

# 7. ArduinoBlocks

- [¿Qué es Arduino Blocks?](#)
- [Crear cuenta](#)
- [Cuentas alumnado](#)
- [Arduino blocks connector](#)
- [Empezando un proyecto](#)

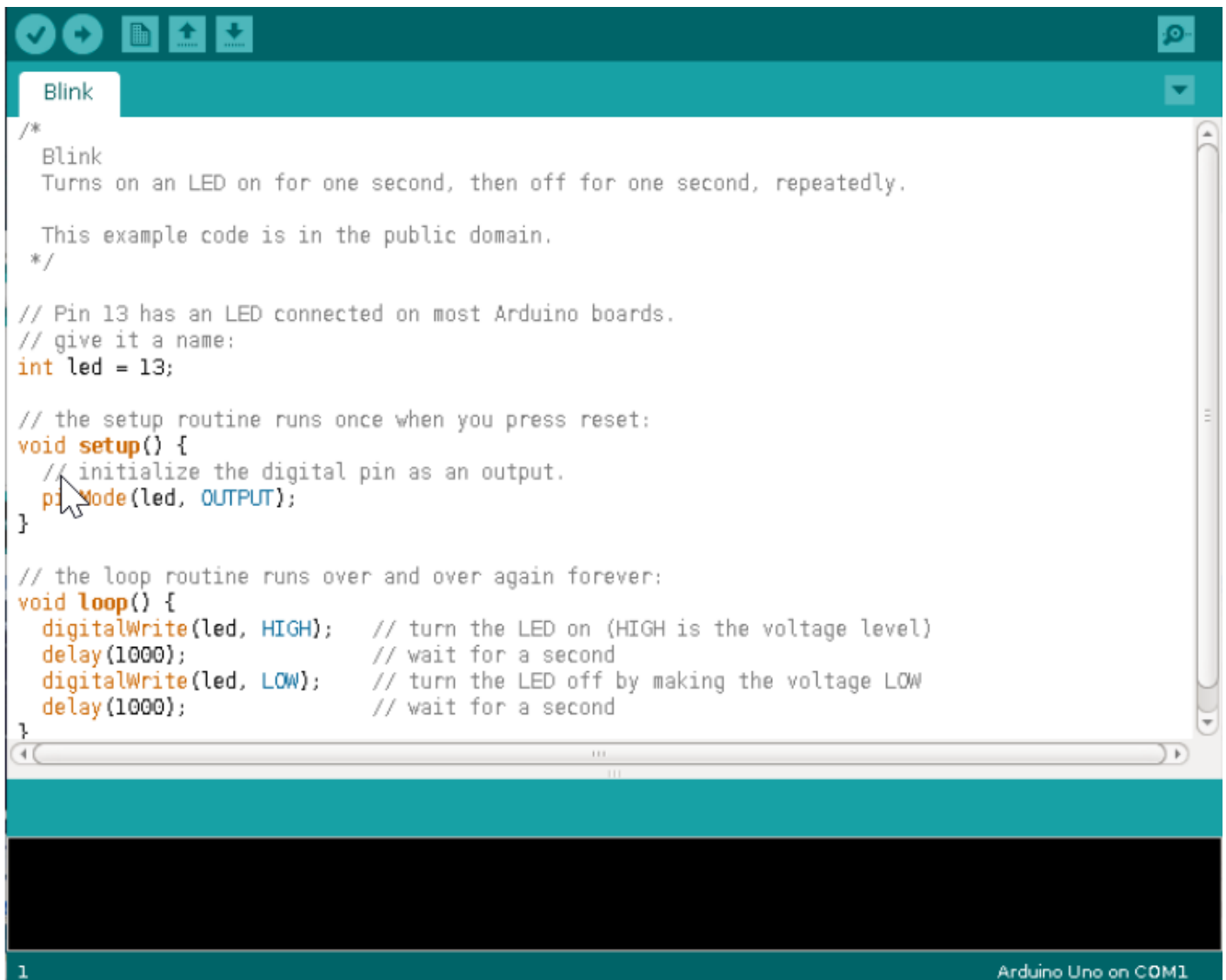
# ¿Qué es Arduino Blocks?

Esto no pretende ser un tutorial exhaustivo de ARDUINOBLOCKS, sino una guía rápida. ARDUINOBLOCKS es un programa que tiene muchas posibilidades. Si quieres saber más sobre ARDUINOBLOCKS tutoriales, ejemplos, foro.... te recomendamos

<http://arduinoblocks.didactronica.com/> o el libro [Arduino blocks - libros y tutoriales](#)

## ¿Por qué una programación con bloques?

Arduino se programa en lenguaje C++ (con algunas variaciones para simplificarlo). Para programar normalmente se utiliza el IDE ("Integrated Development Environment"/"Entorno de Desarrollo Integrado") de Arduino, que permite escribir el código, compilar el programa (crear el programa binario para el procesador Arduino) y grabarlo en la placa Arduino a través del puerto USB. El IDE de Arduino se puede descargar [desde la web oficial](#). Es totalmente libre (José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

The image shows the Arduino IDE interface. At the top, there's a toolbar with icons for checking, uploading, and downloading. Below it, a tab labeled 'Blink' is active. The main text area contains the C++ code for the Blink program. The code includes comments explaining the purpose and pin configuration, followed by the setup and loop functions. The setup function initializes pin 13 as an output, and the loop function turns the LED on and off with 1-second delays. A mouse cursor is pointing at the 'pinMode' function call. At the bottom, a status bar shows '1' on the left and 'Arduino Uno on COM1' on the right.

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

Sin embargo pensando en edades más tempranas se han desarrollado formas más sencillas e intuitivas de programar Arduino como son los **lenguajes de programación por bloques**. De todos estos lenguajes cabe destacar **ARDUINOBLOCKS**.

Gracias a este lenguaje visual podemos programar las placas Arduino sin necesidad de escribir ni una sola línea de código, de esta forma podemos empezar a realizar proyectos con Arduino de una forma muy rápida y a edades más tempranas. La única desventaja es que el lenguaje por código tiene todo el potencial que requiere la programación de un experto.

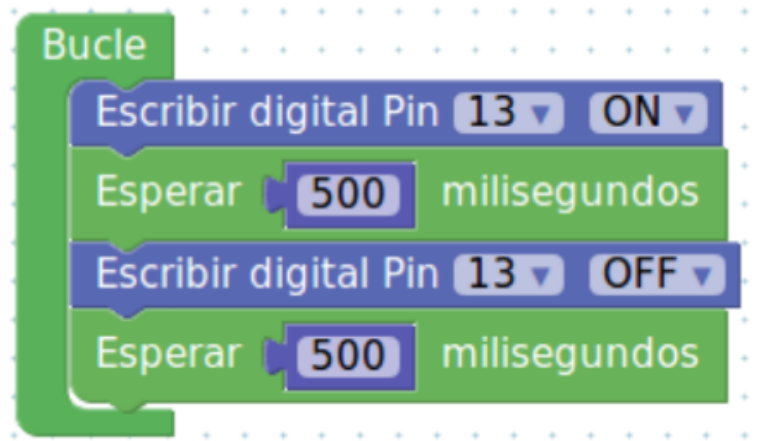
Mismo programa creado con el IDE de Arduino en C++ (imagen de la izquierda) y con Arduinoblocks (imagen de la derecha).

(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

```

void setup()
{
  pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}

```



*(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)*

Para trabajar con Arduinoblocks debemos ir a su página web <http://www.arduinoblocks.com/> desde cualquier navegador y para cualquier sistema operativo (Windows, Linux, Mac). *(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)*

## ArduinoBlocks



Arduinoblocks es un programa creado por el profesor Juanjo López. Gracias a su entorno gráfico facilita la programación de placas Arduino a todos los niveles. Esta herramienta permite programar a personas sin conocimientos previos de programación, pero su versatilidad y potencia es tan grande que expertos programadores también pueden utilizarlo. *(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)*



De [Juan José López Almendros](#) CC-BY-SA

La programación en **ArduinoBlocks** se realiza con bloques al estilo AppInventor o Scratch, se puede utilizar a partir de 8 años. No tenemos que escribir líneas de código y no nos permitirá unir bloques incompatibles evitando así posibles errores de sintaxis. La plataforma ArduinoBlocks genera, compila y sube el programa a la placa Arduino por medio de la conexión USB. Una vez subido el programa, la placa el Arduino no necesitará de la conexión al PC para funcionar pudiendo alimentarla con baterías o una fuente de alimentación para que funcione de forma autónoma.

ArduinoBlocks actualmente funciona con todos los navegadores de última generación: Mozilla Firefox, Google Chrome, Opera, Safari,...

*(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)*

Por otro lado, tal y como se describe en la [Wiki](#) de **Vitalinux**, ArduinoBlocks funciona perfectamente con todos los sistemas operativos, pudiendo ser fácilmente instalable en equipos individuales y a nivel de centro dentro del soporte de Vitalinux.

## Ayuda en Arduinoblocks

Además hay chat de Telegram con una comunidad de profesores y técnicos de la empresa que apoya Arduinoblocks donde puedes encontrar proyectos, enlaces interesantes y lo más importante: Puedes preguntar tus dudas o problemas

[https://t.me/innovadidactic\\_comunidad](https://t.me/innovadidactic_comunidad)



arduinoblocks + InnovaDidàctic

553 members

Los contenidos de esta página se han extraído del curso [Rover Marciano con Arduinoblocks e Internet de las Cosas \(IoT\)](#) de Aularagón.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



Financiado por  
la Unión Europea  
NextGenerationEU



Plan de Recuperación,  
Transformación  
y Resiliencia



GOBIERNO  
DE ESPAÑA

MINISTERIO  
DE EDUCACIÓN  
Y FORMACIÓN PROFESIONAL



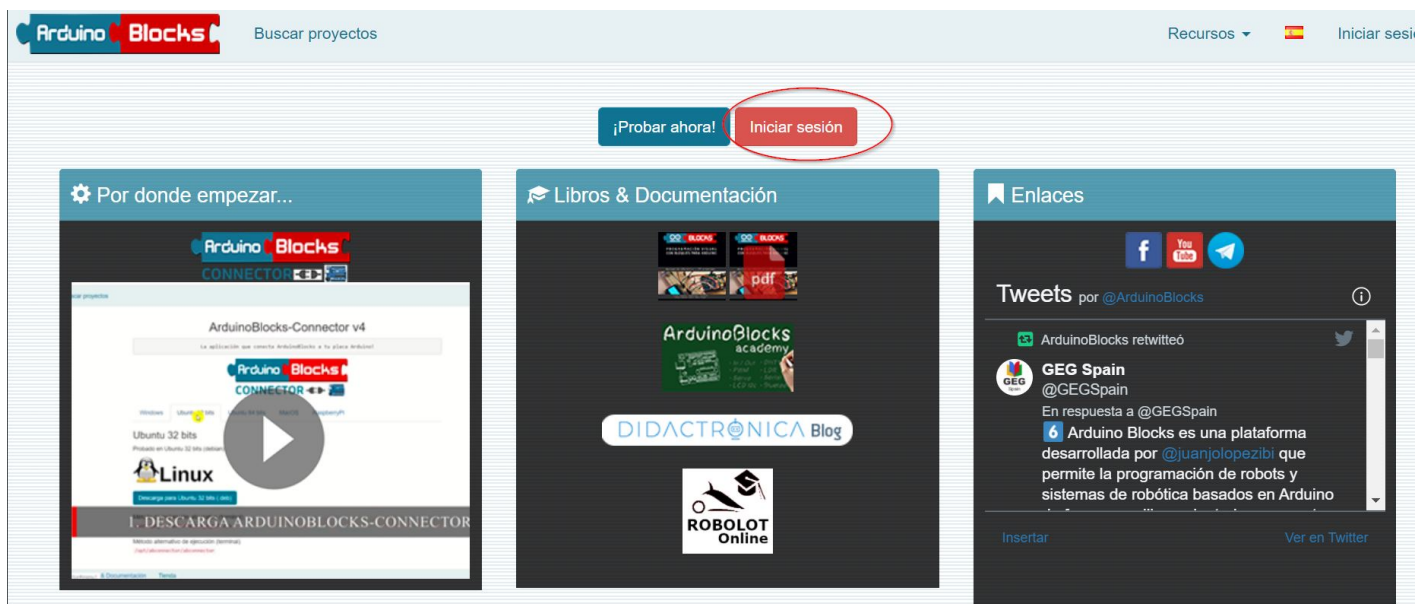
GOBIERNO  
DE ARAGON

# Crear cuenta

Registrándonos como usuarios de la plataforma ArduinoBlocks podemos aprovechar todas estas posibilidades:

- Guardar tus proyectos en la nube de ArduinoBlocks.
  - Añadir información al proyecto: descripción, componentes utilizados, imágenes, etc.
  - Añadir archivos adjuntos relacionados con el proyecto: esquemas, fotos, archivos para impresión 3D, aplicaciones, etc.
  - Compartir proyectos con el resto del mundo.
  - Importar proyectos compartidos por otros usuarios.
  - Valorar y comentar proyectos.
  - Programar directamente Arduino desde el propio navegador (con la aplicación: **ArduinoBlocks-Connector**).
  - Utilizar la consola serie desde el propio navegador.
- (José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

Entramos en <http://www.arduinoblocks.com/> e iniciamos sesión



Y rellenamos el formulario

## Nuevo usuario

\*\*\* Recommended **GMail** accounts (Review SPAM folder) \*\*\* (Hotmail, Msn,... may not work due to spam filters)

Correo electrónico

Confirmación de correo electrónico

Clave

Confirmación de clave

Nombre

Apellidos

País

Ciudad

Los contenidos de esta página se han extraído del curso [Rover Marciano con Arduinoblocks e Internet de las Cosas \(IoT\)](#) de Aularagón.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



# Cuentas alumnado

Tal y como dice el tutorial de Juanjo López :

*Permite a un usuario registrado con email, crear y administrar nuevas cuentas de usuario dentro de una organización, centro educativo o institución.*

<https://www.youtube.com/embed/zdzKX0NX60Y>

Si lo quieres en papel, te recomendamos el tutorial de Juanjo López son 12 diapositivas muy bien explicados [https://github.com/arduinoblocks/documentacion/blob/master/usuarios\\_gestionados.pdf](https://github.com/arduinoblocks/documentacion/blob/master/usuarios_gestionados.pdf)

El contar con esta funcionalidad es especialmente interesante en el caso de alumnado menor de 14 años, de forma que no sea necesario ceder ningún tipo de datos de ellos, ni recabar consentimientos parentales para la utilización de la plataforma. En general por criterios de protección y privacidad de datos, siempre es preferible trabajar con aplicaciones que solo requieren registro por parte del profesorado.

Los contenidos de esta página se han extraído del curso [Rover Marciano con Arduinoblocks e Internet de las Cosas \(IoT\)](#) de Aularagón.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



# Arduino blocks connector

**Espera !!!** Aún no conectes tu placa (Arduino, ESP32, NodeMCU, KeyStudio TDR STEAM...)

## PRIMER PASO Descargar e instalar ArduinoBlocks Connector

Para poder usar la herramienta **Arduinoblocks** tenemos que ejecutar antes **Arduinoblocks conector**. Lo descargamos de la misma página de ArduinoBlocks según el sistema operativo que usemos: Windows (**W7 E INFERIORES NO FUNCIONA**), Linux ....



Lo descargamos y lo **instalamos**.

En el caso de tener equipos Vitalinux, es fácilmente accesible e instalable desde la aplicación **Vitalinux Play** o si se desea una instalación masiva en el centro a través de su página de [soporte](#):



## SEGUNDO PASO: INSTALAR LOS DRIVERS


Si no hacemos estos pasos, cuando conectamos la placa, siempre sale en el COM1, le damos a subir y sale erro

En <http://www.arduinoblocks.com/web/site/abconnector5> tenemos abajo ARDUINO SERIAL DRIVERS

RECOMENDAMOS EL PRIMER ENLACE Y EL TERCERO

### AB-Connector v5

WindowsUbuntuMacOSChromebook

 **Windows**

v5.3 (Windows 64)

v5.3 - Descarga para Windows 64 (Installer .exe)

v5.3 - Descarga para Windows 64 [Portable .zip]

v5.1 (Windows 32/64)

[Descarga para Windows \(Installer .exe\)](#)

[Descarga para Windows \[Portable zip\]](#)

¡Desactiva el antivirus si la descarga falla!

**Arduino serial drivers:**

- Arduino + FTDI usb-serial converter
- Arduino + CH340G usb-serial converter
- Arduino + CP2102 usb-serial converter

**En el primero** el instalador está en este enlace

[https://cdn.sparkfun.com/assets/learn\\_tutorials/7/4/CDM21228\\_Setup.exe](https://cdn.sparkfun.com/assets/learn_tutorials/7/4/CDM21228_Setup.exe)

sparkfun

START SOMETHING

SHOP

LEARN

BLOG

CUSTOM KITS

PRODUCT MENU

find products, tutorials, etc...

Find a Retailer

Need Help?

0

LOG IN

REGISTER

TODAY'S DEALS

SPARK

FORUM

HOME / TUTORIALS / HOW TO INSTALL FTDI DRIVERS

## How to Install FTDI Drivers

CONTRIBUTORS PAUL SMITH

FAVORITE 12 SHARE

### Windows - Quick and Easy

**Note:** The screen shots in this tutorial are from Windows 7. The process should be very similar for other versions of Windows. For most late versions of windows, such as 8.1 through Windows 10, the hardware may work fine without any driver install. If you can't locate a COM port for your hardware, then the set of instructions below is the possible fix. The exception to this is Windows 8. For instructions on how to disable device driver signatures, [please visit this tutorial](https://learn.sparkfun.com/tutorials/disabling-driver-signature-on-windows-8).

**Note for Educators:** You will most likely need to obtain administrative privileges from your network or IT administrator in order to install these drivers.

1. By default, windows does not have FTDI drivers installed. If you plug in your FTDI, open the Arduino IDE, go to Tools -> Serial Ports, and see nothing, you need the drivers! Let's go get them!

2. Download a copy of the v2.12.28 FTDI VCP Driver Executable here:

WINDOWS FTDI VCP DRIVER EXECUTABLE - V2.12.28 (CDM21228.SETUP.EXE)

Otherwise, visit FTDI's VCP Drivers page for the latest download of the Windows FTDI Driver executable and clicking on the Window's "Available as a setup executable" link. Make sure to unzip the executable before proceeding in the next step.

Pages

Introduction

Meet the FT232RL

**Windows - Quick and Easy**

Windows - In Depth

Mac

Linux

Resources and Going Further

Comments 14

Single Page

Print

License

tutorials are CC BY-SA 4.0

**El segundo** sólo si quieres utilizar Arduinos no oficiales, de fabricantes chinos, que tiene el CH340g y hay que leerse la página, paciencia

**El tercero** es necesario el 2102 si utilizas el ESP32 el instalador esta en este enlace, es una carpeta comprimida, la descomprimes y está el ejecutable instalador

[https://www.silabs.com/documents/public/software/CP210x\\_Windows\\_Drivers.zip](https://www.silabs.com/documents/public/software/CP210x_Windows_Drivers.zip)

## Download and Install VCP Drivers

Downloads for Windows, Macintosh, Linux and Android below.

\*Note: The Linux 3.x.x and 4.x.x version of the driver is maintained in the current Linux 3.x.x and 4.x.x tree at [www.kernel.org](http://www.kernel.org).

## Software Downloads

Software (11)

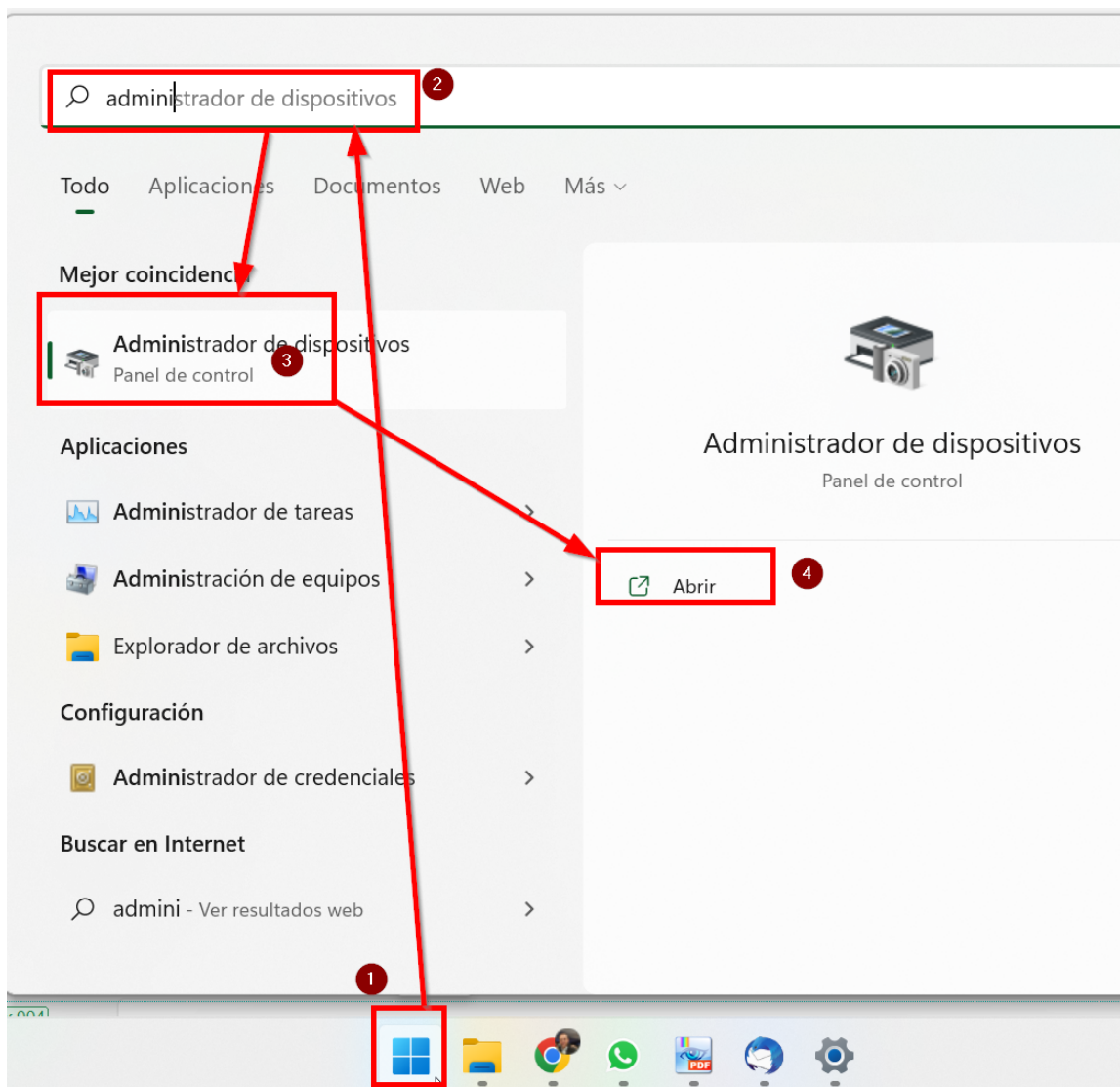
### Software · 11

CP210x Universal Windows Driver	v11.3.0 6/24/2023
CP210x VCP Mac OSX Driver	v6.0.2 10/26/2021
CP210x VCP Windows	v6.7 9/3/2020
CP210x Windows Drivers	v6.7.6 9/3/2020
CP210x Windows Drivers with Serial Enumerator	v6.7.6 9/3/2020

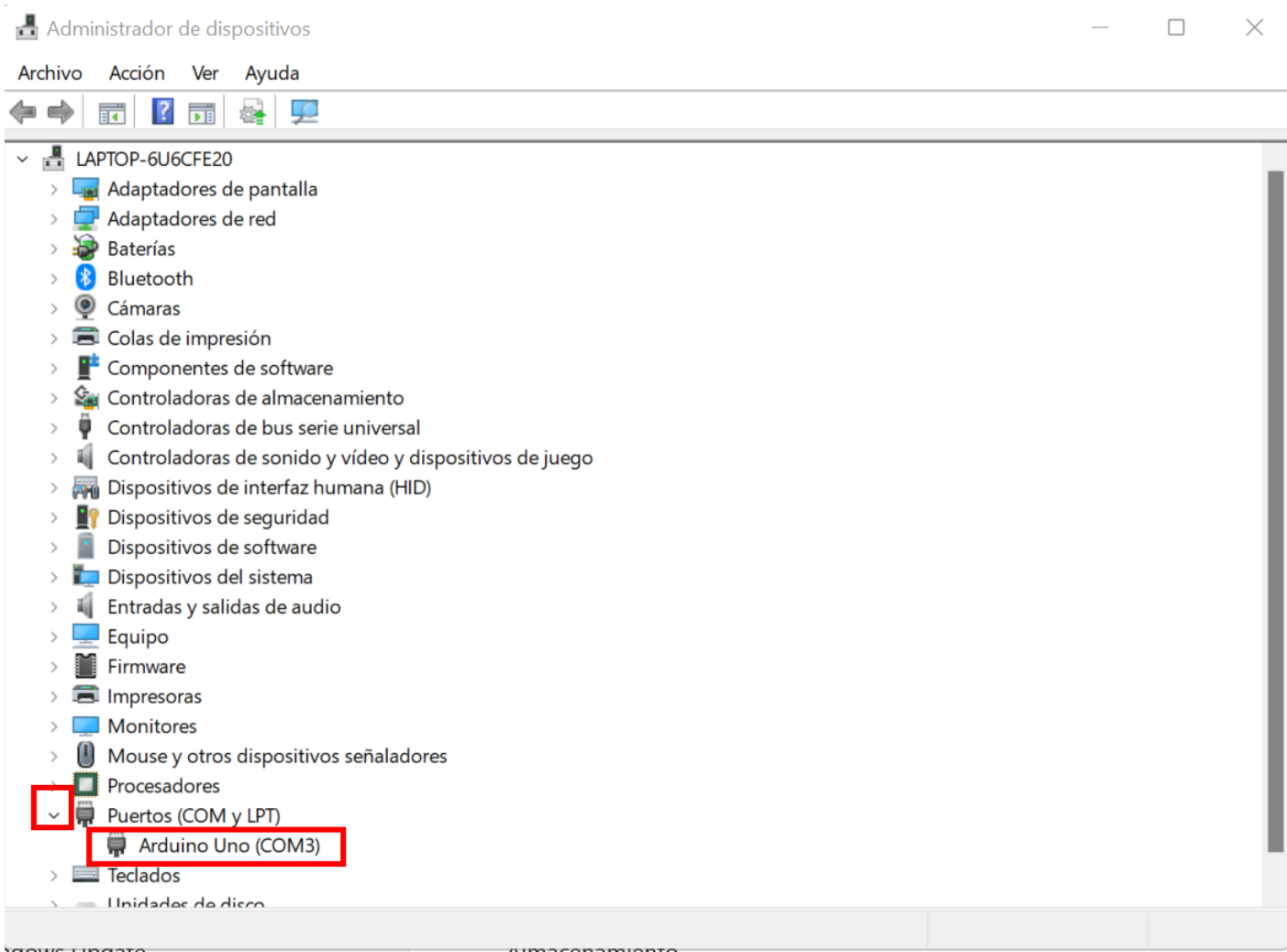
[Show 6 more Software](#)

## COMPROBAR QUE DETECTA LA PLACA

Ahora **conectamos la placa** (Arduino, NodeMCU, KeyStudio TDR STEAM...) a nuestro ordenador, y observamos si lo detecta, en Windows entramos en Administrador de dispositivos:



Y vemos que en los puertos COM se ha detectado correctamente la placa:



En el caso de que no aparezca, es que no se han instalado correctamente los *drivers* de Arduino. Entonces vamos a la página oficial de Arduino y descargamos el programa **ARDUINO IDE** : <https://www.arduino.cc/en/software> y lo instalamos. Al instalar este programa se instalan los drivers en nuestro ordenador. No hace falta ejecutarlo.

En el caso de equipos con sistema operativo Linux (como Vitalinux) el puerto serie tiene la forma **/dev/XXXX**

## YA PUEDES EJECUTAR ARDUINOBLOCKS CONNECTOR

Ahora buscamos el programa ArduinoBlocks connector que hemos descargado e instalado en el primer paso y lo **ejecutamos**.



```
12:36:51> AB-Connector v5.3
12:36:51> Path: C:\Program Files\abconnector\bin
12:36:51> Port: 9987
12:36:51> Arduino-CLI: 0.35.3
12:36:51> ['arduino:avr', 'esp32:esp32', 'esp8266:esp8266']
12:36:51> Checking/updating libs...
12:36:51> Libraries version: 56
```

**ATENCIÓN** No podemos cerrar la ventana mientras utilizamos *Arduinoblocks*, la minimizamos simplemente.

En caso contrario, Arduinoblocks no se puede comunicar con nuestra placa Arduino, NodeMCU, KeyStudio, etc ....

## YA PUEDES EJECUTAR ARDUINOBLOCKS

Entramos en la web ARDUINOBLOCKS <http://www.arduinoblocks.com/> nos logueamos e iniciamos un proyecto, Vemos que en el editor que aparece ya los puertos COM (si no te aparece, dale a la rueda actualizar)

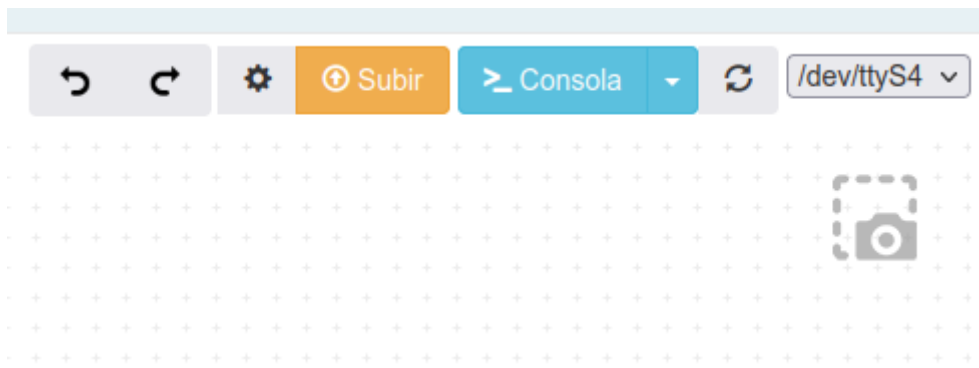
Aparecen varios COM, **elegir el último que tiene que coincidir con el que has visto en el segundo paso**, no necesariamente es el COM más alto.

Si se queda una ruleta de espera demasiado tiempo, entonces, actualizar la página o darle a actualizar el botón 1 de la figura :



Una vez elegido el COM ya puedes darle al botón amarillo **SUBIR** cuando has realizado tu proyecto pero antes de subir, por si acaso dale a **guardar** el proyecto que has realizado.

En el caso de equipos con Linux veremos algo así:



### ¿Tengo que hacer los cuatro pasos cada vez?

No, sólo la primera vez para asegurar los drivers del Arduino, las siguientes veces que te conectes lo único que tienes que hacer es el tercer y cuarto paso

**IMPORTANTE: TENER EL SOFTWARE ARDUINOBLOCKS ACTUALIZADO** para que funcionen los nuevos bloques que se incorporan en Arudinoblocks

Los contenidos de esta página se han extraído del curso [Rover Marciano con Arduinoblocks e Internet de las Cosas \(IoT\)](#) de Aularagón.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU

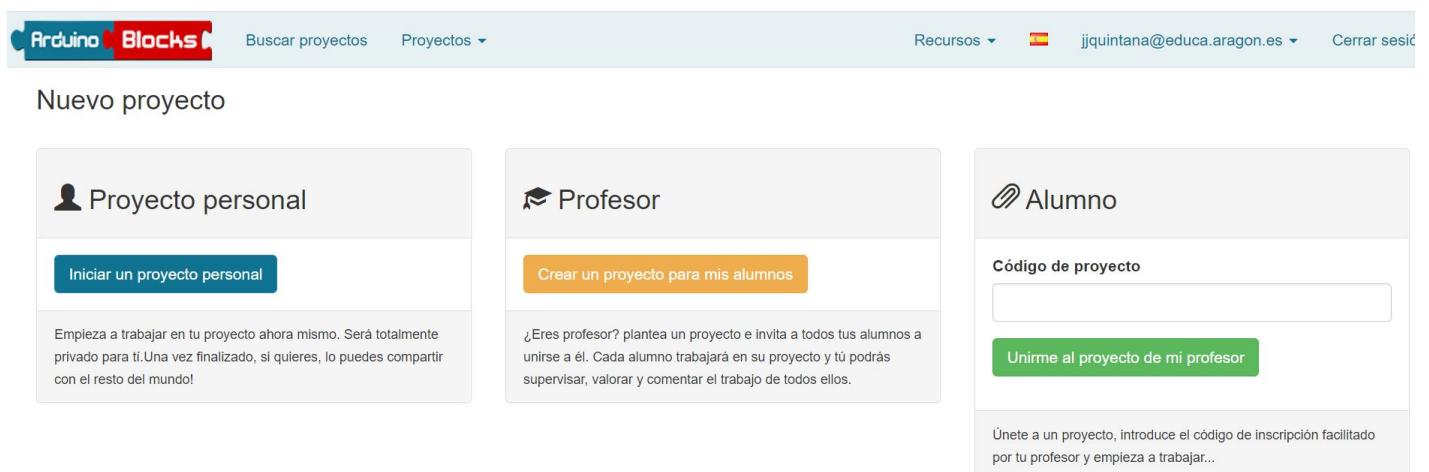


# Empezando un proyecto

Entramos en Proyectos y podemos ver nuestros proyectos creados como también empezar uno.



Y nos aparece tres opciones :



En esta ventana podremos elegir qué tipo de proyecto vamos a realizar:

- **Proyecto Personal:** Iniciar un nuevo proyecto que sólo será accesible para el usuario. Posteriormente se puede compartir al resto de la comunidad si se desea.
- **Proyecto Profesor:** Iniciar un proyecto como profesor. De esta forma no se inicia un proyecto como tal, sino que se especifican los datos del proyecto y se genera un código para que los alumnos se puedan suscribir al proyecto. El profesor podrá supervisar y valorar los proyectos de sus alumnos.
- **Alumno:** De esta forma nos unimos a un proyecto planteado por el profesor. Nosotros realizaremos el proyecto como si de un proyecto personal se tratara, pero el profesor podrá supervisar y valorar nuestro trabajo.

Adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

Lo primero que tenemos que elegir es para qué tipo de placa se hace el proyecto.

- En el caso de que estés con el kit de CATEDU **Rover marciano con Arduinoblocks** el tipo de proyecto es para **ESP8266 / NodeMCU**
- En el caso de que estés con el kit de CATEDU **Arduino con Arduinoblocks** el tipo de proyecto es para **Arduino UNO**
- En el caso de que estés con el kit de CATEDU **ArduinoBlocks en el aula** tienes dos opciones totalmente válidas:
  - **ArduinoUno**
  - **ArduinoUno + Imagina TdR STEAM**
- En el caso de que estés con el kit de CATEDU **ESP32 en el aula** tienes dos opciones totalmente válidas:
  - **ESP32 STEAMakers**
  - **ESP32 STEAMakers + Imagina TdR STEAM**

## Nuevo proyecto personal

Tipo de proyecto	<input type="text"/>
Nombre	Arduino Uno
Descripción	Arduino Nano / ATmega328 Arduino Nano / ATmega328 (new bootloader) Arduino Mega / 2560 Arduino Leonardo Arduino UNO + Imagina TdR STEAM 3dBot / Imagina-Arduino Keyestudio EasyPlug Keyestudio KeyBot Otto DIY / Nano Otto DIY / Nano (new bootloader) ESP8266 / NodeMCU v2 ESP8266 / WeMos D1 ESP32 / WROOM ESP32 STEAMakers ESP32 STEAMakers + Imagina TdR STEAM ESP32 STEAMakers + 3dBot
Componentes	

**ATENCIÓN** luego **NO** se puede cambiar. Es decir, un proyecto realizado para un tipo de placa, no se puede cambiar a otro tipo de placa (la razón es simple: las instrucciones cambian)

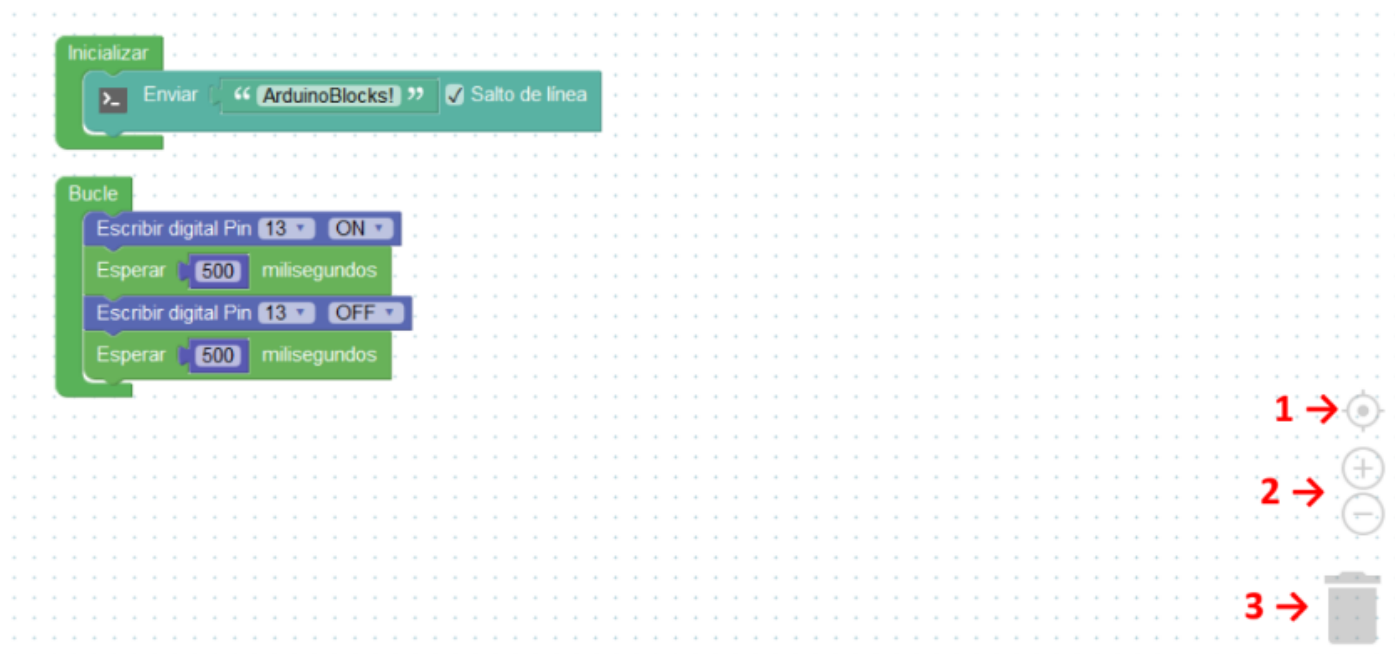
Luego el nombre y el resto de campos es optativo pero **importante y buena costumbre** rellenarlos, sobre todo si el proyecto lo **compartimos**:

- Descripción
- Componentes
- Comentarios

# Área de programación del proyecto






Este es el área sobre el que se trabaja en Arduinoblocks. En esta área arrastraremos y colocaremos los bloques que vamos a utilizar para crear nuestro programa.

En el área de trabajo hay un Zoom (2) para ampliar o reducir la imagen, un icono para centrar (1) y un icono donde podremos borrar los bloques que no utilicemos (3).



Adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

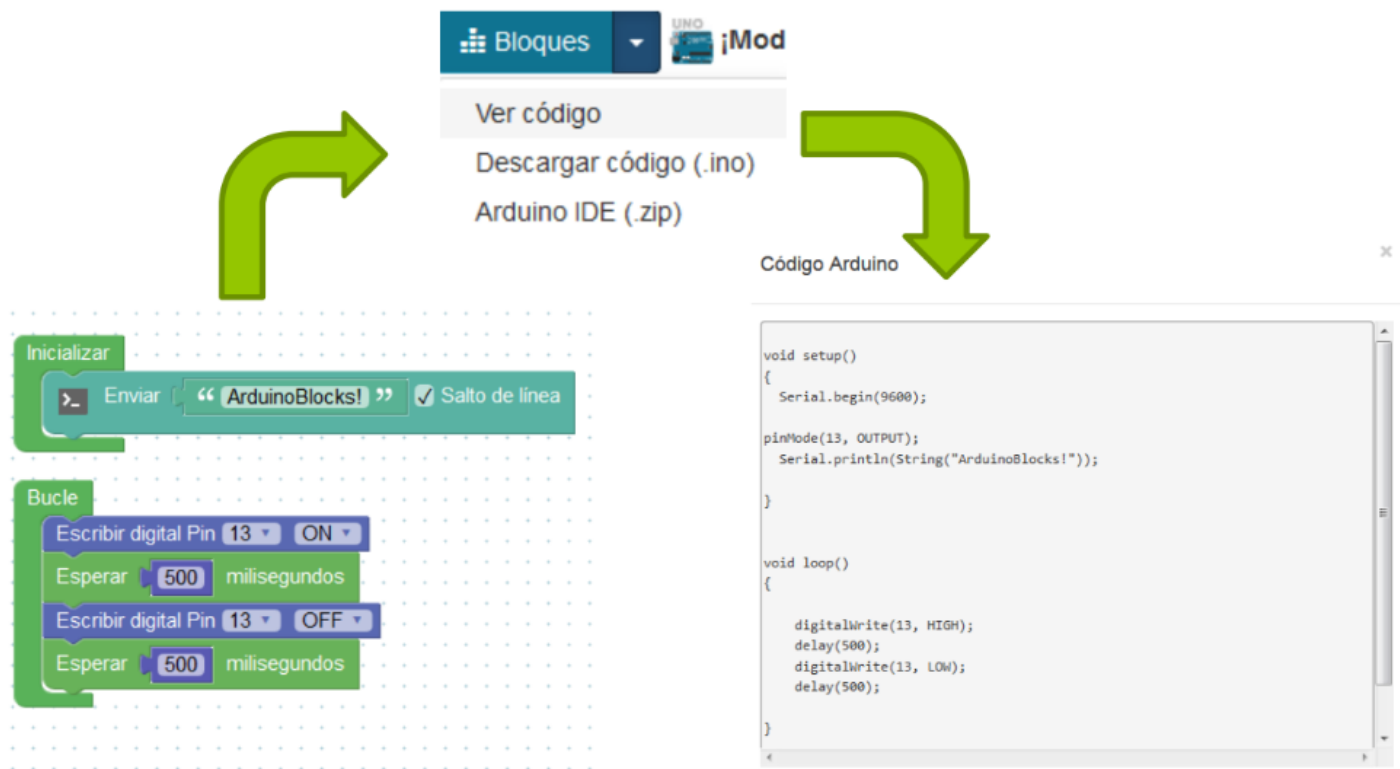
Las principales secciones del área de programación son las siguientes :

Herramientas	Área de programa	Opciones
<div> <div>Lógica</div> <div>Control</div> <div>Matemáticas</div> <div>Texto</div> <div>Variables</div> <div>Listas</div> <div>Funciones</div> <div>Entrada/Salida</div> <div>Tiempo</div> <div>Puerto serie</div> <div>Bluetooth</div> <div>Sensores</div> <div>Actuadores</div> <div>Pantalla LCD</div> <div>Memoria</div> <div>Motor</div> <div>Motor-Shield</div> <div>Keypad</div> <div>Reloj RTC</div> <div>GPS</div> </div>	<p>Bloque de iniciación</p>  <p>Bloque de bucle del programa principal</p> 	<p>Subir el programa a la placa Arduino conectada:</p>  <p>Puerto de conexión de la placa Arduino:</p>  <p>Mostrar la consola serie:</p> 

Adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

## Ver el código

**ArduinoBlocks** genera el código de Arduino a partir de los bloques. El programa se puede compilar y subir directamente a la placa Arduino gracias a la aplicación **ArduinoBlocks-Connector**, sin embargo si deseamos ver o descargar el código podemos realizarlo desde el área de bloques.



Adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

Siempre, desde un **lenguaje de programación en bloques** podemos obtener su equivalente a **Código de Arduino IDE** (de hecho es lo que hacen los programas), y luego con las funciones de **Código de Arduino IDE** el software lo pasa a **lenguaje máquina** que es la que se graba el Arduino, **pero no al revés** es decir, no existen programas que dado un código máquina o código Arduino IDE lo pasen a bloques gráficos, (igual que no hay programas que lean el código máquina que hay grabado en un Arduino y lo pasen a código Arduino IDE). Esto no es del todo 100% verdadero pues la Ingeniería inversa en informática trata pues de eso: obtener la fuente aunque sea parcial, pues si obtienes el código legible, puedes alterar lo que quieras.

Cuando compras un programa comercial, te dan el lenguaje máquina ilegible. Mientras que los programas de software libre se publica el código fuente legible para que todo el mundo pueda mejorarlo.

Por ejemplo en la siguiente figura, el programa gráfico **mBlock** que se utiliza en Arduino, mBot, etc... pasa sus instrucciones de lenguaje de programación de bloques parecido a Scratch a lenguaje de **Código de Arduino IDE** y Arduino IDE graba instrucciones binarias de **lenguaje máquina** al Arduino.



**¡¡A disfrutar!!**

Consejo: Te recomendamos visitar el canal de Youtube de Arduinoblocks

<https://www.youtube.com/c/ArduinoBlocks>

Los contenidos de esta página se han extraído del curso [Rover Marciano con Arduinoblocks e Internet de las Cosas \(IoT\)](#) de Aularagón.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU

