

# Entorno de programación por código

Necesitarás el **entorno de desarrollo Arduino IDE** (IDE, Integrated development environment) (aquí <https://www.arduino.cc/en/Main/Software> para descargarlo)

OJO, existe **la versión online** del editor <https://create.arduino.cc/editor>.

Es una buena solución si trabajas en varios equipos y quieres que tus proyectos estén disponibles en cualquier equipo.

**ATENCIÓN** para usar la versión online, tienes que instalar en tu ordenador el software **AGENT**

<https://create.arduino.cc/getting-started/plugin/welcome>

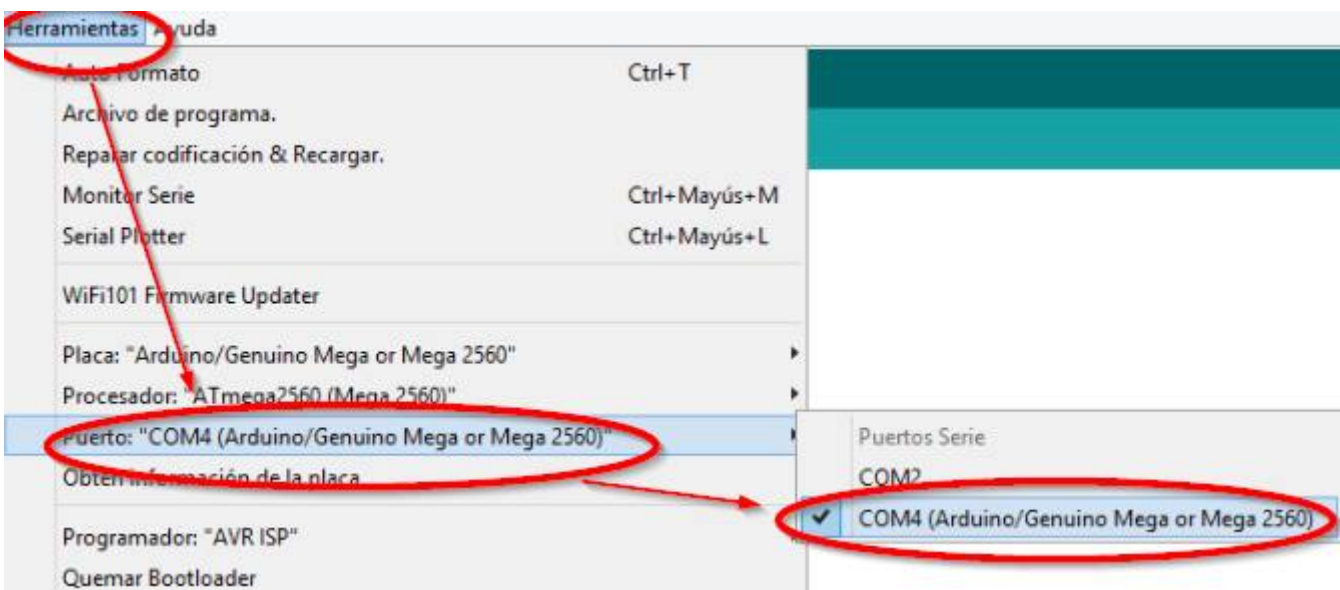
En Linux puede salir este mensaje "can't open device "/dev/ttyUSB0": Permission denied" donde 0 puede ser otro número, la solución [aquí](#)

Está constituido por un **editor de texto** para escribir el código, un **área de mensajes**, una barra de herramientas con botones para las funciones comunes, y una serie de menús.

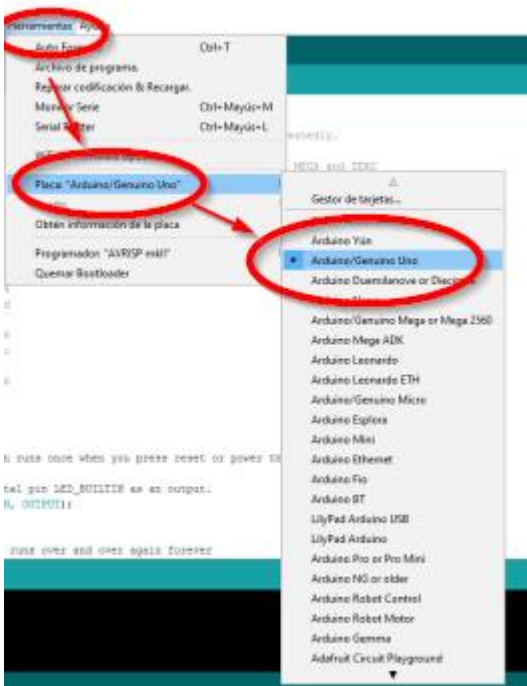
Arduino utiliza para escribir el código fuente o programa de aplicación lo que denomina "sketch" (programa). Estos programas son escritos en el editor de texto. Existe la posibilidad de cortar/pegar y buscar/remplazar texto.



Permite la conexión, por USB, con el hardware de Arduino para cargar los programas y comunicarse con ellos.



Y permite varias placas, tenemos que elegir la nuestra, en el KIT de CATEDU es Arduino UNO pero si tienes otro modelo este curso seguro que puede ser válido:



En el área de mensajes se muestra información mientras se cargan los programas y también muestra errores.

**Lo importante es cuando pinchemos en la flecha de subir nuestro programa, no salga ningún error, sino simplemente "Subido".**



## ¿Cómo se programa Arduino?

Las partes principales de un programa hecho en Arduino son: Bloque de inclusión de módulos y declaración de variables, bloque de configuración **void setup()** donde se indica el modo de funcionamiento de los pines (entrada y salida), comunicación serie, etc... y bloque de ejecución continua **void loop()**, en este bloque se incluyen las acciones que queremos que realice el programa. Se ejecutará línea a línea de forma secuencial y continua. Cuando llegue a la última instrucción incluida en la función **loop()** volverá a ejecutar la primera y continuará en un bucle infinito.

The image shows a screenshot of an Arduino IDE window with a teal title bar labeled 'Knob'. The code is for controlling a servo motor using a potentiometer. The code is annotated with four colored boxes and labels:

- Red box (top):** Contains the header comment: `// Controlling a servo position using a potentiometer (variable resistor)  
// by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>`. To its right is the label 'Descripción del programa'.
- Green box (middle):** Contains the include and variable declarations: `#include <Servo.h>  
  
Servo myservo; // create servo object to control a servo  
  
int potpin = 0; // analog pin used to connect the potentiometer  
int val; // variable to read the value from the analog pin`. To its right is the label 'Módulos y declaración de variables'.
- Red box (lower middle):** Contains the setup function: `void setup()  
{  
 myservo.attach(9); // attaches the servo on pin 9 to the servo object  
}`. To its right is the label 'Bloque de configuración'.
- Black box (bottom):** Contains the loop function: `void loop()  
{  
 val = analogRead(potpin);  
 val = map(val, 0, 1023, 0, 179);  
 myservo.write(val);  
 delay(15);  
}`. To its right is the label 'Bloque de ejecución continua'.

The full code in the IDE is as follows:

```
// Controlling a servo position using a potentiometer (variable resistor)
// by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>

#include <Servo.h>

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  val = analogRead(potpin);
  val = map(val, 0, 1023, 0, 179);
  myservo.write(val);
  delay(15);
}
```

## ¿Arduino tiene que estar continuamente conectada a un ordenador?

Sólo es necesario que esté conectado al ordenador mediante el USB para cargar los programas o para visualizar en tiempo de ejecución datos del programa mediante **la consola serie**. El ordenador proporciona la energía eléctrica suficiente para que funcionen los programas, pero una vez cargado el programa en la memoria del microcontrolador de Arduino se puede desconectar del USB y alimentar a la tarjeta mediante una fuente externa mediante el jack de alimentación con un margen de (5 a 20 Voltios). El programa cargado en Arduino queda grabado permanentemente aunque cese el suministro eléctrico.

Para una mayor información y manejo de la instalación del entorno de programación, lenguaje de programación y librerías se encuentra en la página web de la comunidad Arduino:

- [www.arduino.cc](http://www.arduino.cc) (portal en inglés, más actualizada).
- [www.arduino.es](http://www.arduino.es) (portal en español).

Revision #7

Created 1 February 2022 11:20:04 by Equipo CATEDU

Updated 23 December 2023 09:38:04 by Javier Quintana