**Guía Introductoria de la Tarjeta FirstMakers con el Software SNAP**

Tabla de contenido

[Introducción 2](#_Toc449022075)

[La Tarjeta FirstMakers 2](#_Toc449022076)

[El lenguaje de programación Snap4Firstmakers 3](#_Toc449022077)

[La interfaz de usuario. 4](#_Toc449022078)

[Conexión de la tarjeta FirstMakers al puerto USB del computador 5](#_Toc449022079)

[Prueba de los sensores y actuadores 6](#_Toc449022080)

[Principios de Programación en Snap4Firstmakers 8](#_Toc449022081)

[Movimiento del sprite en el escenario. 8](#_Toc449022082)

[Uso del Lápiz 9](#_Toc449022083)

[Repetición del bloques 10](#_Toc449022084)

[Operadores 11](#_Toc449022085)

[Programación de los sensores y actuadores 12](#_Toc449022086)

[Decisiones 13](#_Toc449022087)

[Construcción de una alarma con sonido 14](#_Toc449022088)

[Ejercicios 16](#_Toc449022089)

[El menú Apariencia 18](#_Toc449022090)

[Variables 19](#_Toc449022091)

[Sonidos 20](#_Toc449022092)

[Guardar y abrir Proyectos 20](#_Toc449022093)

[Sensores y actuadores incorporados en la tarjeta FirstMakers: 21](#_Toc449022094)

[Solución a errores frecuentes 22](#_Toc449022095)

Abril 2016

# Introducción

Esta guía es una introducción a la programación de la tarjeta electrónica FirstMakers utilizando el software Snap4FirstMakers[[1]](#footnote-1).

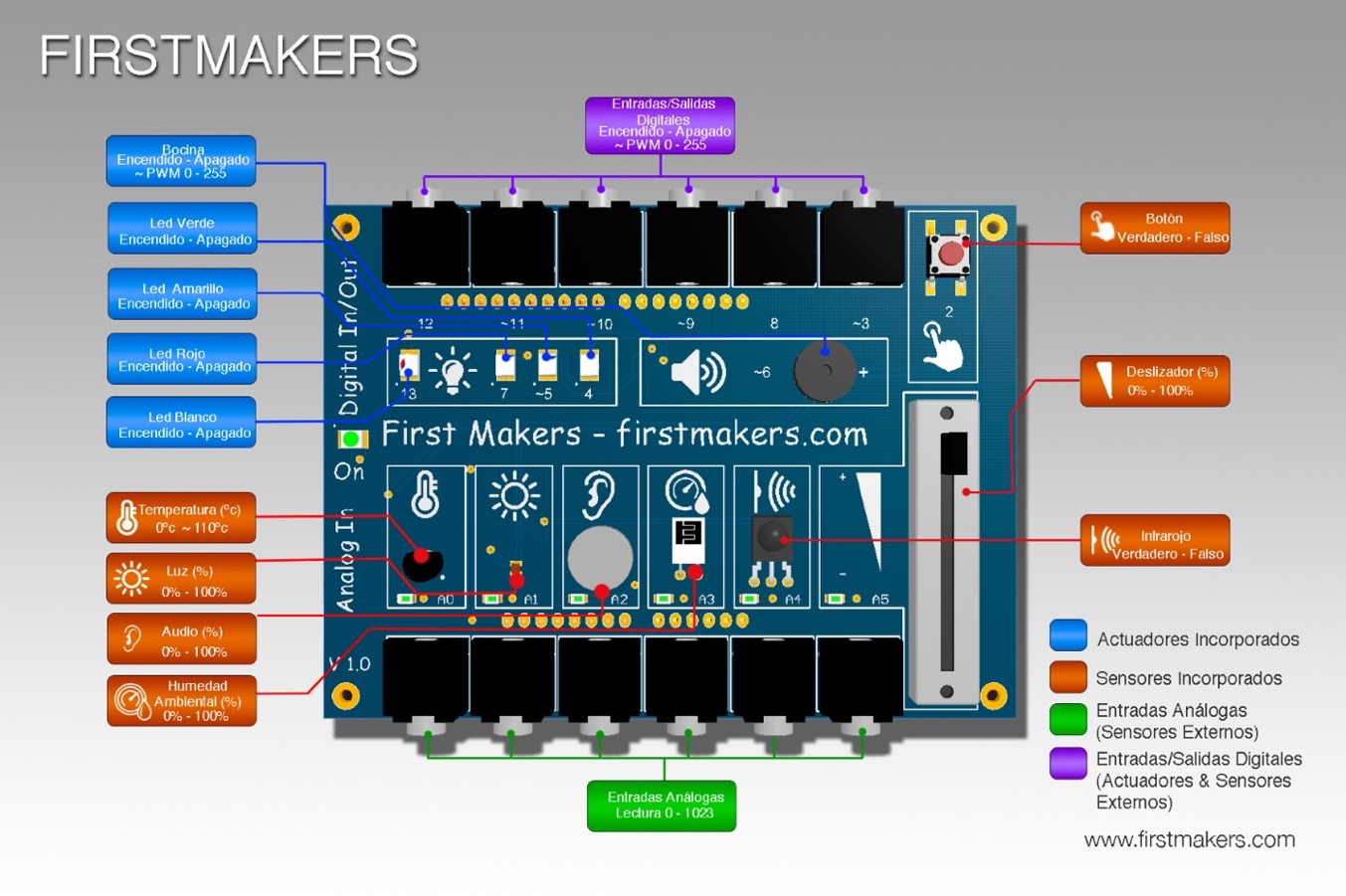
FirstMakers es una tarjeta electrónica que permite la creación de proyectos, prototipos, inventos, juegos y simulaciones interactivas. Consta de una tarjeta electrónica que incorpora sensores y actuadores y que se conecta a través de un cable USB a un computador el cual debe llevar un software que se comunica con la tarjeta.

Para comprender esta guía, no se requiere experiencia previa en programación, en uso de sensores o de la tarjeta FirstMakers

El **software** Snap4FirstMakers para trabajar con la tarjeta se puede descargar gratuitamente del sitio: <http://www.firstmakers.com>

## La Tarjeta FirstMakers

La tarjeta FirstMakers es un dispositivo electrónico que utiliza un Arduino como base e incorpora sensores: temperatura, luminosidad, audio, humedad ambiental, infrarrojo, potenciómetro y switch. En la parte superior se encuentran los actuadores y están compuestos por 4 Leds (blanco, rojo, amarillo y verde) y una bocina o buzzer. La tarjeta incluye conectores para la conexión de sensores y actuadores externos.



Es importante distinguir los **sensores** (temperatura, luz, audio, humedad e infrarrojo, además del potenciómetro) de los **actuadores** de la tarjeta que son los leds y la bocina. De los sensores sólo se puede leer su valor (ej: el valor de la temperatura), en cambio sobre los actuadores se puede actuar (por ejemplo, encender un led o activar la bocina).

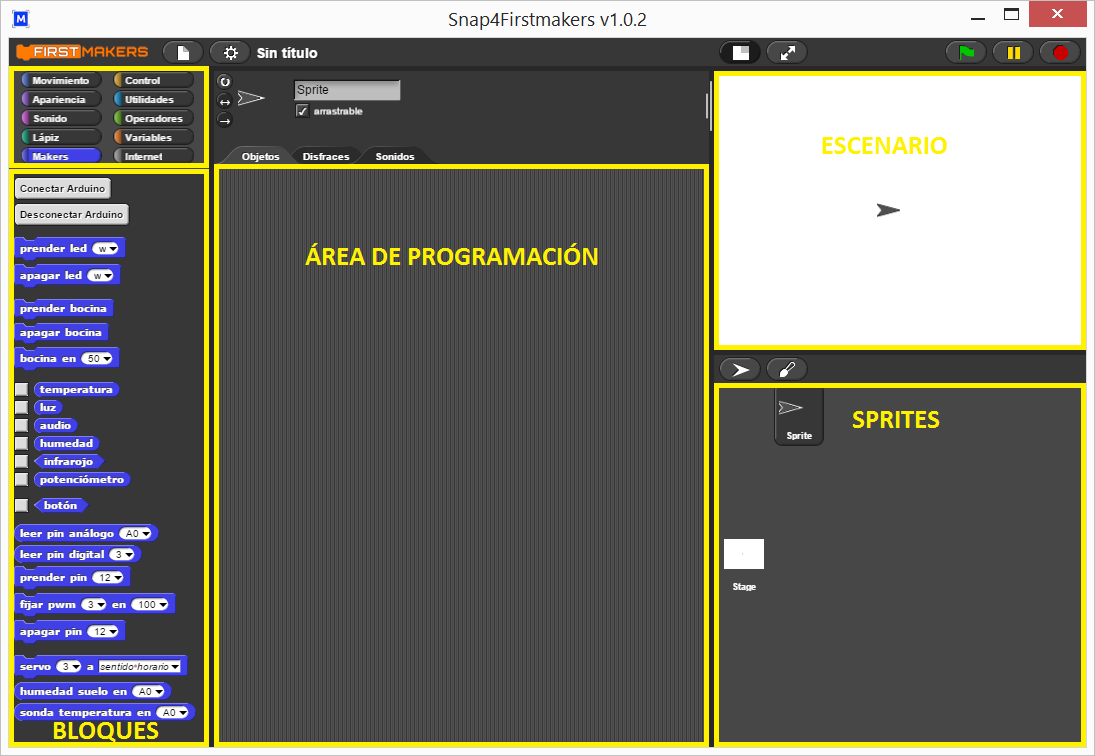
Posteriormente le conectaremos a la tarjeta sensores y actuadores externos a través de las entradas análogas y de las entradas y salidas digitales (esto se describe en otras guías que puedes revisar en <http://www.firstmakers.com>.

## El lenguaje de programación Snap4Firstmakers

El lenguaje de programación Snap4Firstmakers es una adaptación del lenguaje SNAP[[2]](#footnote-2) para interactuar con la tarjeta Firstmaker. SNAP es un lenguaje de programación de computadores, descendiente de SCRATCH, ampliamente conocido en el mundo escolar. A su vez, SCRATCH proviene del antiguo lenguaje LOGO, el cual fue el primer lenguaje de computadores diseñado para usos en educación.

## La interfaz de usuario.

Al activar Snap4Firstmakers, aparece en la pantalla una interfaz como la siguiente. Familiarízate con ella, reconociendo sus diversas partes.



En la parte superior izquierda están los **menús de bloques** (“Movimiento”, “Apariencia”, “Sonido”, “Makers”, etc.). Cada **menú** tiene diversos “**bloques**” para acciones específicas. Para acceder a los bloques de un menú, haz clic sobre el **menú**. Por ejemplo, el **menú** “Movimiento” tiene los bloques “**mover**”, “**girar**”, etc.

En la parte central está el **área de Programación**, a la cual se arrastran los bloques que formarán un “**script**”. Un script es una secuencia de bloques unidos que ejecutarán una acción específica.

A la derecha hay una zona en blanco que es el **Escenario** donde se traducirán muchas de las acciones que programemos. La flecha negra sobre el escenario se denomina **Sprite** y puede moverse y dejar marcas sobre el escenario según los bloques que usemos.

En la parte superior izquierda del escenario hay una pequeña barra para modificar el tamaño del escenario y del área de programación.

En SNAP, un programa (denominado **script**) se compone de diversos **bloques** de instrucciones que se procesan en secuencia. Cada **bloque** realiza una acción específica. Los bloques deben quedar bien conectados entre sí para crear un **script** (se atraen como imanes y se va conectando un bloque bajo el otro). Al estar cerca un bloque de otro, listo para conectarse, se dibuja una raya blanca entre ellos (en ese instante los bloques hacen “SNAP” entre sí).

En la siguiente figura se muestra un **script** de 5 **bloques** de color y a su derecha un **bloque**.

**Script** de SNAP que dibuja un cuadrado. **Bloque** “**mover**” del Menú **Movimiento**

Para activar un script, es decir, para que el software procese los bloques, se hace “clic” sobre la bandera verde del primer bloque o bien sobre cualquier bloque del script.

Después veremos que es conveniente iniciar cada script con el bloque “Al presionar” (con la banderita verde) del menú Control.



# Conexión de la tarjeta FirstMakers al puerto USB del computador

Activa el software Snap4FirstMakers (debes haberlo descargado previamente del sitio Web: <http://www.firstmakers.com>)

En el menú Makers de Snap4FirstMakers, haz clic en la opción:



Si la tarjeta está bien conectada, verás el siguiente mensaje:



Es posible que tu computador te pregunte en cual puerta USB tienes la tarjeta (si es que tiene varias), en cuyo caso puedes o bien averiguar cual es esa puerta o probar una a una.

Si recibes un mensaje de error, sigue los siguientes pasos:

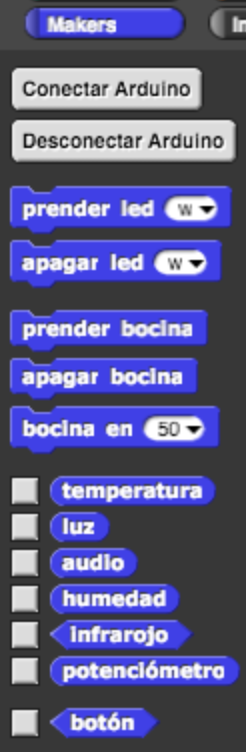
Desconecta y vuelva a conectar la tarjeta del puerto USB. Luego vuelve a seleccionar la opción “Conectar Arduino”.

Si persiste el mensaje de error, sale del programa Makers y actívalo nuevamente, repitiendo el proceso.

Si el error persiste, tendrás que reiniciar el computador.

## Prueba de los sensores y actuadores

Selecciona el menú Makers en la parte superior para desplegar los bloques que interactúan con la tarjeta y luego haz clic sobre los sensores para observar sus valores en el “**escenario**”:



Haz clic en estos sensores y observa su valor en el **escenario**

Observa la tarjeta e identifica los diferentes **sensores** que contiene. Prueba lo siguiente:

* Coloca un dedo sobre el sensor de luz y observa cómo cambia su valor. Agrégale más luz (ej: con un celular) observa el cambio en el valor.
* Aprieta suavemente con dos dedos el sensor de temperatura y observa cómo cambia su valor. Este sensor varía lentamente su valor ante cambios de temperatura, lo cual hay que considerar al realizar los experimentos.
* Cambia el valor del potenciómetro (desplaza la manilla), deberá modificar su valor entre 0 y 100. Esta condición servirá para simular valores rápidamente durante un experimento, para ver sus efectos.
* Emite sonidos sobre el sensor de audio y observa como varía su valor. Esto permitirá hacer experimentos con diversos sonidos del medio ambiente.
* Puedes humedecer levemente el sensor de humedad para ver su variación. La tarjeta no debe mojarse pero este sensor puede ser ligeramente humedecido (solo el sensor) con el dedo o un paño y no se afectará la tarjeta.
* Observa que el botón situado en un esquina de la tarjeta puede tener sólo 2 valores: verdadero (al estar apretado) o falso. Esta condición servirá para una serie de experimentos, de modo de activar o desactivar un proceso.

Para probar los **actuadores** de la tarjeta, haz clic sobre el bloque “prender bocina”. Deberá sonar un pito. Para apagarlo haz clic sobre el bloque “apagar bocina”

Puedes prender y apagar las luces (leds) de la tarjeta con los siguientes bloques:





El signo “W” significa “luz blanca” (White en inglés). Si apretas la flecha junto a la “W”, podrás encender y apagar las luces roja, verde y amarillo de la tarjeta.

Ahora ya conoces todos los elementos básicos y puedes comenzar a programar los sensores y actuadores de la tarjeta.

# Principios de Programación en Snap4Firstmakers

## Movimiento del sprite en el escenario.

Para mover el **sprite** en el escenario se utilizan bloques del menú **Movimiento**. Para que el **sprite** dibuje una línea al desplazarse, se utilizan los bloques del menú **Lápiz**.

Selecciona del menú **Movimiento** el bloque “mover 10 pasos” (es el primer bloque del menú) y arrástralo al área de programación. Luego haz clic repetidas veces sobre el bloque y observa cómo se desplaza el **Sprite** por el **escenario**. En este caso, se ha desplazado **10 pasos** en cada clic.



Puedes modificar cuanto se desplaza el Sprite cambiando su valor actual (10) que está sobre el fondo blanco, por cualquier otro número positivo o negativo. Por ejemplo, para hacerlo retroceder puedes escribir (-20) en el espacio con fondo blanco. Puedes también arrastrar el Sprite con el ratón para dejarlo en alguna posición.

Muchos bloques tienen un espacio con fondo blanco donde va un número o un texto como “**dato de entrada**” para el bloque. El dato de entrada para el bloque **mover** es el número de pasos que el Sprite debe desplazarse; en el caso del bloque **girar**, el dato de entrada es el número de grados que debe girar el Sprite. Cada vez que veas un boque con un espacio con fondo blanco, podrás incorporar un nuevo valor de entrada.

Prueba con diferentes **datos de entrada**, positivos y negativos para familiarizarte con el desplazamiento del bloque “mover” y los giros del bloque “girar”.

También puedes insertar en el **dato de entrada** de un bloque **el valor de cualquier sensor.** Por ejemplo, en el siguiente bloque se ha insertado el valor del **sensor de temperatura** del menú **Makers** como dato de entrada. Eso hará que el sprite se mueva según la temperatura ambiente. Más adelante verás que podrás combinar datos del medio ambiente con figuras y sonidos en el **escenario**.



Una situación frecuente es que el **Sprite** salga del **escenario**, por ejemplo, luego de un número de avances y giros.

Puedes volver a la posición inicial con el siguiente bloque:



También puedes imaginar que el **escenario** es un **sistema de coordenadas**, donde el Sprite siempre tiene una posición según el eje x y el eje y. En Snap, el centro del escenario está en x=0; y=0

Puedes ir al centro del escenario (sin modificar la dirección en que apunta el Sprite) con el siguiente bloque:



Prueba cambiando los **datos de entrada** de x e y observando donde queda el **Sprite** en cada caso. Esto te servirá al dibujar figuras en el escenario. Por ejemplo:





Prueba girar y desplazar el **Sprite** con los bloques **mover** y **girar**. Si juntas ambos bloques, al hacer clic en ellos se ejecutarán ambos en secuencia. Con esto, tienes tu primer **script** de 2 bloques. Por ejemplo, este **script** avanza 50 pasos y luego gira 60º a la derecha.



Para desacoplar dos o más bloques, simplemente arrastra el bloque (o los bloques) de más abajo del script, hasta que la raya blanca desaparezca.

Para desplazar un script dentro del área de programación, arrástralo con el bloque superior.

Para eliminar un bloque o un script del área de programación, arrástralo hacia la zona de bloques y suéltalo ahí. Puedes eliminar cualquier bloque de un script haciendo clic con el botón derecho del ratón sobre el bloque que deseas eliminar y seleccionar la opción borrar.

Prueba esto, ya que lo usarás muchas veces.

## Uso del Lápiz

El Sprite puede dibujar en la medida que avanza, como si tuviese un lápiz en su punta. Para ello, utiliza el bloque **bajar lápiz** del **menú Lápiz**, para que deje una raya en su camino. Puedes borrar el **Escenario** con el bloque **borrar**.

El siguiente **script** de 2 **bloques** baja el lápiz y dibuja una raya negra en el escenario al moverse el Sprite 50 pasos. La raya se dibuja desde la punta del Sprite.



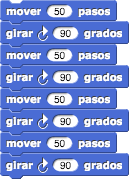
Al comenzar a dibujar un figura, conviene borrar todo el escenario con el bloque 

## Repetición del bloques

El bloque de **Control** “**repetir**”, es útil para repetir un conjunto de bloques las veces que necesites. Todos los bloques que colocas en el interior del bloque **repetir** (en su “boca”) serán repetidas las veces que indica el dato de entrada.



Por ejemplo, para dibujar un cuadrado, puedes colocar 4 veces los bloques **mover** y **girar** o bien utilizar el bloque **repetir** para indicar esto una sola vez.

Dos **scripts** para dibujar un cuadrado. Observa lo simple que es usar el bloque **repetir**.

En el **script** anterior, al hacer clic sobre el bloque **repetir**, el Sprite se desplazará (muy rápidamente) 4 veces.

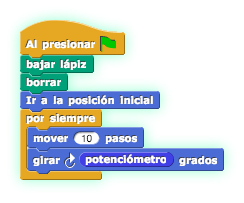
Es conveniente iniciar los script con el bloque del menú **Control** “**Al presionar”**. Más adelante verás la utilidad de esto.



Para observar mejor las acciones de un script, agrega el bloque **esperar** del menú **Control**, para detenerse un tiempo después de ejecutar un bloque. Por cierto, puedes cambiar el **dato de entrada** con el número de segundos que desees que espere.

Un bloque de repetición muy útil es **por siempre** del menú **Contro**l. Este bloque te permite dejar a la tarjeta trabajando por muchas horas si lo deseas, repitiendo el **script** que contiene.

Observa la figura que realiza el siguiente script variando el valor del potenciómetro:



Puedes agregarle el bloque de control  para ver con mayor lentitud qué sucede.

## Operadores

En el menú **Operadores**, encuentras varios bloques para hacer operaciones aritméticas y comparaciones. En el **script** anterior ya utilizamos un operador de resta. .

Puedes usar el siguiente bloque para hacer operaciones aritméticas: image.png Al hacer clic sobre el bloque verás que indica 50.

Por cierto, puedes colocar como dato de entrada el valor de un sensor para hacer operaciones aritméticas. Por ejemplo, si quieres usar el sensor de audio y la señal de audio es muy baja, puedes aumentar su valor:



# Programación de los sensores y actuadores

Para comenzar la programación de sensores y actuadores, haremos un pequeño proyecto consistente en graficar el valor de un sensor en el **escenario** utilizando el movimiento del lápiz.

La idea es que en el escenario aparezca una línea cuyo largo dependa del valor del sensor.

Para ello, construiremos un **script** en el área de **programación**, que consistirá de un proceso repetitivo de lectura del valor de un sensor con 1 segundo de espera. Luego de cada lectura, vamos a dibujar en el escenario una línea de largo igual al valor del sensor.

Usaremos bloques del menú de **Control** para iniciar la lectura, para la repetición y para la espera de 1 segundo. Para la lectura del sensor usaremos un bloque del menú **Makers**. Para graficar con el **sprite** en el escenario, usaremos bloques del menú **Movimiento** y del menú **Lapiz**.

El “algoritmo” (la secuencia de pasos) del **script** será más o menos así:

Comenzar el script

Colocar el **sprite** en una posición del escenario

Asociarle un lápiz al sprite (se usa el bloque “bajar lápiz”)

Borrar el escenario (para borrar gráficos anteriores)

Repetir lo siguiente:

Mover el sprite según el valor del sensor dibujando una línea en su camino.

Deshacer el camino del sprite, retrocediendo

Girar el sprite 90º a la derecha

Avanzar el sprite un pequeño tramo para dejar un espacio entre cada línea

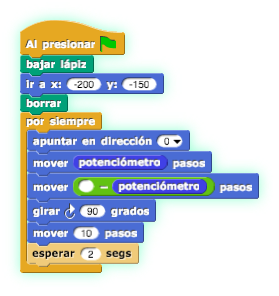
Volver a girar el sprite para que apunte hacia arriba, listo para el siguiente paso

Para probar el algoritmo, primero usaremos el valor del potenciómetro, luego cambiamos el potenciómetro por cualquier otros sensor.

Para retroceder el **sprite** el mismo valor que avanzó, usamos un bloque del menú **Operadores** que permite hacer operaciones aritméticas. En este caso usamos la operación de restar:



Primero, ubicamos el **sprite** en el borde inferior izquierdo del **escenario** para tener espacio para dibujar los valores del sensor. En cada repetición, el **sprite** avanza según el valor del potenciómetro, luego retrocede ese mismo valor y se desplaza un poco a la derecha. Antes de una nueva repetición, espera 2 segundos.

. 

Observa el efecto activando el script y cambiando los valores del potenciómetro. Hay muchas maneras diferentes de hacer este **script**.

Ensaya con el mismo **script** pero con el valor de otros sensores para ver el efecto de los cambios. Por ejemplo, cambia el potenciómetro por el sensor de luz y haz variar la luz que recibe el sensor (con un celular, o tapándolo).

Otra posibilidad es que dejes el **script** tomando muestras cada 10 minutos (600 segundos) en una habitación para ver como cambia la temperatura en la medida que transcurre el día.

También puedes hacerlo con el sensor de sonido para graficar el nivel de “ruido” de una sala, el patio o la calle.

## Decisiones

Para tomar decisiones usa el siguiente bloque del menú **Control**. Con este bloque, si la condición incluida luego de la palabra “si” es verdadera, se ejecutan los bloques en su interior, en caso contrario, estos bloques se saltan (no se procesan):

image.png

Puedes usar el siguiente bloque del menú **Operadores** para hacer comparaciones y determinar si el resultado es verdadero o falso:

image.png

Por ejemplo, este bloque compara el valor del sensor de temperatura y si es mayor a 20 ºC, enciende la luz blanca (W = led blanco) de la tarjeta.



Practica con los operadores y comparaciones en scripts con decisiones, más adelante usarás mucho estas opciones en tus proyectos.

## Construcción de una alarma con sonido

A continuación programaremos los actuadores de la tarjeta: los leds (las luces pequeñas) y el buzzer o generador de sonido.

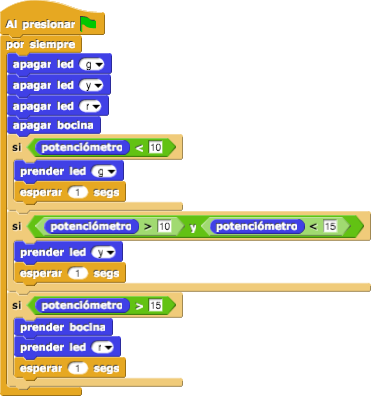
Para esto diseñaremos un **script** que simplemente repita por siempre la lectura de un sensor (comenzaremos usando el potenciómetro pero luego podemos cambiarlo por cualquier sensor), si el valor del sensor sobrepasa un cierto valor bajo, encenderemos la luz verde; si el valor es mediano, encenderemos la luz amarilla y si el valor es alto, encenderemos la luz roja y la bocina hasta que el valor baje.

Para efectos de este script, los valores serán:

Bajo: < 10

Medio: 11-15

Alto: > 15



Script con alarma de luces y sonido conectada al valor del potenciómetro

Ahora puedes mejorar esta alarma con otros sensores y otros valores. Por ejemplo, si la luz en tu pieza se enciende, puede hacer sonar la alarma. A futuro podrás asociarle un sensor de movimiento para detectar si algo o alguien se está desplazando cerca del sensor.

Hay muchos proyectos que puedes hacer con la tarjeta FirstMakers.

Snap4FirstMakers es muy poderoso y en esta guía trabajaste sólo una introducción, aunque con ello ya puedes realizar muchos proyectos.

Por ejemplo, ahora podrías conectarle a la tarjeta sensores externos, motores, luces y generadores de sonidos para programar muchas cosas en tu casa, escuela o en el patio.

Hay muchos jóvenes inventando y programando “artefactos inteligentes” con tarjetas como las que usaste, sensores y Snap4FirstMakers.

Puedes ver ejemplos en <http://firstmakers.com/>

# Ejercicios

Practica formando varios scripts, eligiendo y uniendo diversos bloques (recuerda que queden bien conectados entre sí, mostrando una raya blanca al estar cerca). Cambia los valores de entrada de los bloques y observa su efecto.

Utiliza los bloques **repetir**, **mover** y **girar** para dibujar figuras de diferentes lados

* ¿Cómo puedes dibujar un triángulo? ¿de cuantos grados debe ser el giro y por qué?
* ¿Cómo dibujar un hexágono?
* ¿Cómo hacer cualquier figura regular de “n” lados? ¿hay alguna regla general?

Para dibujar cualquier figura regular, la suma de sus ángulos externos debe ser de 360º. Por ello, el giro es igual a 360 dividido por el número de lados.

Así, un cuadrado tiene un giro de 90º (360 / 4); un triángulo tiene un giro de 120º (360 / 3) y un hexágono tiene un giro de 60º (360 / 6).

Scripts para dibujar un triángulo equilátero, un hexágono y un dodecaedro,

Observa que en todas las figuras anteriores se cumple que el ángulo del giro multiplicado por el número de repeticiones es igual a 360.

Crea diversas figuras regulares y luego dibújalas incorporando un bloque **repetir** dentro de otro, como en el siguiente **script**.

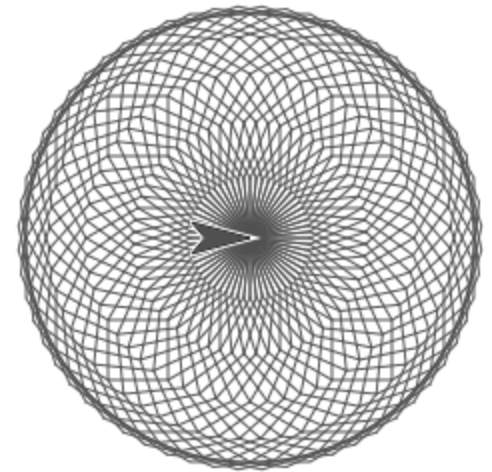
El primer repetir es de 20 y el último giro es de 18º: 18 x 20 = 360.

El segundo repetir es de 6 su giro es de 60º: 60 x 6 = 360.

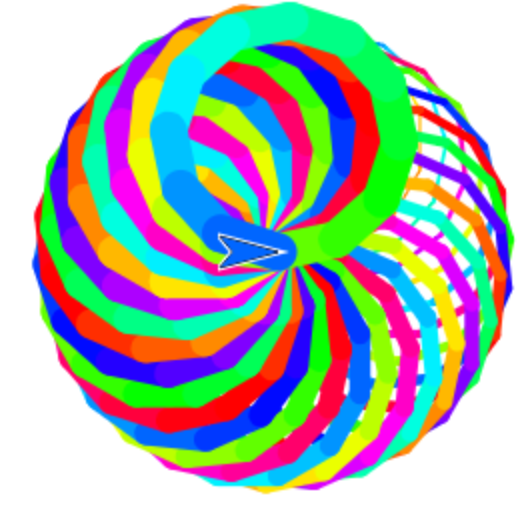


Puedes realizar distintas figuras regulares variando ángulos y movimientos que consideren la regla que el número de repeticiones x los grados de giro da 360.

Practica scripts con repeticiones que incluyan repeticiones … que a su vez incluyan repeticiones….



En el menú **Lápiz** puedes cambiar el color y tamaño del lápiz, pruébalo introduciendo nuevos bloques y dibujando figuras de diversos colores y con diferentes tamaños de lápiz. Ejemplo:



## El menú Apariencia

Puedes escribir un mensaje en el **escenario** de la pantalla utilizando bloques del menú **Apariencia**”.

Crea el siguiente script, actívalo y observa lo que sucede en el **Escenario**. Cambia los datos de entrada de los bloques **decir** y **pensar**. Cambia el texto con lo que tú quieras que se escriba.



Si usas el bloque “pensar”, el texto queda encerrado en una “nube”, como si fuera un pensamiento.

Practica con diferentes frases, saludos o mensajes con los bloques “decir” y pensar”.

En este menú puedes también ****y ****el Sprite. Por ejemplo, si al final de un dibujo no deseas que sea vea el Sprite, lo escondes con el bloque **esconder**.

## Variables

Una variable es como una caja que tiene un nombre y que puede contener un valor y ese valor podemos modificarlo. Por ejemplo, podemos crear una variable con el nombre “alfa” y asignarle el valor 10. Luego podemos modificar el valor de alfa.

Primero se crea la variable en el menú **Variables** y se le asigna un nombre.



Luego de crearla, al principio del área de bloques aparece el bloque con el nombre de la variable. En este caso le pusimos el nombre **alfa**. Nota que puedes mostrar el valor de alfa en el escenario haciendo clic en la cajita blanca a la izquierda del nombre de la variable.



Ahora le asignamos valores a esta variable. En este ejemplo, seleccionamos la variable alfa en el bloque **fijar** (haces clic sobre la flechita negra) y luego le dimos el dato de entrada 15.



El siguiente ejemplo dibuja una espiral en el escenario utilizando la variable alfa para modificar el largo de los pasos en cada repetición.



Script para dibujar una espiral con una variable

## Sonidos

Si tu computador tiene parlantes, asegúrate que está activado y que puedes escuchar sonidos. En el menú **Sonido**. Puedes tocar una nota con el bloque.



Si quieres probar como se van emitiendo los sonidos puedes usar **scripts** como estos:



Por cierto, ahora puedes probar con figuras que al dibujarse vayan emitiendo sonidos según el valor de los sensores.

## Guardar y abrir Proyectos

En el ícono de archivo  en la parte superior del software, junto al Logo, encuentras opciones para guardar un script con un nombre y posteriormente, en otra sesión, abrirlo para continuar con él.

Conviene crear una carpeta especial para guardar scripts de proyectos ordenadamente.

# Sensores y actuadores incorporados en la tarjeta FirstMakers:

|  |  |  |  |
| --- | --- | --- | --- |
| Pin | Sensores | Unidad de Medida | Rangos valores |
| A0 | Temperatura | º Celsius | 0ºc a 110º c |
| A1 | Luminosidad | % Porcentaje | 0% a 100% |
| A2 | Audio | % Porcentaje | 0% a 100% |
| A3 | Humedad | % Porcentaje | 0% a 100% |
| A4 | Infrarrojo | Binario | Verdadero y Falso |
| A5 | Potenciómetro | % Porcentaje | 0% a 100% |
| 2 | Botón | Binario | Presionado y No Presionado |
| Pin | Actuadores | Rangos valores | | |
| 4 | Led Verde | Encendido y Apagado | | |
| 5 | Led Amarillo | Encendido y Apagado | | |
| ~6 | Bocina / Buzzer | Encendido, Apagado y 0 a 100 | | |
| 7 | Led Rojo | Encendido y Apagado | | |
| 13 | Led Blanco | Encendido y Apagado | | |

# Solución a errores frecuentes

A continuación se describen los errores conocidos que pueden presentarse al usar la tarjeta con el software.

Aunque no es un error, hay que aclarar que cada vez que se conecta la tarjeta, ésta se asocia directamente a un único **sprite**, por lo tanto cuando se cree un nuevo proyecto o se importe uno existente, deberá hacer clic en conectar Arduino. En caso contrario le advertirá que la tarjeta no está conectada al intentar ejecutar un **script**.

|  |  |
| --- | --- |
| Mensaje o Error | Solución |
| Se desconectó la tarjeta de la puerta ... | Revisa que la tarjeta esté conectada correctamente al puerto USB. Revisa que el cable esté en buenas condiciones. |
| Arduino no está conectado | Conecta el cable USB al computador y haz clic en Conectar Arduino en el menú Makers. |
| No se pudo comunicar con Arduino en la puerta… Revisar si tiene cargado el firmware *Firmata*. | Este error puede ocurrir cuando el Arduino no tiene el firmware correcto o se ha reemplazado por uno nuevo. Si este error persiste se debe actualizar el firmware del Arduino. Pasos:   * Descargar desde <http://arduino.cc/en/Main/Software> Arduino 1.x.x e instalar. * Abrir el software descargado e ir a Archivo/Ejemplos/Firmata/StandarFirmata. * Se abrirá una nueva ventana, haga clic en el botón “cargar” y espere a que finalice el proceso de actualización. * En caso de error, desconecte todo, vuelva a conectar el Arduino al usb del computador y repita el proceso. |
| La tarjeta no responde al iniciar un programa. | Verifica los cables de conexión, intenta prender y apagar un led desde el software. Si los problemas continúan, cierra el software y desconecta la tarjeta del USB, vuelve a conectar e inicia nuevamente el software. |

1. Snap4FirstMakers es una versión especial del software de programación SNAP, adaptada para funciona con la tarjeta FirstMakers. A su vez, SNAP es una derivado del software Scratch y es muy similar a éste. Si ya sabes Scratch, entonces también sabes Snap. [↑](#footnote-ref-1)
2. Puedes ver más información sobre SNAP en: <http://snap.berkeley.edu> [↑](#footnote-ref-2)