

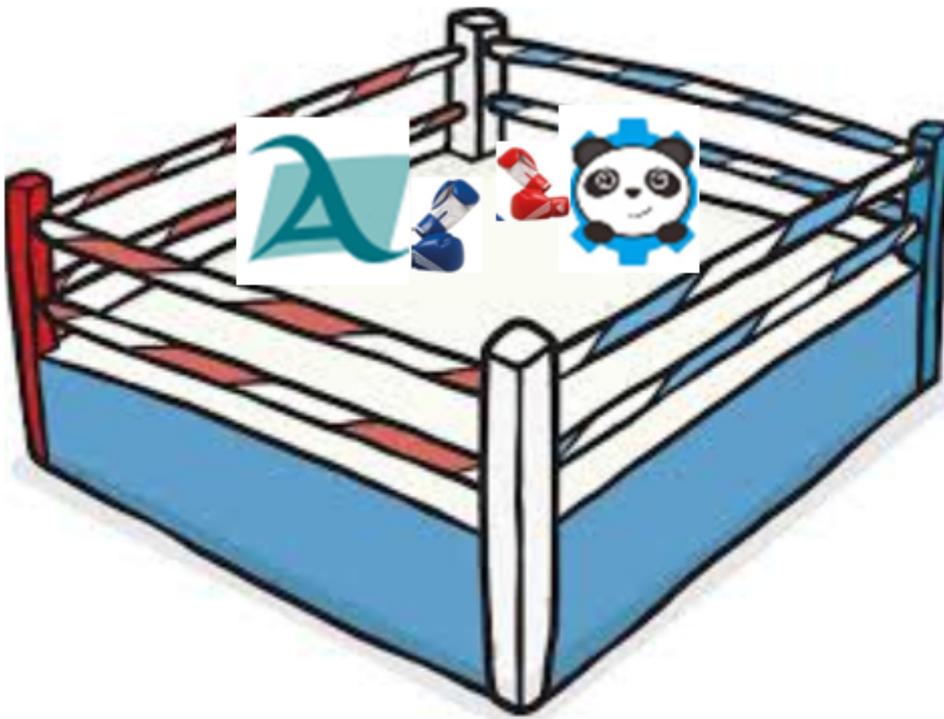
mBlock vs Snap4Arduino

¿cuál es el mejor?

Conclusión: mejor el lenguaje gráfico

¿Pero cual? Siempre es mejor moverse en estándares, y actualmente el lenguaje gráfico que está más extendido y donde hay más proyectos es **Scratch**, luego vamos a centrarnos en dos programas que se basan en Scratch: mBlock y Snap pero ¿cual es mejor? Snap4Arduino o mBock

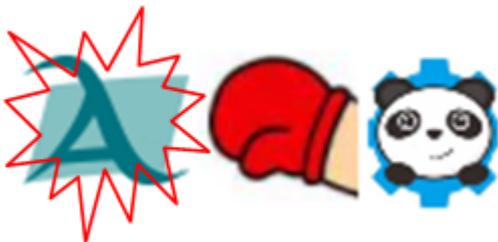
SNAP VS MBLOCK



<https://giphy.com/embed/UkrZKxuX8wjeM>

via [GIPHY](#)

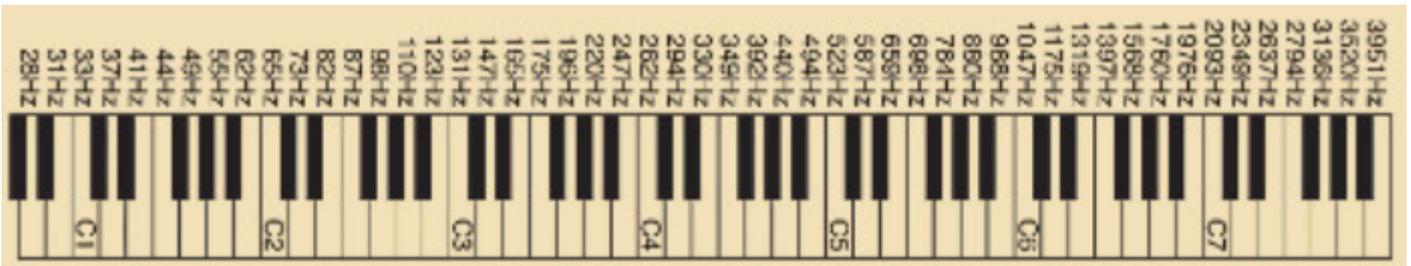
Instrucciones específicas para Arduino



Si entramos en la sección de instrucciones específicas para Arduino, en Snap4Arduino sólo tenemos las instrucciones básicas simples, a partir de estas piezas tienes que hacer el puzzle. Mientras que en mBlock tenemos más instrucciones que nos simplifican los programas pues la misma instrucción implican varias de las simples básicas:

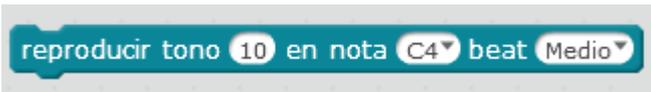


Esto parece una tontería pero no lo es: por ejemplo, supongamos que queremos reproducir el tono C4 es más o menos 262Hz:



Luego tiempo que dura la onda es $=1/262 = 0.0038$ segundos es decir el tiempo de “encendido HIGH” es $0.0038/2 = 0.0019$ seg y el tiempo de “apagado LOW” es $0.0035/2 = 0.0019$ seg.

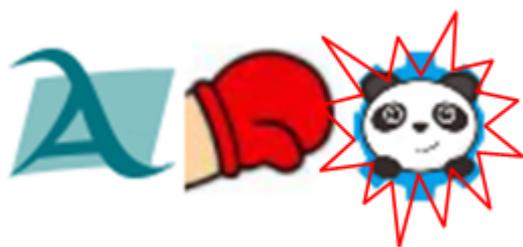
o sea... que para hacer esta instrucción en mBlock



necesitas hacer todo esto en Snap4Arduino :

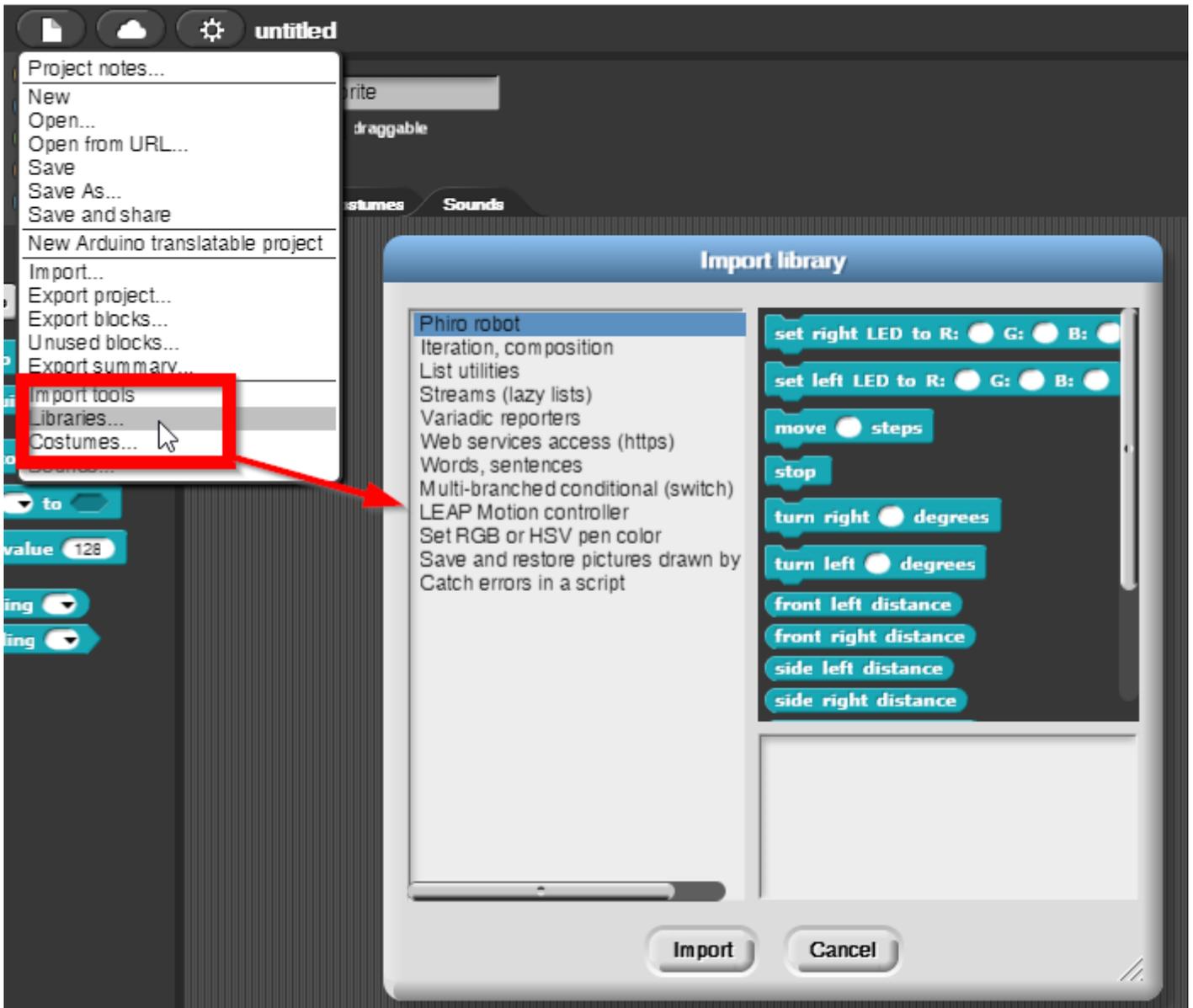


Importación de librerías



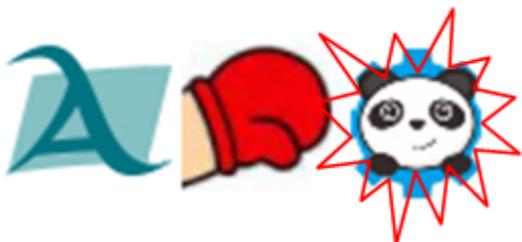
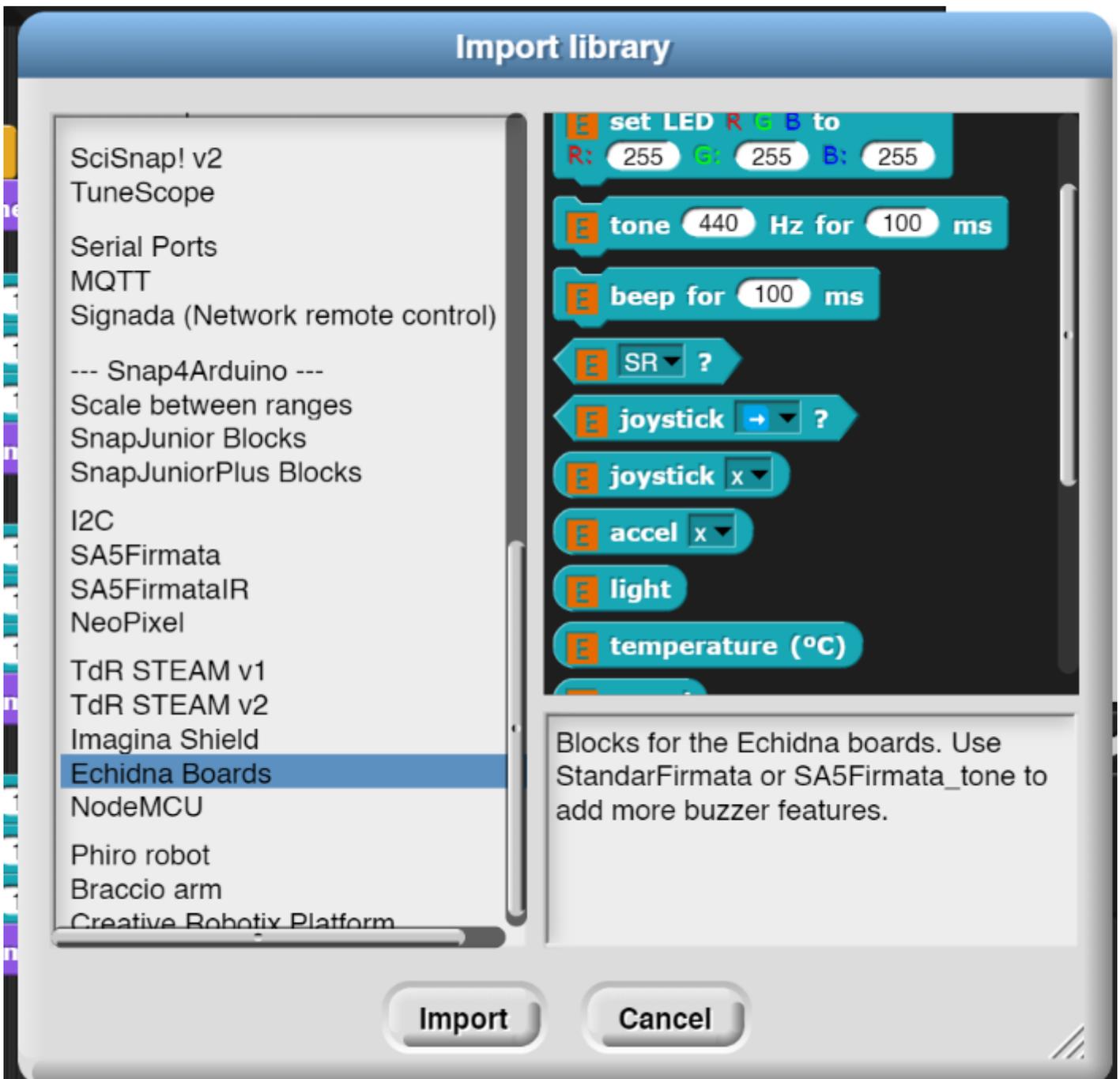
Snap4Arduino puede importar instrucciones o librerías que pueden aumentar el repertorio de instrucciones.

mBlock también tiene en Extensiones-Administrar extensiones pero no son tan específicas para el Arduino.

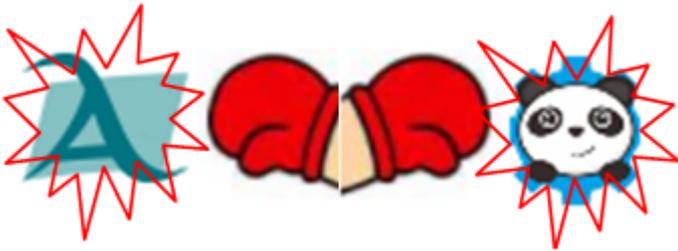


Librería propia para Echidna

Snap4Arduino al ser un software libre, estará siempre ligado a los prototipos que se lancen en hardware libre, y en este caso lo tenemos



Reutilizar proyectos desde Scratch



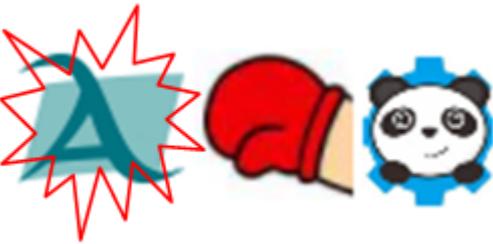
Si vemos un proyecto en [Scratch](#) que nos gusta y lo queremos reutilizar lo descargamos y si lo queremos reutilizar...

En mBlock lo podemos abrir, pero **no** siempre va bien.

En Snap4Arduino no acepta ficheros Scratch, pero hay conversores online como [Snapin8r](#), y también da problemas.

Conclusión: tortazo a los dos!!

Entorno amigable



En Snap4Arduino no existe un menú arriba, las opciones no se encuentran tan inmediatamente

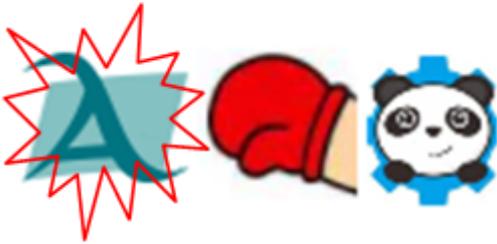


Mientras que en mBlock es más intuitivo:



Importante CARGAR A ARDUINO

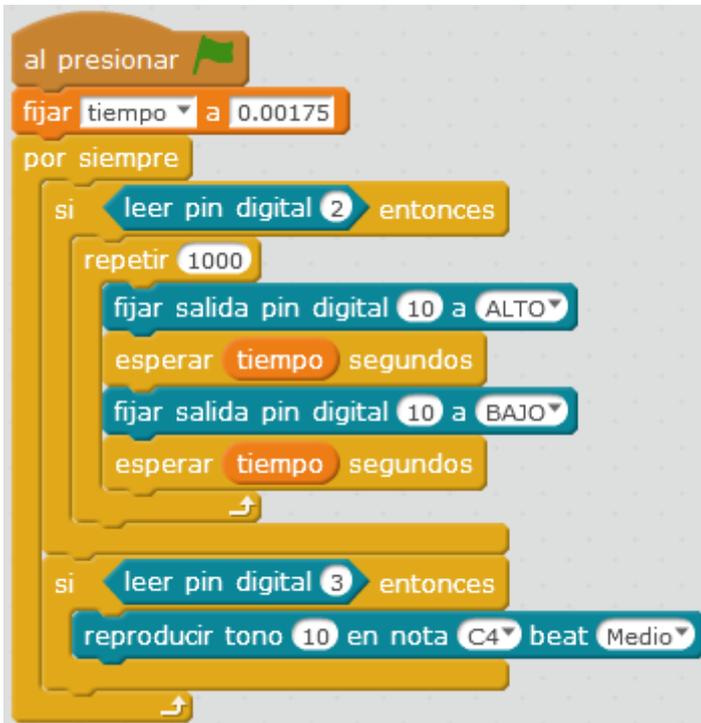
En mBlock podemos poner la vista de Modo Arduino Y NOS MUESTRA NUESTRO PROGRAMA ESCRITO EN LENGUAJE GRAFICO CONVERTIDO A LENGUAJE CÓDIGO



Esto nos permite **SUBIR AL ARDUINO** es decir, nuestro programa grabarlo en el Arduino como si fuera un Firmware y ganar velocidad aquí tienes una captura y puedes observar cómo traduce el programa gráfico en código:

A screenshot of the mBlock software interface. The top navigation bar includes 'makeblock | mBlock', 'Sin título', 'Guardar', 'Publicar', 'Curs...', 'Tutoria...', and 'Comenta'. The main workspace is divided into a left sidebar with 'Dispositivos', 'Objetos', and 'Fondo' tabs, and a central area with a block palette and a workspace. The 'Subir' button in the 'Dispositivos' section is highlighted with a red box. A red arrow points from this button to a code block in the workspace. The code block, also highlighted with a red box, contains the following logic: 'cuando Arduino Uno se inicia' (when Arduino Uno starts), followed by a 'para siempre' (forever) loop containing: 'pon el pin digital 9 a alto' (set digital pin 9 to high), 'espera 1 segundos' (wait 1 second), 'pon el pin digital 9 a bajo' (set digital pin 9 to low), and another 'espera 1 segundos' (wait 1 second). The block palette on the left includes categories like Pin, puerto serie, Datos, Sensor, Eventos, Control, Operador, Variables, and extensión.

QUEREMOS EXPLICARTE LO IMPORTANTE QUE ES ESTO El anterior programa funciona bien si lo hacemos subir al Arduino, pero si lo hacemos de forma normal (cambiamos el evento “Programa Arduino” por evento bandera):



NO FUNCIONA BIEN

¿por qué?: por la lentitud.

Si apretamos el pulsador 2 le pedimos al ordenador que encienda y apague D10 a una velocidad de 0.00175 segundos esto lo tiene que procesar el ordenador, pasa por el cuello de botella del puerto serie y al final se ejecuta en el Arduino a una velocidad mucho más baja, más o menos 0.5 segundos, en vez de oírse un tono, se oye tut,tut,tut,tut....

Si apretamos el pulsador 3 sí que se oye bien, porque mBlock manda el código del bucle de golpe para que se oiga bien.

Volveremos a este programa varias veces, no te preocupes si no lo has entendido del todo bien.

Desventajas de subir un programa al Arduino:

1. **NO PODEMOS INTERACTUAR CON EL ORDENADOR** es decir, no podemos hacer que el sprite (el oso por defecto) se mueva o que al pulsar una tecla del ordenador... porque EL ARDUINO ES INDEPENDIENTE DEL ORDENADOR, incluso podemos desconectarlo y sigue funcionando !!!
2. **Nos hemos cargado el Firmware** que tenía antes !! luego al finalizar tu programa, tenemos que volver a configurar el Arduino para que haga caso al programa mBlock visto en [esta presentación](#).

Ventajas de subir un programa al Arduino:

1. **Rapidez**, tenemos la ventaja del lenguaje con código, el programa se ejecuta a la velocidad del Arduino sin tener el ordenador como intermediario.

2. **Independencia:** Alimentamos el arduino con una batería ¡¡ y funciona !! esto es esencial para proyectos que impliquen movimiento y no queramos tener el ordenador encendido para que funcione.

Aunque...: Rapidez en la simulación

Aunque Snap4Arduino no sube el programa, sólo simula, es mucho más rápido en la simulación que mBlock



En este [apartado](#) veremos las diferencias.

Estan algo reñidos, pero le vamos a dar el premio a mBlock por dos razones

- Subir al Arduino
- Entorno amigable



Todo es cuestión de gustos, las capturas y descargas de programas de este curso serán en mBlock, pero algunas veces utilizaremos Snap4Arduino cuando nos interesa rapidez en la simulación,.