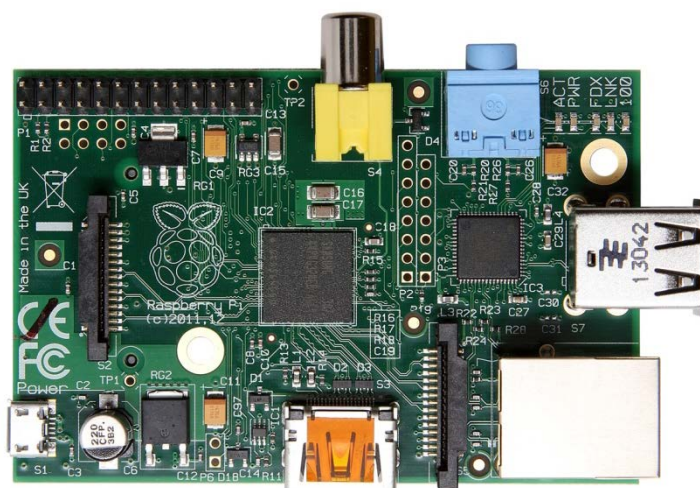


# El Raspberry Pi, una alternativa a los ordenadores convencionales en el aula y su uso para la introducción a los lenguajes de programación.

## MATERIAL ELABORADO



Profesorado de Tecnología, de Matemáticas y de apoyo al Área Práctica del IES Río Órbigo de Veguellina de Órbigo (León)

Profesorado de Tecnología del IES Fernando I de Valencia de Don Juan (León)



## Contenido

¿Qué es Scratch? .....	5
Primeros pasos con Scratch.....	7
Primeros pasos con Scratch - 2.....	11
Vamos a hacer un programa con Scratch.....	15
Scratch y GPIO .....	19
PROGRAMACIÓN DE LA RASPBERRY PI EN C Y MANEJO DE SU GPIO CON UNA LIBRERÍA.....	23







## ¿Qué es Scratch?

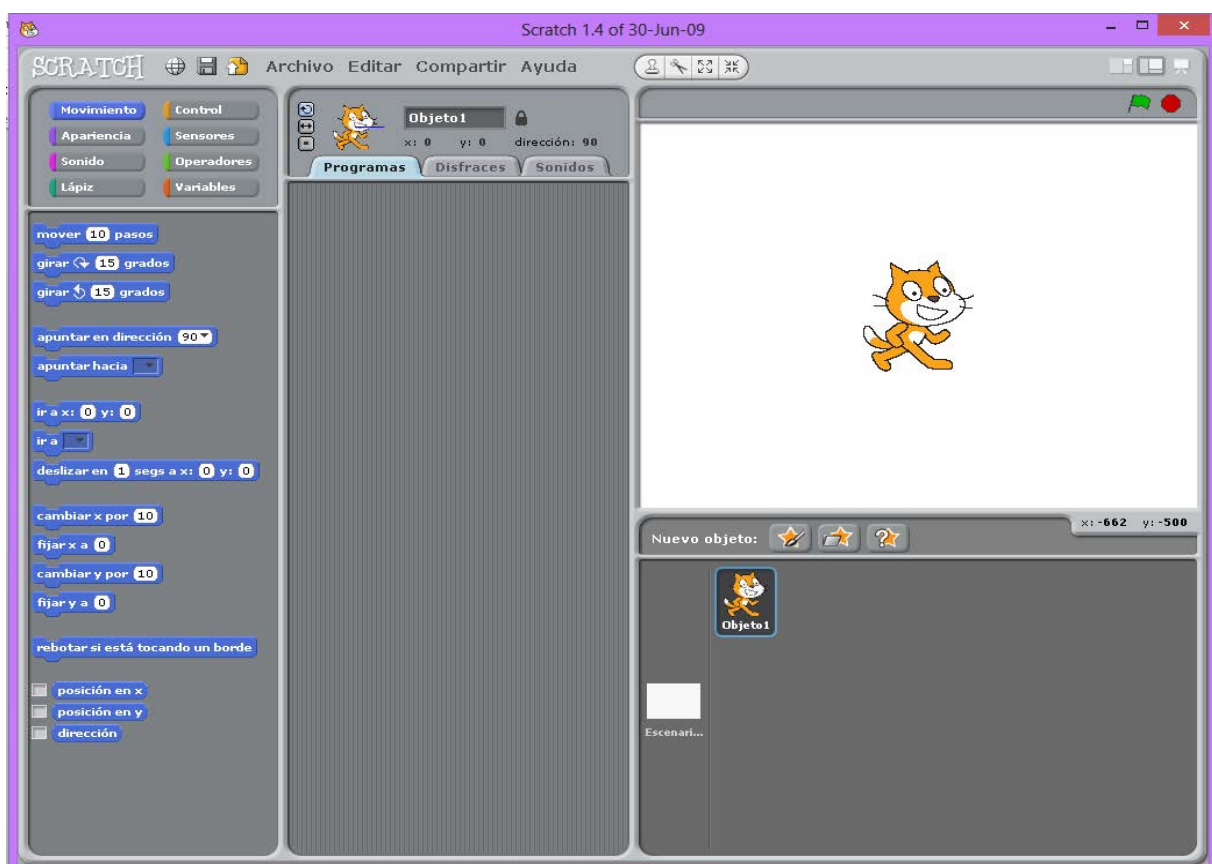
Scratch es un entorno de programación visual. Con él se pueden crear animaciones, juegos y trabajos artísticos interactivos a la vez que se iniciarse en algunos principios técnicos de programación. Es una forma sencilla de introducirse en la programación.

Los datos que procesa un ordenador se almacenan en código binario, es decir, ceros y unos. Los programas son conjuntos de instrucciones que operan con esos datos en binario, y por tanto, esas instrucciones también estarán en código binario. Pero es muy difícil programar en código binario, por lo que se utilizan lenguajes de programación de alto nivel, más cercanos a nuestro lenguaje. Luego, esas instrucciones son traducidas al código binario para que el ordenador las pueda ejecutar.

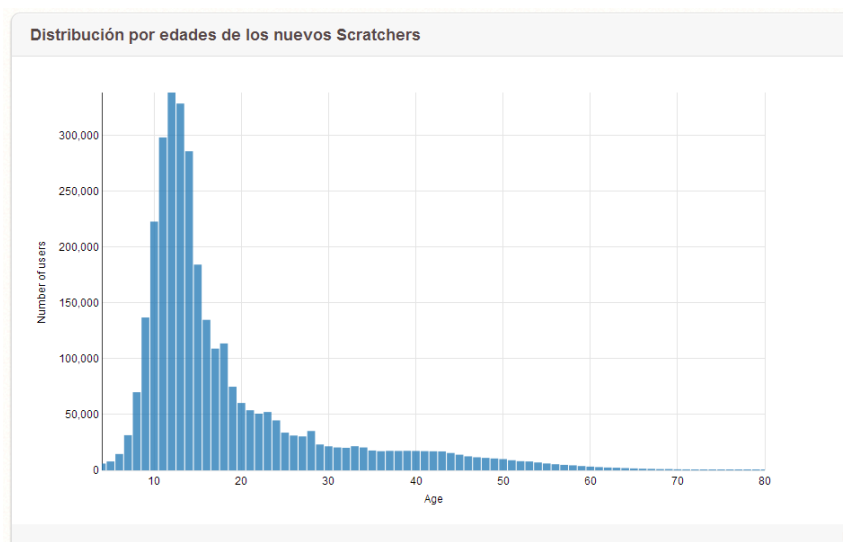
Scratch está basado en el lenguaje LOGO, muy utilizado hace unos años en educación.

El entorno gráfico de Scratch permite la programación mediante bloques funcionales, por lo que no es necesario el aprendizaje de un lenguaje de programación específico. Los bloques se van uniendo para formar el programa, que se ejecuta directamente mediante un intérprete.

Scratch significa arañar, pero también significa empezar algo desde cero. El icono de Scratch y el personaje que aparece por defecto es un gato.



Está creado por el Grupo Lifelong Kindergarten del Laboratorio de Medios del MIT, es gratuito y está destinado a jóvenes entre 8 y 16 años aunque es usado por jóvenes de todas las edades. Si nos fijamos en las estadísticas proporcionadas por la comunidad de usuarios registrados de Scratch, los que más lo utilizan son jóvenes entre 11 y 14 años, lo que lo hace ideal para su uso en la ESO.



El proyecto Scratch integra una comunidad que, en mayo de 2014, cuenta con 5.527.194 proyectos compartidos, 3.318.508 de usuarios registrados, 26.811.515 comentarios publicados y 431.024 estudios realizados. Los proyectos están abiertos, es decir, podemos ver cómo se han programado.

Explorar las posibilidades de Scratch:

[http://scratch.mit.edu/explore/?date=this\\_month](http://scratch.mit.edu/explore/?date=this_month)

La versión actual es Scratch 1.4, disponible para Windows, Mac OS y Debian/Ubuntu. Se puede descargar desde la Web:

[http://scratch.mit.edu/scratch\\_1.4/](http://scratch.mit.edu/scratch_1.4/)

También se puede programar online:

<http://scratch.mit.edu/projects/22322450/#editor>

Se puede obtener más información en la dirección:

[www.scratch.mit.edu](http://www.scratch.mit.edu)

El manual oficial para empezar con Scratch:

<http://download.scratch.mit.edu/ScratchGettingStartedv14.pdf>

Un buen manual en castellano:

<http://www.eduteka.org/pdfdir/ScratchGuiaReferencia.pdf>

Todos los recursos publicados en la comunidad Scratch:

<http://scratched.media.mit.edu/resources/all>



### Primeros pasos con Scratch



1. En la columna izquierda de la ventana del programa aparecen todos los mensajes que podemos enviar a los objetos que se encuentran en el escenario. Arriba están los grupos y abajo las ordenes de cada grupo.

El primer programa que vamos a realizar es el siguiente:



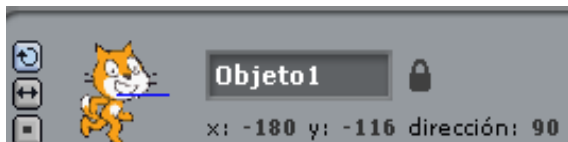
Para que el programa se ponga en marcha pulsa la bandera verde que hay encima del escenario. Como ves los cambios son muy pequeños cada vez que haces que se ejecute el programa. Vamos a realizar algunos cambios en el programa anterior:



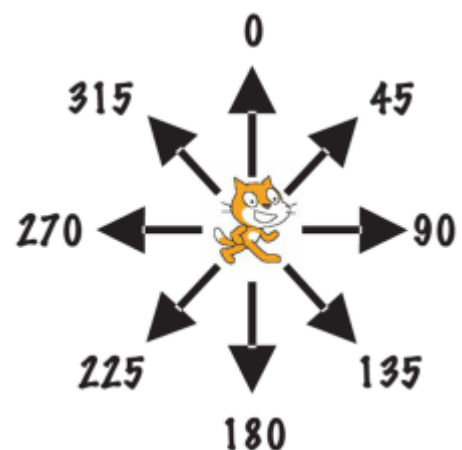
a) *¿Cuál es la diferencia entre estos dos primeros programas?*

b) *¿Cómo cambiarías el programa para que el gato solo diga "miau" al final de su camino? Pruébalo en la Raspberry y luego escríbelo aquí.*

2. En la parte de arriba de la pantalla, se muestra la posición del objeto sobre el escenario, a través de coordenadas (X e Y; en este caso, X= -180 e Y= -116. Es aproximadamente la esquina inferior izquierda):



También nos dice la dirección en la que mira el objeto (90). Las direcciones son:



Utiliza los siguientes bloques para crear un programa, que haga que el gato se posicione al principio en la esquina inferior izquierda del escenario. Luego se mueva 300 pasos a la derecha y diga al final "Estoy muy cansado":

Escríbalo aquí:



3. Vamos a cambiar de escenario. Pulsa sobre escenario. Ahora aparece lo siguiente:



Pulsa el botón "importar", busca una imagen del suelo de la luna y elígela como escenario.

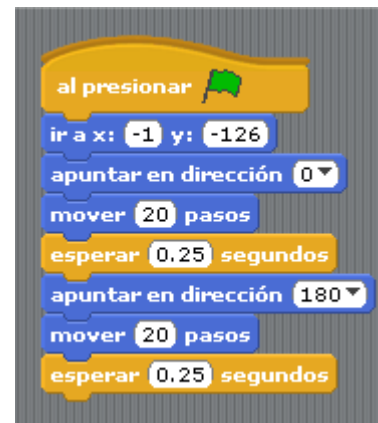
Vamos a y elige "el balón"



cambiar de objeto. Para ello pulsa el botón del medio de: para que te quede algo parecido a lo siguiente:



Prueba el siguiente programa sobre el balón:



a) ¿Qué tendrás que cambiar para que los botes del balón sean más grandes?. Realiza los cambios necesarios en el programa para que el balón llegue al menos a la parte negra. Escribe a continuación el resultado.





**b) ¿Cómo afecta al programa eliminar los dos bloques?**



**c) Añade al programa anterior uno de los bloques que ya hemos visto para que el balón bote 20 veces seguidas. Escribe aquí como ha quedado el programa.**

4. Modifica el programa del ejercicio anterior para que el balón bote en diagonal tres veces, sobre el escenario de la luna y escribe como queda el programa.



5. Elige un escenario y un balón nuevos. Ahora sobre el objeto introduce los siguientes bloques. Podemos duplicar un bloque o conjunto de bloques pulsando sobre ellos con el botón derecho del ratón y eligiendo duplicar.

**¿Qué ocurre al pulsar las teclas de flecha?**

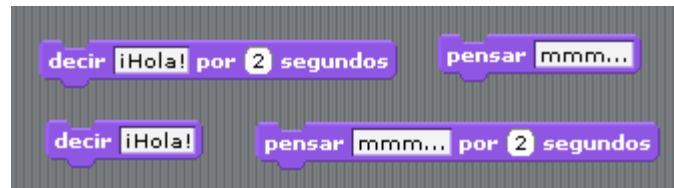






## Primeros pasos con Scratch - 2

1. Con las siguientes órdenes puedes hacer que los personajes u objetos que tengas en el escenario digan algo o que lo piensen, a través de bocadillos.



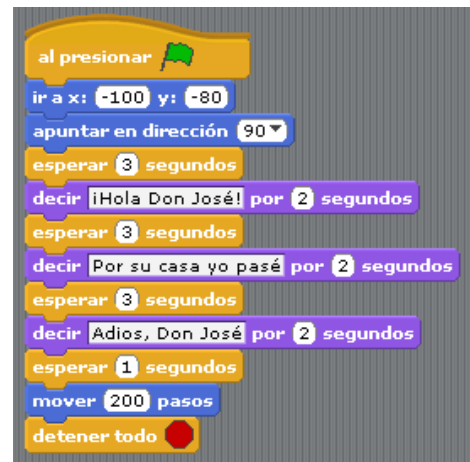
a) Pon un escenario y dos personajes, como en la imagen siguiente:



**El personaje 1 dirá:**



**El personaje 2 dirá:**



Prueba el programa en Scratch. ¿Por qué es necesario meter las órdenes

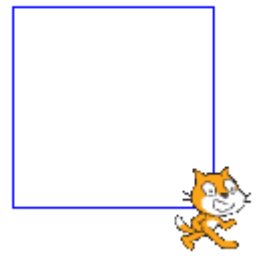


Quita las ordenes de espera y ejecuta el programa. ¿Qué ocurre?

b) Pon un fondo de escenario. Saca dos personajes (objetos) y crea una pequeña historia como la anterior. Ha de tener un dialogo, y algo de movimiento. Escribe a continuación el programa de cada personaje.

2. El gato pintor. Si ejecuto el siguiente programa (con las ordenes de "Lápiz"), el resultado es el dibujo de la derecha:

```
al presionar 
  ir a x: -10 y: -90
  apuntar en dirección 90
  repetir 4
    girar ↻ -90 grados
    bajar lápiz
    mover 100 pasos
  subir lápiz
```

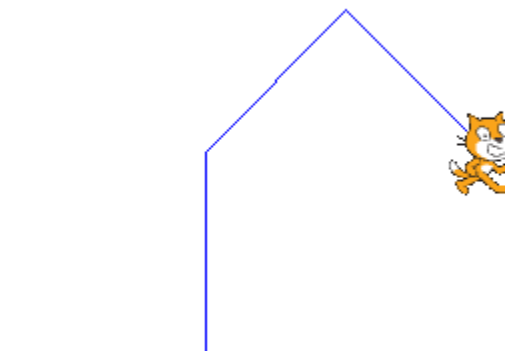


a) Sería bueno añadir al final del programa anterior lo siguiente:  
¿Qué efecto tendrán estas dos órdenes?

```
esperar 5 segundos
borrar
```

- b) Cambia el valor del ángulo de giro y pon -60. ¿Qué figura incompleta dibuja ahora el gato?
- c) ¿Qué modificación tendrás que hacer en el programa anterior para que se complete la figura?

3. Escribe el programa que dibuje una figura como la siguiente (todos los segmentos son de 100 pasos):



4. El bloque **rebotar si está tocando un borde** hace que cuando un objeto toque el borde cambie su dirección en sentido opuesto. Por ejemplo si iba hacia la izquierda se girará hacia la derecha.



Utiliza únicamente los siguientes bloques para hacer que el un objeto se mueva indefinidamente, de izquierda a derecha, por la pantalla, al presionar la bandera verde. Escribe aquí el programa:



5. a) Coloca un escenario y saca una pelota. Realiza un programa para que la pelota haga un bote (primero sube y luego baja).

b) Escribe las siguientes órdenes:



y mete dentro del bucle “repetir” el código que hace que la pelota bote una vez. Explica qué hace el programa.

6. Las variables son cajones en los que podemos guardar un valor (numérico o texto). Luego podemos manejar esos cajones por su nombre

Si vas a “Variables” y pulsas en “Nueva variable”, lo primero que te pide que le pongas nombre al cajón en el que se va a meter la información.

a) Crea una variable que se llame “Pueblo” y escribe el siguiente programa:



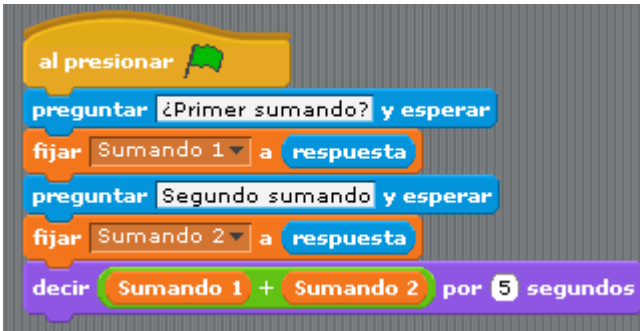
¿Cuál es el nombre del cajón (o variable) en este ejemplo?

¿Qué información has metido en el cajón?

b) Crea un programa como el anterior, para preguntar algo y que luego el gato nos responda con la información que hemos metido en la variable. Escríbelo a continuación.

7. Las ordenes que van en “Operadores”, entre otras cosas nos permiten realizar operaciones matemáticas.

**a) Explica que hará el siguiente programa:**



**b) Crea un programa que reste dos números y escríbelo aquí.**



## Vamos a hacer un programa con Scratch

**El gato protagonista se presentará y preguntará si quieres jugar con él. Luego realizará dos preguntas relacionadas con áreas y volúmenes**

Para empezar a trabajar con la placa Raspberry Pi hay que conectarla a una pantalla, a un teclado, a un ratón y si, quieres, a una conexión de datos. Por último, conecta el alimentador a la corriente eléctrica. Se empezará a cargar el Sistema Operativo del Raspberry Pi. Luego te pedirá algún dato:

**Login: pi**

**Password: raspberry**

Y por último escribe **startx** para que funcione la interfaz gráfica.

Luego iniciamos el programa Scratch, que tiene un acceso directo en el escritorio.

1. Arrastra el bloque MOVER al área de Programas y cambia 10 por 15 pasos (situándote encima de la cifra y haciendo clic podrás cambiarla). Al gato se moverá un poco hacia adelante.

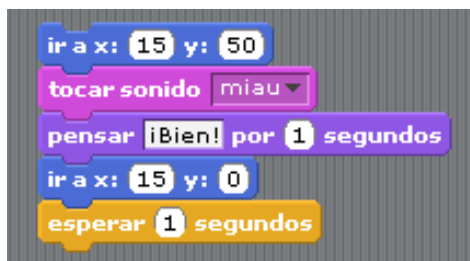


2. Haz clic en APARIENCIA y arrastra el bloque DECIR hasta que encaje con el anterior. Encaja un segundo bloque DECIR y cambia la frase para que el gato diga cómo se llama y pregunte *¿quieres jugar conmigo?* Cambia la duración a 3 segundos.



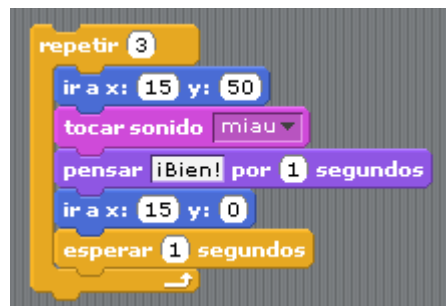
3. Ahora vamos a hacer que el gato muestre su alegría saltando, diciendo miao y que le salga un bocadillo de la cabeza en el que ponga ¡Bien!

Primero haremos que el gato suba 50, para ello utilizamos IR A, luego TOCAR SONIDO, PENSAR y ESPERAR (si no ponemos un tiempo de espera cuando el gato suba no veríamos el movimiento. Luego le mandamos bajar y que espere otro segundo. Ensambla todas estas instrucciones pero no las unas todavía a las anteriores.



Encontrarás IR A en la categoría de MOVIMIENTO  
 TOCAR SONIDO está en la categoría de SONIDO  
 PENSAR está en la categoría APARIENCIA  
 ESPERAR está en la categoría de CONTROL

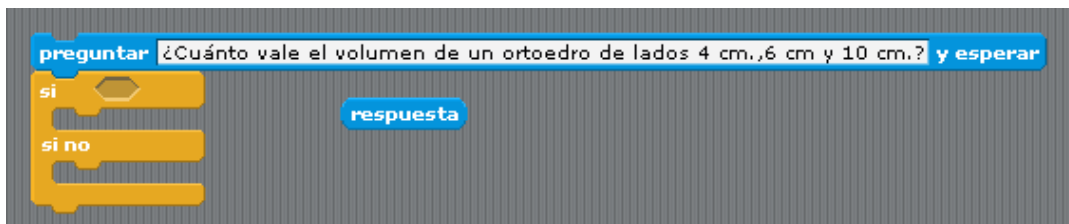
4. Para hacer que el gato repita 3 veces sus saltos de alegría utilizamos la orden REPETIR de la categoría de CONTROL. Introducimos dentro de esta orden el bloque de instrucciones que acabas de hacer en la parte de arriba y cambiamos el número 10 por un 3.



5. Antes de continuar, es buen momento para guardar el proyecto. Haz clic en Archivo – Guardar como y guarda el programa en el escritorio con tu nombre.
6. Ahora vamos a añadir un detalle para mejorar el programa. Para empezar vamos a utilizar la orden AL PRESIONAR, del bloque de CONTROL. Así deberemos de presionar la bandera verde para iniciar el programa y el círculo rojo para pararlo.



7. Ahora el gato va a hacer la primera pregunta sobre áreas y volúmenes. Para ello tenemos que utilizar la orden PREGUNTAR Y ESPERAR, que encontrarás en el bloque de SENSORES. Arrastra también la RESPUESTA, que la utilizaremos después. Encaja debajo la orden SI, SI NO del bloque de control.



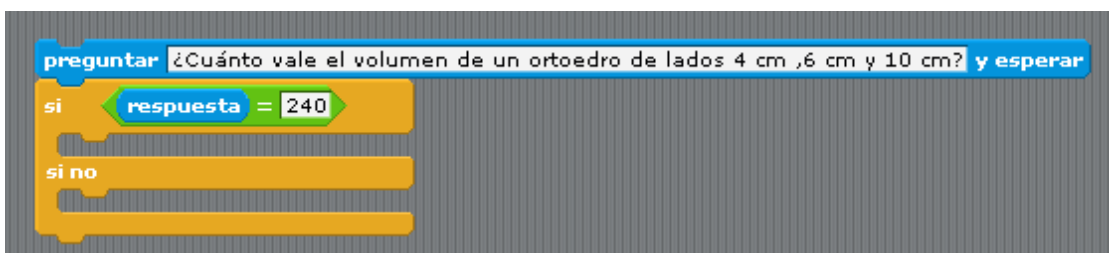
8. La respuesta será correcta si es igual a  $4\text{ cm} \times 6\text{ cm} \times 10\text{ cm} = 240\text{ cm}^3$ , no pediremos la unidad de medida. Para que el programa sepa si es correcta arrastramos el operador IGUAL de la zona de OPERADORES.



En el primer miembro de la igualdad incrustamos la RESPUESTA y en el segundo ponemos el resultado esperado, 240.



Ahora incrustamos todo al lado del SI







Ahora tenemos que completar la instrucción indicando que ocurrirá. Por ejemplo, si la respuesta es correcta, en esta secuencia de instrucciones el gato maulla, cambia de color, de tamaño, te felicita y luego vuelve a quedar como estaba. Luego incrustamos la secuencia entera debajo del SI.

```

tocar sonido miau
cambiar efecto color por 90
cambiar tamaño por 40
decir Muy bien, lo has hecho muy bien. por 2 segundos
pensar Chico listo... por 1 segundos
esperar 1 segundos
quitar efectos gráficos
fijar tamaño a 100 %
    
```

```

preguntar ¿Cuánto vale el volumen de un ortoedro de lados 4 cm
si respuesta = 240
    tocar sonido miau
    cambiar efecto color por 90
    cambiar tamaño por 40
    decir Muy bien, lo has hecho muy bien. por 2 segundos
    pensar Chico listo... por 1 segundos
    esperar 1 segundos
    quitar efectos gráficos
    fijar tamaño a 100 %
si no
    
```

Y si la respuesta es incorrecta, habrá que dar una serie de instrucciones que se deberán de realizar debajo del SI NO. Por ejemplo:

```

cambiar efecto color por 15
tocar sonido Dog1
decir ¡Mal! por 2 segundos
pensar Mmm... No se lo sabe... por 2 segundos
quitar efectos gráficos
    
```

```

si respuesta = 240
    tocar sonido miau
    cambiar efecto color por 90
    cambiar tamaño por 40
    decir Muy bien, lo has hecho muy bien. por 2 segundos
    pensar Chico listo... por 1 segundos
    esperar 1 segundos
    quitar efectos gráficos
    fijar tamaño a 100 %
si no
    cambiar efecto color por 15
    tocar sonido Dog1
    decir ¡Mal! por 2 segundos
    pensar Mmm... No se lo sabe... por 2 segundos
    quitar efectos gráficos
    
```



Para introducir un nuevo sonido, en este caso el "Dog 1", tienes que pinchar en la pestaña "Sonidos" - Importar y busca en la carpeta Animals el sonido de un perro.

9. Ahora tendrás que idear tu cómo hacer la segunda pregunta... Puedes hacerlo de una forma muy sencilla. Coloca la nueva pregunta debajo del bloque SI, SI NO. Ahora situa el ratón sobre el bloque SI, SI NO, haz clic con el botón derecho del ratón y escoge la opción duplicar. Cambia el segundo término de la igualdad por el resultado adecuado y ya estará.

10. Para terminar, el gato deberá despedirse y dar las gracias. Y el programa terminará con la orden DETENER TODO del bloque de CONTROL.



11. Si te queda tiempo puedes investigar y poner un escenario. Busca la opción ESCENARIOS (en la ventana inferior derecha), luego haz clic en FONDOS e IMPORTAR. Por ejemplo, en la imagen se ha añadido el escenario Woods. Para volver al programa haz clic en el gato (Objeto 1) y en Programas.



ahora puedes probar a hacer los cambios que quieres. Y si te gusta el programa, lo puedes utilizar online desde la página Web del MIT (Instituto Tecnológico de Massachusetts):

<http://scratch.mit.edu/>

O descargarlo e instalarlo para Windows: [http://info.scratch.mit.edu/es/Scratch\\_1.4\\_Download](http://info.scratch.mit.edu/es/Scratch_1.4_Download)





## Scratch y GPIO

Para controlar GPIO es necesario añadir un parche a Scratch. Es un programa en Python que se conecta con Scratch y permite a este controlar el GPIO.

Para descargar el parche, debemos ejecutar el siguiente comando:

```
sudo wget http://goo.gl/dANpKr -O isgh4.sh
```

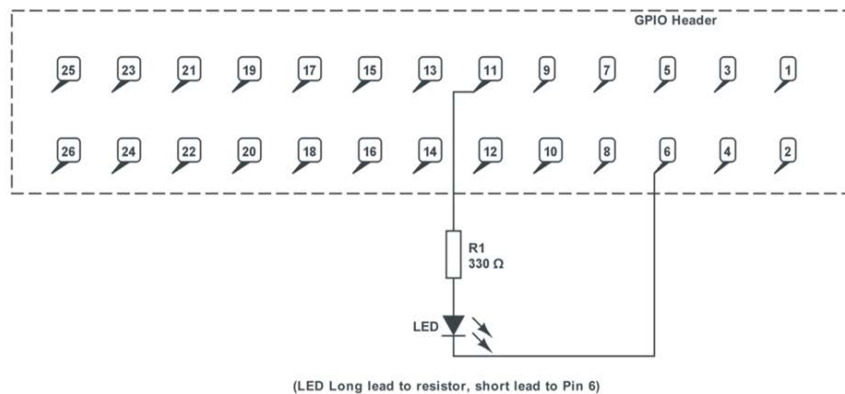
con este, conseguimos descargar el instalador del parche.

Ahora ejecutamos el instalador:

```
sudo bash isgh4.sh
```

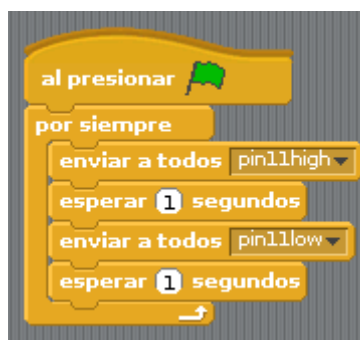
### Manejo de salidas

Montamos un LED en una placa board, junto con una resistencia limitadora, conectada



como se ve en la figura:

Ejecutamos ScrachGPIO. Con el siguiente programa conseguimos que el LED conectado al pin 11 se encienda y se apague durante un segundo:



Se pueden controlar seis pines como salidas (Pins 11,12,13,15,16 y 18) y el resto como entradas (22,7,3,5,24,26,19,21,23,8 y10).

Para poner en alto (3,3V) los pines de salida utilizaría las órdenes:

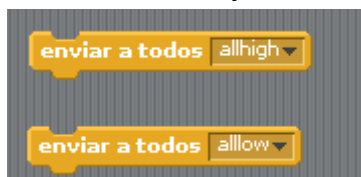


y para ponerlos en bajo (0V):

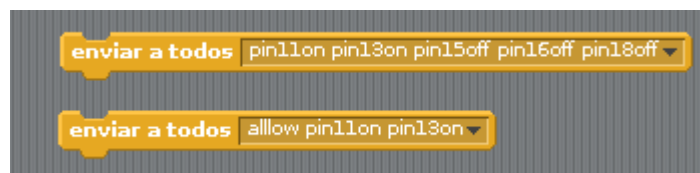


Podemos sustituir **high** por **on** y **low** por **off**.

Para poner todas las entradas en alto o en bajo:



También, si quiero poner en alto el pin 11 y el 13, y los demás en bajo, son validas las siguientes ordenes:



El siguiente mensaje pone los pines 11 y 13 en alto y los demás en bajo :  
(pinpatternPIN18PIN16PIN15PIN13PIN12PIN11)



## Manejo de entradas

Para comprobar una entrada, debemos ir a las órdenes de sensores y allí aparece la siguiente orden de escucha:

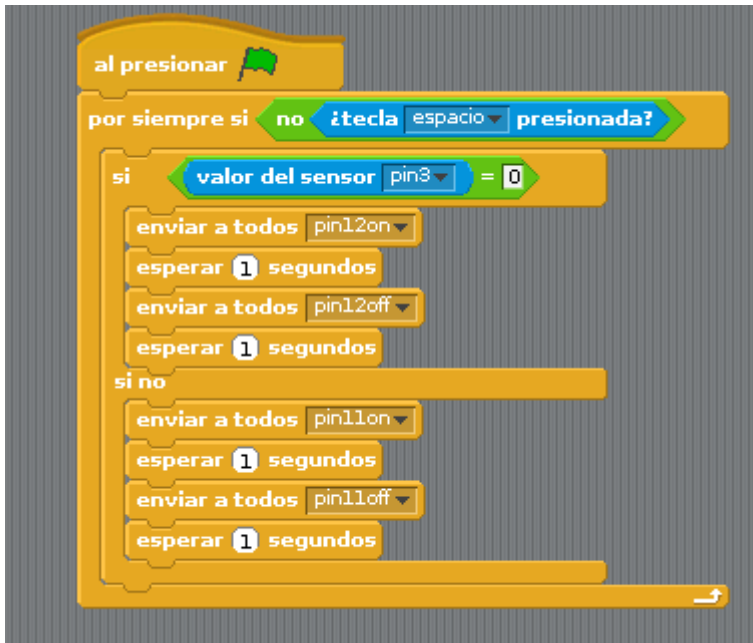




que al pulsar sobre la flechita nos permite elegir el pin que escucharemos. Con el operador de igualdad podemos controlar el valor de pin:



Un programa de ejemplo:



Este controla una entrada colocada en el pin 3. Si está a cero se ilumina un diodo, en el pin 12, y se apaga. Si está a uno se ilumina otro diodo en el pin 11.

La entrada se ha hecho con una resistencia de 1 kΩ en pull-up.

El programa se estará ejecutando siempre que no esté pulsada la barra espaciadora.

### Uso de variables para manejar salidas.

Basta con crear variables con los nombres pin 11, pin12, pin13, pin15, pin16 o pin18 y luego darles el valor 0, off o low, para ponerlas a cero; o darles el valor 1, on o high para ponerlas a 1.

El programa de la derecha pone a 1 y a 0 primero el pin11 y luego el pin12 y así siempre.

La web consultada ha sido:

<http://cymplecy.wordpress.com/2013/04/22/scratch-gpio-version-2-introduction-for-beginners/>



# Raspberry Pi Rev2 - P1 Connector



- Power 3.3V maximum current draw 50mA
- Power 5V maximum current draw Model A - 500mA, Model B - 300mA
- Ground
- UART
- I2C pulled-up with 1K8 resistor to 3.3V
- GPIO
- SPI

[c] 2013 combinatorialdesign.com - License Attribution-ShareAlike CC BY-SA  
[http://combinatorialdesign.com/boards/Raspberry\\_Pi/P1](http://combinatorialdesign.com/boards/Raspberry_Pi/P1) Rev 1.1 - 4/19/2013

## PROGRAMACIÓN DE LA RASPBERRY PI EN C Y MANEJO DE SU GPIO CON UNA LIBRERÍA.

### 1. Compilador gcc

gcc es un compilador múltiple que viene con Raspbian.

Vamos a escribir y compilar un Hola Mundo con gcc (1er. programa)

Basta con teclear el programa con un editor sencillo como nano. Así, en el terminal, habrá que teclear:

```
nano hola.c (previamente habremos creado una carpeta para contener los archivos)
```

Después escribimos el programa:

```
// añade comentarios de línea, /* ... */ comentarios de varias líneas

#include <stdio.h> // incluimos la librería estándar de E/S para entrada/salida por consola

int main(){ // las llaves { ... } contienen las instrucc. de la función pral. (main) del programa:

// en C todo programa es un función

// int se refiere al valor entero que devolverá esta función pral. (puede no ponerse)

// sangramos: es adecuado por lectura

    printf("Hola, Mundo.\n"); // las instrucciones en C acaban en ;

return 0;

} // la llave } indica final de la función main
```

El programa, sin comentarios, sería:

```
#include <stdio.h>

int main(){

    printf("Hola, Mundo.\n"); // las instrucciones en C acaban en ;

return 0;

}
```

Ahora compilamos:

```
gcc -o hola hola.c (hola sería el ejecutable final)
```

Para ejecutar y probar:

```
./hola
```

### 2. Geany.

Geany es un entorno de desarrollo (IDE)-Compilador de C y de otros muchos lenguajes (sólo le falta la ejecución paso a paso)

Se instala:

```
sudo apt-get install geany (en el Terminal, como siempre)
```

y después de esto, lo tendremos en el escritorio.

Entrando en él, la sintaxis (C u otra) se elige en Document|Set Filetype|...Languages

### 3. Librería BCM2835

Esta librería permite usar el puerto GPIO de la Raspberry Pi: el acceso puertos, I2C, PWM, etc.

Ejecutamos sucesivamente:

```
cd; (va al raíz)
```

```
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.xx.tar.gz (xx es la versión: p.e. 36, que sería la última a día 24 Abril 14)
```

```
tar xvfz bcm2835-1.5.tar.gz (xvfz son los permisos)
```

```
cd bcm2835-1.xx (nos movemos al directorio bcm2835-1.xx)
```

```
./configure
```

```
make
```

```
sudo make check
```

```
sudo make install
```

Probamos con un ejemplo (parpadeo LED):

En <http://airspayce.com/mikem/bcm2835/> hay varios ejemplos Así: blink.c

```
// blink.c
```

```
//
```

```
// Example program for bcm2835 library
```

```
// Blinks a pin on an off every 0.5 secs
```

```
//
```

```
// After installing bcm2835, you can build this
```

```
// with something like:
```

```
// gcc -o blink blink.c -l bcm2835
```

```
// sudo ./blink
```



```
//  
  
// Or you can test it before installing with:  
  
// gcc -o blink -I ../../src ../../src/bcm2835.c blink.c  
  
// sudo ./blink  
  
//  
  
// Author: Mike McCauley  
  
// Copyright (C) 2011 Mike McCauley  
  
// $Id: RF22.h,v 1.21 2012/05/30 01:51:25 mikem Exp $  
  
#include <bcm2835.h>  
  
// Blinks on RPi Plug P1 pin 11 (which is GPIO pin 17)  
  
#define PIN RPI_V2_GPIO_P1_13  
  
int main(int argc, char **argv){  
  
// If you call this, it will not actually access the GPIO  
  
// Use for testing  
  
// bcm2835_set_debug(1);  
  
if (!bcm2835_init())  
  
return 1;  
  
// Set the pin to be an output  
  
bcm2835_gpio_fsel(PIN, BCM2835_GPIO_FSEL_OUTP);  
  
// Blink  
  
while (1){  
  
// Turn it on  
  
    bcm2835_gpio_write(PIN, HIGH);  
  
        // wait a bit  
  
bcm2835_delay(500);  
  
    // turn it off  
  
bcm2835_gpio_write(PIN, LOW);  
  
        // wait a bit
```

```
bcm2835_delay(500);  
    }  
bcm2835_close();  
return 0;  
}
```

Para que funcione con la versión 2 de la Raspberry Pi el programa anterior, debemos sustituir en el ejemplo de la web

RPI\_GPIO\_P1\_11 por RPI\_V2\_GPIO\_P1\_13 y así nos soporta la versión 2 de R. Pi y utilizamos el pin 13, que tiene una masa a su lado.