

**Sobre los Entrenadores**

* 

*Eduardo Yallico*

Eduardo Yallicos es un Electrónico graduado con experiencia en diversos Microcontroladores y Electrónica Industrial.

* 

*Juan Ñahui*

Juan es un estudiante de Ingeniería Mecatrónica, apasionado por la Electrónica y los últimos avances tecnológicos, asi como de la implementación de estos en nuestras vidas.

* 

*Josset Gutierrez*

Josset es un apasionado de la Electrónica, te guiará durante el CoreUpgrade 2016, no dudes en contactarlo.

**Programas Necesarios**

[Google Chrome](https://www.google.com/chrome/browser/desktop/index.html), [IDE Arduino](https://www.arduino.cc/en/Main/Software)

### ****Teoría****Introducción al Mundo de la Electrónica.

### ****Fundamentos de la electrónica****

Antes que nada, quisiera compartir con ustedes cierta frase que me ha estado acompañando desde hace ya bastante tiempo:

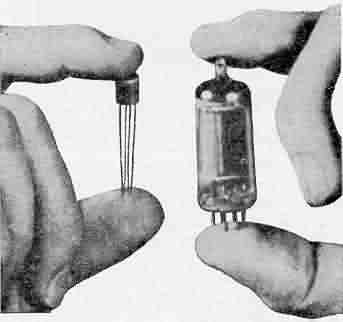
"Me lo contaron y lo olvidé, lo vi y lo aprendí, lo hice y lo entendí."

Dicho esto, bienvenidos a esta sección del entrenamiento, en la que nos enfocaremos en la electrónica. Así que daremos un vistazo realmente rápido a la evolución que ha tenido esta rama.

### Érase una vez...

Desde los años 1800 con la invención de la pila voltaica (cortesía de Alessandro Volta), empezaron los avances en este campo. Otros grandes avances, hoy obsoletos y ligeramente olvidados, fueron el diodo al vacío (1904, Fleming), el primer dispositivo rectificador de corriente; y el triodo (1906, Lee), el primer componente amplificador electrónico. Sin embargo, todo dio un salto enorme desde la invención del transistor, en 1947. Esto redujo enormemente el espacio y los recursos necesarios para implementar un circuito.

El desarrollo de los semiconductores pronto permitiría que en un solo chip puedan entrar 6 transistores dando nacimiento al primer circuito integrado, en 1958. Luego, en 1970 se desarrolló el primer microprocesador. Años después, la electrónica ha dejado (en gran parte) de ser una ciencia solo para aplicaciones industriales, para convertirse en parte de lo cotidiano. Poseemos celulares, radios, computadoras y un sinfín de equipos que consideramos parte de nuestras vidas.



Si bien la intención de este entrenamiento no es desarrollar cosas con tal magnitud de complejidad, deseamos que obtengan las herramientas básicas para prototipar circuitos, pierdan el miedo a lo desconocido y puedan realizar los proyectos que necesiten, desde el ámbito universitario hasta la domótica.

### Conozcamos un poco más de este nuevo mundo!

Para decirlo en términos sencillos la electrónica es ciencia que se encarga del estudio, el control y la aplicación de la producción y el procesamiento de de las señales eléctricas. Hoy en día la tenemos presente en absolutamente TODO, esta ciencia ejerce un gran aporte en la sociedad y cada día nos va invadiendo más.

¿Por qué debemos aprenderlo? Porque YOLO, porque te da poder y porque entenderás cómo funcionan las cosas que cotidianamente usas y no solo eso, lograrás implementar pequeñas cosas que aprenderas a hacer en este curso y obtendras nociones en programación que te permitirán seguirte adentrando en el universo de la Tecnología.

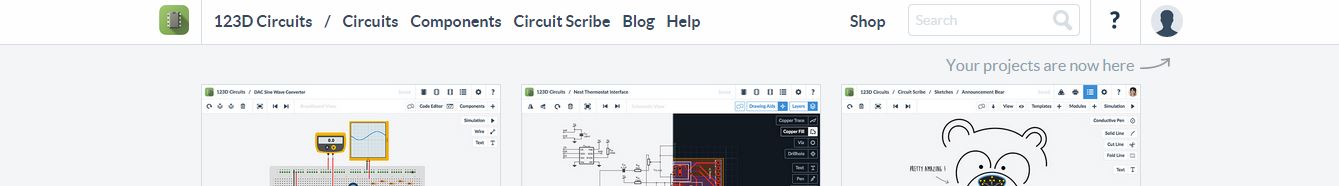
Como ya mencionamos les proporcionaremos herramientas que facilitarán el ingreso al mundo de la Electrónica. Para ellos utilizaremos dos herramientas que nos ayudaran: El mundialmente conocido [Arduino](https://www.arduino.cc/en/Main/Software" \t "_blank)y el simulador [123D Circuits](https://123d.circuits.io/).

### ****Practica****Conociendo un Nuevo Entorno: 123D Circuits.

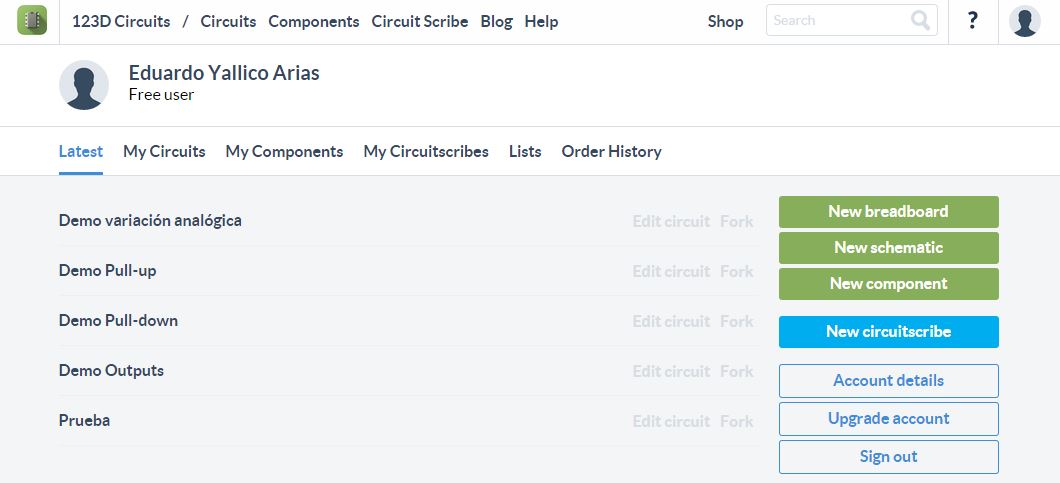
### Iniciando en 123D Circuits

Primero lo primero. Para poder usar el 123D Circuits a plenitud, necesitamos una cuenta, así que a registrarnos. Ingresar a [http://123d.circuits.io](http://123d.circuits.io/) y crear una cuenta.

Al logearnos, en la esquina superior derecha podemos observar la ubicación de nuestros proyectos. Click en el sujeto gris:

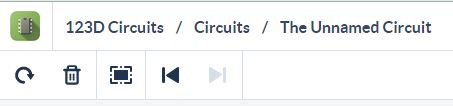


Esta es mi cuenta, hasta el momento. Pueden observarse los proyectos que tengo disponibles, y un par de opciones a la derecha de los proyectos:

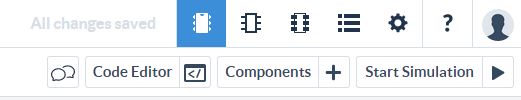


Click en **New breadbord**, o nuevo protoboard.

Daremos una mini explicación del entorno:



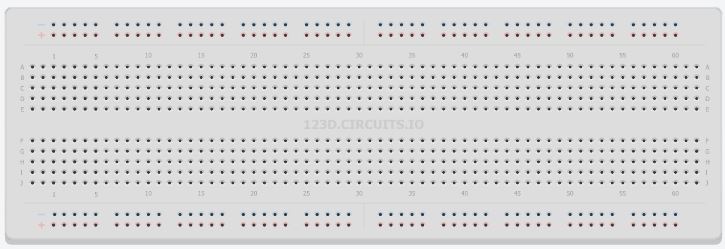
Aquí tenemos el nombre del circuito que estamos diseñando. El nombre por default es "The Unnamed Circuit". Luego, debajo, de izquierda a derecha, tenemos los botones **Rotate**, que se gira componentes en el protoboard; **Delete**, cuya función es eliminar componentes; **Zoom to fit**, que ajusta el zoom para que podamos ver todos los objetos puestos en el área de trabajo; y por último tenemos las opciónes **Undo** y su contraparte, para deshacer cambios recientes.



Del otro lado, arriba, tenemos botontes bastante peculiares. Al lado de **"All changes saved"** podemos observar 3 botones que parecen pequeños chips. Estos son las vistas disponibles. Tenemos la **vista en protoboard**, la **vista del esquemático** y la **vista en el PCB**. Usaremos para el entrenamiento la vista en protoboard (breadboard). El siguiente botón corresponde a la **cuenta por materiales**, que nos entrega un precio promedio de los componentes requeridos. Finalmente tenemos la **vista de configuraciones**, donde podemos cambiar el nombre de nuestro proyecto. Finalmente, podemos ver los **botones de feedback** y nuestro **usuario**.

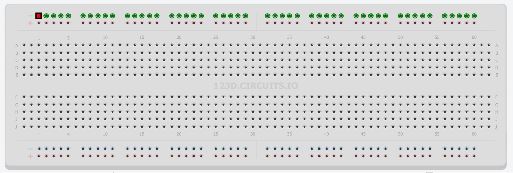
Debajo tenemos los **botones de chat**; el **editor de código**, donde pondremos el código para Arduino; la**librería de componentes**, que podremos colocar en el protoboard; y el más importante, **Iniciar simulación**, que nos permite probar nuestro circuito.

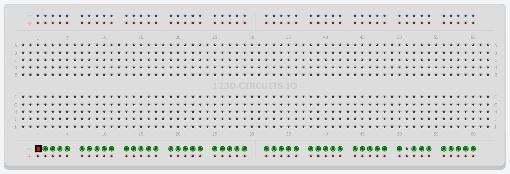
Luego tenemos el **área de trabajo**, que tiene un protoboard.



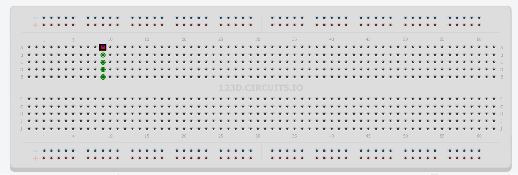
**¿Cómo funciona el protoboard?** Tiene conexiones internas que facilitan el cableado. Estas están dispuestas de la siguiente manera:

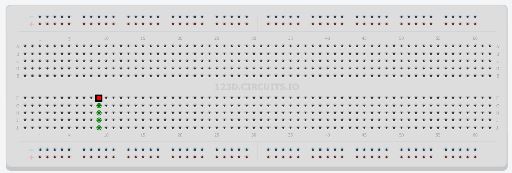
Los extremos tienen conexión horizontal interna. Cualquier cosa que se conecte a alguno de los pines en esa línea, también estará conectada a otra en la misma línea.





La parte interna tiene conexiones verticales, pero la línea central funciona división, es decir, las dos zonas centrales están aisladas entre sí.





Con esto hemos visto lo necesario para conocer nuestro entorno en el que trabajaremos estas semanas. Ahora solo nos queda conocer algunos conceptos básicos y componentes para sacarle el jugo a esta plataforma. Click en el botón de abajo para aprender estas cosas :D

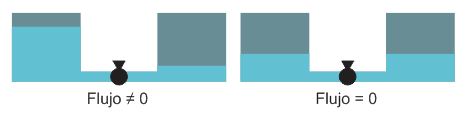
### ****Teoría****Conceptos Básicos de la Electrónica

### ****Conceptos claves para entender un poco más de Electrónica****

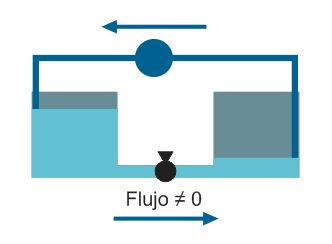
Antes que nada daremos una breve explicación de la teoría básica necesaria.

El **voltaje, o diferencia de potencial**, puede ser asimilado como la diferencia de nivel entre dos tanques conectados.

Si no hay diferencia de nivel, no puede haber flujo. Lo mismo ocurre con la corriente. Si no hay diferencia de potencial, no hay corriente. Sin embargo, hay otro requisito, y es que el flujo sea permitido por las conexiones entre los tanques. Es decir, mientras nadie le permita el paso, a pesar de tener voltaje, la corriente será 0.



Ahora bien, contamos con una diferencia de potencial estable. Para que esto ocurra, usamos una fuente de voltaje, que puede ser asemejada a una bomba de agua que mantiene la diferencia de nivel, permitiendo un flujo constante por diferencia de potencial.



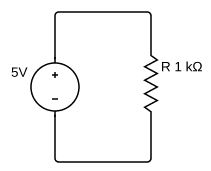
Sin embargo, la cantidad de agua que circula depende de la tubería. Mientras más angosta sea, menos agua podrá circular. Esto se asemeja a las resistencias, que limitan el flujo de la corriente.

Es hora de ponerle unidades. La diferencia de potencial (diferencia de nivel) se mide en **voltios (V)**; la intensidad de corriente (flujo de agua), en **amperios (A)** y la resistencia (el ancho de la tubería), en **ohmios (Ω)**.  
La relación entre las variables anteriormente mencionadas es descrita por la famosa **ley de Ohm**:

Voltaje = Corriente x Resistencia  
V = I X R

Puesto que un amperio es una unidad bastante grande de corriente, lo normal es trabajar con el**miliamperio (mA).**

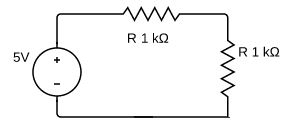
Realizaremos el cálculo correspondiente para el siguiente circuito:



Puesto que tenemos como datos el voltaje (5V) y la resistencia (1 KΩ), calcularemos la corriente, mediante la ley de ohm. Así de simple, obtenemos la respuesta: **5 mA** (o también 0.005 A, aplicar conversion de unidades, 1A = 1000 mA).

Siguiente concepto importante, la resistencia equivalente. Se sigue aplicando la ley de Ohm, pero con una consideración, en este caso:

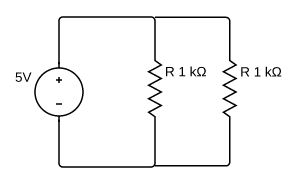
La resistencia equivalente de "n" resistencias en***serie***es igual a la suma de dichas resistencias

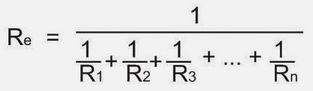


http://hackspace.la/img/ejemplos/electronica1.4.5.jpg

Entonces ambas resistencias pueden ser reemplazadas por su equivalente, es decir, una resistencia de 2KΩ. Realizando nuevamente los cálculos, obtenemos el nuevo valor de la corriente: 2.5 mA

La resistencia equivalente de "n" resistencias en***paralelo***es igual a la inversa de la suma de sus inversas"





Entonces ambas resistencias pueden ser reemplazadas por su equivalente, es decir, una resistencia de 500Ω. Realizando nuevamente los cálculos, obtenemos el nuevo valor de la corriente: 10 mA.

Notamos aquí algo curioso, en ambos casos tuvimos la misma cantidad de resistencias, y estas fueron del mismo valor. La unica diferencia fue la manera de conectarlas.

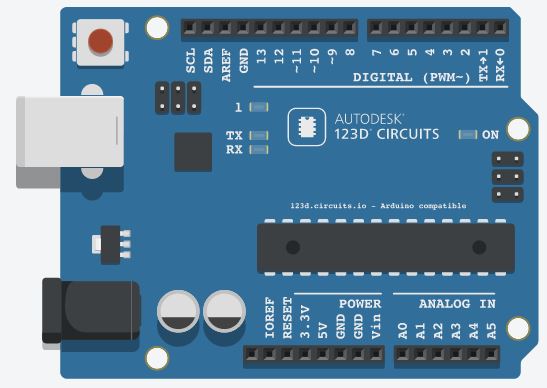
### ****Digital y Analógico (Orientado a Arduino)****

Lo digital puede ser comprendido como **1's y 0's, encendido y apagado**. Pero, ¿Cómo hace el Arduino para determinar si está leyendo un "1" o un "0"? Lo que hace en realidad es leer el voltaje en el pin deseado, y si el valor es mayor a cierto punto, lo considera como **"1"(HIGH)**, y si es menor, lo considera como **"0"(LOW)**. Puesto que el mundo del Arduino está comprendido entre los 0 y 5V, si le otorgamos un valor cercano a 5V lo considerará como alto, y si le otorgamos un valor cercano a 0V lo considerará como bajo. El **valor umbral está cercano a los 2V**, pero para no causar confusiones en la lectura digital, se recomienda que el voltaje entregado no esté cerca al valor mencionado, ya que podría generar datos erróneos.

Lo **analógico se comprende como variación**. Es decir, el Arduino lee el pin seleccionado y almacena el valor de voltaje con un número entre **0 y 1023**, escalando los valores de 0V a 5V. Para esto, 0V equivale a 0 y 5V a 1023, y cualquier valor intermedio se puede calcular por una regla de tres simple.

Ambos casos se podrían ilustrar con una escalera. Mientras que a la lectura digital solo almacena si estás abajo o estás arriba (verdadero o falso), la lectura analógica almacena en qué escalón estás (variación).

Pasemos al hardware entonces. El arduino tiene **13 pines digitales**, pero puesto que los dos primeros son utilizados para la comunicación serial, normalmente no se usan, así que los pines digitales 0 y 1 no serán usados de momento. Estos pines pueden ser configurados como entradas, o como salidas.



Luego están los pines analógicos, del **A0 al A5**. Estos sólo pueden ser usados como entradas. Por otro lado, tenemos una **salida de 5V, una salida de 3.3V, tres conexiones a 0V o GND (tierra)**.

Ya conocemos un poco del Arduino, llego la hora de practicar! Dale click al botón de abajo!

### ****Practica****Manos a la obra!

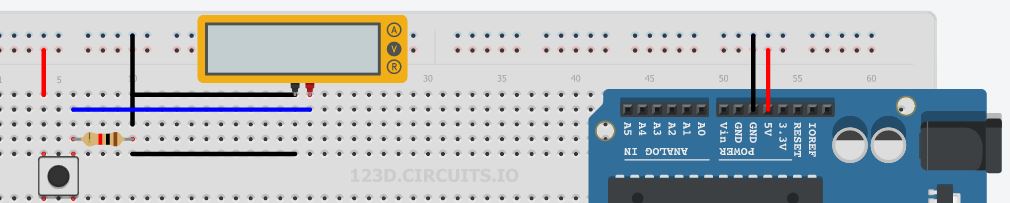
### Por fin algo de acción

Como ya hemos mencionado, el Arduino puede leer datos de dos maneras, digital y analógica. Empezaremos con lo digital. **¿Cuál es la manera correcta de darle un 1 o un 0?** Si pensaste en botones y switchs, tienes el 50% de la respuesta correcta. El otro 50% radica en la forma de conectarlos. Tenemos entonces 2 formas (efectivamente, ponerle un cable directo también funciona, pero significaría tener que mover cables para cambiar el valor, y es riesgoso):

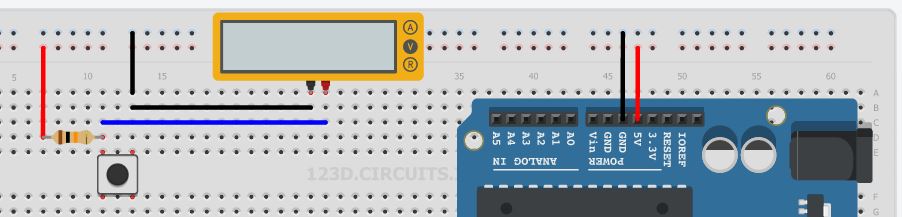
Tenemos entonces 2 configuraciones para los pulsadores: **pull-up y pull-down**

Consideraciones: En los siguientes ejemplos emplearemos una especie de "pantallita" conocida técnicamente como multímetro. Esta nos permite observar valores como la corriente, el voltaje, y la resistencia. Cuando empecemos a usar el Arduino, la usaremos con menos frecuencia. Por el momento usamos el Arduino solo como una fuente de voltaje, no tiene nada programado.

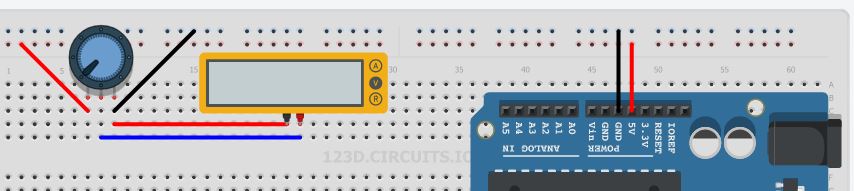
Pull-up entrega constantemente al pin un "0" lógico, y cuando se presiona el pulsador, entrega un "1". Las conexiones son como se muestran:



Pull-down entrega constantemente al pin un "1" lógico, y al presionar el pulsador, entrega un "0". Las conexiones son las siguientes:



Y, **¿Cómo entregamos un valor analógico?** La respuesta está en el siguiente diagrama de conexión: se emplea un **potenciómetro.**



Con esto ya estamos listos para un pequeño reto! Clickea en el botón de abajo.

### ****Reto 1****Primer reto del entrenamiento de Electrónica

### ****Reto N°1 de Electrónica****

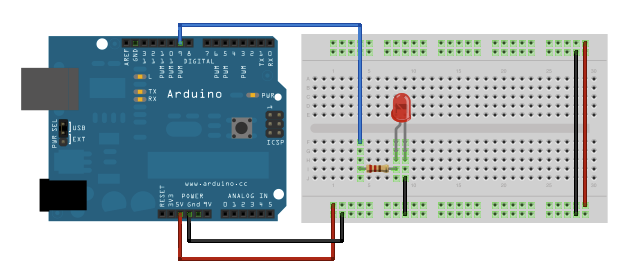
Para este primer reto tenemos un pequeño cuestionario para ti y un reto de implementación en 123D Circuits.

### Cuestionario:

1. ¿Cuál es la unidad para medir corriente?
2. La configuración Pull-up entrega un ... al presionar el pulsador.
3. El protoboard posee en total ... zonas aisladas.
4. ¿123D Circuits permite probar códigos en Arduino?
5. ¿Cuántas salidas analógicas posee un Arduino?¿Cuántas digitales?

### Desarrollo:

Para la siguiente parte del reto tenemos el siguiente circuito que debemos simular en 123D Circuits:



En en Code Editor coloquemos lo siguiente:

void setup() {

pinMode(9,OUTPUT);

}

void loop() {

digitalWrite(9,HIGH);

delay(1000);

digitalWrite(9,LOW);

delay(1000);

}

Para terminar clickea en **Upload and Run**.Por el momento no es muy necesario que entiendas el código que te mandamos, te enseñaremos a codear a lo largo de estas cuatro semana, lo importante es hacer la simulación y darse cuenta del resultado. Al realizar la simulación **tomen un pantallazo**, súbelo a<http://imgur.com/> (o tu hosting favorito) y adjúntenlo junto a las respuestas del cuestionario.

Hasta la proxima semana!

#### Semana 2 : Arduino

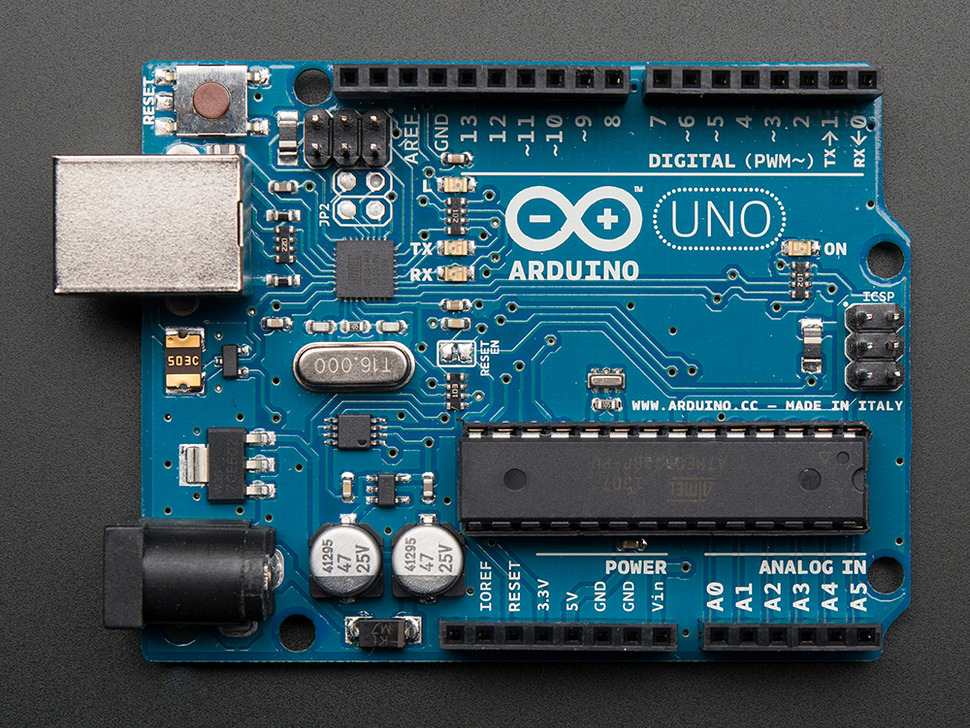
### ****Teoría****El Arduino.

### ****Un Nuevo Aliado****

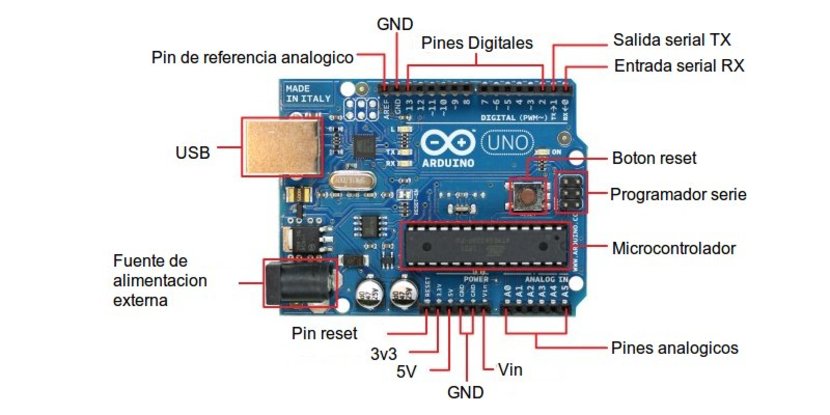
Continuando con este entrenamiento ya nos toca conocer más a fondo la electrónica y para esto nos ayudaremos de una herramienta fácil de aprender, moderna y segura, con la cual entenderemos más de la electrónica: **Arduino**

Arduino es toda una plataforma (incluye Software y Hardware) de electrónica libre orientada para personas interesadas en crear modelos o entornos interactivos, fáciles de elaborar, flexibles y de bajo costo. Al mencionar electrónica libre me refiero a que tú podrás elaborar tu propio Arduino, ya que está bajo licencia libre.

En la actualidad existen una infinidad de lenguajes de programación que nos agilizan la forma de trabajar. El Arduino no se escapa de esto, ya que se programa en C/C++ o incluso existen otras formas de darle funcionamiento al Arduino, como por ejemplo [S4A](http://s4a.cat/index_es.html) (Scratch para Arduino).



### ****Partes del Arduino****



**Una imágen vale más que mil palabras :D**

Lo que más usaremos de todo esto serán los pines de entrada/salida, los pines de alimentación y debes en cuando el botón de reset.

Mucho ojo: en el Arduino hay pines de e/s digitales y analógicas, ya explicamos qué es lo analógico y digital, ya debes tener una idea de que podemos hacer con estos pines.

¡Ya puedes clickear abajo y pasar a la siguiente parte!

### ****Teoría****Programación en Arduino.

### ****Revisando el Reto Anterior...****

El reto anteriormente planteado fue colocado para ver si podían entender cómo usar el 123D Circuits. Como sé que lo lograron y les pareció fácil, pasaré a explicarles el código que colocamos y luego revisaremos la estructura de cómo hacer un algoritmo para el Arduino.

Ahora si pasemos a revisar el código del reto anterior, que fue un "Blink" (parpadeo en español), elHOLA MUNDO de todo electrónico.

void setup() {

pinMode(9, OUTPUT);

}

void loop() {

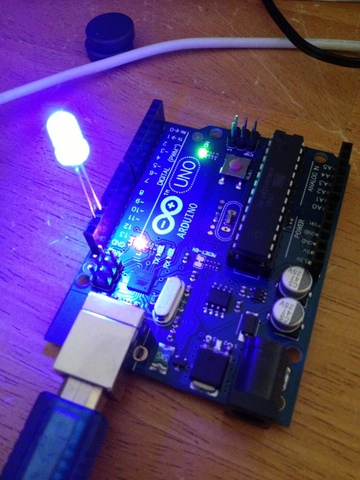
digitalWrite(9, HIGH);

delay(1000);

digitalWrite(9, LOW);

delay(1000);

}



### ****Vayamos por partes!****

Partes principales del Sketch:

void setup() {

/\* Aquí colocaremos todas nuestras configuraciones

Cuáles seran nuestros pines de entrada, cuáles seran

los de salida, configuraciones iniciales, etc.

Se ejecuta una sola vez \*/

}

void loop() {

/\* Esta función se ejecuta luego del void setup().

Contiene la secuencia de pasos que sigue el Arduino.

No se detiene hasta que se desconecte la fuente de

alimentación del Arduino. \*/

}

Como ya mencionamos, en el void setup() vamos a colocar las configuraciones que necesitaremos en nuestro código. En este caso para prender el led necesitabamos la siguiente línea:

**pinMode(9, OUTPUT);**

Con esto le estamos diciendo que el Pin 9 sea una OUTPUT (o salida, en español).En otras palabras este pin emitirá el pulso cuando se lo ordenemos.

Dentro del void loop() tenemos las 4 siguientes líneas:

**digitalWrite(9, HIGH);**

**delay(1000);**

**digitalWrite(9, LOW);**

**delay(1000);**

**Explicación:**

Con el 'digitalWrite(9,HIGH);' le estamos diciendo al Pin 9 que emita un '1' lógico, y por ende prenda el led. El 'delay(1000)' se refiere a tiempo de ejecución, en este caso 1000 microsegundos (medida por defecto del arduino). Lo mismo ocurre con las dos siguiente líneas, a excepcion que en lugar de dar un pulso de '1' lógico, nos da uno de un '0', lo cual apagaría el led por otros 1000 microsegundos, equivalente a 1 segundo. Como todo esto está en la función loop, no se detendrá salvo que desconectes el Arduino.

### ****Practica****Hora de aprender a codear!

### ****Entendiendo un poco más de código****

/\*

Encender LED con un pulsador

Código para que al presionar

el pulsadorel led se encenderá

\*/

//------------------------------------

//Declara puertos de entradas y salidas

//------------------------------------

int pulsador=2; //Pin donde se encuentra el pulsador, entrada

int led=13; //Pin donde se encuentra el LED, salida

//------------------------------------

//Configuración

//------------------------------------

void setup() // Se ejecuta cada vez que el Arduino se inicia

{

pinMode(pulsador, INPUT); //Configurar el pulsador como una entrada

pinMode(led,OUTPUT); //Configurar el LED como una salida

}

//------------------------------------

//Loop

//------------------------------------

void loop() // Esta funcion se mantiene ejecutando

{ // cuando este energizado el Arduino

//Condicional para saber estado del pulsador

if (digitalRead(pulsador)==HIGH)

{

//Pulsador oprimido

digitalWrite(led,HIGH); //Enciende el LED

}

else

{

//Pulsador NO oprimido

digitalWrite(led,LOW); //Apaga el LED

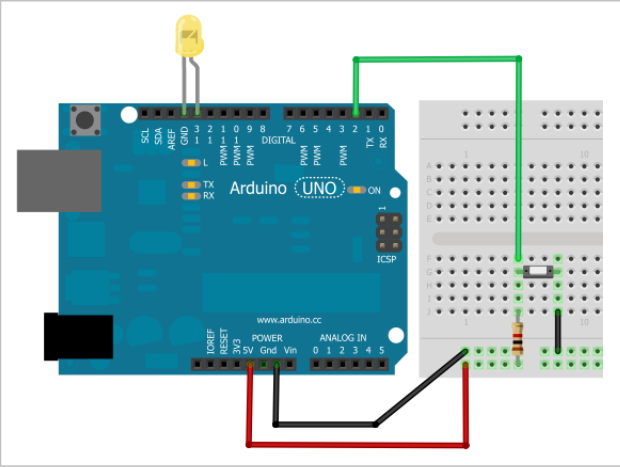
}

}

//Fin programa

Te dejamos un código totalmente comentado para que entiendas un poco más de como declarar variables y el orden que debes llevar para codear en Arduino.

Los comentarios (lo que está despues de // o entre /\* \*/ ) son textos que no son leídos por el Arduino, pero nos permite dejar palabras clave o frases en partes de código que podemos olvidar para qué son.



Ahora sí, tú mismo eres. Aqui arriba tienes el esquemático del circuito con todas las conexiones correspondientes al código planteado. Tú mismo eres, simúlalo en 123D Circuits :D

En la siguiente sección te dejaré un video con un resumen de lo visto hasta ahora.

### ****Reto 2****Segundo reto del entrenamiento de Electrónica

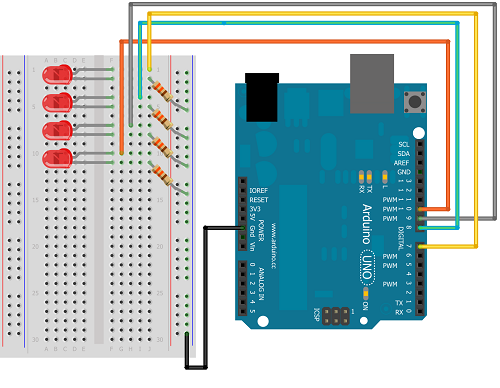
### ****Reto N°2 de Electrónica****

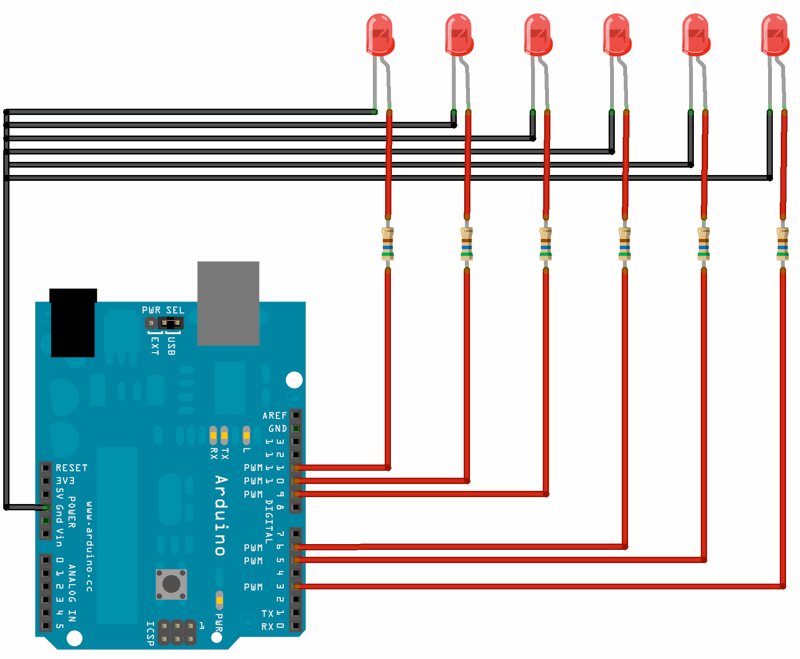
Para el segundo reto tendrán que hacer un conjunto de luces secuenciales.

Para que lo entiendan más fácil, los leds tendrán que prender secuencialmente. Pueden googlear, pero será mejor que ustedes creen el código.

Aqui les dejo una imágenes del esquemático. **Ready, Set, Go!**

Envía la foto de tu proyecto terminado.





[Entregar Reto](https://hackspaceper.typeform.com/to/ZkJTM7)