ARDUINO en la Enseñanza de la ELECTRÓNICA, PROGRAMACIÓN, TECNOLOGÍA

Introducción

Sesión realizada por: Ángela Diez Diez
Dpto: Ingeniería Eléctrica y de Sistemas y Automática

UNIVERSIDAD DE LEÓN

Objetivos

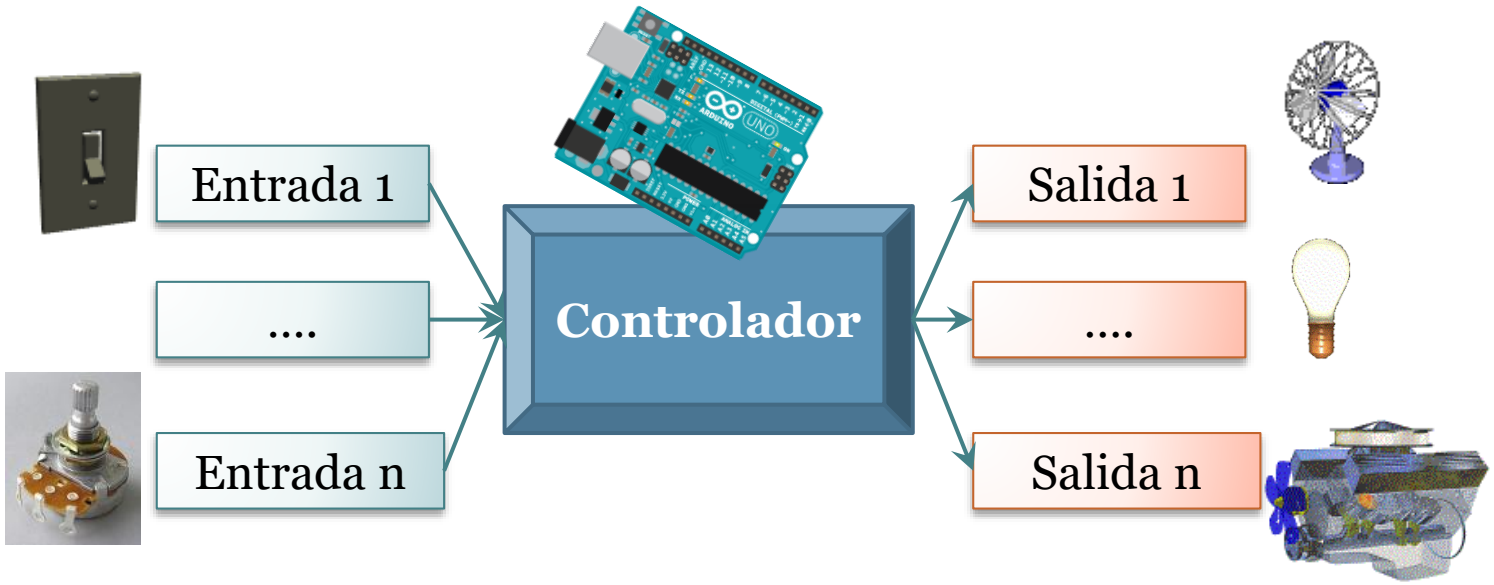
- Dar a conocer a los alumnos de Tecnología de Primaria de una herramienta Open Source.
- Aprende a :
  - **P**ensar
  - **A**nalizar,
  - **P**robar tus ideas
- Te mostramos una herramienta.
- Pero, recuerda hay otras muchas que puedes usar.

Arduino ¿Qué es?

- Dispositivo
- Tarjeta
- Controlador
- Placa

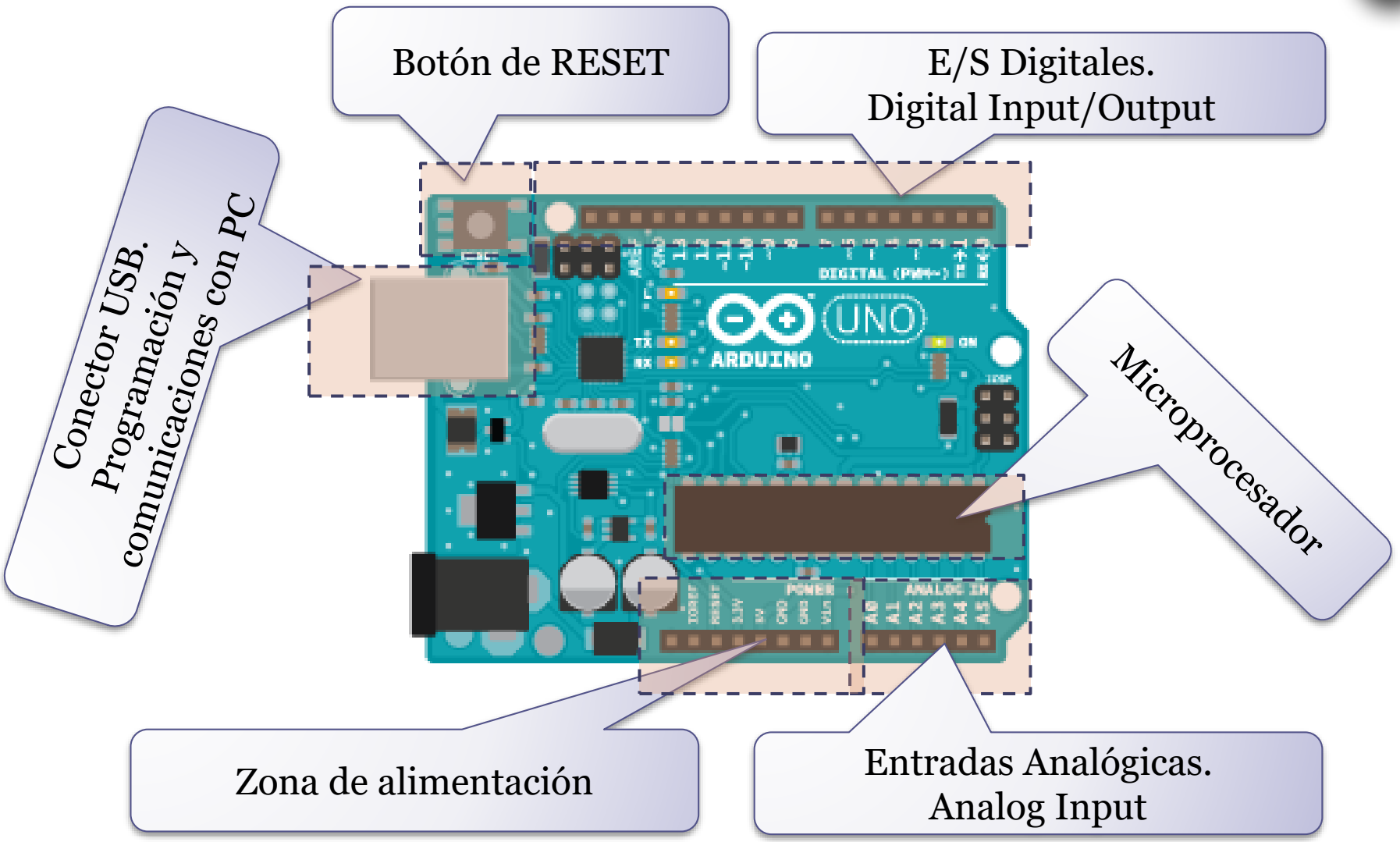
- Es un **elemento programable** que posee entradas y salidas digitales, analógicas.
- Nos permite realizar pequeños proyectos domésticos tomando como base la electrónica, la robótica y la automática.

Controladoras: Elementos



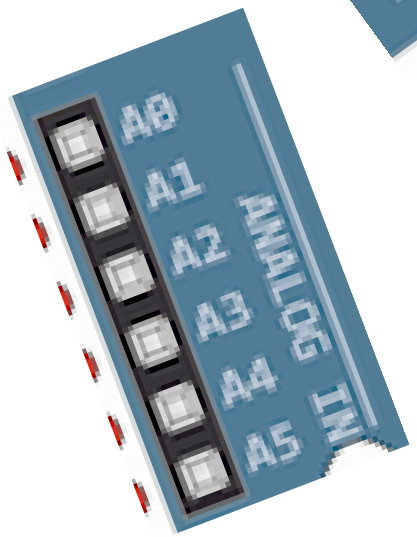
- Señales **entrada**
  - Pulsadores
  - Interruptores
  - Sensores luz, …
- Señales de **Salidas**
  - Luces
  - Sirenas
  - Motores, ….

Arduino: HARDWARE



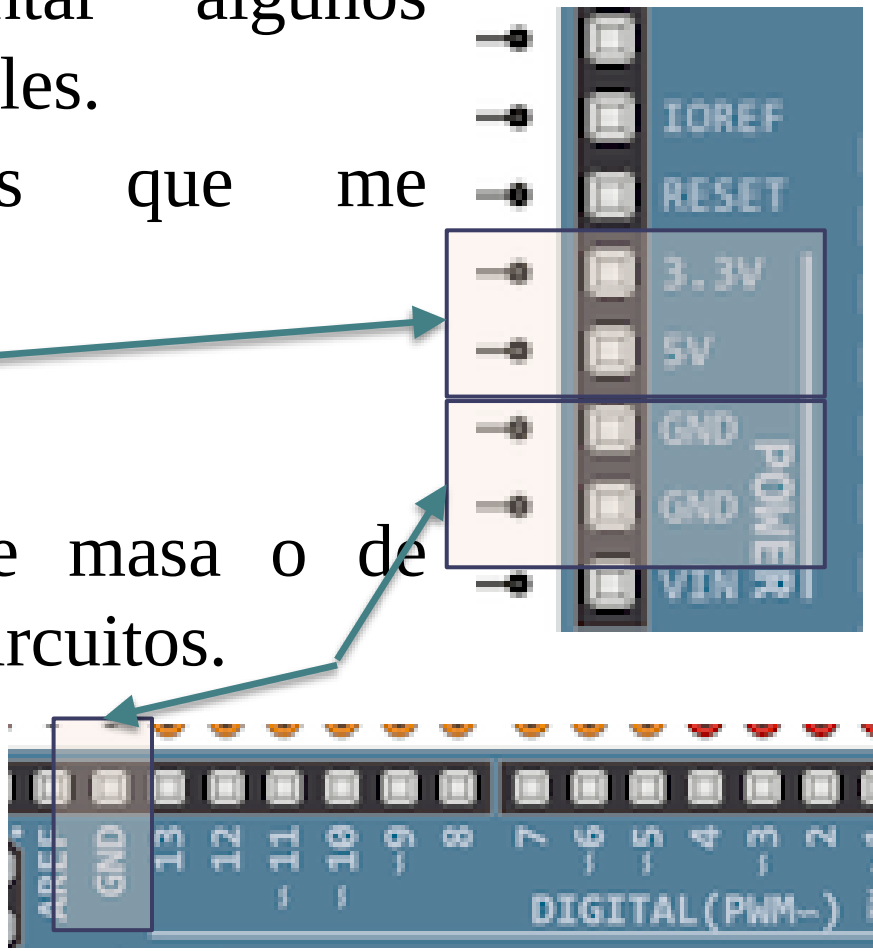
ARDUINO: HARDWARE

- Pines digitales:
  - Se **configuran**: **INPUT** (Entrada), **OUTPUT** (Salida).
  - Tienen un número Identificativo: 13, 12, ,,,,,2. El 0 y el 1 lo utiliza internamente para comunicarse con el PC.
- Pines analógicos
  - Son todas Entradas
  - Tienen un número que las identifica.



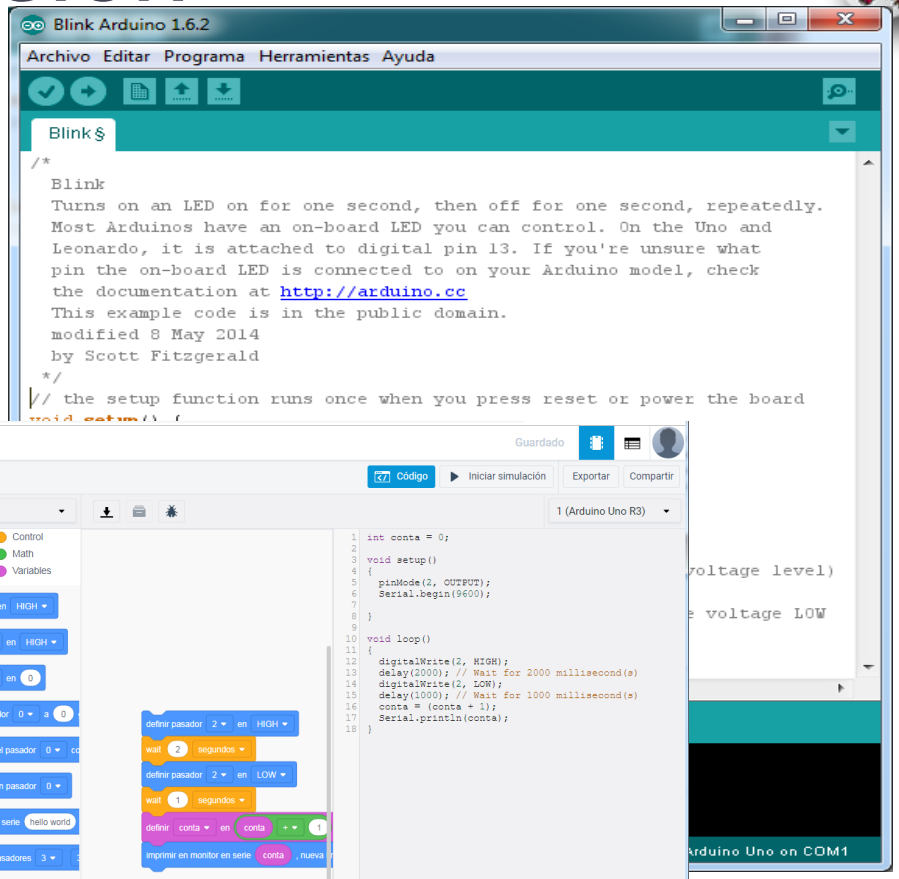
ARDUINO: HARDWARE

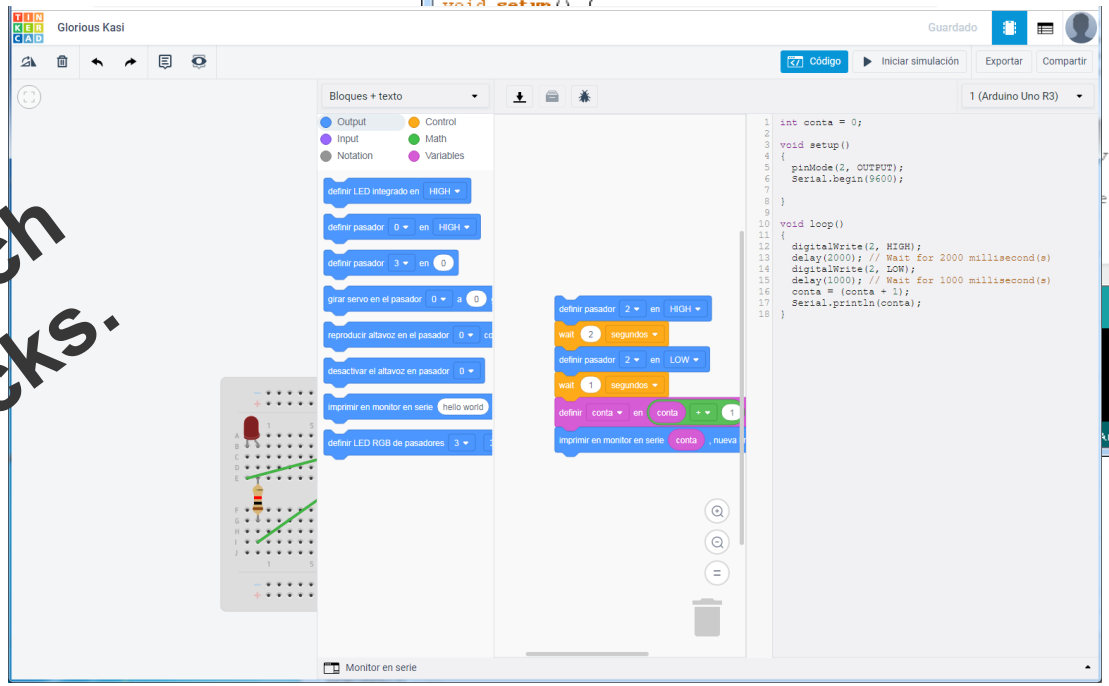
- Arduino permite alimentar algunos dispositivos o sensores simples.
- Las tensiones eléctricas que me proporcionan son:
  - 5V
  - 3,3V
- También me da líneas de masa o de referencia (GND) para los circuitos.



Entornos programación

- TEXTUAL
- GRÁFICOS
- BLOQUES

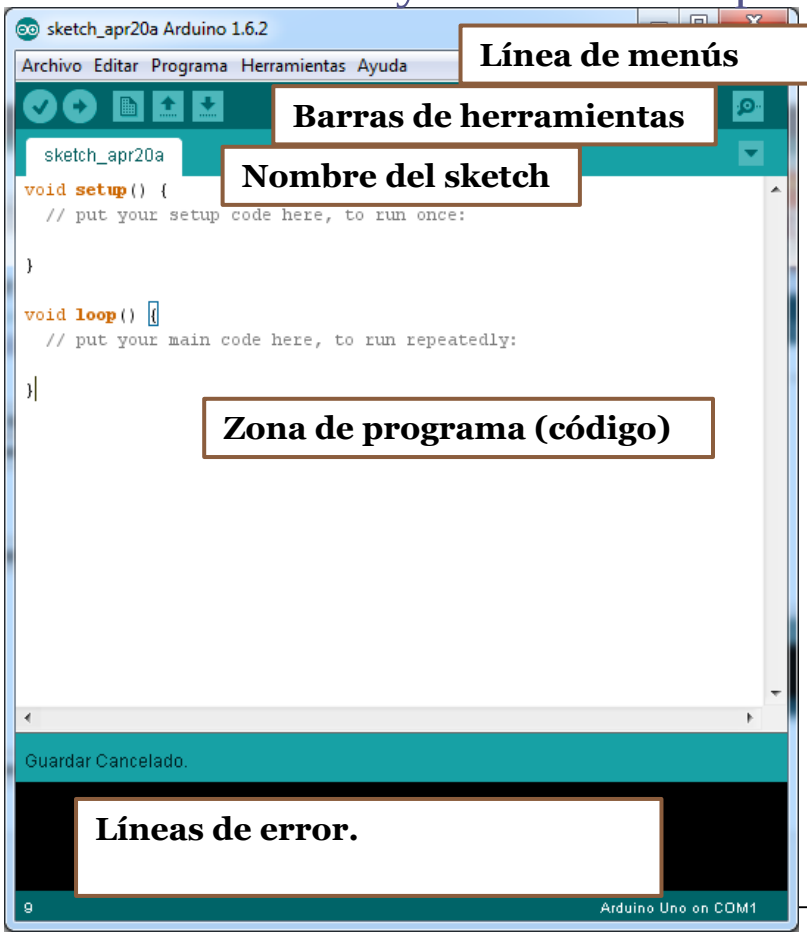


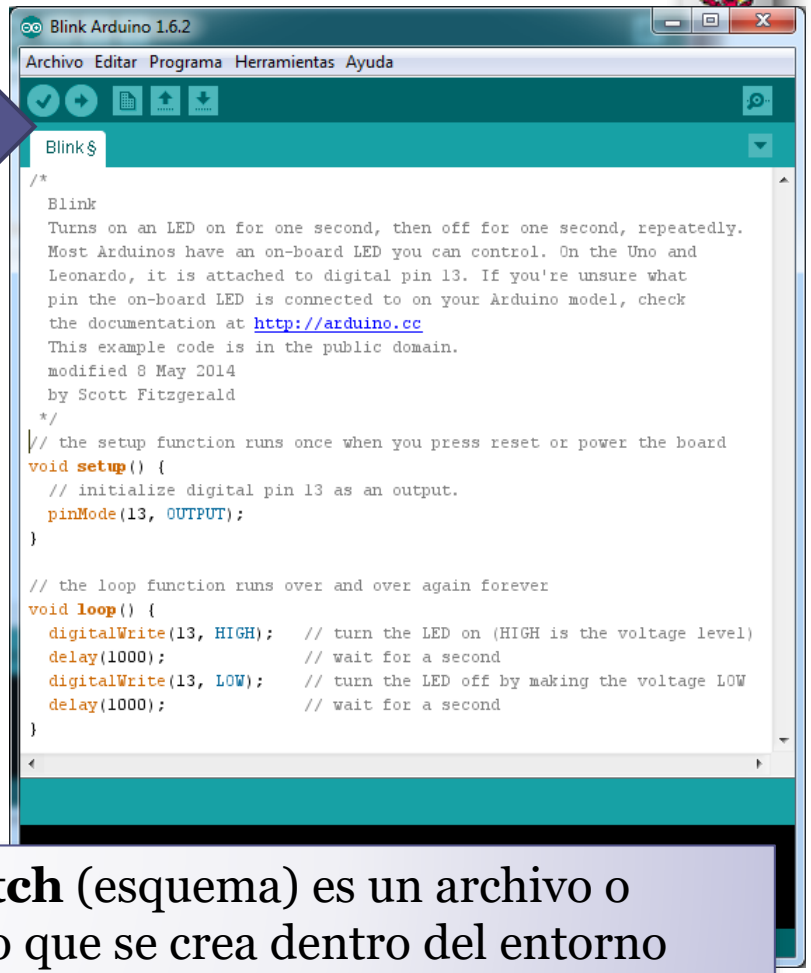


Scratch Blocks.

Arduino: Entorno programación

- IDE de programación.
  - IDE=Entorno de desarrollo integrado
    - Editor y el entorno de programación





- Un **sketch** (esquema) es un archivo o proyecto que se crea dentro del entorno IDE del Arduino.
- La primera vez que abrimos un nuevo proyecto aparecerá en blanco.

ARDUINO: Software: barra de herramientas

New. Crea un sketch nuevo, en blanco.

VERIFICAR. Comprueba si el código es correcto.

Open: Abrir archivo

Save Crea/Salva el archivo. La extensión es .pde // .ino.

Subir: Descarga el programa en el controlador y se ejecuta.

Nueva versión IDE v2

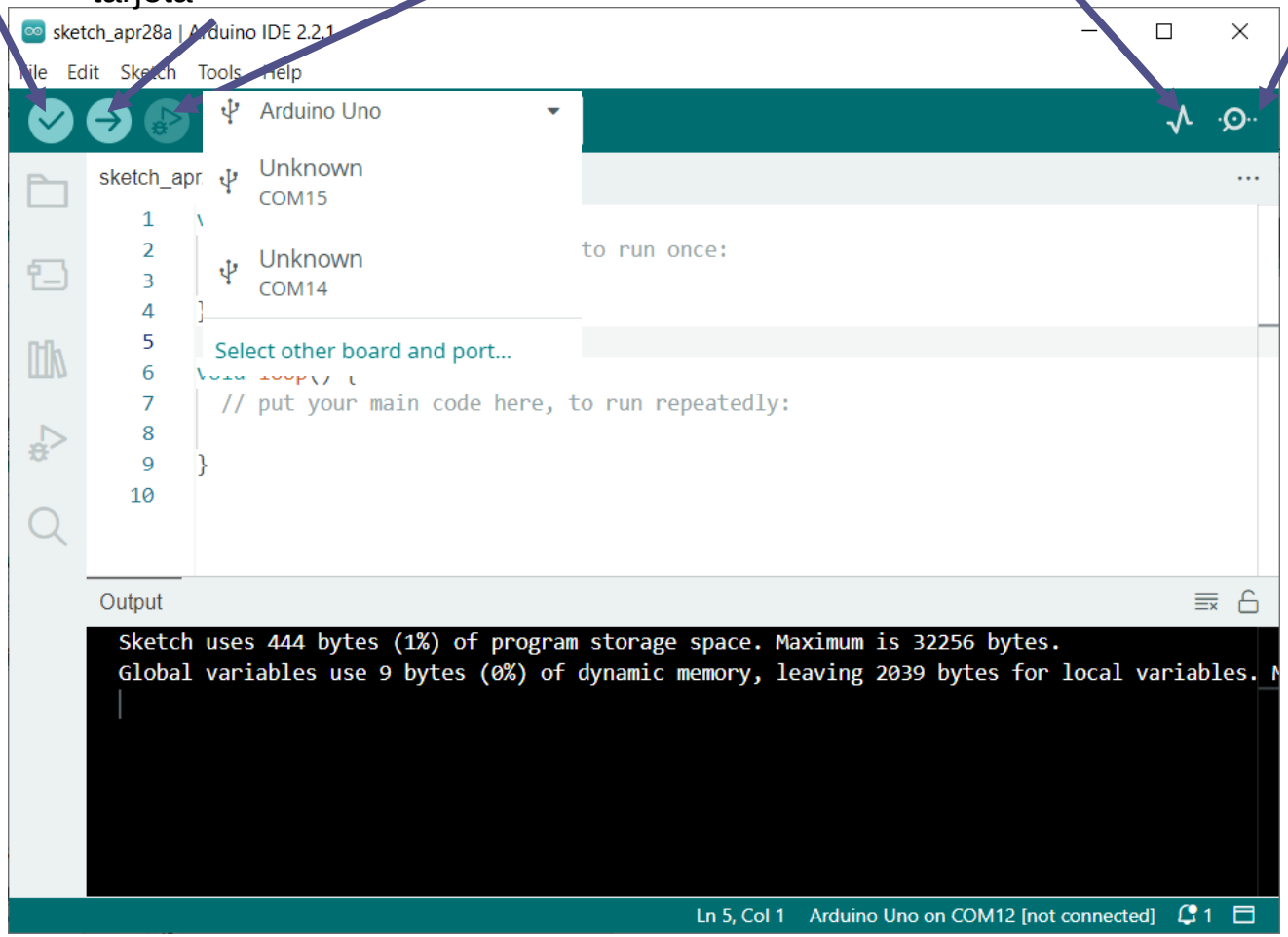
Verifica el programa

Descarga el programa en la tarjeta

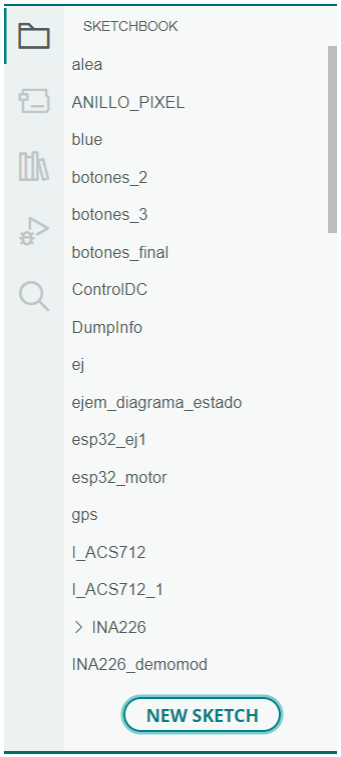
Depura el programa

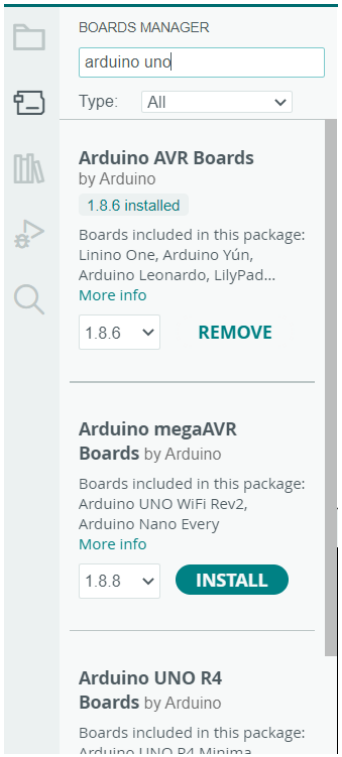
Plotter serie

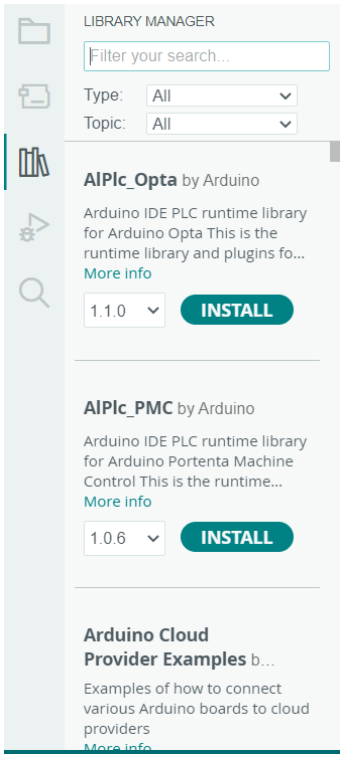
Monitor serie

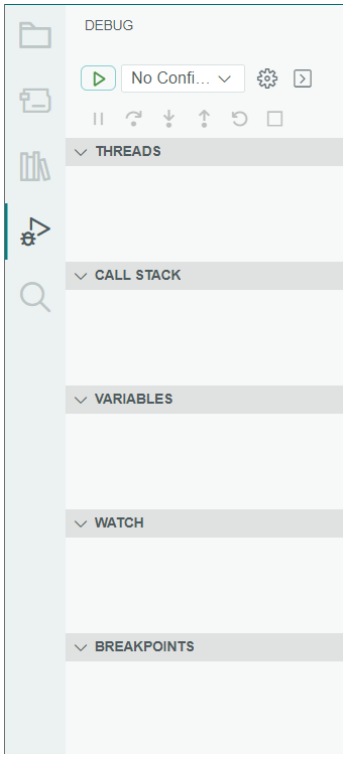


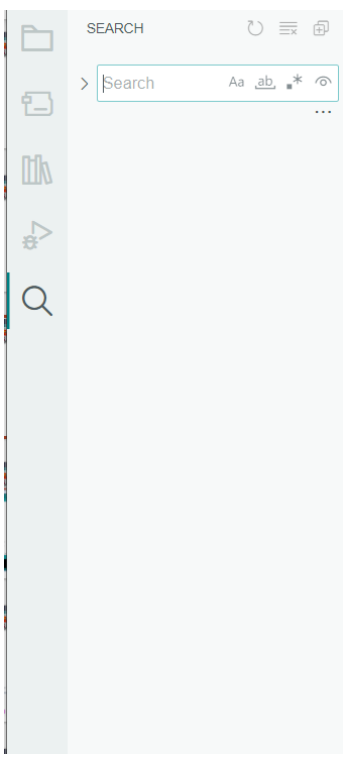
Opciones del IDE v2











ARDUINO SOFTWARE

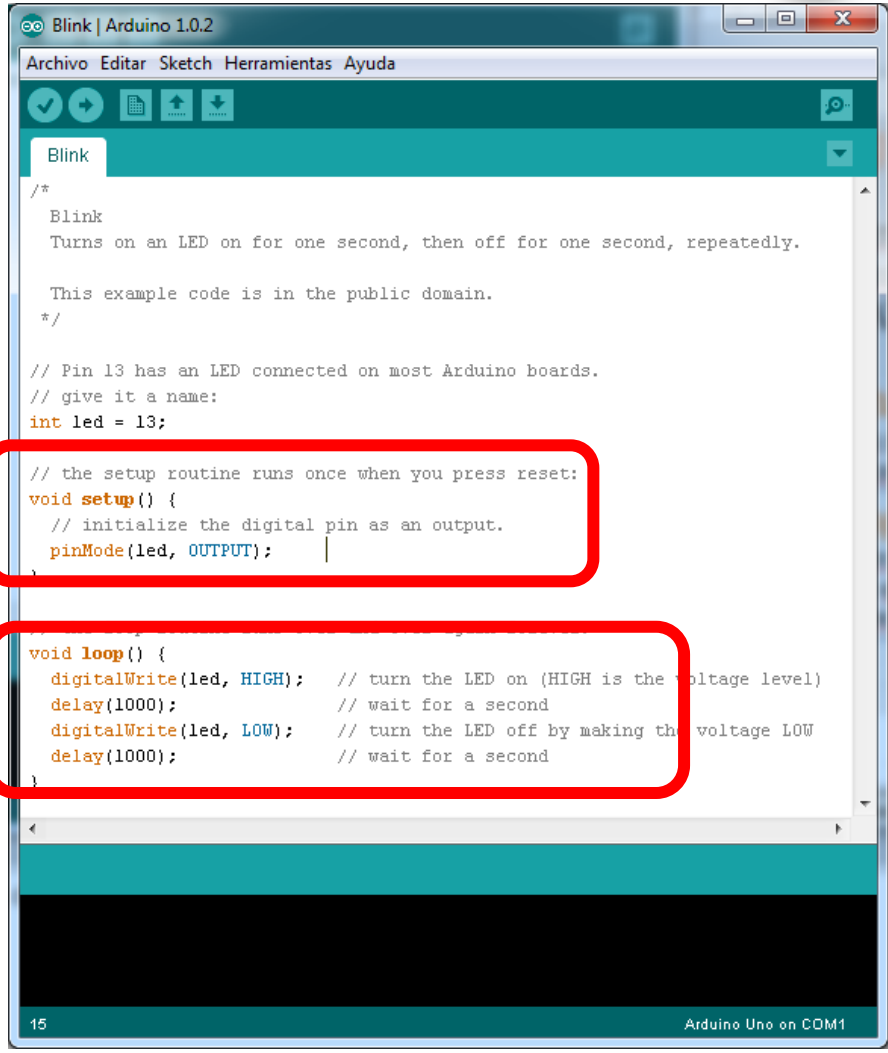
Routines

Cada programa Arduino se suele denominar SKETCH y tiene dos funciones OBLIGATORIAS , denominadas **Rutinas**.

void setup ( ) { } - Todo el código interno dentro de los corchetes se ejecuta **UNA ÚNICA VEZ** cuando se carga inicia el controlador. **Permite configurar e inicializar nuestro sistema**.

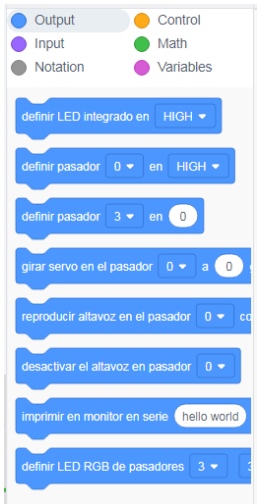
void loop ( ) { } - Esta función se ejecuta DESPUÉS que la función setup ha finalizado.

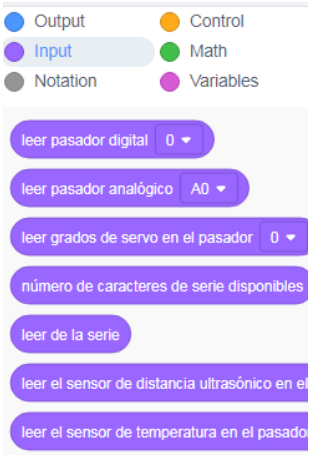
El código dentro de los corchetes se **ejecutan de forma continua una y otra vez** hasta que se elimine la alimentación de la tarjeta.

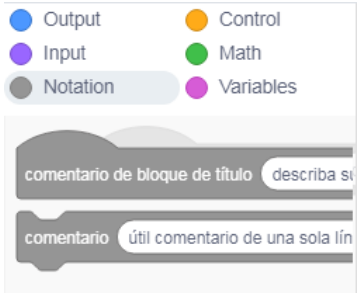


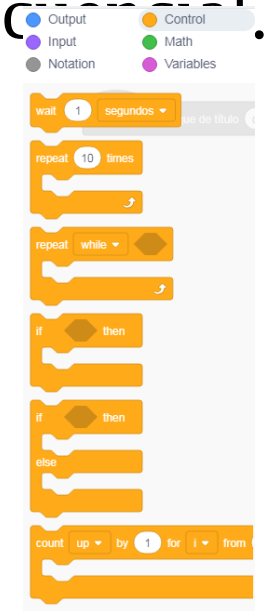
BLOQUES

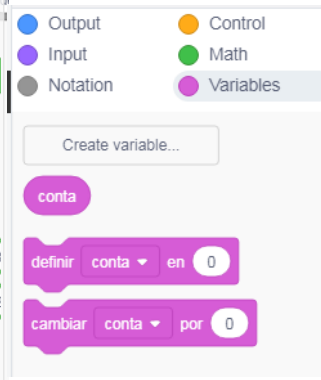
- Lenguajes están basados en una lógica de bloques donde el color identifica el tipo de bloque, y la forma permite que encajen entre sí.
- Permiten completar una secuencia de operaciones.
- Orientado a programación secuencial.





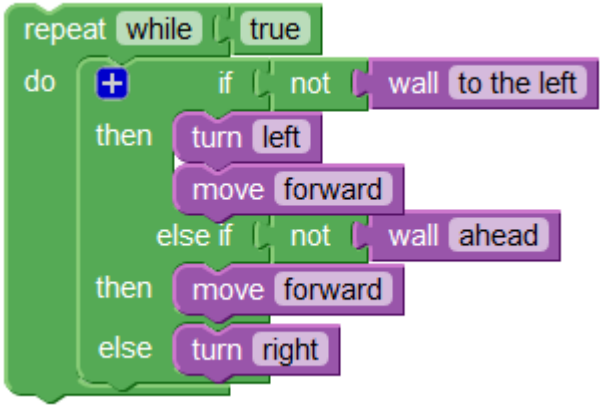






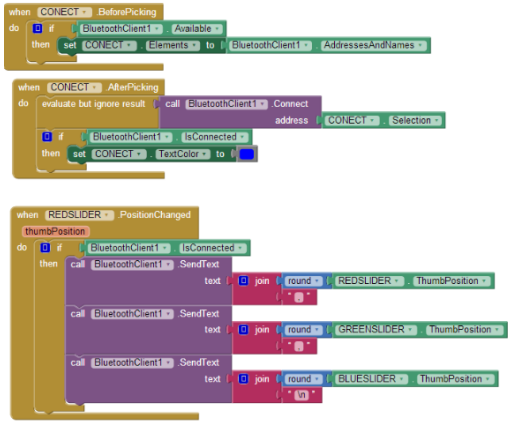
Secuencia gráfica

- Tipo scratch



App Inventor

MIT App Inventor



App Inventor Blocks Editor

Tinkercad

ARDUINO

MI PRIMER EJEMPO

¿Qué QUIERO HACER?

- Necesitamos saber describir lo que queremos hacer.
- No te preocupes. Es un ejercicio de redacción muy simple.

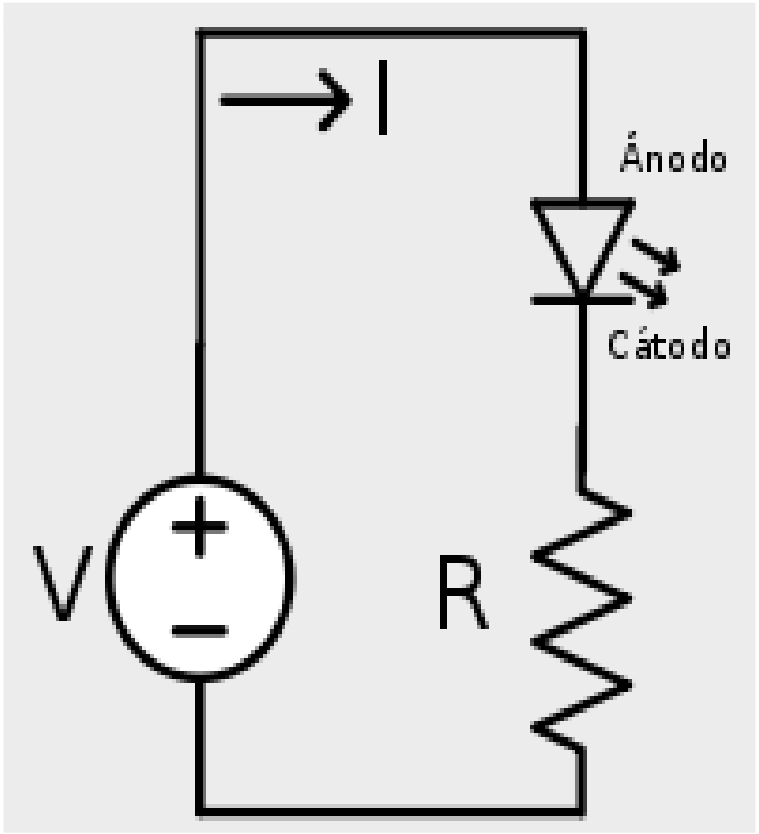
EJEMPLO:
Necesitamos hacer **parpadear** una pequeña **luz** durante un determinado **tiempo**.

¿Qué DISPOSITIVOS o elementos NECESITO?

- Podemos usar un LED, como fuente de luz.
- Necesitamos algo que cuente o gestione tiempo (segundos).

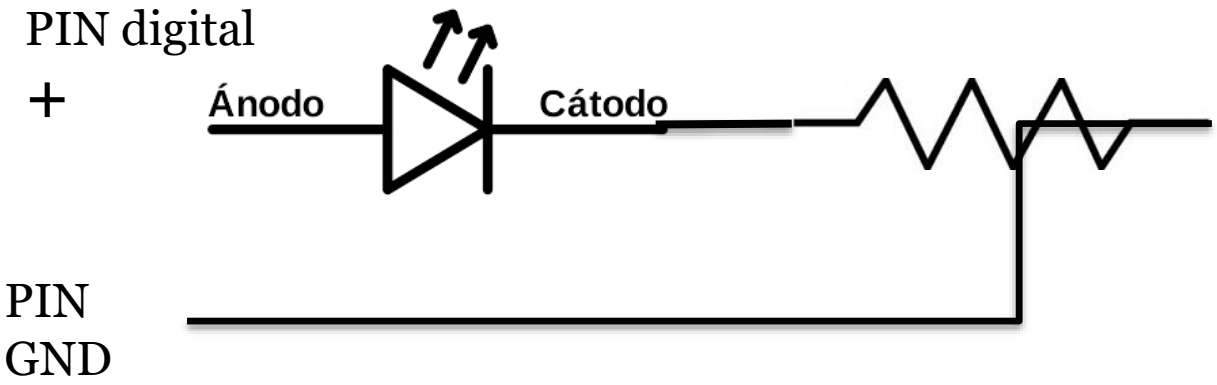
¿Cómo funciona LED?

- Para hacer esto tengo que documentarme sobre el dispositivo LED.
  - Podemos usar los libros de clase
  - Podemos usar la red de Internet. (!!no te creas todo lo que pone!!). Selecciona páginas del **entorno educativo**.

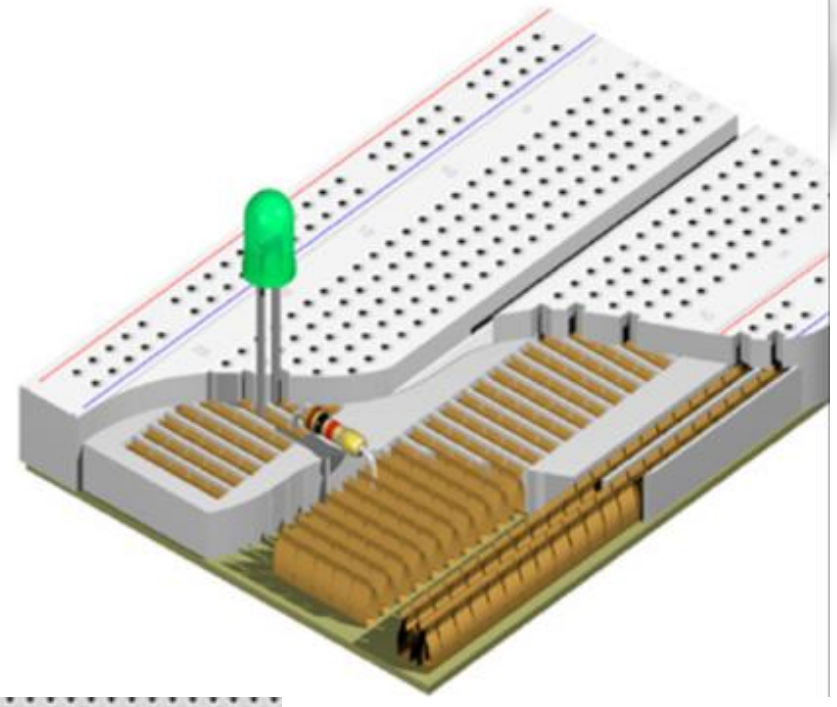


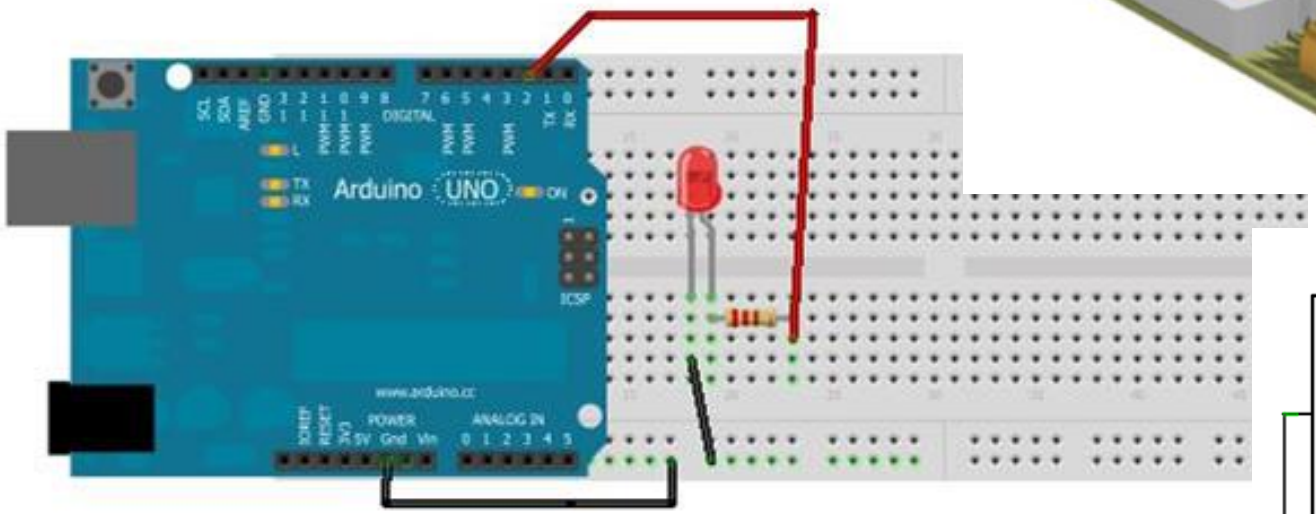
Conexión LED a Arduino

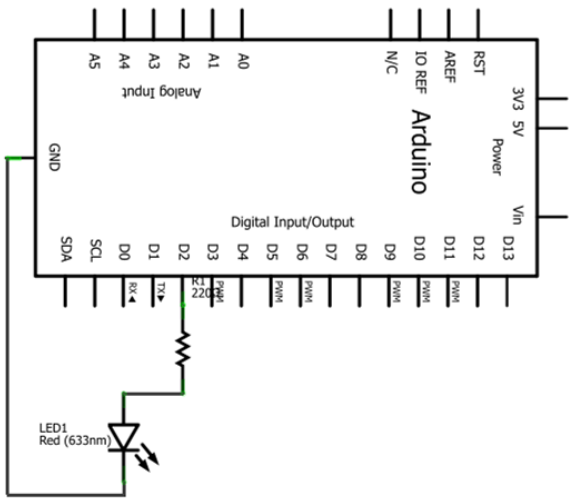
- Usaremos un pin que podamos manejar desde el micro.
- Los pines digitales los podemos poner a 5V, que se denomina HIGH, o a 0V LOW. Decimos que lo ponemos a 1 o a 0.
- Sustituimos la pila y el pulsador por la entrada digital.



LED: conexión a Arduino







¿Cómo defino la secuencia de acciones que tiene que realizar el MICROCONTROLADOR?

- Para definir la secuencia de acciones, primero pensamos en los distintos pasos que tenemos que dar.
- Por ejemplo pueden ser del tipo:
  - Inicializar comunicaciones
  - Configurar pin digital como entrada,….. O como salida
  - Poner a 1 o en Alto el pin ….
  - Poner a 0 o en Bajo el pin …..
  - Leer la entrada analógica nº….
  - Leer el valor del sensor de luz de la entrada analógica ….
  - Esperar un tiempo: segundos, milisegundos…
  - Leer puerto serie …
  - Escribir puerto serie…
  - Activar motor
  - Parar motor
  - Girar a derecha
  - Girar a izquierda
  - ……

Creamos el programa

- A partir del análisis anterior, generamos las órdenes de nuestro microcontrolador.
- Para ello tenemos que conocer las instrucciones que tenemos y su **Sintaxis**.

ANÁLISIS	**PROGRAMA**

INICIALIZAR
Configuramos el pin 2 como salida digital
FIN_INICIALIZAR
 	
Bucle
Poner pin digital 2 en ALTO
Esperar 1seg
Poner pin digital 2 en Bajo
Esperar 1seg
Volver a Bucle
	void **setup**() {
	pinMode(2, OUTPUT);
	}
	
	void **loop**() {
	digitalWrite(2, HIGH);
	delay(1000);
	digitalWrite(2, LOW);
	delay(1000);
	}

Documentamos el programa.

- Ponemos comentarios en el programa para indicar las acciones que realiza.
- Acciones:
  - Guardar el programa
  - Verificar programa
  - Si todo correcto entonces lo cargamos en el controlador.

Sintaxis lenguaje de programación

Nombre de la instrucción (parámetros, ….) **;**

- Recuerda que todas las instrucciones terminan en **;**
  - **Excepto:**
    - **setup(){},**
    - **loop(){},**
    - **y las que se llaman estructuras de control. if(){}…., for(){}…**

pinMode(2, OUTPUT);

Instrucción

Parámetros. Separados por coma

- loop()

- HIGH | LOW
- INPUT | OUTPUT|

- pinMode()
- digitalWrite()
- digitalRead()

Analog I/O

- analogReference()
- analogRead()
- analogWrite() - *PWM*

Advanced I/O

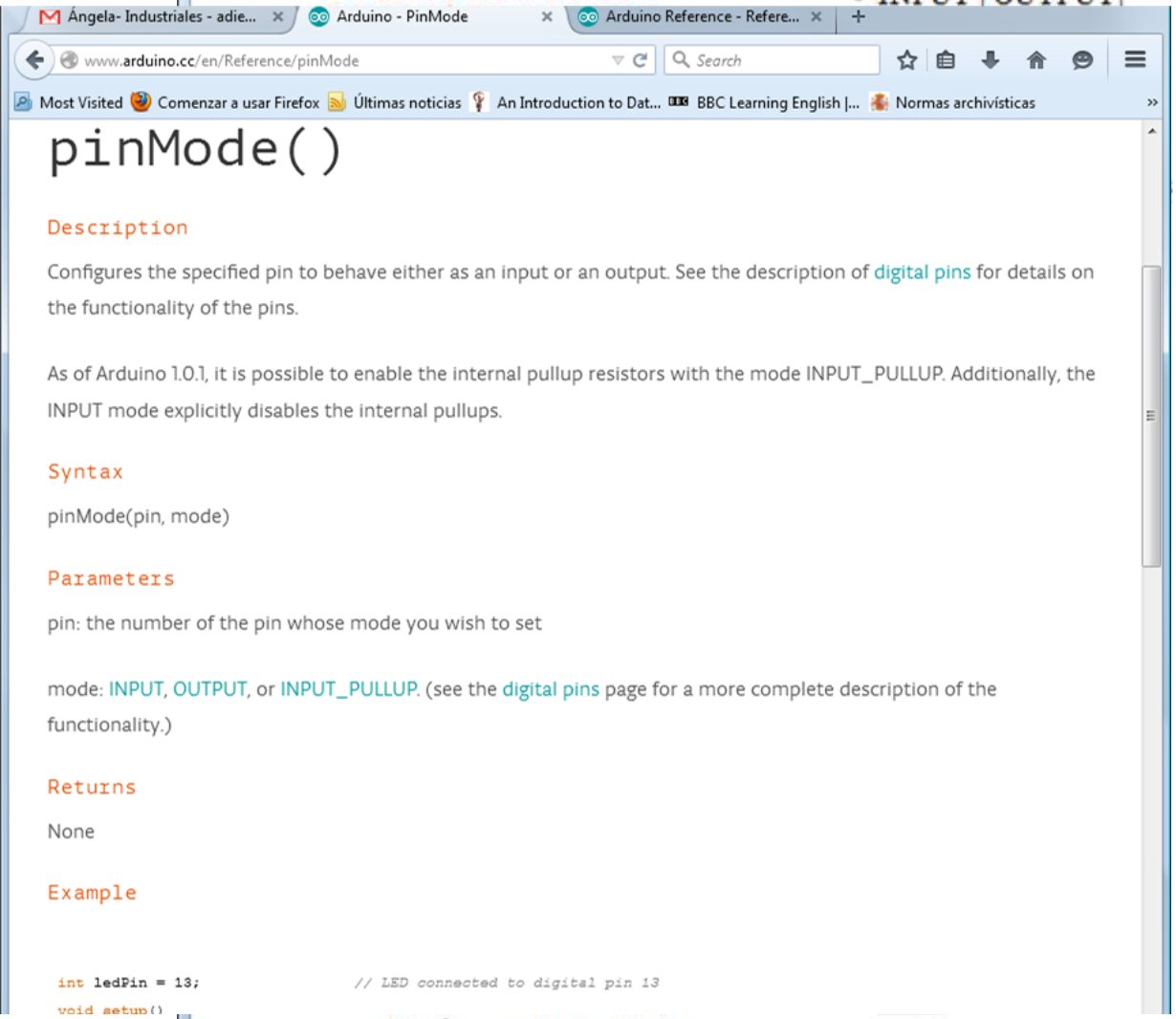
- tone()
- noTone()
- shiftOut()
- shiftIn()
- pulseIn()

Time

- millis()
- micros()
- delay()
- delayMicroseconds()

Math

- min()
- max()
- abs()
- constrain()

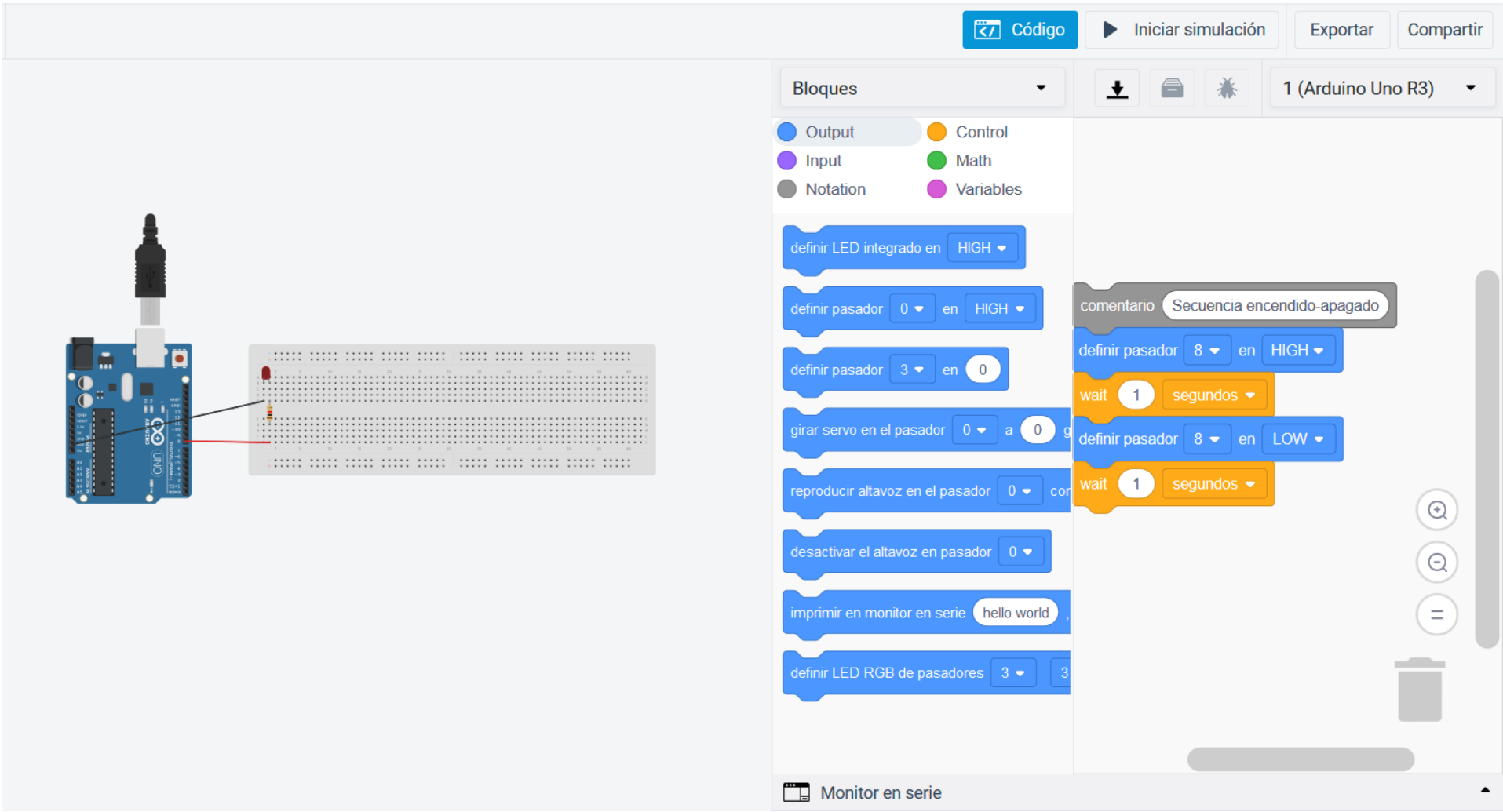


- = (assignment operator)
- + (addition)
- - (subtraction)

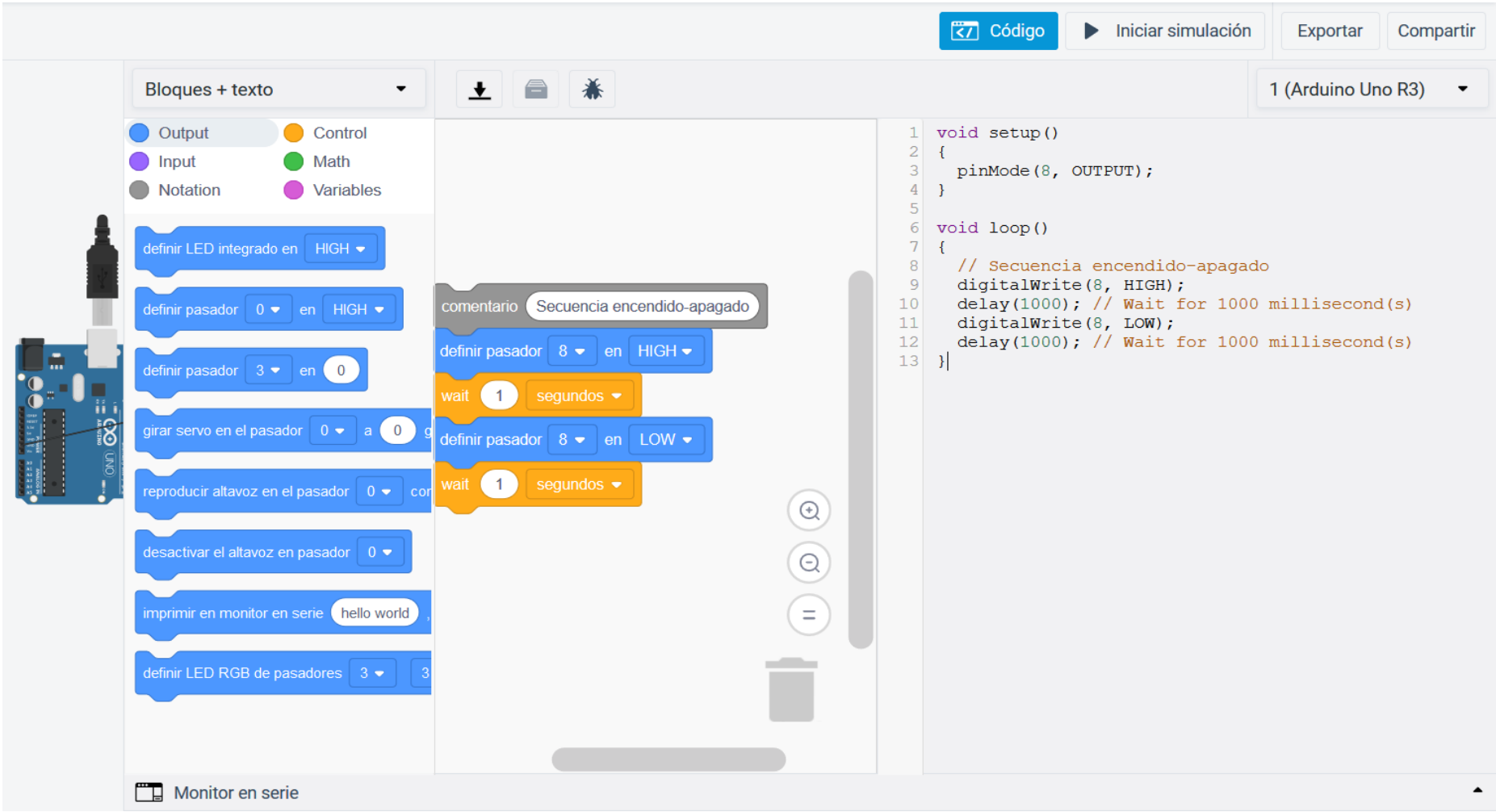
Conversion

- char()

Bloques



Bloques / Texto



Referencias

- Web oficial de Arduino. http://www.arduino.cc/ [On line]. Consultado [Abril 2015]
- Referencia Software. Web oficial Arduino < http://www.arduino.cc/en/Reference/HomePage> Consultado [Abril 2015]
- Referencia Hardware Arduino UNO. Web oficial Arduino < http://www.arduino.cc/en/Main/ArduinoBoardUno> Consultado [Abril 2015]
- Imagen con la estructura de arduino UNO. http://i.stack.imgur.com/wKz2l.png Consultado [Abril 2015]
- AUTODESK: TINKERCAD- Circuits. Entorno de desarrollo online de simulación de Arduino/Electronica/3D. (Software gratuito) Web que require inscripción. https://www.tinkercad.com, [Diciembre 2018]
- Blockly: Google for education. A JavaScript library for building visual programming editors. https://developers.google.com/blockly/ [Diciembre 2018]