



# Tutorial Scratch para Arduino (S4A)

S4A es una modificación de Scratch que permite programar la plataforma de hardware libre Arduino.

---

COLECCIÓN DE APLICACIONES GRATUITAS PARA CONTEXTOS EDUCATIVOS

---

Plan Integral de Educación Digital

Gerencia Operativa Incorporación de Tecnologías (InTec)

[buenosaires.gob.ar/educacion](http://buenosaires.gob.ar/educacion)   [educacionBA](#)



**Buenos Aires Ciudad**

Ministerio de Educación del Gobierno de la Ciudad de Buenos Aires  
01-02-2026



**EN TODO ESTÁS VOS**

## Prólogo

Este tutorial se enmarca dentro de los lineamientos del Plan Integral de Educación Digital (PIED) del Ministerio de Educación del Gobierno de la Ciudad Autónoma de Buenos Aires que busca integrar los procesos de enseñanza y de aprendizaje de las instituciones educativas a la cultura digital.

Uno de los objetivos del PIED es “fomentar el conocimiento y la apropiación crítica de las Tecnologías de la Información y de la Comunicación (TIC) en la comunidad educativa y en la sociedad en general”.

Cada una de las aplicaciones que forman parte de este banco de recursos son herramientas que, utilizándolas de forma creativa, permiten aprender y jugar en entornos digitales. El juego es una poderosa fuente de motivación para los alumnos y favorece la construcción del saber. Todas las aplicaciones son de uso libre y pueden descargarse gratuitamente de Internet e instalarse en cualquier computadora. De esta manera, se promueve la igualdad de oportunidades y posibilidades para que todos puedan acceder a herramientas que desarrollen la creatividad.

En cada uno de los tutoriales se presentan “consideraciones pedagógicas” que funcionan como disparadores pero que no deben limitar a los usuarios a explorar y desarrollar sus propios usos educativos.

La aplicación de este tutorial no constituye por sí misma una propuesta pedagógica. Su funcionalidad cobra sentido cuando se integra a una actividad. Cada docente o persona que quiera utilizar estos recursos podrá construir su propio recorrido.

# Índice

¿Qué es? .....	4
Requerimientos técnicos.....	4
Consideraciones pedagógicas .....	4
Nociones básicas.....	5
▪ Descarga e instalación última versión de Arduino.....	5
▪ Verificación de conexión Arduino en Windows .....	6
▪ Descarga e instalación de S4A.....	7
▪ Descarga e instalación del firmware S4A.....	8
▪ Instalación del <b>firmware</b> dentro de Arduino .....	9
▪ Verificación del correcto funcionamiento de S4A .....	10
▪ Entorno de Scratch para Arduino.....	13
▪ Objeto Arduino .....	14
▪ Detalle de entradas/salidas en S4A.....	14
▪ Bloques específicos de S4A .....	15
▪ Conocimiento de la electrónica Arduino .....	16
▪ Ejercicios prácticos .....	19
Enlaces de interés .....	25

## ¿Qué es?

**S4A** es una versión modificada de Scratch preparada para interactuar con Arduino. Incorpora nuevos bloques para utilizar sensores y actuadores.

## Requerimientos técnicos

- Puede utilizarse bajo los sistemas operativos Windows y GNU/Linux.
- Placa Arduino
- Cable USB para conectar la placa al equipo.
- Software Arduino (URL de la aplicación: <http://www.arduino.cc>)
- Software S4A (instalador + *firmware*: <http://www.s4a.com>)

## Consideraciones pedagógicas

**Nivel educativo:** Primario - Medio

**Áreas:** todas

- Es posible crear interacciones entre el mundo real (tangible) mediante sensores y actuadores y el mundo virtual (intangibles) dibujos, sonidos, gráficos, cálculos, historias, videojuegos.
- La construcción de estas interacciones mediante la programación en bloques se complementa con el uso de elementos físicos, otorgando una experiencia de adquisición de conocimientos más completa y atractiva.
- Es posible desplegar estas interacciones dentro y fuera de la pantalla, pudiendo realizar montajes donde intervengan sincronizadamente elementos tales como luces de led, potenciómetros, sensores (de presencia, distancia, humedad, tacto, temperatura), pantallas, proyecciones, maquetas.

## Funcionalidades

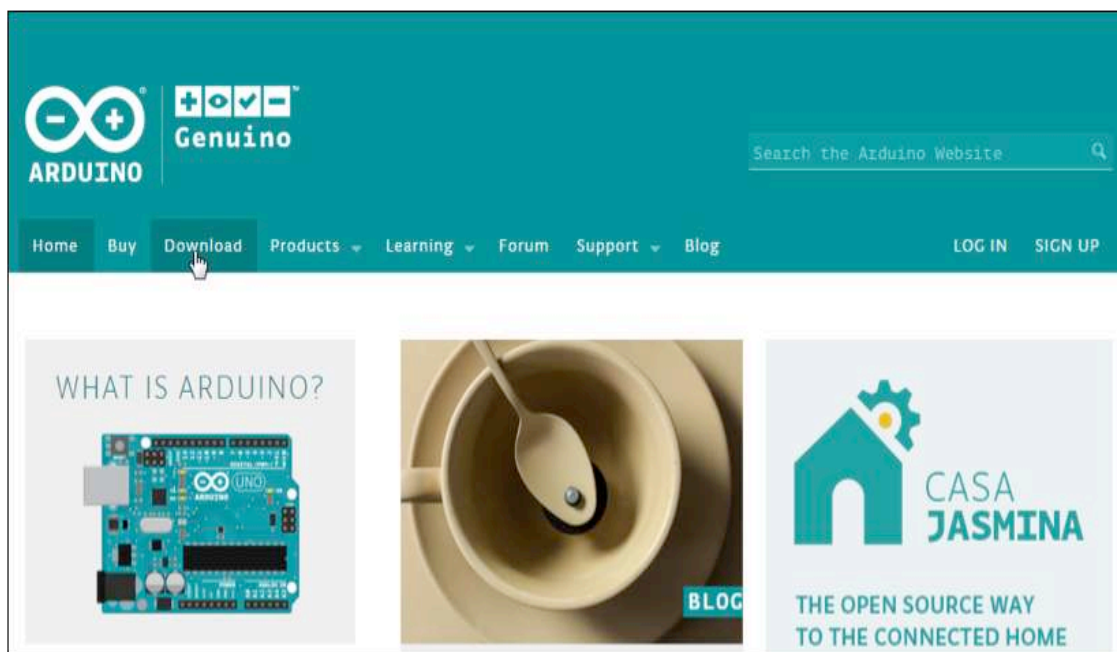
S4A brinda las siguientes funcionalidades:

- Obtener información en pantalla mediante un sensor externo.
- Controlar elementos físicos mediante actuadores externos a través de operaciones en el *software*.

## Nociones básicas

### ► Descarga e instalación última versión de Arduino

Es posible descargar el instalador del programa desde la página oficial de Arduino [www.arduino.cc](http://www.arduino.cc). Es posible encontrarla dentro de la sección **Download**.



Una vez descargado el archivo, será necesario instalar el programa **Arduino**. Más detalles de este procedimiento se encuentran disponibles en el [tutorial de Arduino](#) en Integrar.

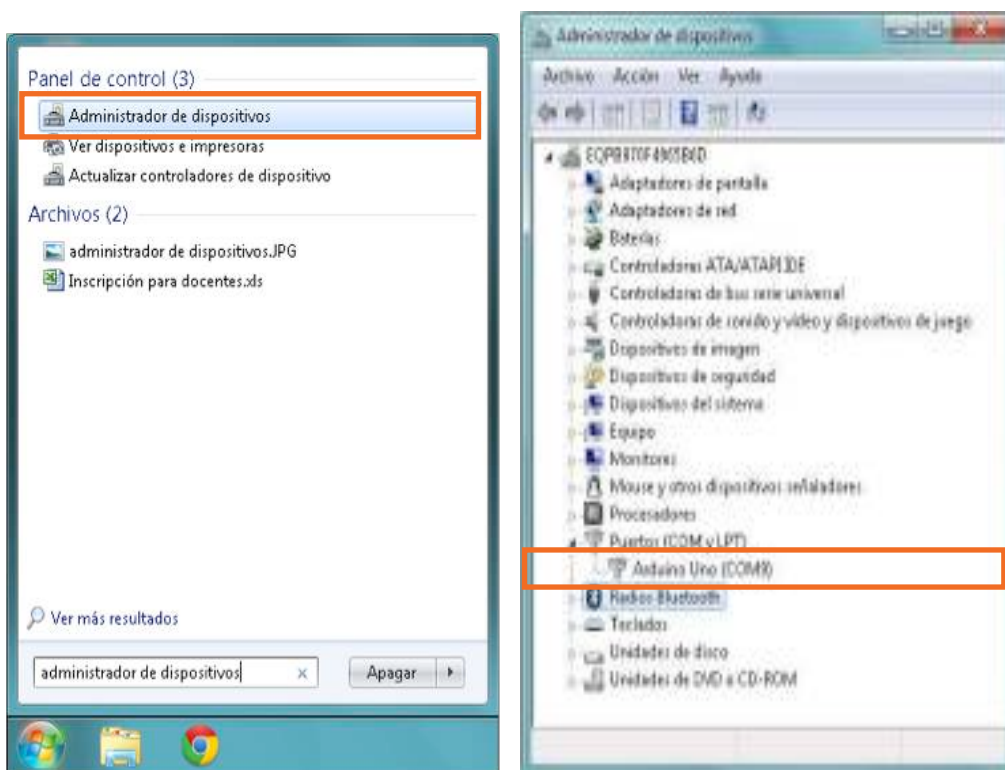
**Nota:** Se recomienda realizar esta instalación a fin de mantener actualizado el entorno de desarrollo (IDE) de Arduino, como así también obtener los últimos drivers necesarios para que el sistema operativo reconozca sin inconvenientes cualquier tipo de placa Arduino al conectarla mediante el cable USB.

### ► Verificación de conexión Arduino en Windows

Conectar la placa Arduino a la computadora mediante el cable USB. Verificar el estado encendido del LED de power.



Abrir el **Administrador de Dispositivos** en el **menú Inicio**. Verificar la aparición de la placa dentro de la sección **Puertos**.



Es importante recordar el número de puerto de comunicación asignado (en el ejemplo de la imagen, COM9). Esto será útil posteriormente para verificar la configuración dentro del *software* de Arduino.

### ► Descarga e instalación de S4A

El instalador se encuentra en el apartado **Descargas** de la página web [http://s4a.cat/index\\_es.html](http://s4a.cat/index_es.html)



**Nota:** Verificar si se dispone de la última versión de S4A instalada en el equipo. Si existe una versión anterior, se recomienda desinstalarla como paso previo antes de instalar la versión más reciente encontrada en la web de S4A.

Proseguir a la descarga e instalación del programa S4A.



## ► Descarga e instalación del firmware S4A

El *firmware* de S4A se encuentra disponible para su descarga, en la misma sección web de descargas del programa S4A [http://s4a.cat/index\\_es.html](http://s4a.cat/index_es.html)

### Instalar el firmware en el Arduino

Este firmware es un programa que debe instalarse en vuestra placa Arduino para poderos comunicar con ella desde S4A.

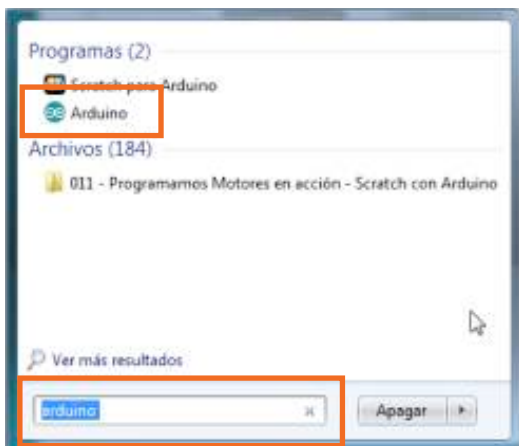
- Descargad e instalad el entorno Arduino siguiendo las instrucciones de <http://arduino.cc/en/Main/Software>. Hay que tener en cuenta que, para trabajar con Arduino Uno, se necesita la versión 0022 o superior.
- Descargad nuestro firmware [aquí](#)
- Conectad la placa Arduino a un puerto USB de vuestro ordenador
- Abrid el archivo de firmware (S4AFirmware15.ino) desde el entorno Arduino
- En el menú Herramientas, seleccionad la versión de la placa y el puerto serie en qué esté conectada la placa
- Cargad el firmware a la placa mediante la opción de menú Archivo > Subir

La extensión del archivo del *firmware* es **.ino** ya que se trata de un archivo con código Arduino para cargar en la placa mediante el *software* de Arduino. Para bajar este código, seleccionar el vínculo con el botón derecho del mouse utilizando la opción **Guardar enlace como**, para así poder guardar el archivo en el disco local para su uso con el programa Arduino. De otra forma, el código se abriría en el mismo navegador mostrándose como texto.



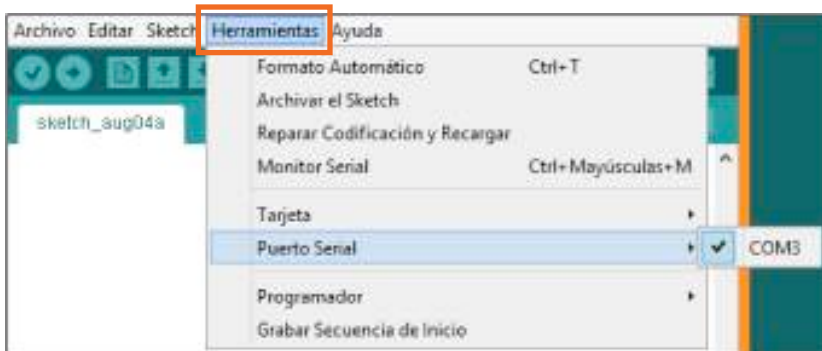
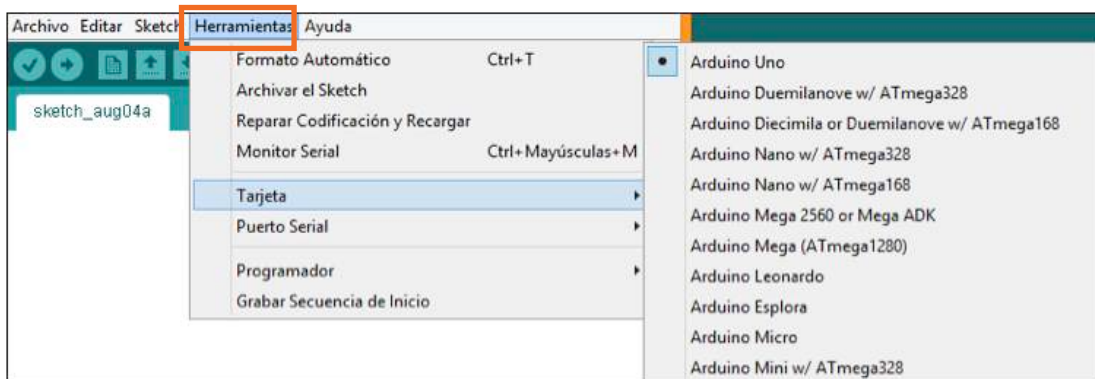
## ► Instalación del *firmware* dentro de la placa Arduino

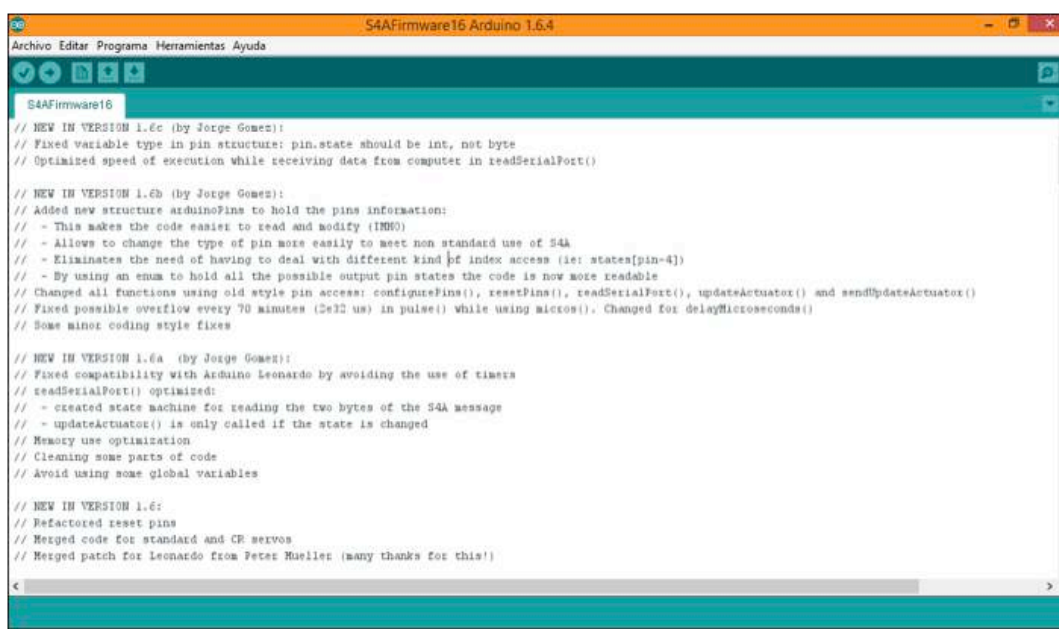
Abrir el programa **Arduino**.



Verificar siempre que el mismo se encuentre configurado correctamente para comunicarse con la placa eligiendo en el menú **Herramientas**:

- **en Placa >** debe figurar el tipo de placa Arduino que se utilice.
- **en Puerto >** debe figurar el puerto de comunicación mencionado anteriormente.





```

S4AFirmware16
// NEW IN VERSION 1.6c (by Jorge Gomez):
// Fixed variable type in pin structure: pin.state should be int, not byte
// Optimized speed of execution while receiving data from computer in readSerialPort()

// NEW IN VERSION 1.6b (by Jorge Gomez):
// Added new structure ArduinoPins to hold the pins information:
// - This makes the code easier to read and modify (IMHO)
// - Allows to change the type of pin more easily to meet non standard use of S4A
// - Eliminates the need of having to deal with different kind of index access (ie: states[pin-4])
// - By using an enum to hold all the possible output pin states the code is now more readable
// Changed all functions using old style pin access: configurePins(), resetPins(), readSerialPort(), updateActuator() and sendUpdateActuator()
// Fixed possible overflow every 70 minutes (2e32 us) in pulse() while using micro(). Changed for delayMicroseconds()
// Some minor coding style fixes


// NEW IN VERSION 1.6a (by Jorge Gomez):
// Fixed compatibility with Arduino Leonardo by avoiding the use of timers
// readSerialPort() optimized:
// - created state machine for reading the two bytes of the S4A message
// - updateActuator() is only called if the state is changed
// Memory use optimization
// Cleaning some parts of code
// Avoid using some global variables

// NEW IN VERSION 1.6:
// Refactored test pins
// Merged code for standard and CE servos
// Merged patch for Leonardo from Peter Mueller (many thanks for this!)

```

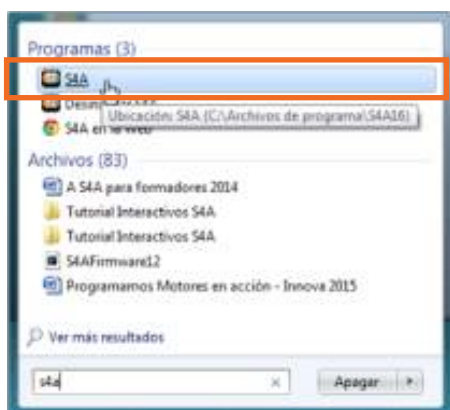
A continuación se deberá cargar el código del *firmware* de S4A dentro de la placa Arduino.

En el programa Arduino seleccionar del menú **Archivo** la opción **Abrir...** y elegir el **archivo .ino** del *firmware* descargado.

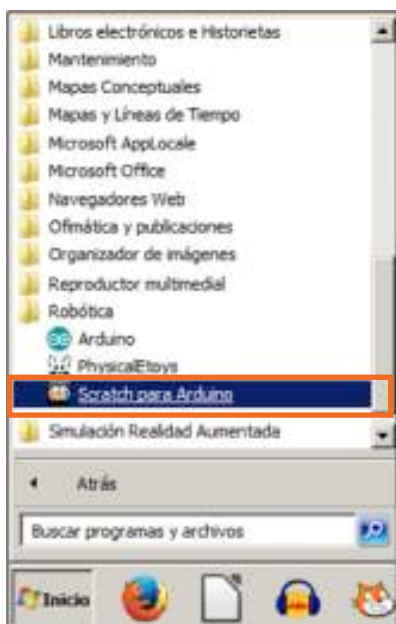
Luego proceder a la carga del *firmware* dentro de la placa. Desde el menú **Archivo** seleccionar la opción **Subir** (o bien desde el botón de carga )

### ► Verificación del correcto funcionamiento de S4A

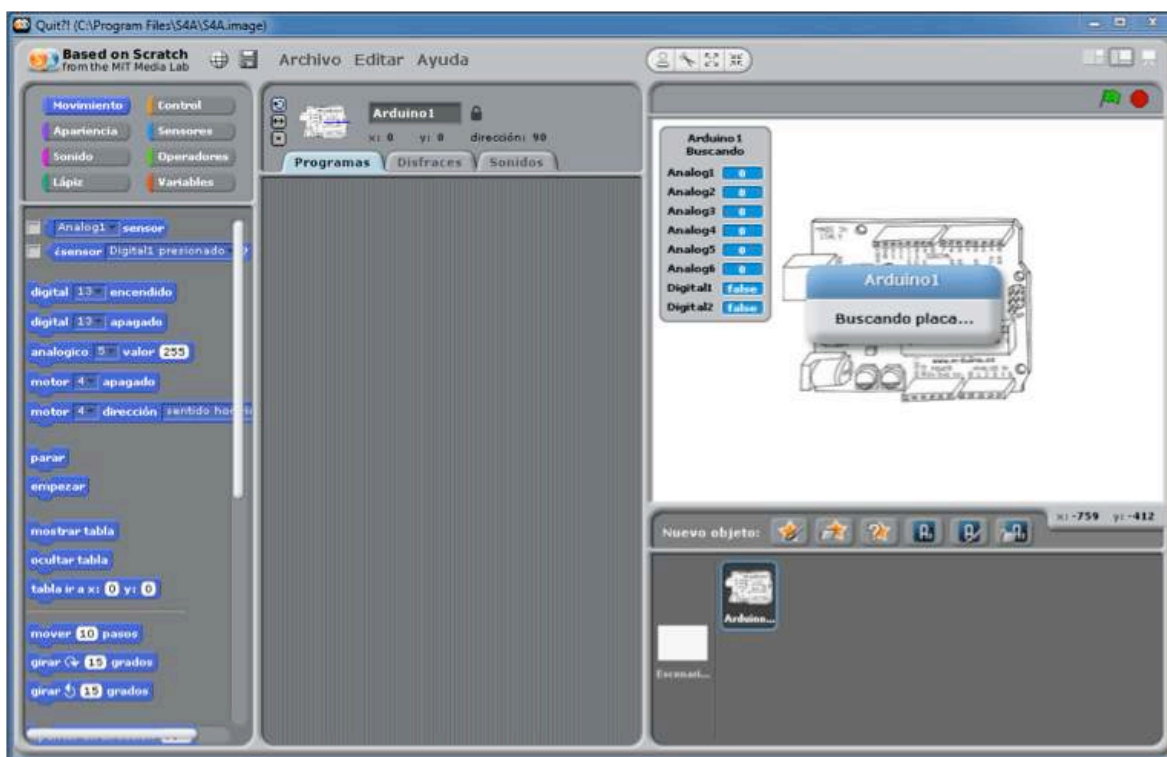
Para comenzar a trabajar es necesario abrir el programa **S4A** desde el menú **Inicio**. Es posible hallar el programa tras escribir S4A en el campo de búsqueda.



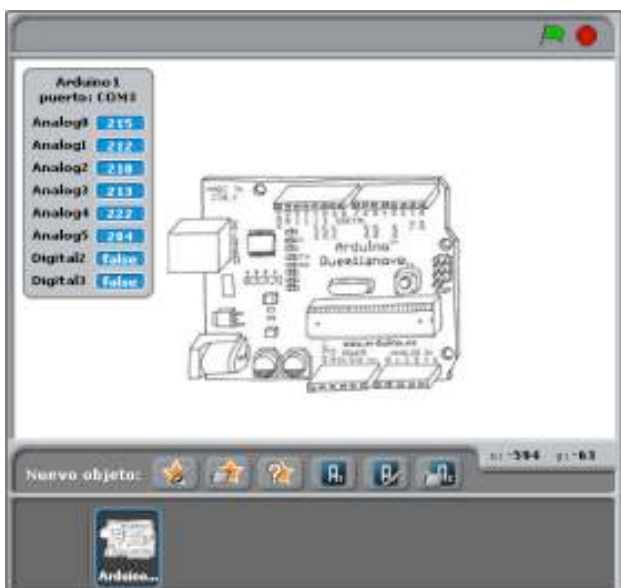
En el caso de los equipos del Plan Sarmiento se encuentra el programa dentro de la categoría **Robótica**.



Cuando se abre el programa puede observarse la siguiente ventana:

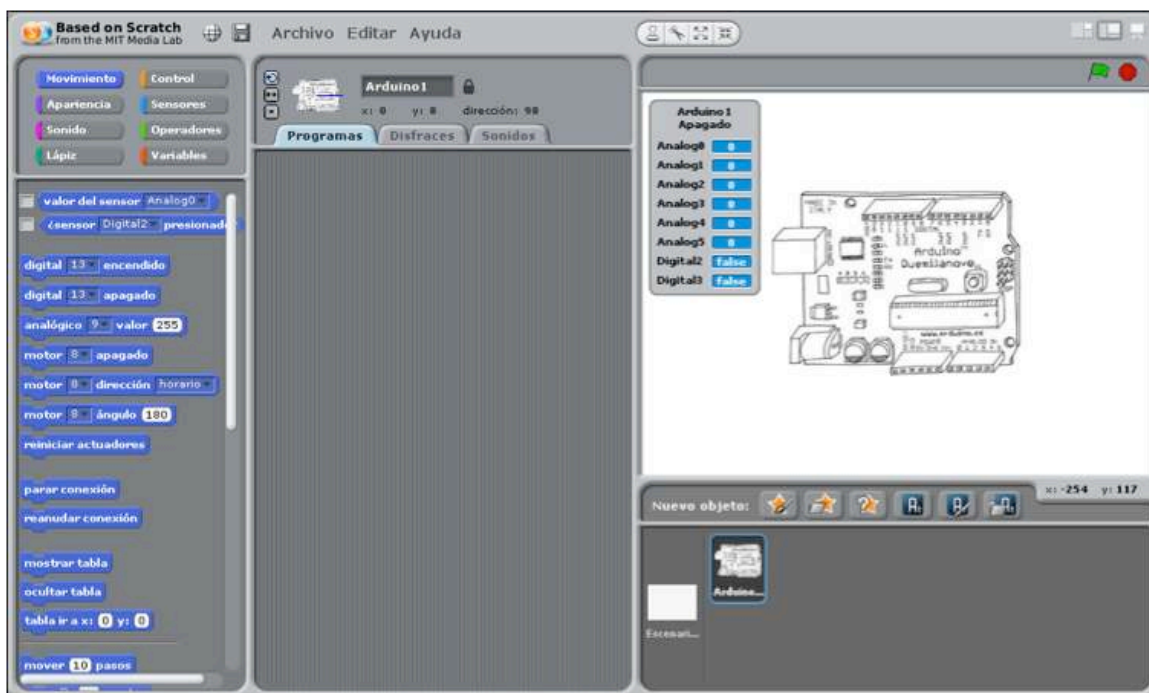


El programa buscará la presencia de una placa Arduino conectada a la computadora. Se observará la carga por defecto de un **Objeto Arduino** en lugar de la tradicional figura del gatito Scratch.



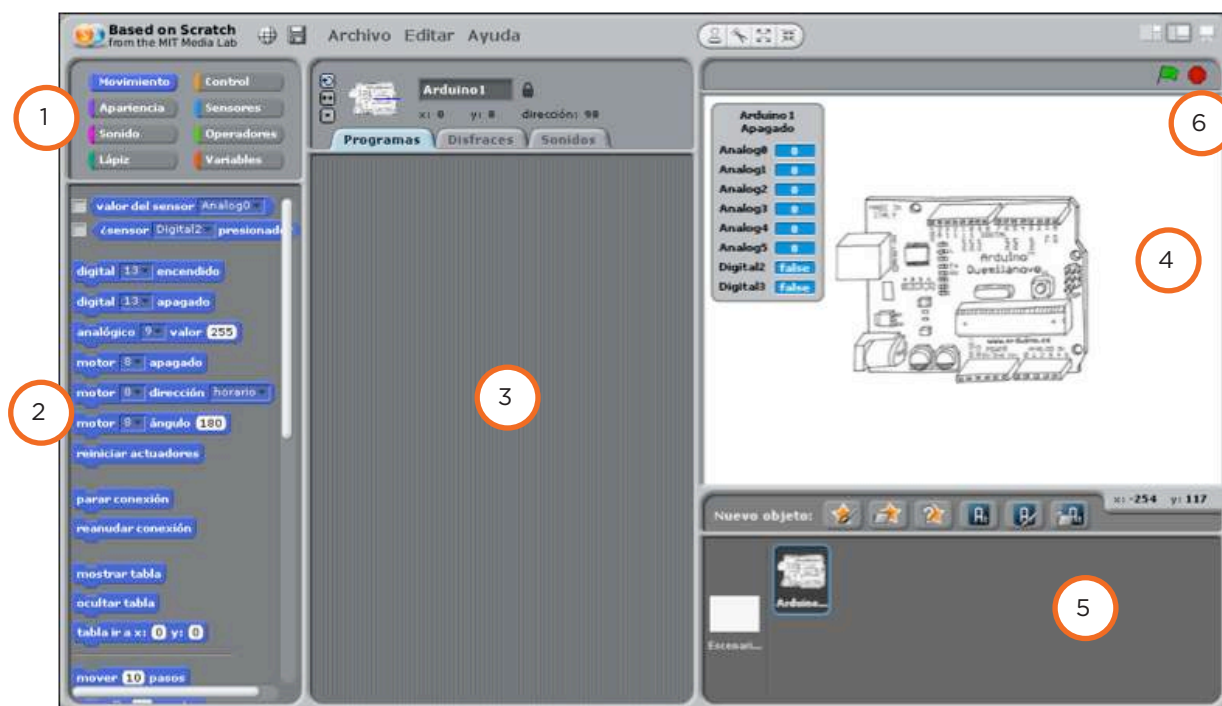
Cuando S4A detecta una placa Arduino conectada, mostrará instantáneamente una tabla con los valores detectados en sus puertos.

Al conectar la placa Arduino a la computadora mediante el cable USB se observará la siguiente imagen:



Comenzará a observarse la lectura de datos que provienen de los pines analógicos y digitales. Esta tabla, puede ocultarse o mostrarse a voluntad en forma total o parcial. S4A permite conectar más de una placa (diferentes “objetos Arduino”)

### ► Entorno de Scratch para Arduino

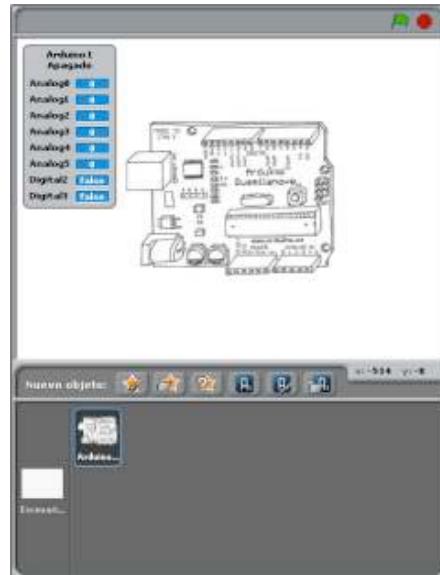


1. Categorías de bloques
2. Bloques
3. Área del código de programación
4. Área de escenario
5. Área de objetos, incluidos a los objetos Arduino.
6. Iniciar/detener el programa

Una de las principales diferencias que se observan respecto al Scratch regular es la presencia del objeto Arduino dibujando en el centro del área de **Escenario**, en lugar del gatito.

## ► Objeto Arduino

Se puede observar también que existen tres íconos adicionales para crear este tipo de objetos llamados **Arduino**.



A este objeto se le puede editar su diseño o cambiar de disfraz, alterando su apariencia. De acuerdo a la programación que se realice, puede coexistir más de un **objeto Arduino** dentro del programa.

A cada **objeto Arduino** le corresponderá una **placa Arduino** vinculada.

Cada **objeto Arduino** trae consigo una **tabla de valores** que muestra el número de placa Arduino conectada, su estado y los valores digitales y analógicos recibidos en sus respectivos pines.

## ► Detalle de entradas/salidas en S4A

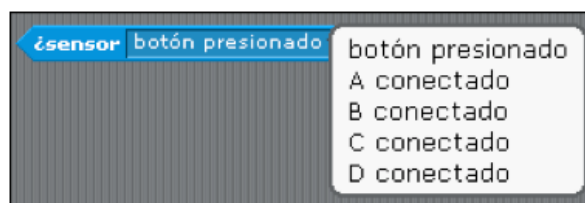
- Salidas digitales (pines digitales 10,11 y 13)
- Salidas analógicas (pines digitales 5, 6 y 9)
- Entradas analógicas (pines 0,1,2,3,4,5)
- Entradas digitales (pines digitales 2 y 3)
- Servomotores RC (pines digitales 4, 7, 8 y 12)



## ► Bloques específicos de S4A

S4A cuenta con bloques adicionales a Scratch, específicos para el trabajo con la placa Arduino, los cuales pueden identificarse en dos grupos: de entrada y de salida de datos.

Dentro del grupo de **Sensores** se encuentran los siguientes bloques de detección de entradas de Arduino.



Dentro del **grupo Movimiento** se encuentran los siguientes **bloques de control** de salidas de Arduino



Nota: en electrónica, las **entradas/salidas digitales** corresponden a la presencia o no de energía sobre el PIN al que se refiere el *software* (encendido o apagado). Ej.: 5 o 0 voltios.

En el caso de las **entrada/salidas analógicas**, en S4A se refiere a un valor que puede hallarse dentro del rango de valores entre los 0 y los 5 voltios. Ej.: 1,0273 voltios. Expresado en S4A mediante valores enteros.

## ► Conocimiento de la electrónica Arduino



La imagen muestra uno de los modelos más populares de la placa de electrónica experimental Arduino. Existen diferentes modelos de placa Arduino compatibles.

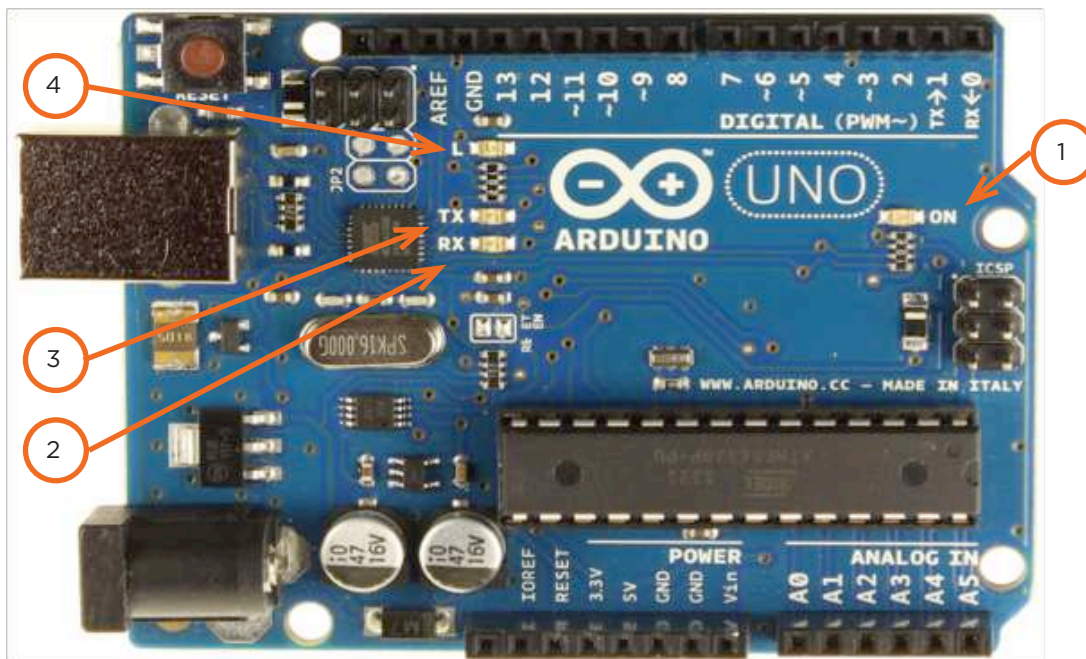
### • Voltaje

Utiliza energía eléctrica de baja potencia, la cual puede ser suministrada al conectarse el cable de transferencia de datos USB o bien utilizando un conector con un portapilas. La unidad de medida de potencia eléctrica es el voltio. La potencia máxima que utiliza Arduino es de 5 voltios. Puede utilizarse una batería de 9 voltios, cuyo voltaje será regulado por la placa a 5v en la entrada.

### • Leds

La placa cuenta con algunos **leds indicadores**.

1. El led **"ON"** encendido, indica si la placa tiene alimentación de energía.
2. El led **"RX"** enciende intermitentemente, indica que la placa está recibiendo datos.
3. El led **"TX"**, enciende intermitentemente, indica que la placa está emitiendo datos.
4. El led **"L"**, es un led regular que está conectado mediante una resistencia al pin 13 en forma permanente a la placa, de modo que puedan realizarse pruebas básicas de programación (conocidas como "Hola mundo") tales como encender y apagar una luz (**"led tilt"**).



### • Pines

Se denominan **pines** a los orificios donde es posible conectar las clavijas de los componentes o cables conductores. Pueden identificarse principalmente tres áreas de pines, tal como menciona la descripción impresa sobre la placa.

#### **Pines de alimentación o Power.**

- Pines **Digitales**.
- Pines **Analógicos**.
- Pines de Tierra: equivalen al polo negativo de la batería. Están indicados en la placa como **GND** (del inglés *ground*, que significa tierra).

Nota: en las placas Arduino se encuentran varios pines señalizados como tierra bajo la sigla GND. Todos estos pines están interconectados entre sí, por lo cual a la hora de seleccionar donde conectar a tierra el negativo de algún elemento como ser un motor o un led, será indistinto seleccionar cualquiera de los pines de tierra (GND).

Ejemplo: Si se quiere conectar un elemento actuador tal como un motor, una sirena o una lámpara de bajo voltaje, por un lado se conectará a tierra (GND) y por otro a un pin de salida de energía, ya sea analógico o digital según el caso.

- **Componentes suplementarios**

S4A permite trabajar con componentes adicionales los cuales trabajan como sensores o sensores dentro del ambiente real.

→ **Sensores**

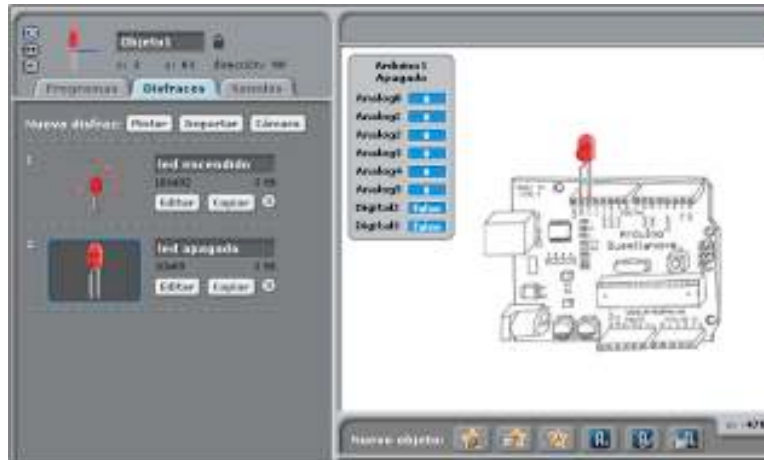
- **Interruptores/pulsadores.** Pueden ser utilizados como sensores de toque.
- **Luz.** Usualmente se utilizan foto resistencias. Similares a las resistencias pero varían su valor como si se tratara de un potenciómetro, de acuerdo a la luz recibida.
- **Temperatura.** Estos componentes electrónicos pueden ser análogos o digitales. Algunos también miden humedad.
- **Audio.** Pueden ser micrófonos o piezoeléctricos.
- **Proximidad.** Usualmente miden distancia calculando el tiempo de eco al emitir/recibir un pulso de audio o de luz.
- **Acelerómetro.** Es utilizado para detectar giros, inclinaciones y cambios de movimiento. Si es de tres ejes de coordenadas, proveerá información analógica en tres valores.

→ **Actuadores**

- **Motores.** Usualmente de corriente continúa. Estos invierten su giro al cambiar la polaridad.
- **Servos.** Los más utilizados en este caso se alimentan en sus bornes positivo y negativo. Mientras que un tercer borne recibe información de pulsos digitales con indicación sobre la dirección y el ángulo que debe conservar.
- **Luminancias.** Pueden ser de filamento tradicional (sin polaridad) o bien tipo LED. En éstos últimos debe respetarse la polaridad de sus pines positivo/negativo. Una principal ventaja de los LEDs es que emiten más luz con menor consumo de energía.

## ► Ejercicios prácticos

### 1. Encendido y apagado de un led ("hola mundo!")



Este programa brinda instrucciones a la placa para que encienda y apague la salida de energía del pin 13, esperando un segundo en cada período.



Nota: La mayoría de las placas Arduino compatibles traen consigo un led conectado en forma permanente sobre el pin 13. Es posible hacer una prueba de funcionamiento del sistema mediante este programa visualizando la luz emitida por el led que trae soldado consigo la placa.

## 2. Salida intermitente con visualización de estado en pantalla y control utilizando la tecla espaciadora como pulsador.

Esta alternativa al programa anterior, refleja el estado de la luz led virtualmente, mediante un gráfico en pantalla.



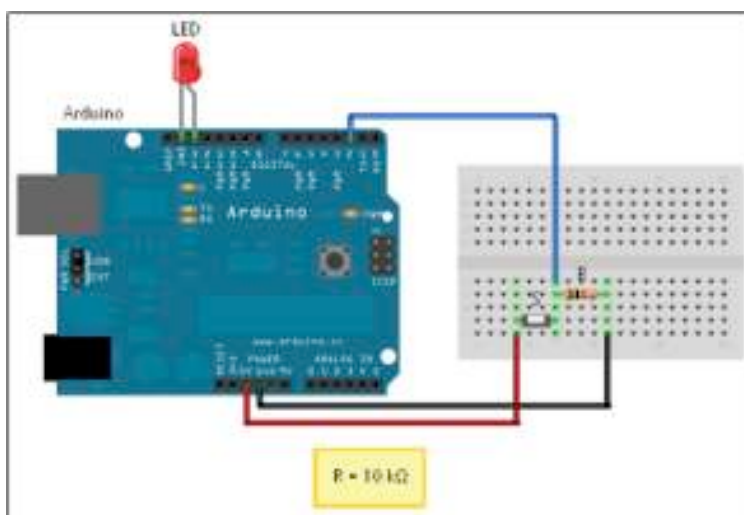
Este diagrama de programación presenta un condicional según la tecla espaciadora se encuentre o no presionada e indica también un cambio de disfraz de acuerdo a su estado.

Las teclas que se definan en el programa funcionan como pulsadores. En caso de que no se presione la tecla indicada, el led permanecerá apagado.

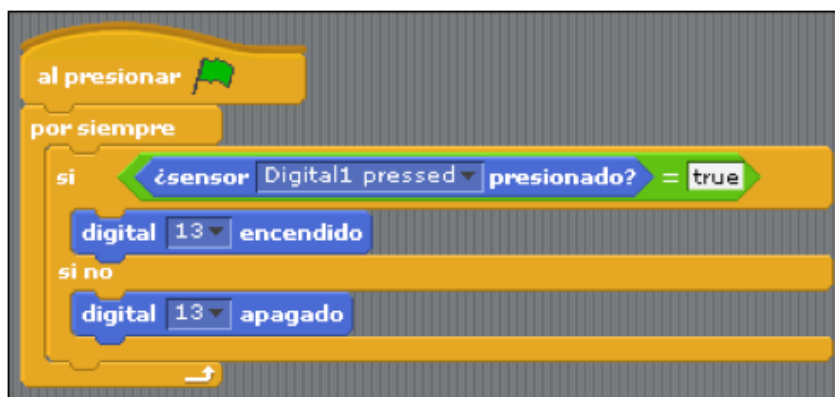
## 3. Control mediante pulsador

El diagrama muestra como conectar un pulsador (S) mediante una resistencia de ( $R=10k\Omega$ ) al pin correspondiente.

Para mejorar la visibilidad del led (L) ya integrado a la placa, se puede conectar un led en paralelo, es decir entre el pin 13 y GND.



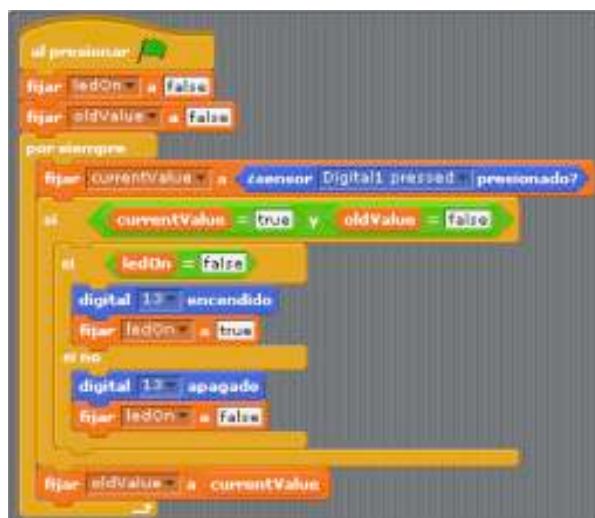




El código representado determina que desde el inicio del programa, de manera continua, si el sensor Digital1 está presionado (true) el pin digital 13 tendrá energía. De lo contrario, se mantendrá apagado.

#### 4. Control mediante pulsador con memoria de estado

Este diagrama muestra una variación de la propuesta anterior.

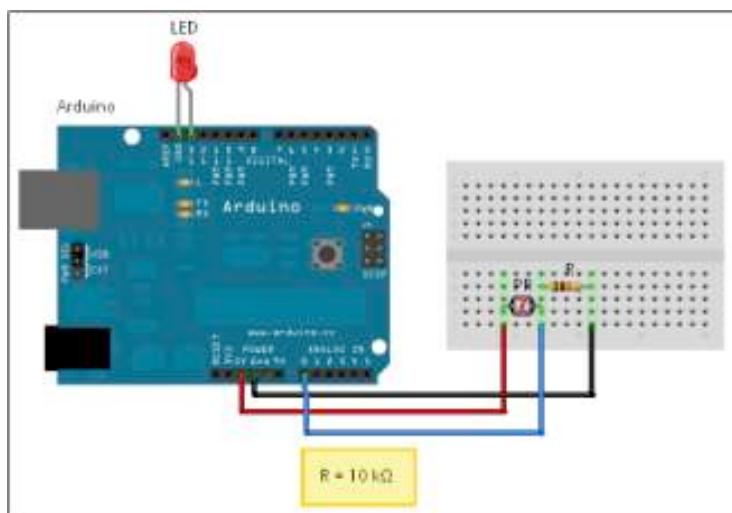


Añadiendo las variables de estado **ledOn**, **oldValue** y **currentValue**, se genera una memoria de conservación del último estado transformando el mismo pulsador en un interruptor.

#### 5. Comandar el estado de una salida digital mediante un control analógico

En la web de S4A se encuentran algunos ejemplos con la posibilidad de bajar el código.

Puede observarse uno que muestra la posibilidad de establecer un valor SI/NO (en este caso encendido / apagado) respecto a un valor fluctuante tomado a partir de un sensor analógico.



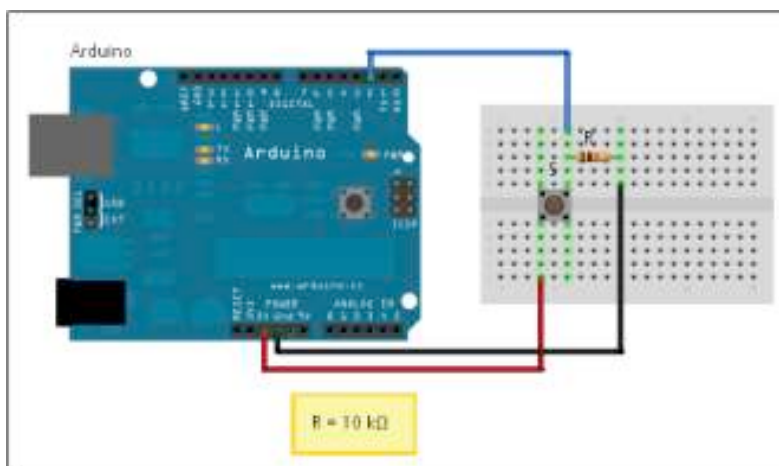
En el código de bloques, se establece que la variable llamada **valor** sea siempre igual al valor que brinde el sensor de la placa. En caso de que este número iguale o supere el número 800, el led digital encenderá. En caso contrario se mantendrá apagado.

Respecto a la conexión electrónica, se aprecia la similitud con la manera en la que se conecta un pulsador o interruptor. Solo que al brindar un rango de valores se utilizará un pin analógico para censar información entrante.

## 6. Realizar un contador

Los contadores son formas útiles de generar interactividad con S4A, ya sea contando elementos detectados al pasar o tocar un nivel, generando estadísticas en pantalla o realizando juegos interactivos.

En la web de S4A, se encuentran documentados ejemplos básicos del uso de S4A. Uno de ellos es **el pulsador**. Es posible desarrollar diferentes tipos de contadores utilizando este mismo principio de conexión.



En este caso se cuentan los impulsos (uso del pulsador, toques) entrantes al pin **Digital2**.



A partir del valor generado en la variable **Cuenta**, pueden establecerse acciones, como por ejemplo si el valor **Cuenta** supera un número establecido, realizar una acción.

Estas acciones pueden ser incrementar distancias, cambiar colores, sonidos o disfraces.

El bloque final de color violeta se utiliza para evitar que el sensor sume un valor extra en forma involuntaria si el sensor volviera a accionarse en forma inmediata.

- **Puesta a cero del contador**

En la mayoría de los casos resulta conveniente contar con un botón de puesta a cero del contador.



El presente diagrama de bloques muestra la forma de hacerlo, volviendo a poner en cero la variable **Cuenta**.

Esta acción podría realizarse al presionar una tecla, o bien como muestra la imagen, adicionando un segundo pulsador en este caso sobre el pin **Digital3**.

- **Sumar y restar valores al contador**

Es posible incrementar o restar valores a la variable creada, como se aprecia en el diagrama de bloques.



Estos son solo algunos de los ejemplos posibles para realizar. Aplicándolos a diferentes proyectos brindan la posibilidad de añadir interactividad a videojuegos, proyecciones o maquetas.

## Enlaces de interés

Sitio oficial software Arduino: <http://www.arduino.cc>

Sitio oficial S4A <http://www.s4a.com>

Contacto:

[ayuda.pedagogico.digital@bue.edu.ar](mailto:ayuda.pedagogico.digital@bue.edu.ar)



Esta obra se encuentra bajo una Licencia Attribution-NonCommercial-ShareAlike 2.5 Argentina de Creative Commons. Para más información visite <http://creativecommons.org/licenses/by-nc-sa/2.5/ar/>



