE .....	17
6. Programación .....	17
6.1. Estructura de programa mBlock .....	17
6.2. Sentencias básicas mBlock .....	18
6.3. Estructura de programa Arduino IDE .....	18
6.4. Sentencias básicas Arduino IDE .....	19

6.5.	Circuito 3: LED programado con mBlock y Arduino IDE.....	20
6.6.	Circuito 4: LED + Pull-Down programado con mBlock y Arduino IDE .....	20
6.7.	Circuito 5: LED + Pull-Up programado con mBlock y Arduino IDE .....	21
7.	Proyecto: Entrenador de Arduino V1.0 .....	21
7.1.	Descripción del producto .....	21
7.2.	Lista de materiales .....	22
7.3.	El MAX7219 .....	24
7.4.	Esquema electrónico.....	26
7.5.	Extensión del MAX7219 para mBlock .....	26
7.6.	Librería del MAX7219 para Arduino IDE .....	27
7.7.	Montaje del Entrenador.....	28
7.8.	Programación del Entrenador .....	28

## 1. Datos generales

- Título: ARDUINO UNIDAD 1: FUNDAMENTOS
- Proyecto: ENTRENADOR DE ARDUINO V1.0
- Organiza: DIGITAL CODESIGN – KREITEK – COLEGIO NURYANA
- Fecha: 02/09/19 – 04/09/19
- Horario: 17:00 – 19:30
- Horas lectivas: 5
- Plazas: 20
- Lugar: MAKERSPACE KREITEK – COLEGIO NURYANA
- Ponentes: MIQUEAS FORTES GONZÁLEZ  
DANIEL ÁLVAREZ EUSEBIO
- Categoría: ELECTRÓNICA Y PROGRAMACIÓN

## 2. Introducción

### 2.1. Objetivos

El presente curso tiene por objeto el fomento de las vocaciones científicas, la creatividad y la actitud emprendedora entre alumnos, profesores y emprendedores, promoviendo un nuevo modelo de trabajo y aprendizaje basado en proyectos, también denominado Educación STEAM (Ciencia, Tecnología, Ingeniería, Arte y Matemáticas) o Educación 4.0, caracterizado por el empleo de las TIC (Tecnologías de la Información y la Comunicación) como recurso para la formación y adquisición de nuevas competencias tecnológicas.

### 2.2. Descripción del curso

Curso de Arduino Unidad 1: Fundamentos e Iluminación, es el primero de una serie de cursos de electrónica, robótica y programación con Arduino, una placa de desarrollo de hardware y software libre muy popular por su facilidad de uso y gran versatilidad, gracias a la cual resulta muy sencillo desarrollar infinidad de proyectos, tanto de propósito educativo como profesional, convirtiendo en fácil, accesible y barato, lo que antes era caro y complejo.

En esta primera unidad veremos los primeros conceptos de electrónica y programación, partiendo de la teoría de circuitos fundamental y los componentes electrónicos más sencillos, para luego centrar la atención en la placa Arduino, lo que implica conocer su arquitectura básica y sus puertos de entrada y salida.

Después de haber visto lo esencial del hardware pasaremos a estudiar, instalar, configurar y manejar el software necesario para programar la placa Arduino. Con el fin de aportar recursos suficientes a los asistentes, recurriremos a dos tipos de programas



diferentes para la programación de la placa en todos los casos prácticos del curso, mBlock (Scratch), un lenguaje de programación visual diseñado para acercar la lógica computacional a los alumnos de primaria y primeros ciclos de secundaria, y por otro lado el IDE de Arduino, un tipo de lenguaje derivado del C++ de medio-alto nivel, ideal para alumnos de los últimos ciclos de secundaria y bachillerato, así como para profesionales y emprendedores.

Empleando ambos tipos de programa estudiaremos programación básica de una forma totalmente práctica, proponiendo diversas situaciones que permitan a los participantes conocer la estructura del código, así como las sentencias elementales y no tan elementales de los dos lenguajes de programación utilizados.

Finalmente y como proyecto del curso, los participantes tendrán la oportunidad de fabricar un Entrenador de Arduino V1.0, un dispositivo compacto con buen acabado, fabricado a partir de un circuito electrónico que los participantes deberán conectar y programar, así como de cuatro piezas que conforman su carcasa, diseñada a medida para el producto e impresa previamente en 3D.



El Entrenador de Arduino V1.0 está pensado para permitir al usuario adquirir los conocimientos básicos de electrónica y programación con Arduino durante la fase de montaje del kit y primera programación, así como para entrenar dichos conocimientos y profundizar en la materia con el producto una vez terminado, ya que éste permite ser reprogramado para realizar múltiples funciones.

Una vez montado, el Entrenador de Arduino V1.0 básicamente consta de una pantalla LED que podemos programar mediante Arduino para mostrar cualquier tipo de mensaje

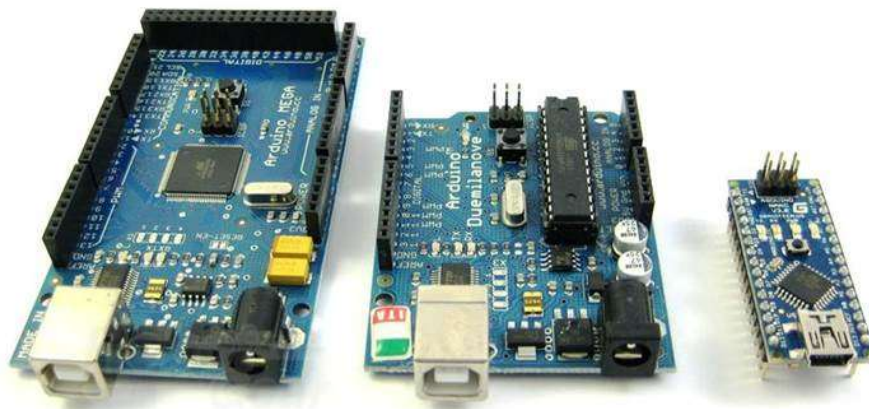
(números, texto, símbolos, dibujos sencillos), que guarde un significado o una función específica para un proyecto o aplicación determinada. En definitiva, un producto compacto que nos permite programar mensajes.

Los recursos que utilizaremos para esta primera unidad serán, en primer lugar, el aula virtual que hemos desarrollado específicamente para este, que permitirá a los participantes y al docente llevar una progresión ordenada y dinámica de los contenidos del curso. Por otro lado, nos será de utilidad la presente guía didáctica en la que se resume el contenido teórico y práctico, así como por supuesto los materiales que emplearemos para la fabricación de los prototipos.

### 3. Fundamentos de Arduino

#### 3.1. ¿Qué es Arduino?

Arduino es una marca de placas de desarrollo Open Source y Open Hardware. La primera placa de Arduino fue desarrollada en el año 2005 en el Instituto IVREA (IDII), en Ivrea (Italia), con el propósito de crear una herramienta de bajo costo y fácil de usar, que permitiera a alumnos y personas sin altos conocimientos técnicos desarrollar todo tipo de proyectos de electrónica, robótica y programación.



#### 3.2. ¿Por qué es tan popular?

Arduino es una de las tecnologías que más ha hecho posible el acercamiento de ciertas materias complejas como la electrónica y la programación de microcontroladores a personas de todas las edades y sin conocimientos específicos, de forma que lo que antes solo podía estar en mano de ingenieros, investigadores y en general de empresas y profesionales cualificados, ahora ha pasado a ser de dominio público.



Las características más relevantes de Arduino que han sido causa de su alta popularidad e influencia, se resumen en los siguientes atributos:

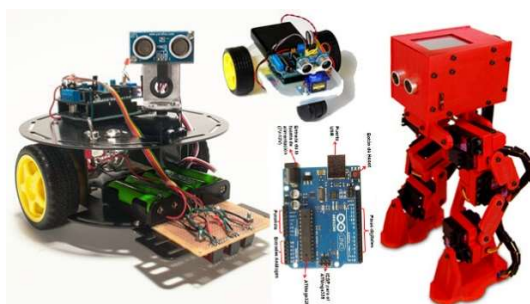
1. FÁCIL DE APRENDER
2. FÁCIL DE USAR
3. OPEN-SOURCE
4. COMPATIBLE CON MUCHAS TECNOLOGÍAS
5. COMUNIDAD INMENSA
6. POSIBILIDADES INFINITAS
7. DIVERTIDO
8. MUY BARATO

Por todo esto, Arduino ha logrado atraer una gran avalancha de usuarios interesados por el desarrollo de proyectos y la creación de tecnología: aficionados, makers, profesores, jóvenes, niños, e incluso empresas y profesionales que han visto en Arduino y otras tecnologías paralelas, la oportunidad de desarrollar nuevos productos de forma mucho más rápida y económica.

### 3.3. ¿Qué tipo de proyectos se pueden hacer?

Tal vez sea mucho más rápido explicar lo que no se puede hacer que lo que sí se puede hacer con Arduino, dado que las posibilidades de esta placa de desarrollo son literalmente infinitas, especialmente teniendo en cuenta la infinidad de dispositivos y módulos electrónicos con los que Arduino es compatible y con los que podemos desarrollar proyecto para aplicaciones muy diversas.

No obstante y para comprender mejor los usos y aplicaciones de esta herramienta, de manera muy resumida podríamos agrupar los diferentes tipos de proyectos más populares de Arduino en las siguientes temáticas:



**ROBÓTICA MÓVIL**



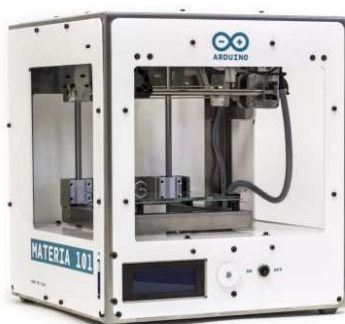
**ROBÓTICA INDUSTRIAL**



**DOMÓTICA**



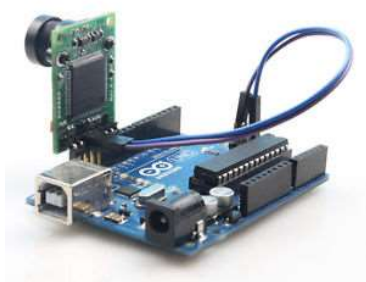
**DRONES**



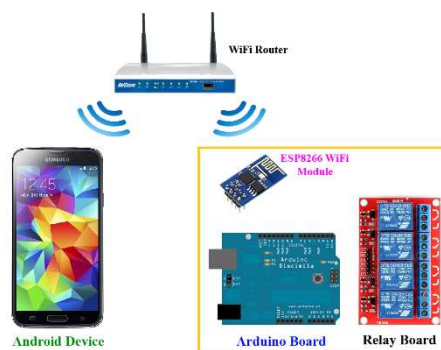
**FABRICACIÓN DIGITAL**



**ROPA INTELIGENTE**



**IMAGEN Y SONIDO**

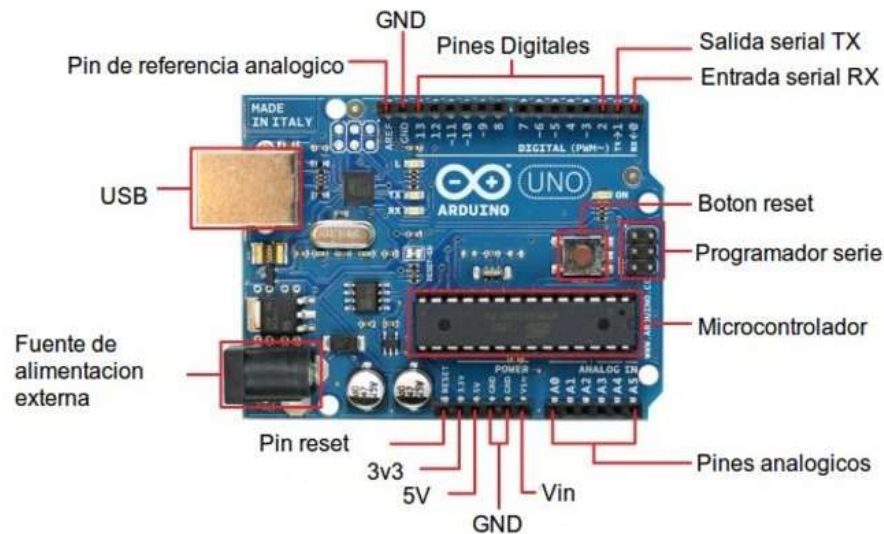


**COMUNICACIONES**



### 3.4. Arduino UNO - Características

Arduino UNO es el modelo estándar de placas Arduino, normalmente la más utilizada cuando se comienza a trabajar en los primeros proyectos. En la siguiente imagen se muestra una placa Arduino UNO destacando sus principales componentes:

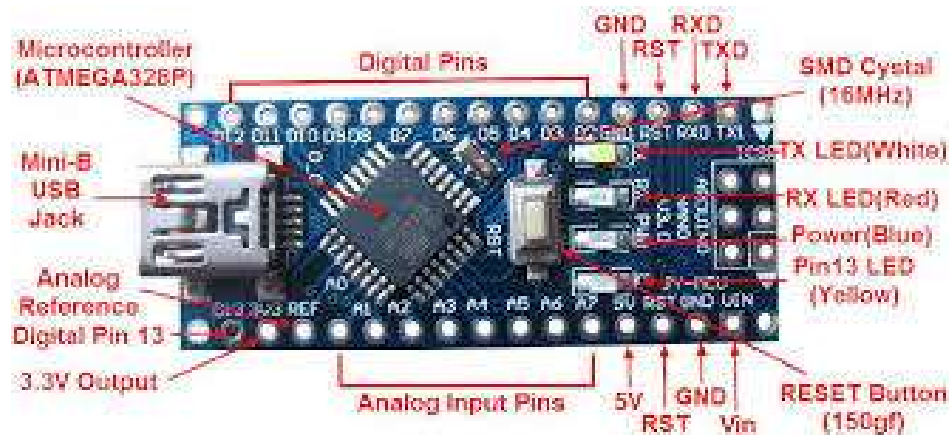


Las principales características del Arduino UNO respecto a otras versiones de Arduino son las siguientes:

- Microcontrolador ATmega328
- Voltaje de entrada 7-12 V
- Tensión de funcionamiento 5 V
- 14 pines digitales de I/O (6 salidas PWM)
- 6 entradas análogas
- 32k de memoria Flash
- Reloj de 16MHz de velocidad
- Dimensiones 68,6 x 53,4 mm

### 3.5. Arduino NANO - Características

Otra versión de Arduino igual de sencillo de manejar que Arduino UNO es el Arduino NANO:



Las principales características del Arduino NANO respecto a otras versiones de Arduino son las siguientes:

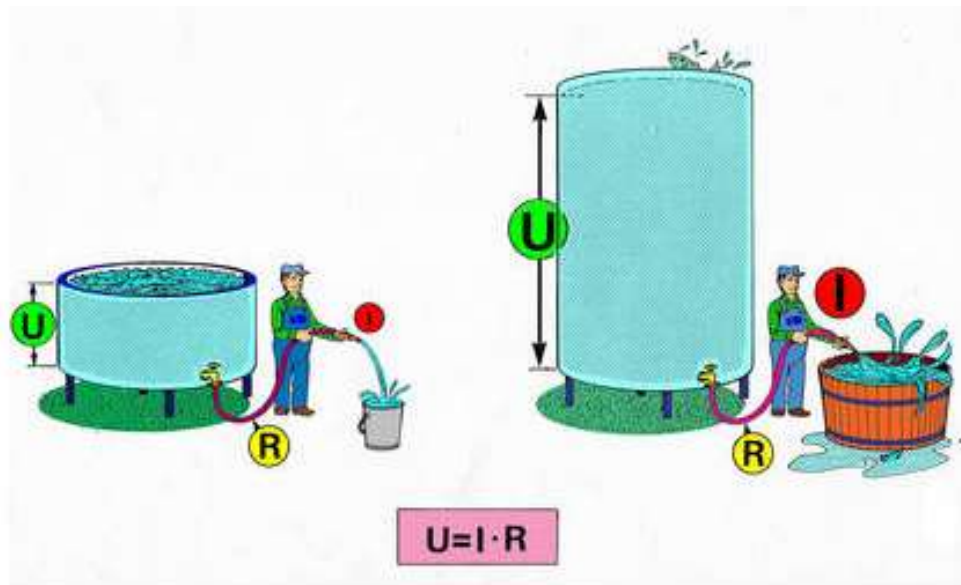
- Microcontrolador ATmega328
- Voltaje de entrada 7-12 V
- Tensión de funcionamiento 5 V
- 14 pines digitales de I/O (6 salidas PWM)
- 8 entradas análogas
- 32k de memoria Flash
- Reloj de 16MHz de velocidad
- Dimensiones 18,5 x 43,2 mm

## 4. Electrónica básica

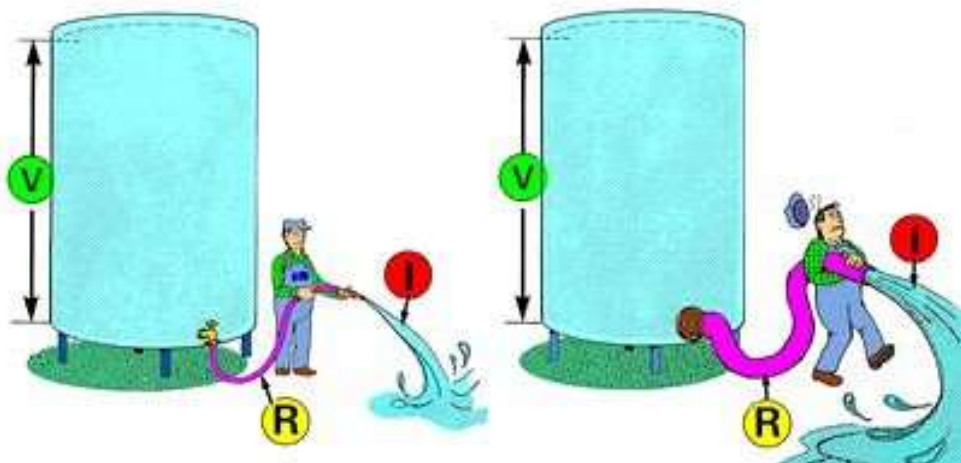
### 4.1. La Ley de Ohm

En electrónica y electricidad existe una ley fundamental, la **Ley de Ohm**, que relaciona las tres magnitudes básicas más importantes de un circuito: **Tensión (Voltaje)**, **Corriente (Intensidad)** y **Resistencia**.

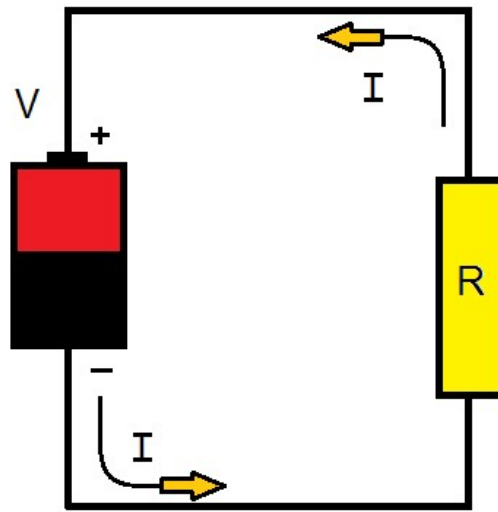
Tratar de comprender directamente el significado de estas magnitudes y su relación puede resultar un poco complicado, pero es bastante sencillo de entender si se hace una analogía entre un circuito eléctrico y un circuito de agua.



- **Tensión (Voltaje):** La altura del tanque que aporta mayor o menor presión de agua, equivale a la tensión o voltaje que aporta una pila o fuente de alimentación en un circuito.
- **Corriente (Intensidad):** El flujo de agua a través de la manguera equivale a la corriente eléctrica que circula a través de los cables.
- **Resistencia:** La resistencia eléctrica que regula el paso de la corriente equivale al diámetro de la tubería que permite que pase mayor o menor caudal de agua.

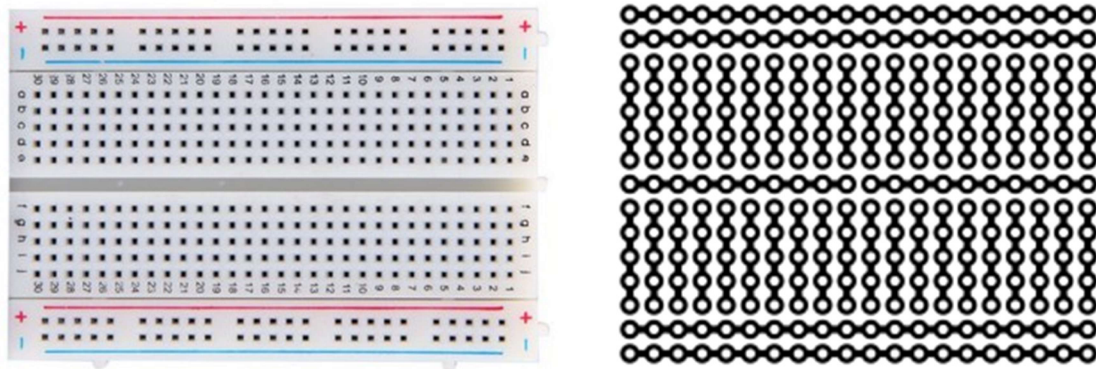


El circuito eléctrico más sencillo que se puede realizar es el que se muestra a continuación, y equivale directamente a la analogía con el agua explicada en los esquemas anteriores:



#### 4.2. La Protoboard

Una placa de pruebas o protoboard es un tablero con múltiples agujeros conectados entre sí, preparado para insertar y conectar componentes electrónicos y cables sin necesidad de soldar, con el fin de construir y prototipar circuitos electrónicos de forma rápida y sencilla.



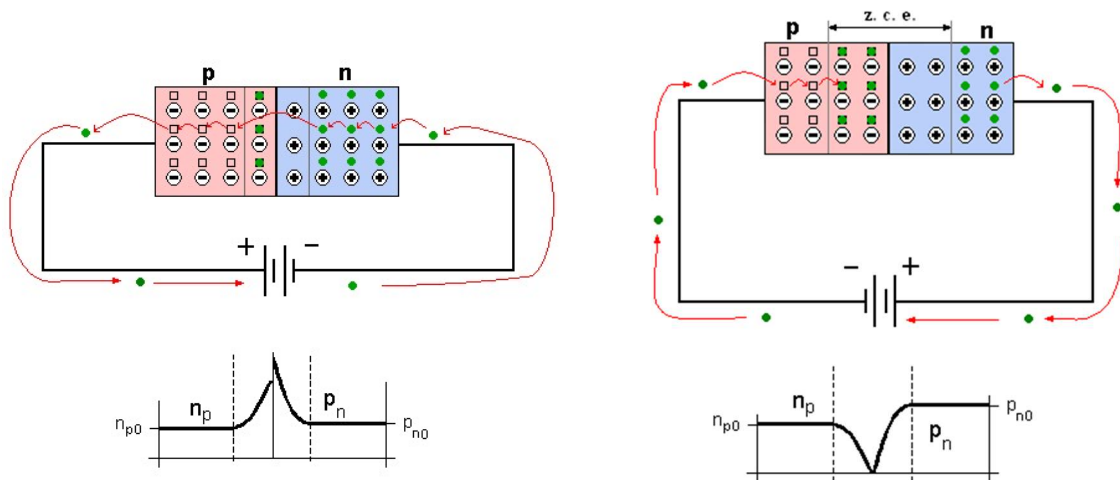
De las dos imágenes anteriores, la de la izquierda muestra el aspecto de una protoboard, mientras que la de la derecha muestra el patrón típico de disposición de las láminas de cobre interiores que conectan los agujeros de la protoboard.

#### 4.3. El LED

Un LED (Diodo Emisor de Luz) es un tipo de diodo con la particular característica de que es capaz de emitir luz. Un diodo es un componente electrónico que permite la

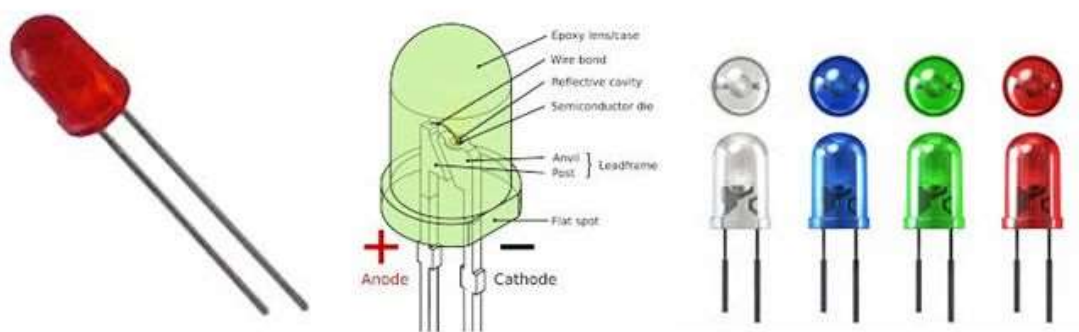


circulación de corriente en un solo sentido. Está formado internamente por un material semiconductor con unión tipo p-n, por tanto cuando se polariza en directa el diodo conduce electricidad y cuando se polariza en inversa bloquea el paso de corriente. Esta manera de funcionar del diodo hace que en cierta forma se parezca a un interruptor controlado por tensión.



En el caso particular del LED, al conducir electricidad el material semiconductor libera energía en forma de fotones, lo que hace que se ilumine.

**Veamos ejemplos reales de diodos led:**



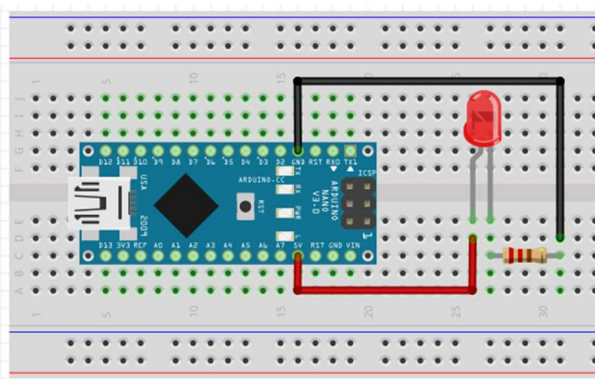
**Veamos el símbolo de diodo led:**



**Nota importante:** Cuando un LED conduce electricidad ofrece muy poca resistencia al paso de corriente, por lo que la fuente de alimentación va a suministrar toda la corriente posible de forma descontrolada, lo que equivale a un cortocircuito que como poco supondrá un alto riesgo de quemar el LED. Para evitar esto, siempre debemos conectar una resistencia de valor adecuado (220 ohm normalmente) en serie con el LED. Así controlamos el paso de corriente y evitamos que nada se rompa.

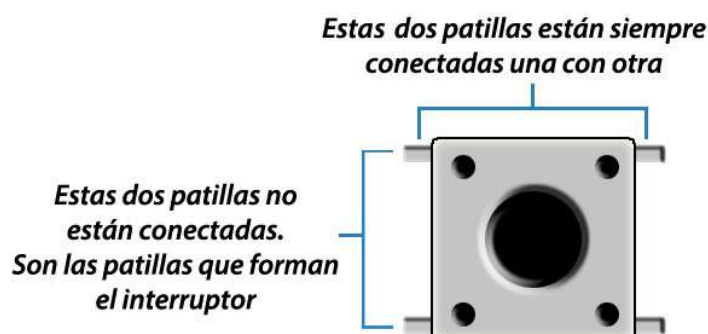
#### 4.4. Circuito 1: Mi primer LED

Podrás acceder a esta actividad desde el aula virtual, dónde encontrarás publicadas las instrucciones del ejercicio, el esquema electrónico del circuito que habrás de montar, las preguntas finales, así como cualquier otro recurso o material asociado a la actividad.



#### 4.5. Pulsadores

Un botón o pulsador es un componente básico en electrónica, que sirve para permitir o interrumpir de forma mecánica el paso de la corriente en un circuito cuando se acciona de forma manual, con un dedo por lo general.

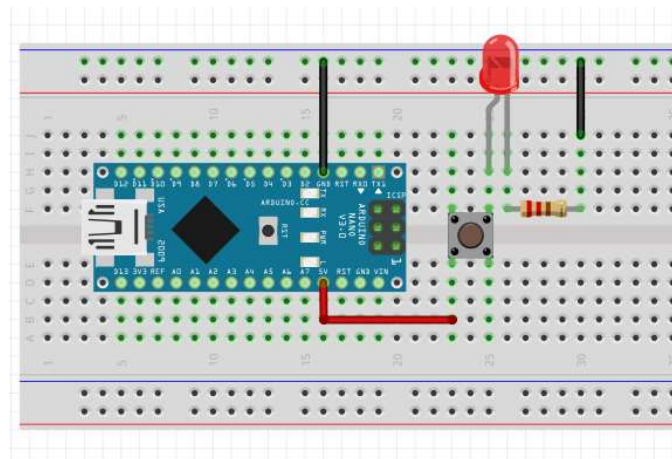


A diferencia del interruptor, el botón o pulsador cuenta con una única posición de reposo, lo que significa que cuando se deja de accionar el contacto vuelve por sí solo a su posición original.

En función de cuál sea su posición de reposo existen dos tipos de pulsadores, con contacto normalmente abierto (NA) o con contacto normalmente cerrado (NC).

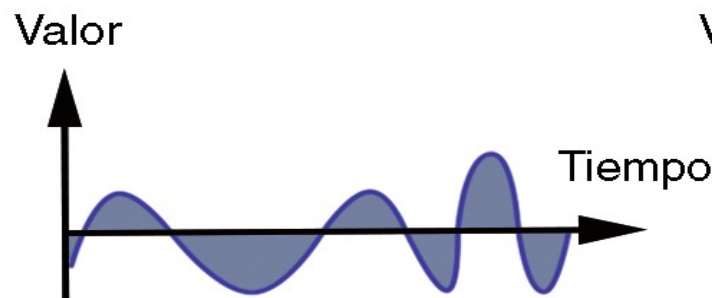
#### 4.6. Circuito 2: LED con pulsador

Podrás acceder a esta actividad desde el aula virtual, dónde encontrarás publicadas las instrucciones del ejercicio, el esquema electrónico del circuito que habrás de montar, las preguntas finales, así como cualquier otro recurso o material asociado a la actividad.



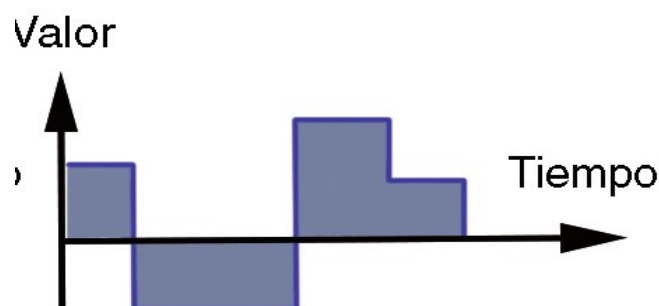
#### 4.7. Señales analógicas y digitales

- Señal analógica: Es aquella que presenta una variación continua en el tiempo por lo que puede tomar infinitos valores en un intervalo de tiempo dado. Este tipo de señales son las que podemos encontrar de forma natural en el medio físico, como por ejemplo el sonido, variaciones en la temperatura, etc...



a. Señal analógica

- Señal digital: Es aquella que presenta una variación discontinua en el tiempo y que solo puede tomar valores discretos, 0-1 en el caso de señales digitales de un bit, 0-1-2-3 en el caso de señales de dos bits, etc... Estas señales no se encuentran en el mundo físico de forma natural, sino en electrónica y sistemas de comunicaciones digitales creados por el hombre.



b. Señal digital

## 5. Software e instalación

Como ya sabemos, Arduino es una placa de desarrollo que dispone de un microcontrolador en cuya memoria podemos cargar un programa definido por el usuario, que permitirá al Arduino realizar una tarea o función específica, comunicándose con otros dispositivos conectados a sus entradas y salidas.

Para programar una placa Arduino podemos utilizar diferentes lenguajes y entornos de programación que podemos clasificar en dos categorías:

### 1) Lenguajes de programación visual o por bloques:



- Scratch 4 Arduino
- Ardublock
- Visualino
- mBlock

## 2) Lenguajes de programación textual:

- Arduino IDE

Los lenguajes de programación por bloque se utilizan sobre todo en el ámbito educativo para que niños y jóvenes puedan dar sus primeros pasos en diseño de videojuegos y programación de microcontroladores, dado que este tipo de lenguajes facilita mucho la comprensión de la lógica computacional, así como a aprender a estructurar algoritmos secuenciales.

El objetivo de los lenguajes de programación por bloques es facilitar por tanto el acceso a los lenguajes de programación textual, los cuales a diferencia de los primeros trabajan directamente con líneas de texto, siendo por tanto un poco más difíciles de comprender al principio, pero que sin duda terminan siendo la forma más óptima para programar.

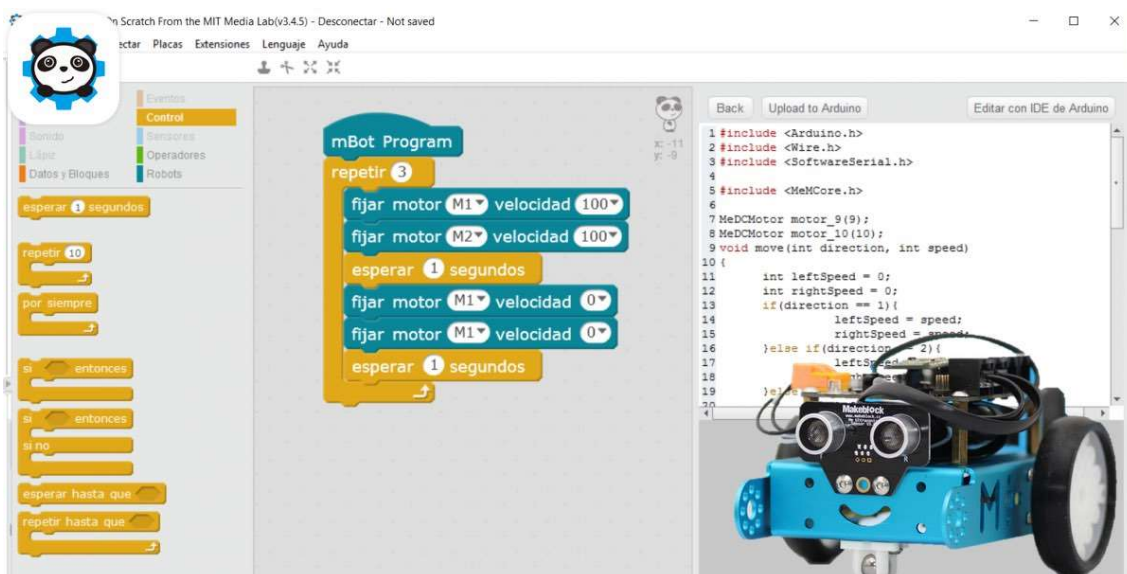
### 5.1. Descripción de mBlock

El software mBlock es un entorno gráfico de programación basado Scratch 3.0, la versión más actualizada de Scratch. Si bien el sistema de Scratch y mBlock son muy similares, el primero está pensado para diseño de animaciones y videojuegos, mientras que mBlock es una adaptación Scratch orientada a la programación de placas de desarrollo Arduino para proyectos de electrónica y robótica.

Las principales ventajas de mBlock respecto a otros programas utilizados para programar Arduino mediante lenguaje de programación por bloques, son las siguientes:

- Está basado en la versión más actual de Scratch (Scratch 3.0), mientras que por ejemplo Scratch 4 Arduino (S4A) está basado en Scratch 1.4.
- El Arduino no necesita estar conectado al PC para poder ejecutar el código cargado.
- Traducción en tiempo real del lenguaje por bloques a lenguaje Arduino.

La siguiente imagen muestra cómo es el entorno de trabajo de mBlock:



Los distintas partes que podemos distinguir de la interfaz de mBlock son las siguientes:

- **Menú superior:** para gestionar el proyecto, seleccionar el modo de trabajo, seleccionar y establecer la comunicación con la placa de desarrollo, gestionar las librerías (extensiones) y seleccionar el lenguaje.
- **Programas:** Tipos de programas disponibles. Cada programa dispone de un conjunto de instrucciones específico.
- **Instrucciones:** Conjunto de instrucciones propio del programa seleccionado.
- **Editor de programación:** Espacio en el que se van ubicando las instrucciones para ir conformando el programa.
- **Visualizador de código Arduino IDE:** Espacio en el que mBlock va traduciendo nuestro programa en bloques a código textual Arduino IDE para comprobar su equivalencia.
- **Monitor serie:** Ventana que nos permite visualizar los datos recibidos y enviados entre el Arduino y el PC.

## 5.2. Instalación de mBlock

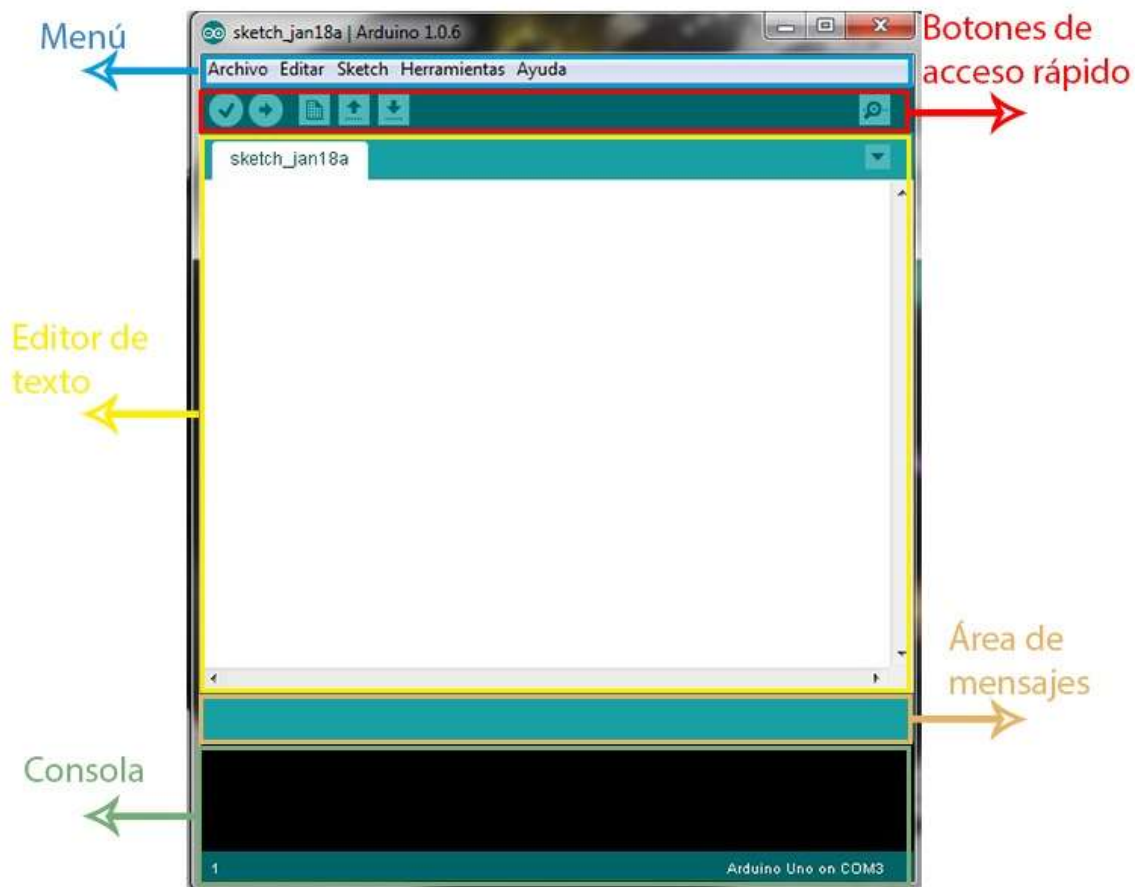
Dirígete al aula virtual para realizar la siguiente actividad que consiste en la descarga e instalación del Software mBlock en tu ordenador. Allí encontrarás las instrucciones al respecto: link de descarga, pasos a seguir y preguntas asociadas a la actividad.

## 5.3. Descripción de Arduino IDE



El software Arduino IDE es un entorno de desarrollo integrado, que como ya hemos adelantado nos permite programar placas Arduino empleando un tipo de lenguaje de programación textual. Las funciones que dispone este software son las siguientes:

- **Editor de texto:** Entorno en el que se desarrolla el código.
- **Compilador:** Traduce el código desarrollado con lenguaje de alto nivel a lenguaje máquina.
- **Depurador:** Detección de errores de sintaxis en el código durante la compilación.
- **Subida de código:** Cargar el código compilado en la memoria flash del microcontrolador del Arduino.
- **Monitor Serie:** Arduino se comunica con el ordenador por puerto serie mediante USB. El monitor serie es una ventana que nos permite visualizar los datos recibidos y enviados entre el Arduino y el ordenador.

El entorno de trabajo del software IDE es el siguiente:



Las herramientas más importantes del software IDE que aprenderemos a manejar son las siguientes:

Símbolo	Descripción
	Crear nuevo proyecto
	Abrir un proyecto
	Guardar proyecto
	Compilar y depurar código
	Cargar programa en la placa de Arduino tras compilar
	Abrir la ventana del monitor serie
-	Selección de placa de desarrollo
-	Gestión de librerías
-	Comentar / Descomentar código
-	Aumentar / Disminuir sangría en código



#### 5.4. Instalación Arduino IDE

Dirígete al aula virtual para realizar la siguiente actividad que consiste en la instalación del Software Arduino IDE en tu ordenador. Allí encontrarás las instrucciones al respecto: link de descarga, pasos a seguir y preguntas asociadas a la actividad.

## 6. Programación

### 6.1. Estructura de programa mBlock

En modo Arduino, mBlock dispone de cuatro categorías de programas:





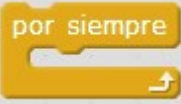

	<b>Control</b>	Instrucciones de tiempo, condicionales y bucles.
	<b>Datos</b>	Creación de variables y bloques



	<b>Operadores</b>	Operadores matemáticos, operadores lógicos, comparadores, operadores de cadenas de texto y generación aleatoria de números.
	<b>Robots</b>	Instrucciones diversas para manejo de placas Arduino y librerías de módulos electrónicos específicos.

## 6.2. Sentencias básicas mBlock

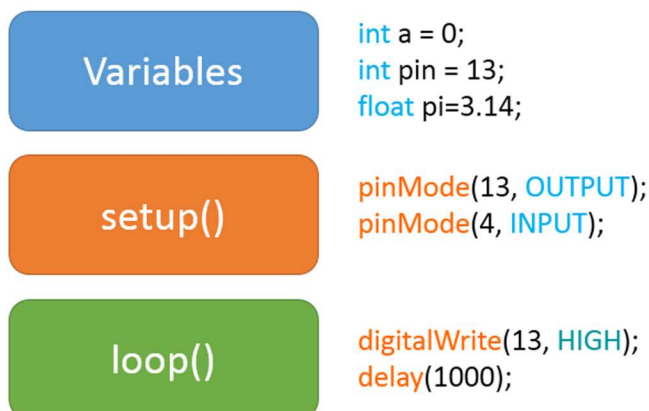
La siguiente tabla muestra las sentencias más elementales de mBlock, que necesitaremos para desarrollar los programas de ejemplo que veremos más adelante:

Sentencia	Descripción
	Inicio de programa
	Declaración de variables y asignación de valores
	Escribe en la salida digital especificada un valor lógico de estado (ALTO) o bajo (BAJO)
	Una función de tiempo. Sirve para pausar la ejecución del programa el tiempo especificado (segundos)
	Bucle infinito. “Por siempre” se ejecutarán las sentencias encerradas dentro del bucle
	Un tipo de control de flujo que ofrece dos posibles caminos a tomar por el programa en función de si se cumplen o no las condiciones especificadas

## 6.3. Estructura de programa Arduino IDE

Como cualquier lenguaje de programación, Arduino IDE tiene su propia estructura básica que es necesario entender y respetar para evitar errores derivados de una falta

de orden o conocimiento sobre las características de las diferentes partes que conforman dicha estructura, y que podemos resumir en las tres siguientes:



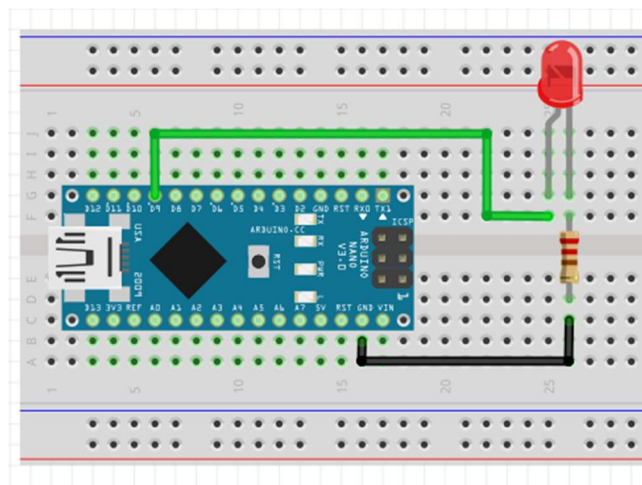
#### 6.4. Sentencias básicas Arduino IDE

La siguiente tabla muestra las sentencias más elementales de Arduino IDE, que necesitaremos para desarrollar los programas de ejemplo que veremos más adelante:

Sentencia	Descripción
<b>int / const int</b>	Declaración de variables y constantes tipo entero
<b>pinMode()</b>	Configura el pin especificado para comportarse como entrada o como salida
<b>digitalWrite()</b>	Escribe en la salida digital especificada un valor lógico de estado alto (HIGH) o bajo (LOW)
<b>digitalRead()</b>	Lee el valor de la entrada digital especificada. Valor lógico HIGH si entrada 5V o LOW si entrada 0V
<b>delay()</b>	Una función de tiempo. Sirve para pausar la ejecución del programa el tiempo especificado (milisegundos)
<b>if / else if / else</b>	Un tipo de control de flujo que ofrece dos o más posibles caminos a tomar por el programa en función de si se cumplen o no las condiciones especificadas.

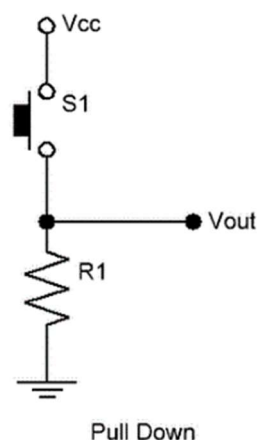
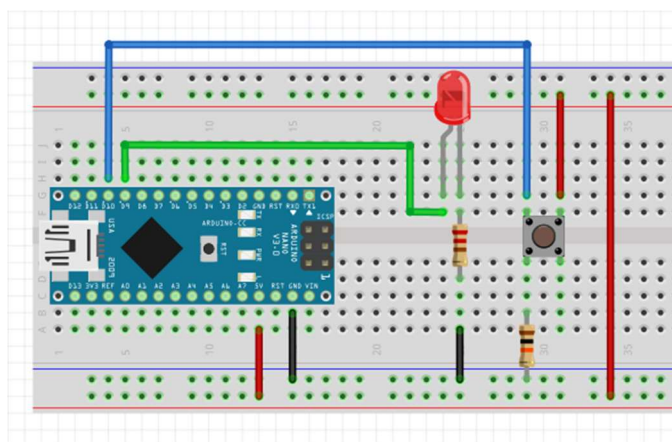
### 6.5. Circuito 3: LED programado con mBlock y Arduino IDE

Dirígete al aula virtual para acceder a las instrucciones del siguiente ejercicio práctico, CIRCUITO 3, que consistirá en realizar un programa básico desde mBlock por un lado y Arduino IDE por otro lado, que logre encender y apagar un LED según el intervalo de tiempo programado, conectado a una de las salidas digitales de la placa Arduino, tal y como se muestra en el siguiente circuito:



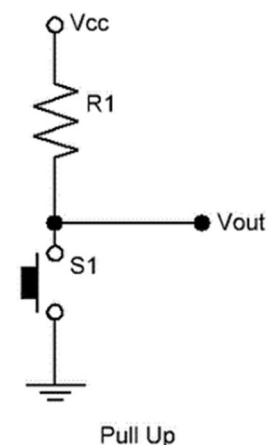
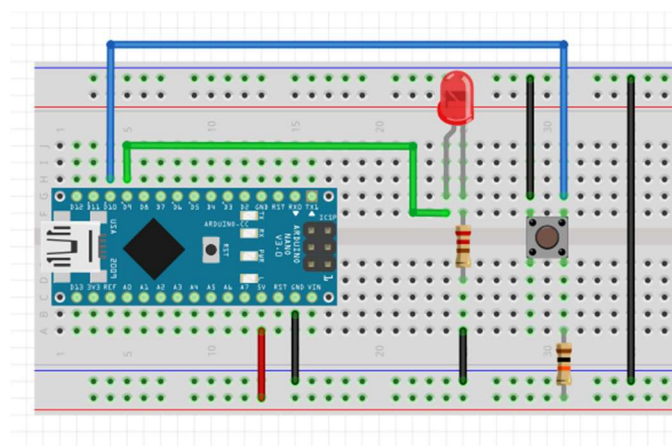
### 6.6. Circuito 4: LED + Pull-Down programado con mBlock y Arduino IDE

Dirígete al aula virtual para acceder a las instrucciones del siguiente ejercicio práctico, CIRCUITO 4, que consiste en añadir al CIRCUITO 3 un pulsador en modo Pull-Down, conectado a uno de los pines digitales del Arduino configurado como entrada. Realiza el programa de forma que al mantener pulsado el botón, el LED se encienda y apague de forma continua a intervalos de tiempo fijo, y que al dejar de pulsar, el LED se mantenga apagado:



## 6.7. Circuito 5: LED + Pull-Up programado con mBlock y Arduino IDE

Dirígete al aula virtual para acceder a las instrucciones del siguiente ejercicio práctico, CIRCUITO 5, que consiste en lo mismo que el CIRCUITO 4, pero conectando el pulsador en modo Pull-Up en lugar de en modo Pull-Down:



## 7. Proyecto: Entrenador de Arduino V1.0

### 7.1. Descripción del producto

El Entrenador de Arduino V1.0 es un producto maker desarrollado para fines educativos dentro del itinerario STEAM, con el propósito de servir como proyecto de iniciación a electrónica y programación con Arduino y otras tecnologías vinculadas al universo open-source.



El producto se ofrece en kit, listo para que el usuario pueda montarlo y posteriormente programarlo siguiendo el programa del curso asociado. El kit incluye una carcasa en cuatro piezas diseñadas a medida del producto e impresas en 3D, lo que le da un acabado robusto, compacto y estético.

El Entrenador de Arduino V1.0 está pensado para permitir al usuario adquirir los conocimientos básicos de electrónica y programación con Arduino durante la fase de montaje del kit y primera programación, así como para entrenar dichos conocimientos y profundizar en la materia con el producto una vez terminado, ya que éste permite ser reprogramado para realizar múltiples funciones.


Una vez montado, el Entrenador de Arduino V1.0 básicamente consta de una pantalla formada por dos matrices LED 8x8 que podemos programar mediante Arduino para mostrar cualquier tipo de mensaje (números, texto, dibujos sencillos), que guarde un significado o una función específica para un proyecto o aplicación determinada. En definitiva, un producto compacto que nos permite programar mensajes.

En resumen, el Entrenador de Arduino V1.0 es un producto diseñado para fines educativos con las siguientes características:

- Dispone de una pantalla LED programable a través de Arduino Nano, para mostrar cualquier tipo de mensaje mediante texto, numeración, símbolos o dibujos sencillos.
- Diseño compacto y resistente mediante carcasa diseñada a medida e impresa en 3D.
- Fácil acceso al conector USB del Arduino para ser reprogramado de forma rápida y sencilla, sin tener que desmontar el producto.
- Alimentado por pila de 9V en su interior, lo que le permite ser un producto portátil y no depender de tener que estar permanentemente conectado a un PC.
- Interruptor de encendido y apagado externo.

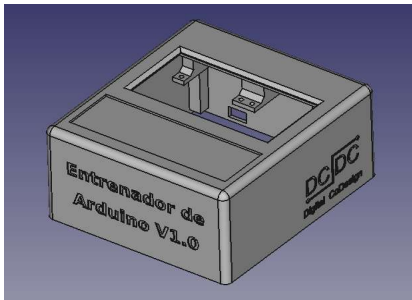
## 7.2. Lista de materiales

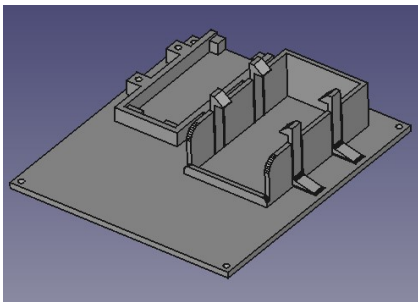
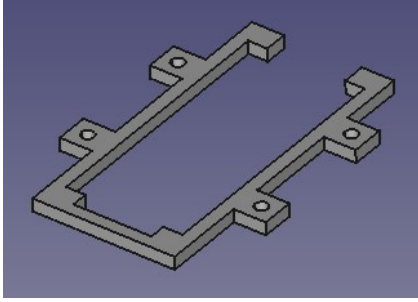

En la siguiente lista se muestra el total de componentes electrónicos necesarios para fabricar el circuito electrónico del Entrenador de Arduino V1.0:

	<p>Dos matrices LED de 8x8, con chip MAX7219, conectadas en serie</p>
---	---

	Arduino Nano
	Cables Dupont hembra-hembra
	Pila 9V con su portapilas
	Interruptor
	Kit de tornillos: - 4 x Tornillo M2 x 10 mm - 12 x Tornillo M2 x 6 mm - 2 x Tornillo M2 x 3 mm

Por otro lado tenemos también el Kit de piezas impresas en 3D que junto con el listado anterior conforman el total de materiales necesarios para fabricar un Entrenador Arduino V1.0:

	Carcasa parte superior
---	------------------------

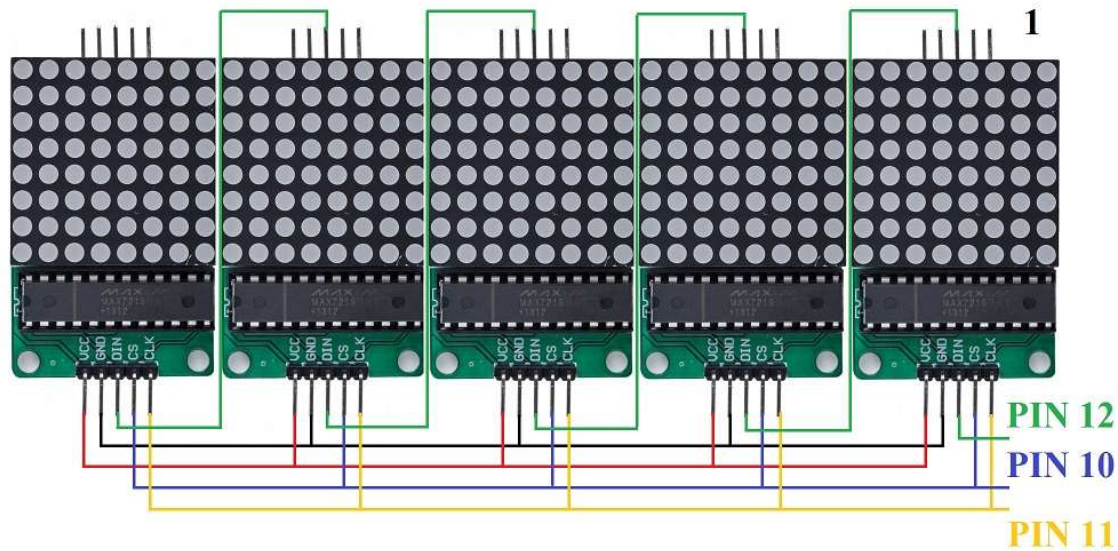
	Carcasa parte inferior
	Soporte Arduino Nano
	Panel identificativo de la Consejería de Educación

### 7.3. El MAX7219

El chip MAX7219 es un controlador LED, que nos permite manejar varios displays de 7 segmentos o matrices LEDs empleando muy pocos pines del Arduino, concretamente tres, gracias a que la comunicación se realiza a través de bus serie SPI.

El bus SPI tiene una **arquitectura de tipo maestro-esclavo**. El maestro (master), en este caso el Arduino, se comunica con uno o varios esclavos (slaves), en este caso los dos chips MAX7219.

La siguiente imagen muestra el esquema de comunicación entre el Arduino y los módulos MAX7219:



La descripción de los pines es la siguiente:

- VCC: Alimentación 5V
- GND: Tierra
- CLK: Reloj comunicación puerto SPI
- CS: “Chip Select” para habilitar el chip con el que se va a comunicar
- DIN: Datos de entrada
- DOUT: Datos de salida

El funcionamiento general es el siguiente:

La comunicación por puerto SPI es un tipo de comunicación síncrona, lo que significa que todos los componentes deben estar sincronizados, y para ello se requiere de una señal de reloj “CLK” que parte del maestro (Arduino) y que es común a todos los esclavos.

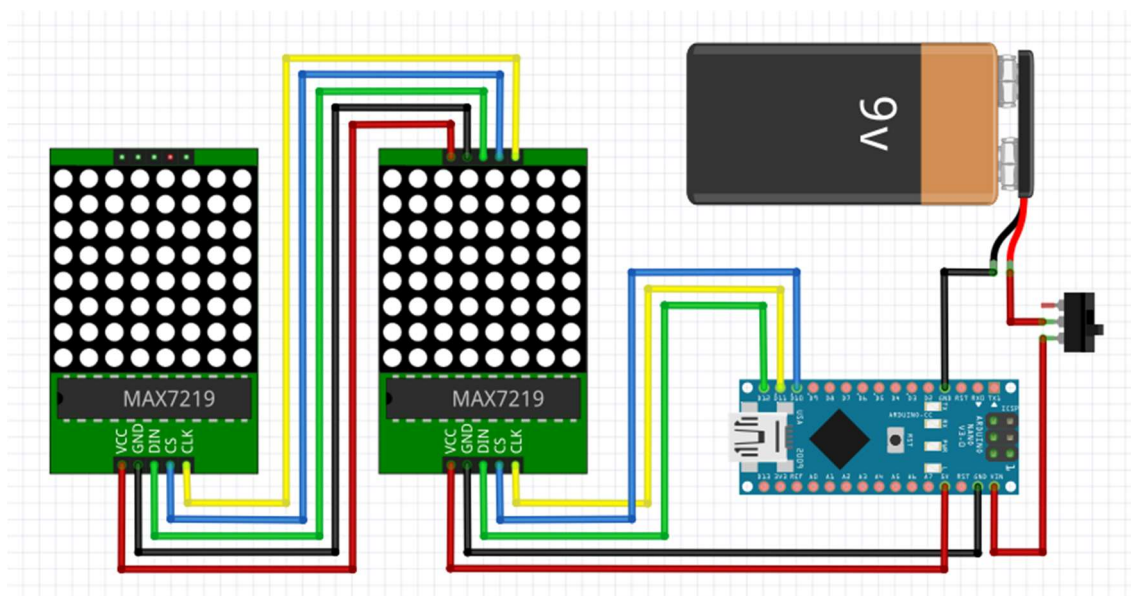
Como el texto que se muestra en pantalla se moverá de derecha a izquierda, pasando de una matriz LED a otra, todos los esclavos deben estar permanentemente habilitados, por ello comparten la misma señal “CS”.

El Arduino solo comunica datos directamente al MAX7219 que se encuentra más a la derecha, a través del pin “DIN”. Luego estos datos se transfieren al segundo chip de la izquierda, conectando los pines “DOUT” del primer chip y “DIN” del segundo chip. De esta forma se logra transferir el texto de una matriz LED a otra.



#### 7.4. Esquema electrónico

La siguiente imagen muestra el esquema electrónico del Entrenador Arduino V1.0:



Para que el Entrenador de Arduino sea portátil, utilizaremos como fuente de alimentación una batería de Li-ion 6F22 de 7,4 V, en serie con un interruptor para que fácilmente podamos encender y apagar el dispositivo.

#### 7.5. Extensión del MAX7219 para mBlock

Para poder tener acceso a un juego de instrucciones que nos permita controlar la Matriz Led MAX7219 de forma sencilla, necesitamos una extensión específica llamada MatrizLed, que podemos instalar siguiendo los pasos que se indican en la actividad correspondiente publicada en el aula virtual del curso.

El juego específico de instrucciones que podrás utilizar con esta extensión son las que se detallan en la siguiente tabla:

Sentencia	Descripción
Configurar: DIN 12 CLK 11 CS 10 NM 1	Configuración general: DIN: pin de datos CLK: pin de reloj CS: pin enable matriz

	NM: nº de matrices
mostrar cifra 0	Muestra una cifra de hasta dos dígitos
mostrar dos letras OK	Muestra una palabra de hasta dos letras
mostrar frase texto velocidad 50	Muestra una frase que se desplaza a la velocidad indicada (milisegundos / frame)
led matriz nº 0 fila 0 columna 0 estado encendido	Determina el estado del led indicado de la matriz
brillo 0	Determina el nivel de brillo de la pantalla, MIN: 0 MAX: 15
encender pantalla	Enciende la pantalla
apagar pantalla	Apaga la pantalla sin perder los datos almacenados en memoria
borrar pantalla	Borra los datos en memoria

## 7.6. Librería del MAX7219 para Arduino IDE

Para poder tener acceso a un juego de instrucciones que nos permita controlar la Matriz Led MAX7219 de forma sencilla, necesitamos una librería específica llamada MatrizLed, que podemos instalar siguiendo los pasos que se indican en la actividad correspondiente publicada en el aula virtual del curso.

El juego específico de instrucciones que podrás utilizar con esta librería son las que se detallan en la siguiente tabla:

Sentencia	Descripción
<b>#include &lt;MatrizLed.h&gt;</b>	Línea de código al comienzo del programa para cargar los recursos de la librería

<b>pantalla.begin(int, int, int, int)</b>	Configuración general. De izquierda a derecha: pin de datos, pin de reloj, pin enable matriz, nº de matrices
<b>pantalla.escribirCifra(int)</b>	Muestra una cifra de hasta dos dígitos
<b>pantalla.escribirFrase("frase")</b>	Muestra una palabra de hasta dos letras
<b>pantalla.escribirFraseScroll("frase", 200)</b>	Muestra una frase que se desplaza a la velocidad indicada (milisegundos / frame)
<b>pantalla.setLed(int, int, int, estado)</b>	Determina el estado del led indicado de la matriz. De izquierda a derecha: número de matriz (empezando por 0), fila, columna, estado (true – false)
<b>pantalla.setIntensidad(int)</b>	Determina el nivel de brillo de la pantalla, MIN: 0 MAX: 15
<b>pantalla.encender()</b>	Enciende la pantalla
<b>pantalla.apagar()</b>	Apaga la pantalla sin perder los datos almacenados en memoria
<b>pantalla.borrar()</b>	Borra los datos en memoria

### 7.7. Montaje del Entrenador de Arduino V1.0

Una vez que tengamos el circuito bien conectado y se haya comprobado que todo funciona correctamente, podemos pasar al montaje final del Entrenador de Arduino V1.0. Para ello disponemos de las partes de la carcasa impresas en 3D, del circuito que acabamos de conectar y de los tornillos necesarios para unir y asegurar las diferentes piezas y componentes que conforman el producto.

Las instrucciones específicas acerca del montaje del Entrenador las explicaremos de forma presencial en el aula.

### 7.8. Programación del Entrenador de Arduino V1.0

Llegado a este punto ya deberías tener tu Entrenador de Arduino V1.0 listo para ser usado, de modo que te animo a probar las diferentes opciones que te ofrecen las librerías que hemos visto, para conocer mejor Arduino y el manejo de las matrices LED, así como a elaborar un programa final totalmente libre y que te llame la atención.