



Arduino

Potenciador de la educación

Roberth Brenes Padilla

Computación física

Es el campo de la computación que estudia la conexión entre el mundo físico y los ordenadores mediante el uso de sensores y actuadores para interactuar con el entorno.

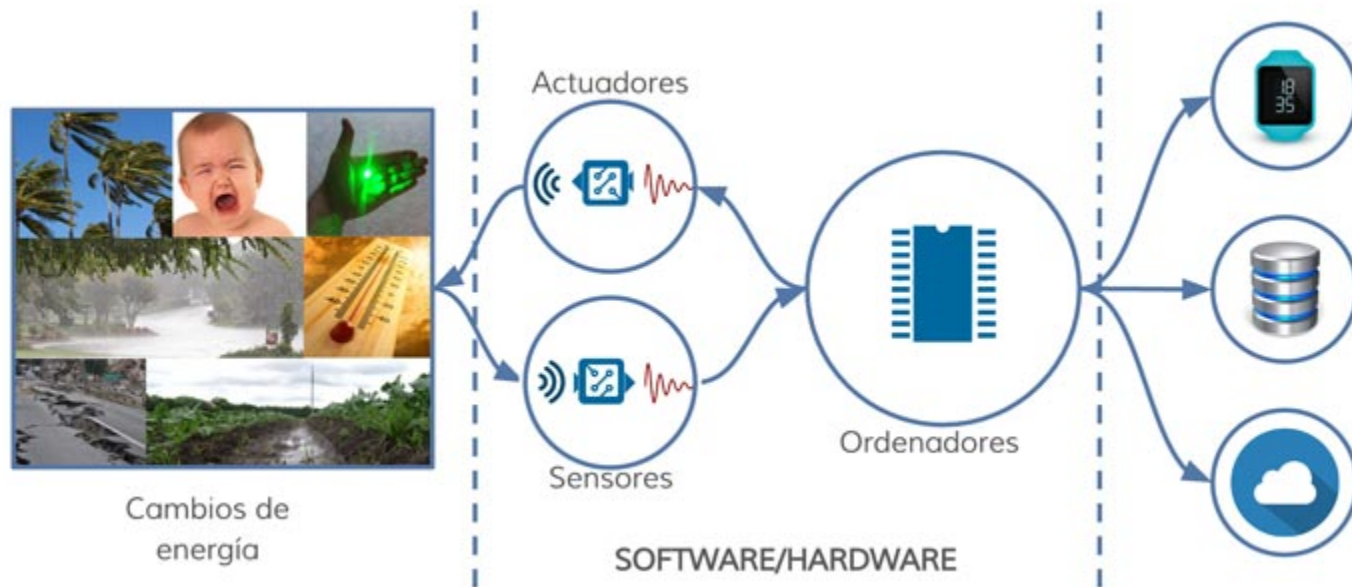


Imagen de programarfacil.com

¿Qué es Arduino?

Arduino es una placa electrónica de código abierto basada en *hardware* y *software* fácil de usar.

La placa es capaz de leer *entradas* y convertirlas en una *salida*.



Fuente: <https://www.arduino.cc>

¿Qué es Arduino?

Arduino utiliza el lenguaje de programación Arduino, basado en **Wiring**, y el software Arduino (*IDE Integrated Development Environment*), basado en el **Processing**.

Utiliza licencia **Creative Commons Attribution Share-Alike**.



Fuente: <https://www.arduino.cc>

¿Por qué elegir Arduino?

Presenta las siguientes ventajas.

- ✓ Precio accesible.
- ✓ Multiplataforma.
- ✓ Amplia documentación.
- ✓ Comunidad muy activa.
- ✓ Software de código abierto.
- ✓ Mayor flexibilidad de componentes y placas.
- ✓ Entorno de programación simple y claro.

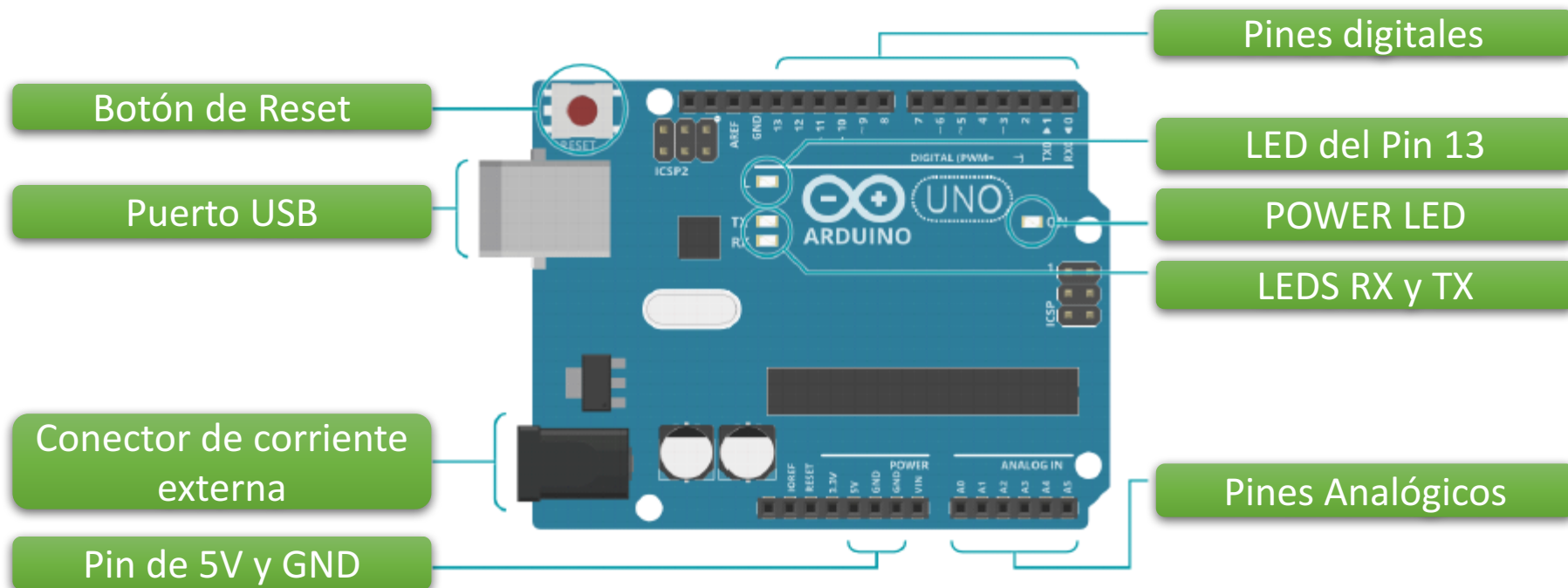


Conceptos importantes

- Hardware.
- Software.
- Entradas (input)
- Salidas (output)
- Programación.
- IDE.
- Código abierto.
- Hardware libre.
- Creative Commons.
- Computación física.



Partes de la placa



El IDE de Arduino

Es un paquete de software que permite crear y descargar programas en placas Arduino y otras placas compatibles.

El software es libre y gratuito. Se puede descargar de la página: <http://arduino.cc/en/Main/Software>



ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.



ARDUINO WEB EDITOR

Start coding online with the [Arduino Web Editor](#), save your sketches in the cloud, and always have the most up-to-date version of the IDE, including all the contributed libraries and support for new Arduino boards. The Arduino Web Editor is one of the [Arduino Create](#) platform's tools.

Try It Now
Getting Started

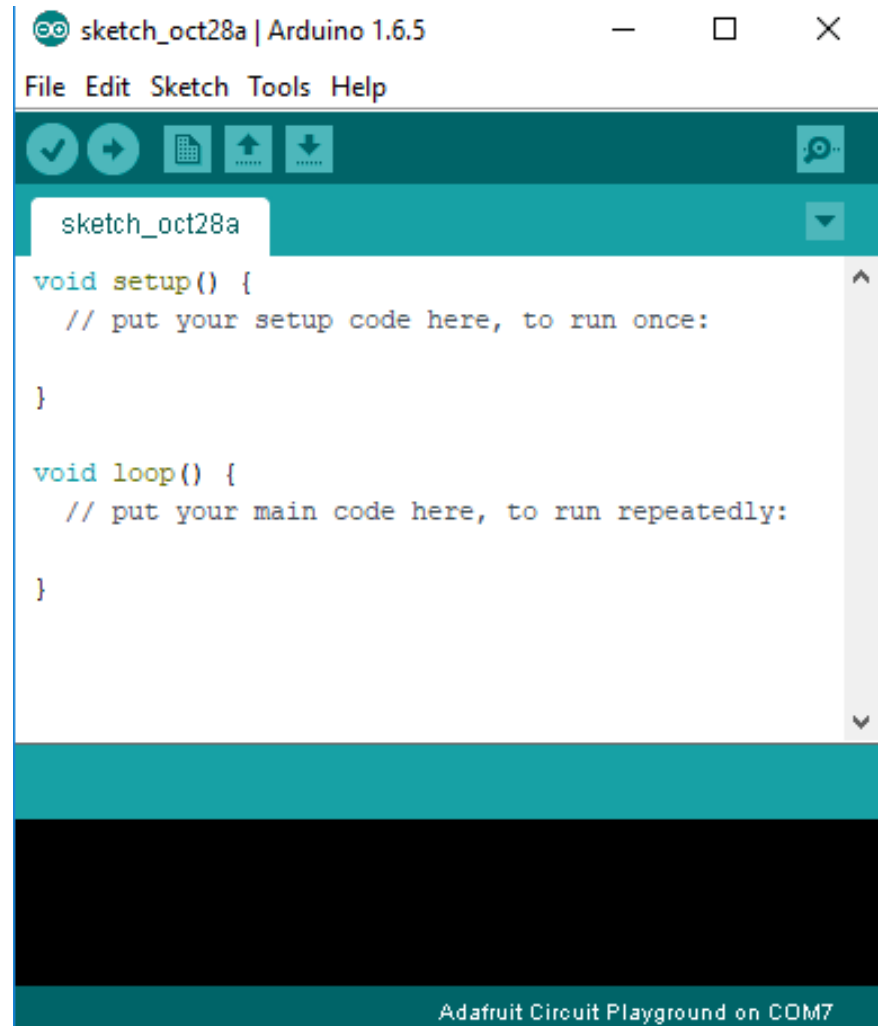
Editor Web de Arduino

The screenshot displays the Arduino Web Editor interface. On the left is a teal sidebar with navigation links: EDITOR, Sketchbook, Examples, Libraries, Monitor, Help, and Preferences. The main workspace is divided into three sections. The top section shows a 'NEW SKETCH' button and a search bar for the Sketchbook. Below this is a list of sketches, with 'sketch_oct23a' selected. A large upward arrow icon is present, with text below it: 'Import your sketches to your online Sketchbook and access them from any device!'. The bottom section of the sidebar features the 'powered by amazon web services' logo. The central editor area shows a sketch titled 'sketch_oct23a' with a dropdown menu set to 'Arduino/Genuino Uno'. Below the menu are tabs for 'sketch_oct23a.ino' and 'ReadMe.adoc'. The code editor displays the following C++ code:

```
1 /*
2
3 */
4
5 void setup() {
6   ...
7 }
8
9 void loop() {
10   ...
11 }
12
```

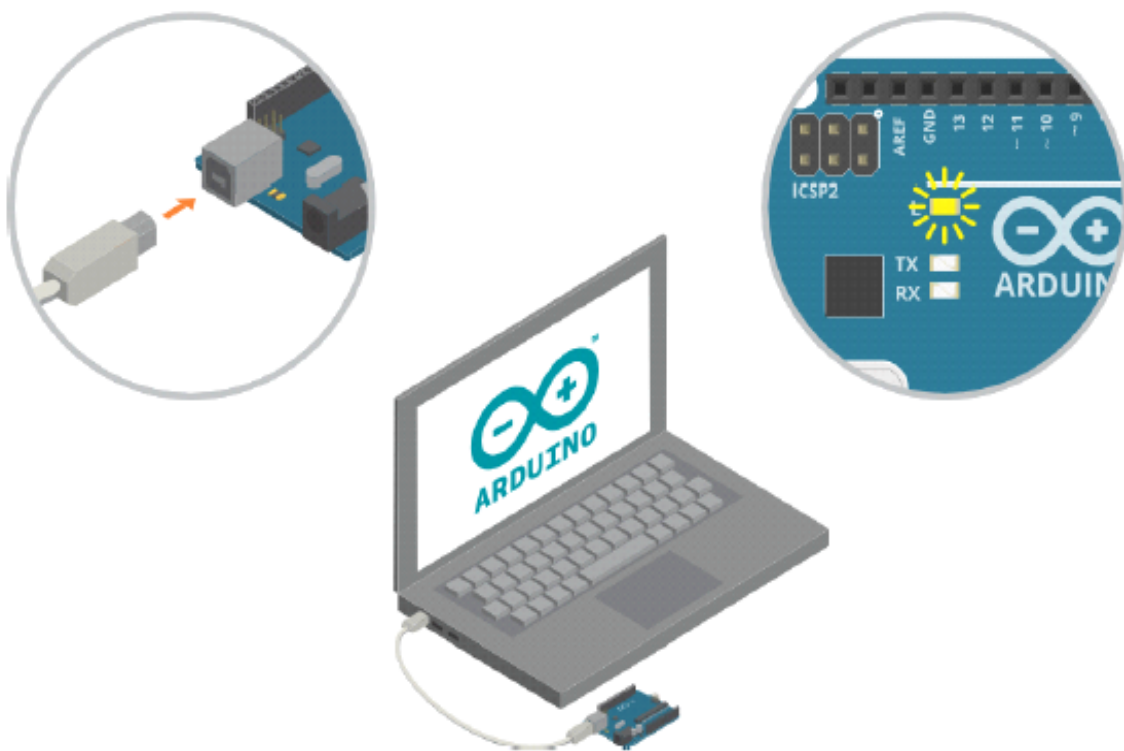
On the right side of the code editor, there are icons for zooming in and out. The bottom of the interface has a black status bar with a document icon on the right.

El IDE de Arduino



¡A trabajar!

Veamos un ejemplo práctico



1 **FILE>**
EXAMPLES>
BASICS>
BLINK

2 **TOOLS>**
BOARD>
ARDUINO UNO

3 **TOOLS>**
SERIAL PORT
Windows OS **COM####**
Mac OS **/dev/ttyusbmodem####**
Linux OS **/dev/ttyACM####**

4 **FILE>**
UPLOAD

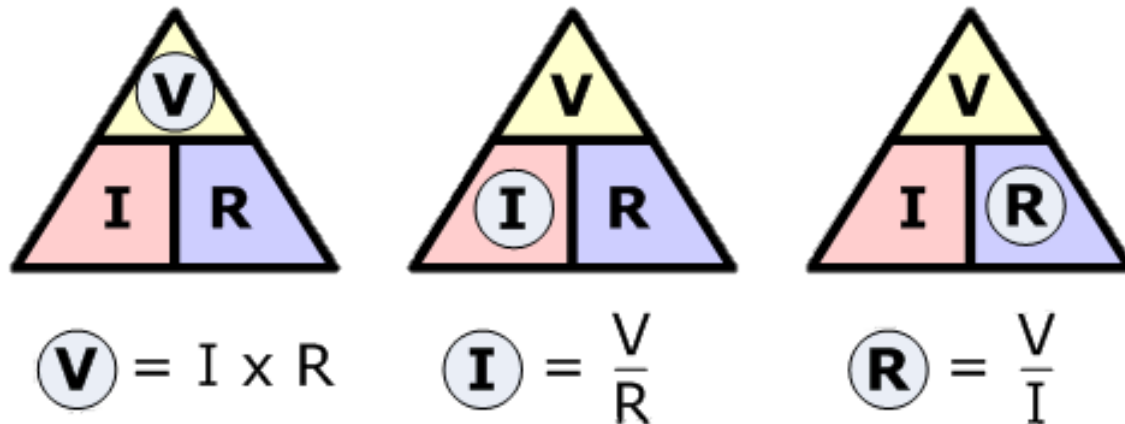


Imagen de: Robótica. Conectando con el mundo físico

Arduino y la ley de Ohm (circuitos)

Esta ley establece las relaciones que existen entre potencial eléctrico (voltaje), corriente eléctrica y la resistencia. Fue propuesta por el físico y matemático alemán **Georg Simon Ohm**.

V = potencia. **R** = resistencia. **I** = corriente.



Arduino y la ley de Ohm (circuitos)

Ejemplo con led que ocupa aproximadamente **3.4V**, y con una corriente de por lo menos **5** mili amperios (**5mA**), aplicamos la formula de la ley de Ohm

$$R = V / I.$$

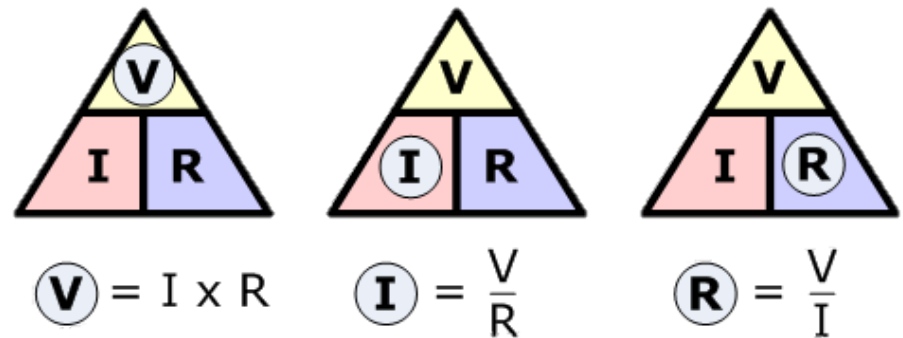
Arduino pasa **5V**, entonces:

$$(5 - 3.4) / 0.005 = 320 \text{ aprox. } *$$

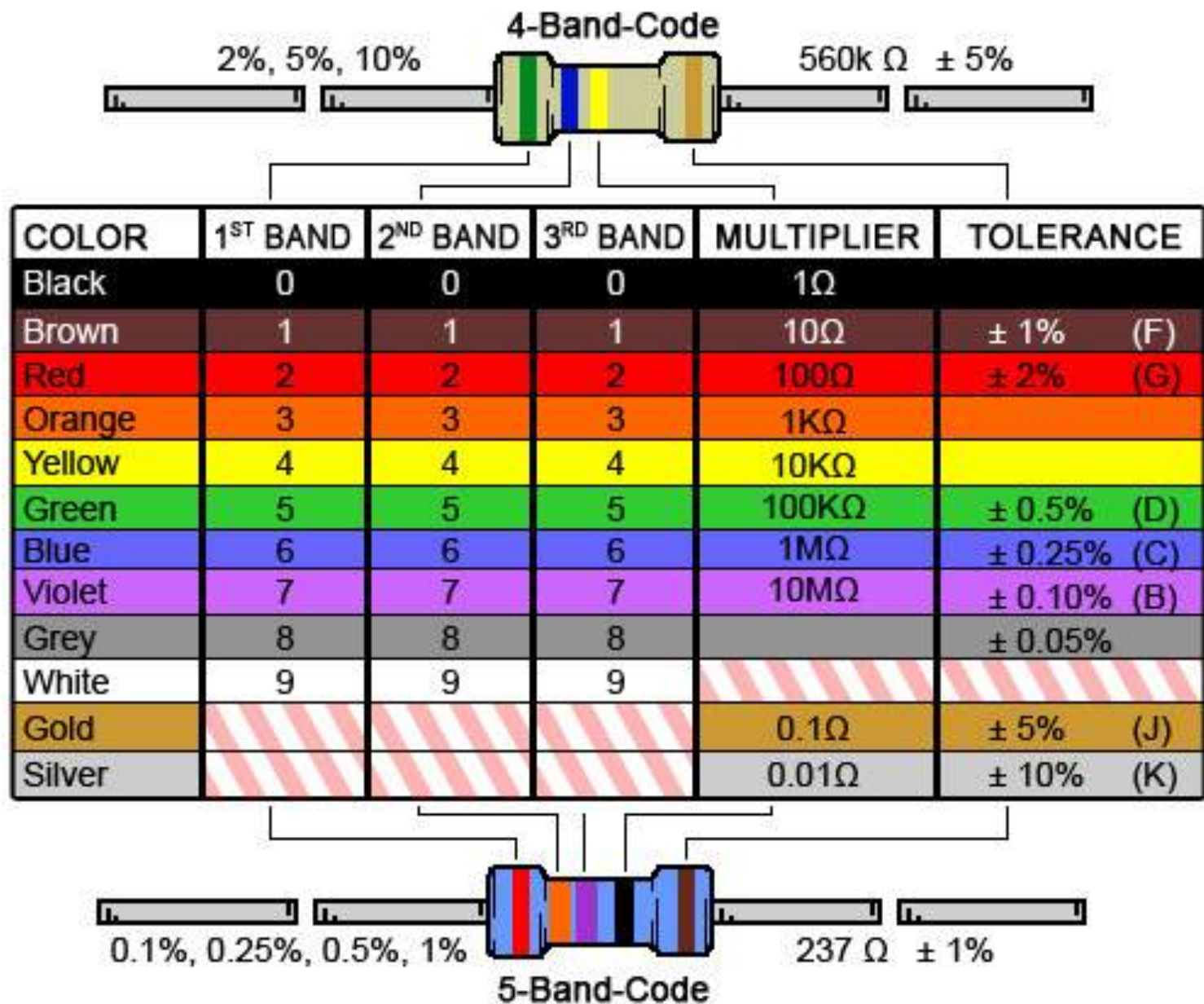
V = potencia.

R = resistencia.

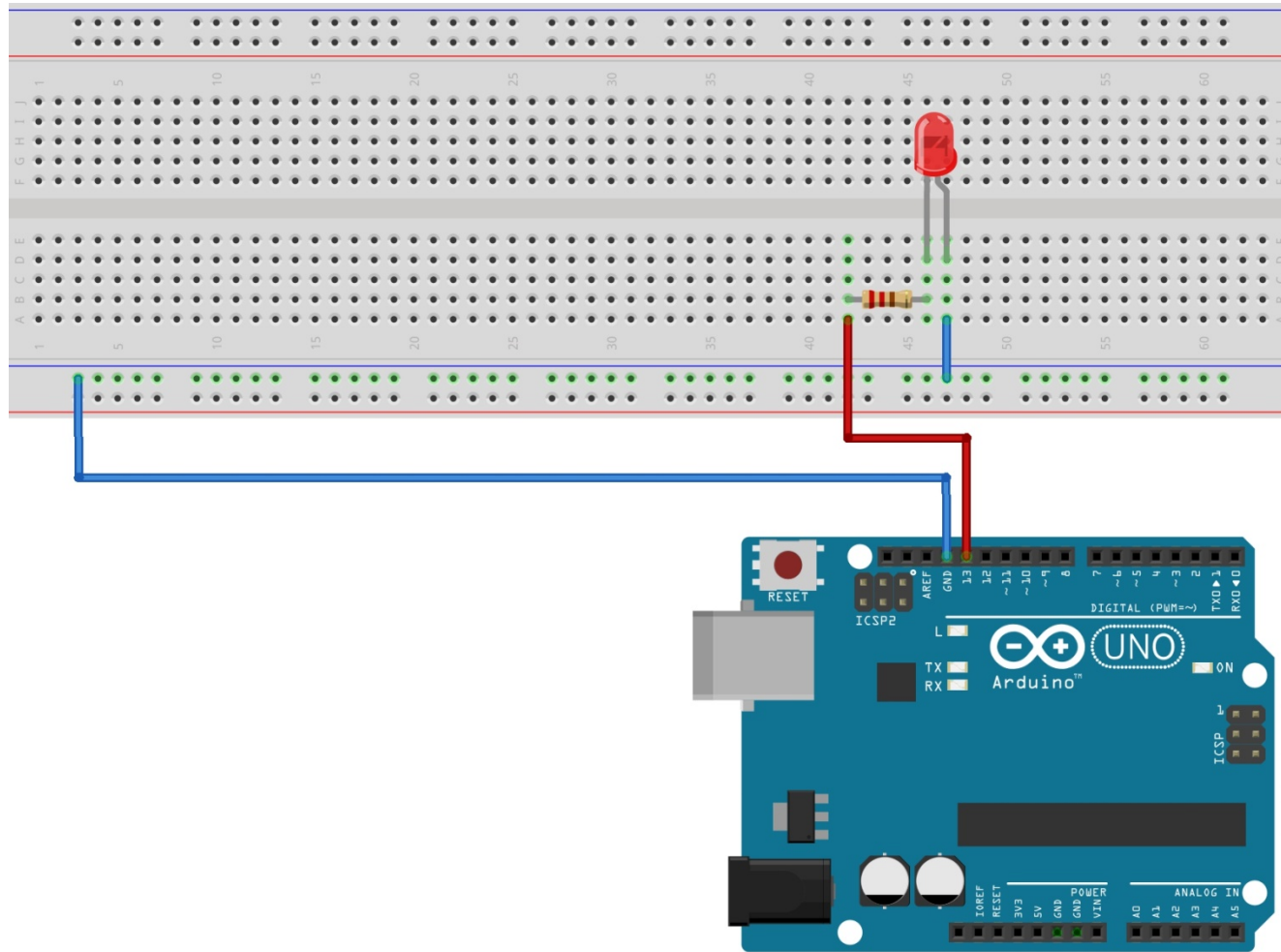
I = corriente.



*<http://www.instructables.com/id/Led-Y-Resistencia/>



Ejemplo del circuito Blink



Elaborado con: fritzing



Reto:

Construir y programar un circuito que simule el funcionamiento de un semáforo.

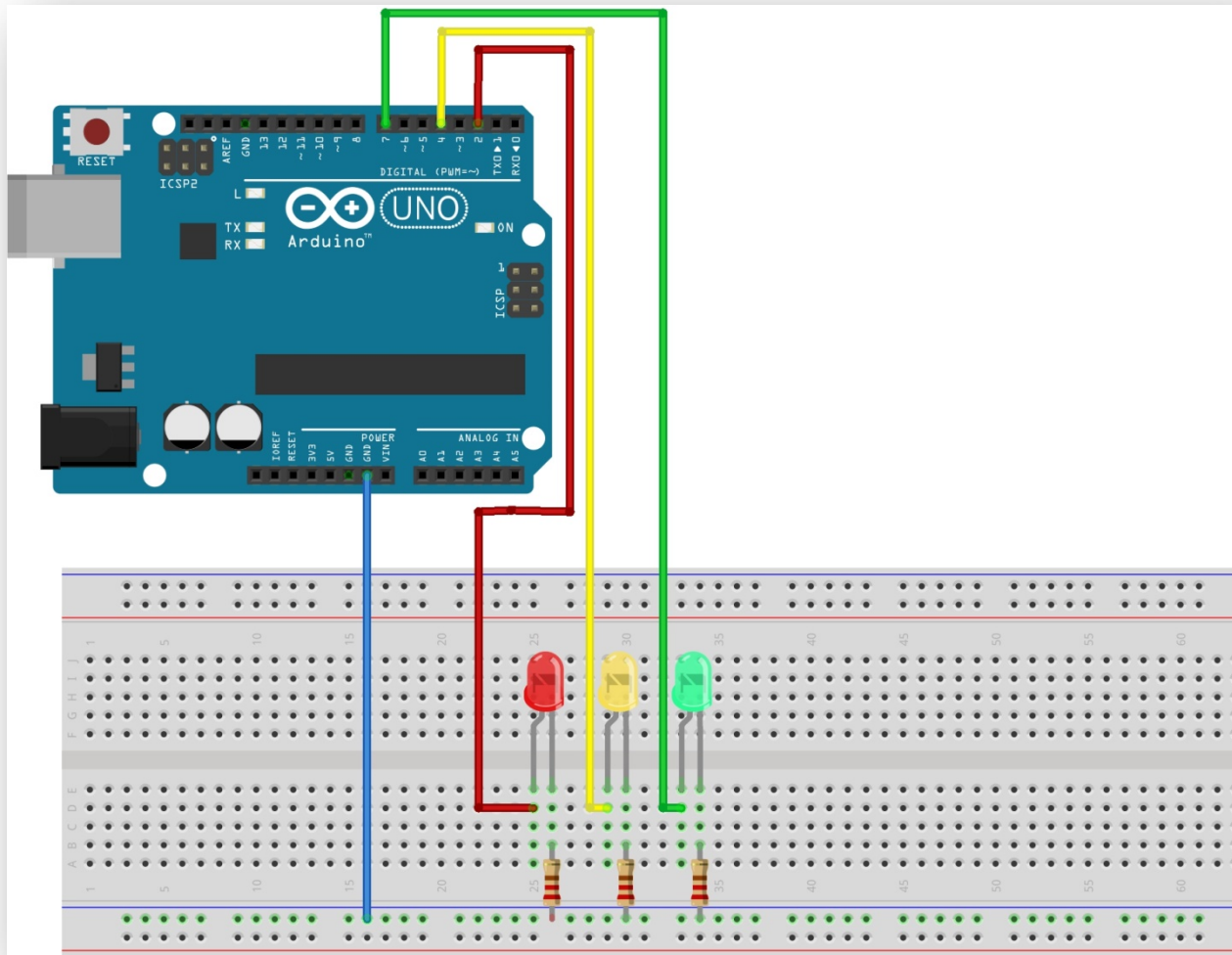


XVII CONGRESO INTERNACIONAL
Innovación y Tecnología en Educación a Distancia



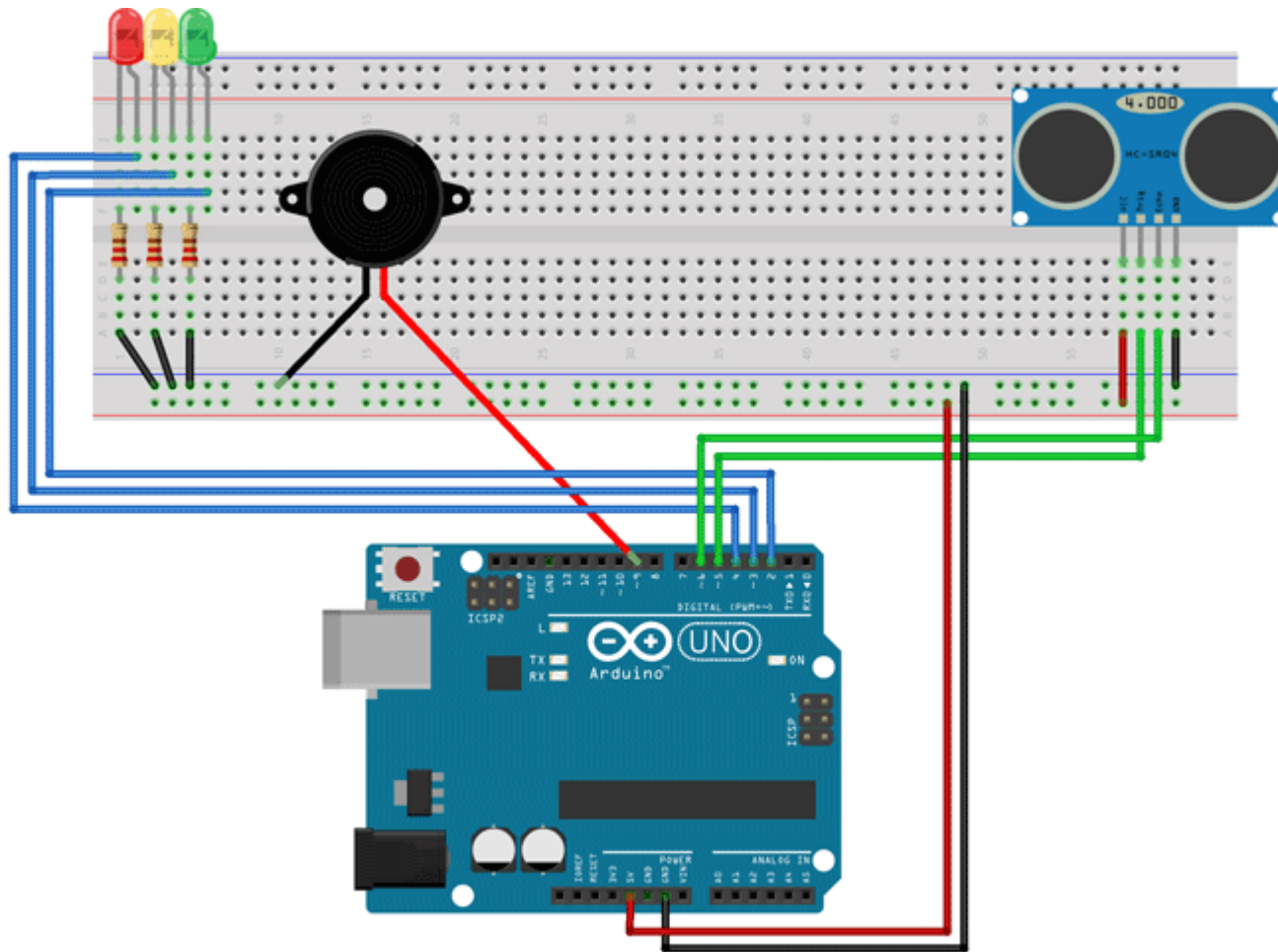
40 Caniversario

Ejemplo del circuito Semáforo



Elaborado con: fritzing

Ejemplo de circuito de seguridad



fritzing

Elaborado con: fritzing

Arduino y matemáticas

Operadores generales

= (operador de asignación)
+ (adición) - (sustracción)
* (multiplicación)
/ (división) % (módulo)
== (igual a) != (desigual a)
< (menor que) > (mayor que)
<= (igual o menor que)
>= (mayor o igual que)
&& (y) || (ó) ! (negación)

Operadores compuestos

++ (incremento)
-- (decremento)
+= (suma compuesta)
-= (resta compuesta)
*= (multiplicación compuesta)
/= (división compuesta)
&= (AND binario compuesto)
|= (OR binario compuesto)

Operadores a nivel de bit

& (AND binario) | (OR binario)
^ (XOR binario) ~ (NOT binario)

Estructuras de control

<i>if / else</i>	<i>For</i>	<i>While</i>	<i>switch / case</i>
<pre> if (condición) { // acción A } else { // acción B } <u>o también:</u> if (condición) { // acción A } else if (condición) { // acción B } else if { // acción C } </pre>	<pre> for (Inicializacion; condicion; incremento) { // acción o acciones; } <u>Ejemplo.</u> parenthesis declare variable (optional) initialize test increment or ↓ ↓ ↓ for(int x = 0; x < 100; x++){ println(x); // prints 0 to 99 } </pre>	<pre> while(expression) { // acción o acciones; } <u>Ejemplo.</u> var = 0; while (var < 200) { // repite 200 veces var++; } </pre>	<pre> switch (var) { case 1: // acción break; case 2: // acción break; default: // opcional si nada cumple break; } </pre>

www.tinkercad.com

TINKERCAD Neat Luulia-Elzing All changes saved

Code Editor Components Start Simulation Export Share

Arduino Uno R3
Name 1

1 (Arduino Uno R3) Upload & Run Block Libraries Download Code Debugger Serial Monitor

```
2 // give it a name:
3 int led = 13;
4
5 // the setup routine runs once when you press reset:
6 void setup() {
7   // initialize the digital pin as an output.
8   pinMode(led, OUTPUT);
9 }
10
11 // the loop routine runs over and over again forever:
12 void loop() {
13   digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
14   delay(1000);             // wait for a second
15   digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
16   delay(1000);             // wait for a second
```

Sitios y recursos de interés

<https://www.arduino.cc>

<http://fritzing.org/>

www.tinkercad.com

<https://www.instructables.com>

s4a.cat

<http://scratchx.org>

bitbloq.bq.com

www.senseitech.org



Muchas gracias por su atención y participación



XVII CONGRESO INTERNACIONAL
Innovación y Tecnología en Educación a Distancia



40 Caniversario