

¿Qué lenguaje gráfico es mejor para Echidna?

EchidnaScratch

- Es un programa específico para Echidna, tiene ya **instrucciones especiales**



- **EchidnaScratch** es un programa que permite usar la placa **EN VIVO** esto nos permite interactuar con el ordenador
- EchidnaScratch esta integrado con la **INTELIGENCIA ARTIFICIAL** LearningML

mBlock

- Es un programa muy extendido multiplaca
- Permite programar **EN VIVO** igual que EchidnaScratch
- Pero también permite **CARGAR EL PROGRAMA EN LA PLACA lo que permite crear proyectos independientes del ordenador**
- En el caso de la EchidnaBlack lo lee como si fuera un **Arduino**
- Para la comunicación entre el Device Echidna y los diferentes Sprites del programa hay que usar **MENSAJES** lo que hace un poco engorroso el programa
- También permite la IA pero con extensiones, ver <https://libros.catedu.es/books/robotica-educativa-con-mbot/page/cognitive-services-servicios-cognitivos>

Snap4Arduino

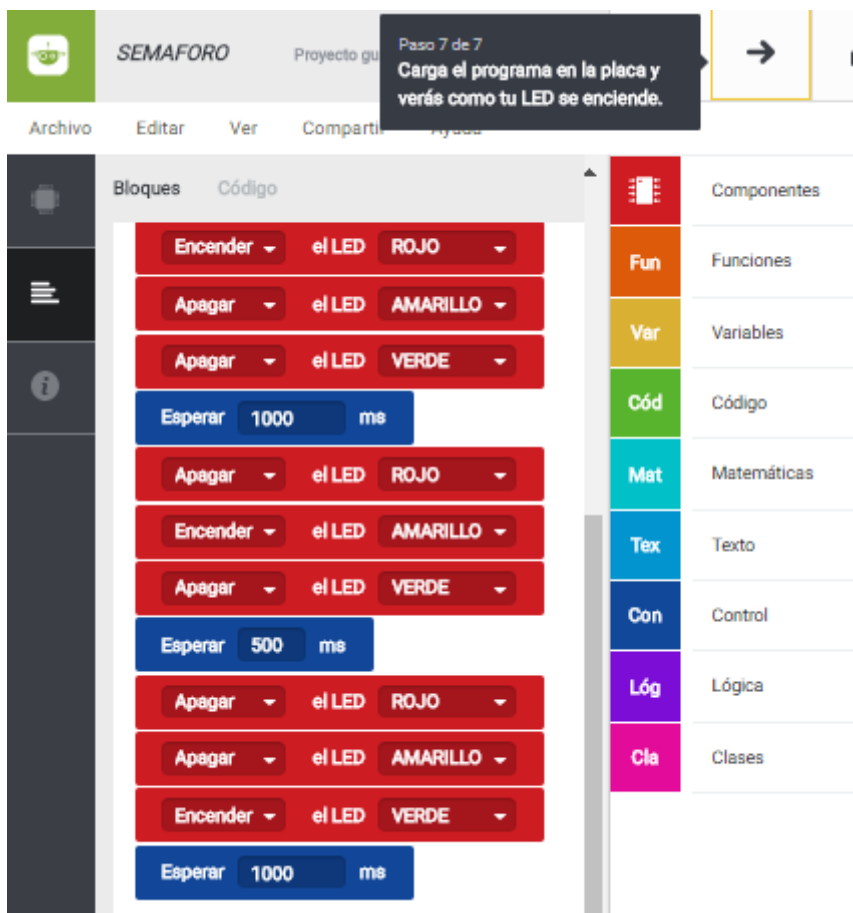
- También es un programa que permite usar EchidnaScratch **en vivo**
- Tiene **instrucciones especiales** para la placa

NOS DECANTAMOS POR ECHIDNASCRATCH aunque en este curso verás también ejemplos con mBlock

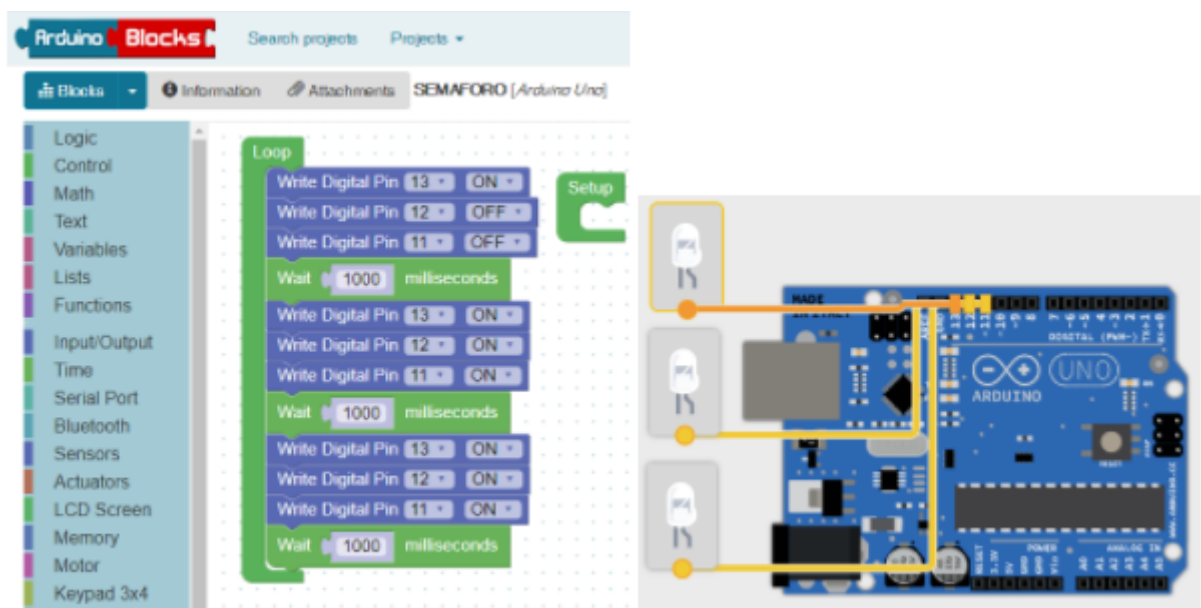
Otros programas gráficos

Existen otros programas gráficos, el ArduinoBlocks, Bitblock... pero no tienen la posibilidad de iteraccionar con Sprites del ordenador, ni tienen el mismo entorno del referente Scratch, por lo que no se utilizarán en este curso, pero son posibilidades interesantes, os mostramos dos capturas de pantalla del mismo programa semáforo visto anteriormente, fíjate que aquí no hay *Sprite*:

Con Bitbloq:



Con ArduinoBlocks:



Y un favorito nuestro el **VISUALINO** pues a la vez que haces programación gráfica, va enseñando el código a la derecha:

Ayuda

The image displays the Arduino IDE interface with two panels. The left panel shows a Scratch-style block-based program, and the right panel shows the corresponding C++ code.

Block-based Program (Left Panel):

- Arduino Inicio** (Green block):
 - LCD** (Green block):
 - CS - DC - RESET: 10 9 8
 - Fondo: [Black square]
- Repetir** (Green block):
 - Declarar variable** (Purple block): Luz = [Luz]
 - Pin analógico** (Red block): A0
 - Var** (Purple block): Luz
 - LCD: imprimir** (Green block):
 - Fondo: [Black square]
 - Texto: [White square]
 - ¿Fijar posición del texto?: [Yes]
 - Esperar: [Yes]

C++ Code (Right Panel):

```
#include <IFTI.h>

/** Global variables */
IFTI IFTIscreen = IFTI(10, 9, 8);

/** Function declaration */

void setup()
{
  IFTIscreen.begin();
  IFTIscreen.background(0, 0, 0);

  pinMode(A0, INPUT);
}

void loop()
{
  int Luz=( (6.0 * analogRead(A0) * 1000.0 )
  IFTIscreen.stroke(255, 255, 255);
  IFTIscreen.text(Luz.c_str(),0,0);
  IFTIscreen.stroke(0, 0, 0);
  IFTIscreen.text(Luz.c_str(),0,0);
}
```

Revision #21

Created 31 March 2022 08:48:42 by Equipo CATEDU

Updated 29 November 2024 08:46:08 by Javier Quintana