STEAMAKERBLOCKS

- ¿Qué es Steamakerblocks?
- Crear cuenta en STEAMAKERBLOCKS
- Cuentas alumnos
- Connector
- Empezando un proyecto

Copyright 2026

¿Qué es Steamakerblocks?

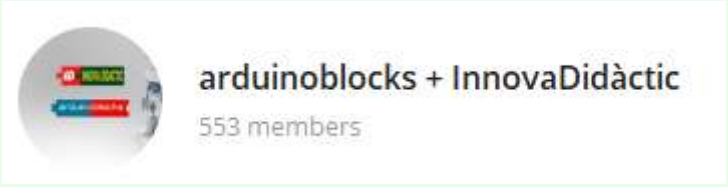
Esto no pretende ser un tutorial exhaustivo de STEAMAKERSBLOCKS, sino una guía rápida. STEAMAKERSBLOCKS es un programa que tiene muchas posibilidades. Si quieres saber más sobre ARDUINOBLOCKS tutoriales, ejemplos, foro.... te recomendamos

http://arduinoblocks.didactronica.com/ o el libro Arduino blocks - libros y tutoriales

IMPORTENTE SABERLO: TENEMOS UN CHAT DE Ayuda en STEMAKERSBLOCKS

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

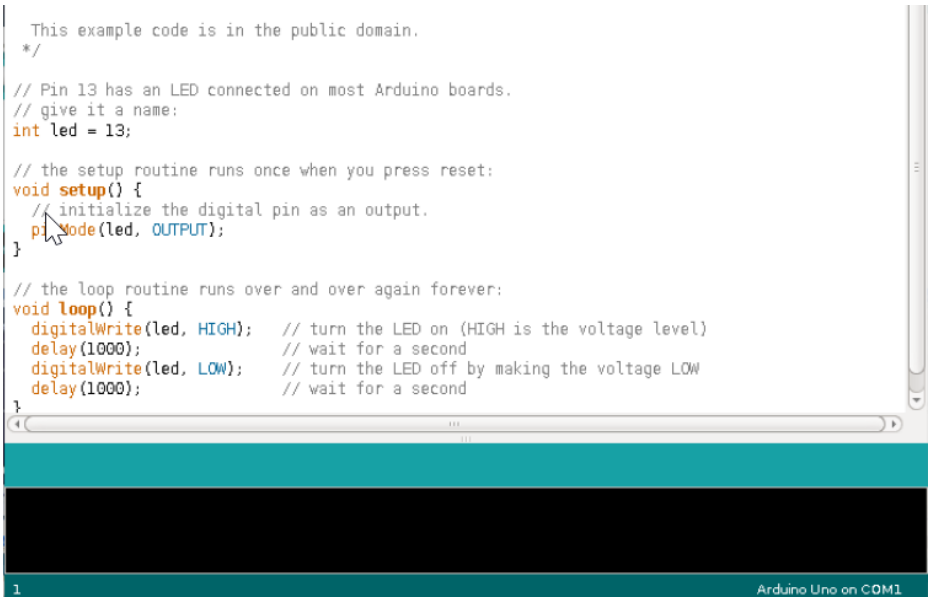
https://t.me/innovadidactic_comunidad



¿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)*

Copyright 2026



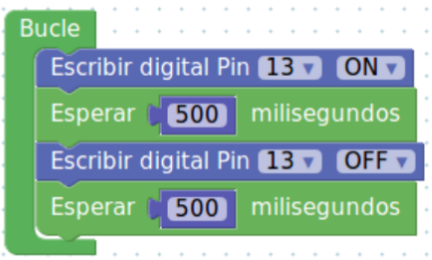
(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

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 **STEAMAKERSBLOCKS**.

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)*

STEAMAKERSBLOCKS (antes Arduinoblocks)

steamakersblocks antes ArduinoBlocks

STEMAKERSBLOCKS 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 **STEMAKERSBLOCKS** 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 **STEMAKERSBLOCKS** 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.

STEMAKERSBLOCKS actualmente funciona con todos los navegadores de última generación: Mozilla Firefox, Google Chrome, Opera, Safari,... *(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)*

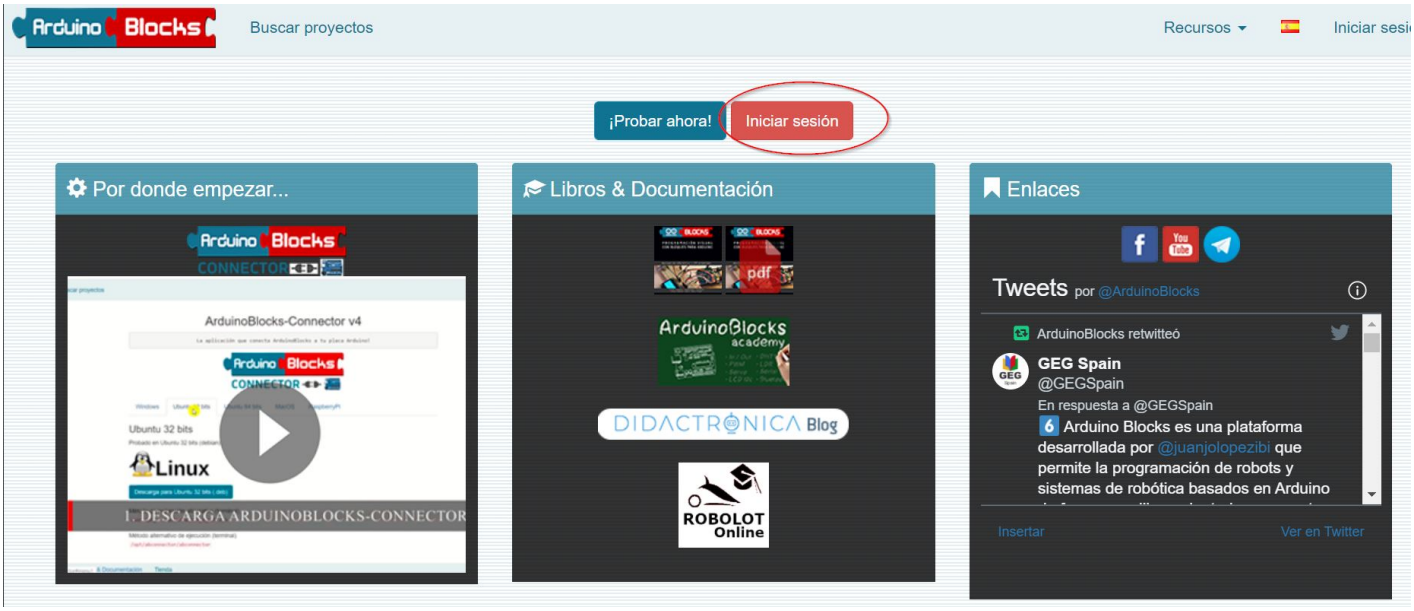
Por otro lado, tal y como se describe en la Wiki de **Vitalinux**, **STEMAKERSBLOCKS** funciona perfectamente con todos los sistemas operativos, pudiendo ser fácilmente instalable en equipos individuales y a nivel de centro dentro del soporte de **Vitalinux**.

Crear cuenta en STEAMAKERBLOCKS

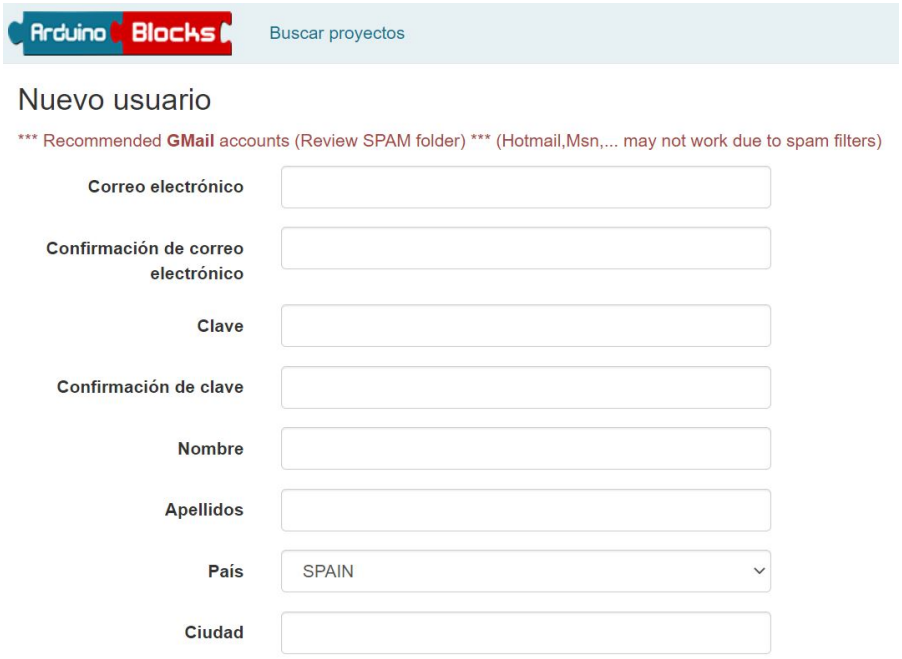
Registrándonos como usuarios de la plataforma **Steamakerblocks** podemos aprovechar todas estas posibilidades:

- Guardar tus proyectos en la nube de **Steamakerblocks** .
- 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: **Steamakerblocks -Connector**).
- Utilizar la consola serie desde el propio navegador.

Entramos en https://www.steamakersblocks.com/ e iniciamos sesión



Y rellenamos el formulario



Autor José Andrés Echevarría @cantabRobots CC-BY-NC-SA

Cuentas alumnos

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.*

- ❤Permite crear usuarios alumnos sin necesidad de ceder ningún tipo de datos
- Puedes pasar proyectos a los alumnos, vacíos o empezados
- Tú puedes ver los proyectos de los alumnos y ponerles comentarios

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

- Si lo quieres en papel, te recomendamos el tutorial de Juanjo López son 12 diapositivas muy bien explicados
  - Repositorio Github
  - Repositorio en steamakerbloks.com
- Prueba con Catedu:
  - https://www.steamakersblocks.com/
  - usuario alumnox.catedu donde x de 1 a 20
  - contraseña [donde esta catedu en minúsculas]

Copyright 2026

Connector

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

PRIMER PASO Descargar e instalar Connector

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

Entra en https://www.steamakersblocks.com/ y en Recursos, tienes la web para descargar este programa:



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:

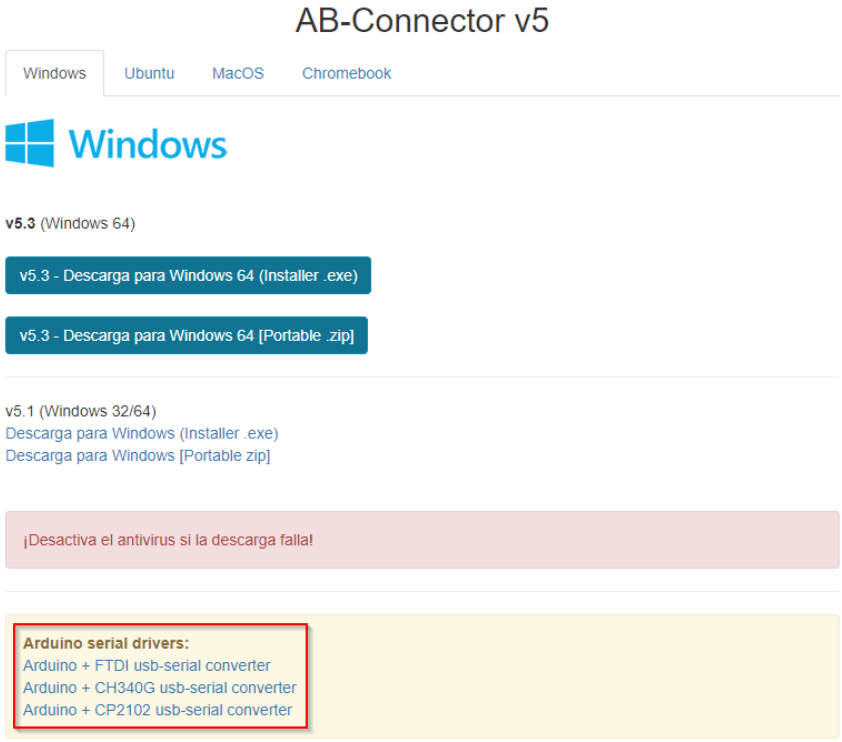


SEGUNDO PASO: INSTALAR LOS DRIVERS

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

En en elnace de connector https://www.steamakersblocks.com/web/site/abconnector tenemos abajo ARDUINO SERIAL DRIVERS

> RECOMENDAMOS EL PRIMER ENLACE Y EL TERCERO



En el primero el instalador está en este enlace

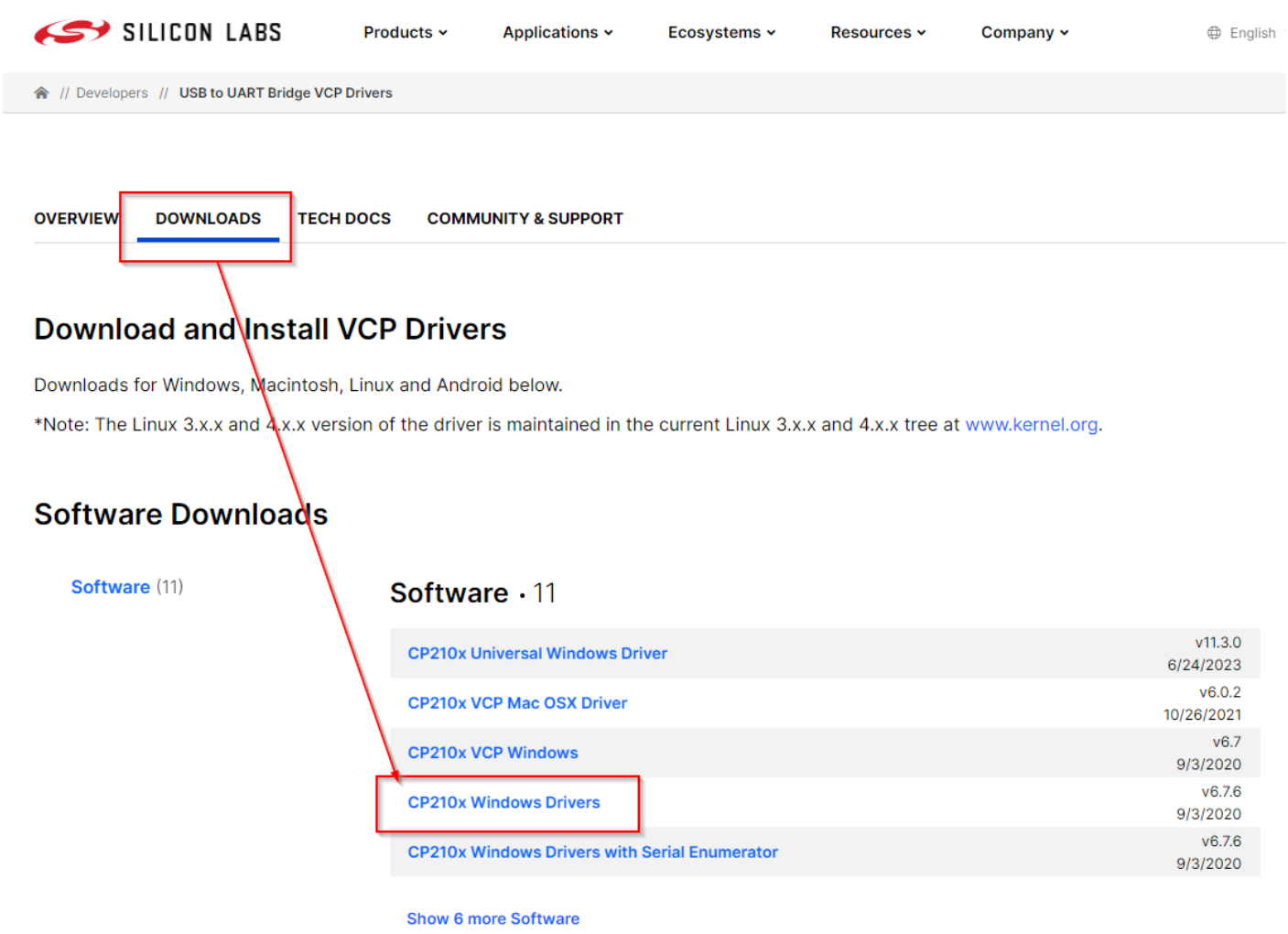
https://cdn.sparkfun.com/assets/learn_tutorials/7/4/CDM21228_Setup.exe



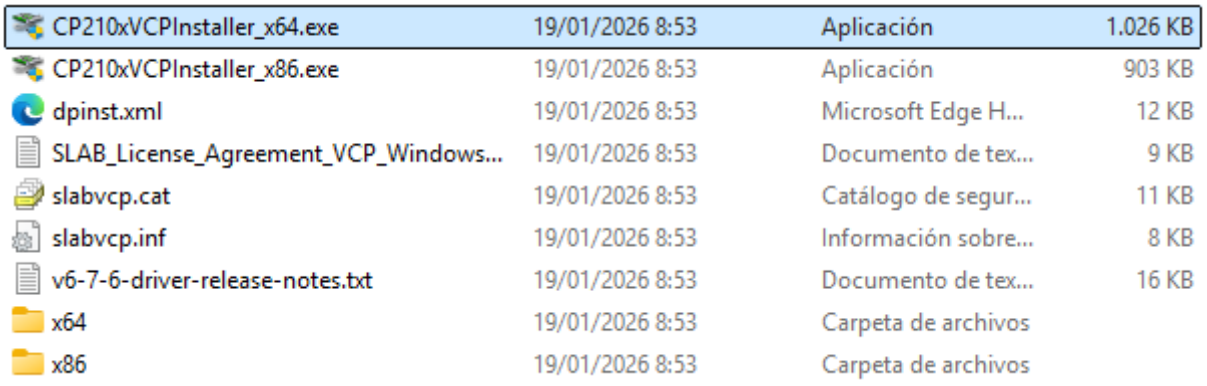
El segundo sólo si quieres utilizar Arduinos no oficiales, de fabricantes chinos, que tiene el CH340g y hay que leerse la página, paciencia

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



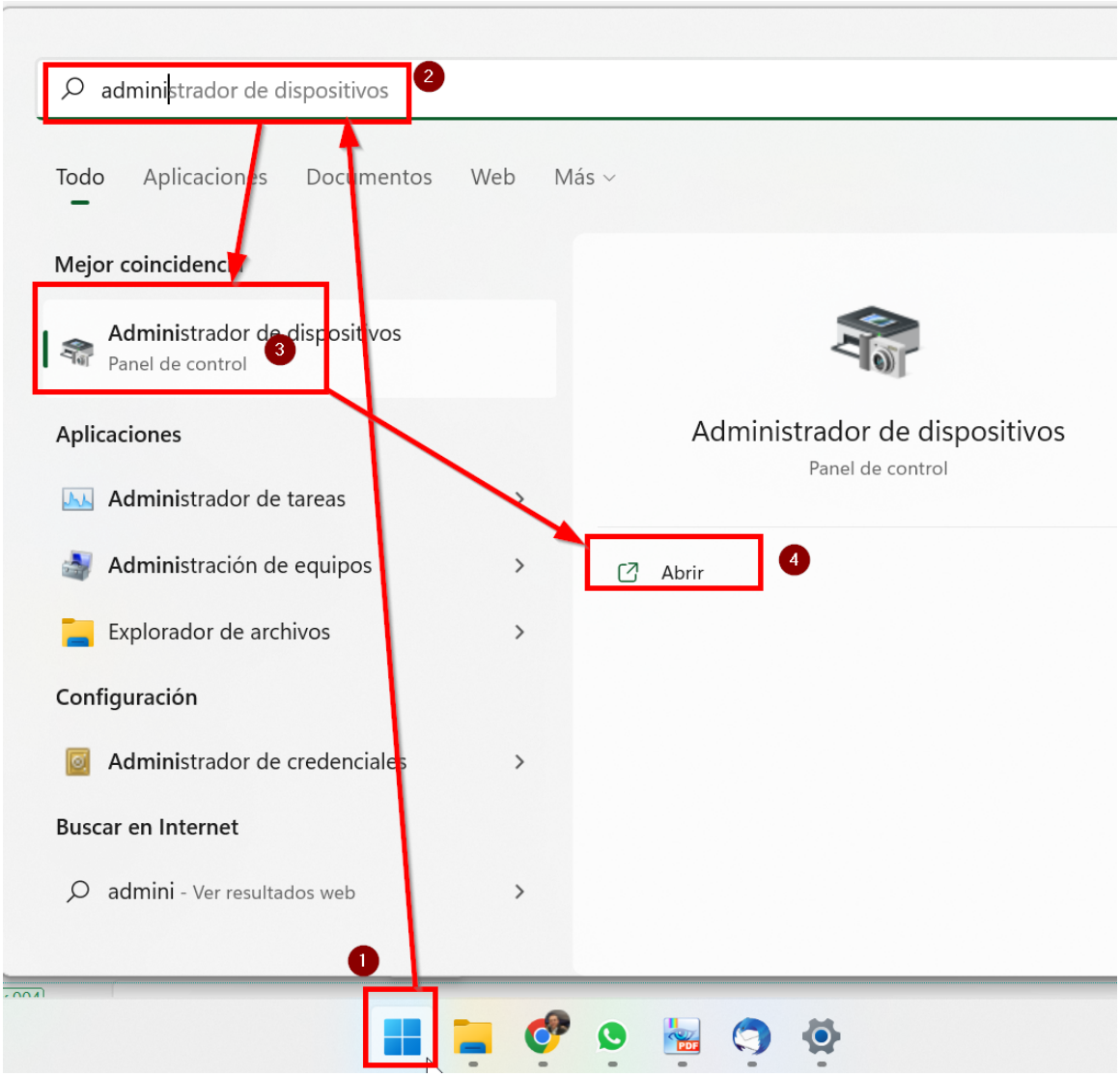
concretamente hay que ejecutar este (al menos que el equipo sea muy viejo de 32bits)



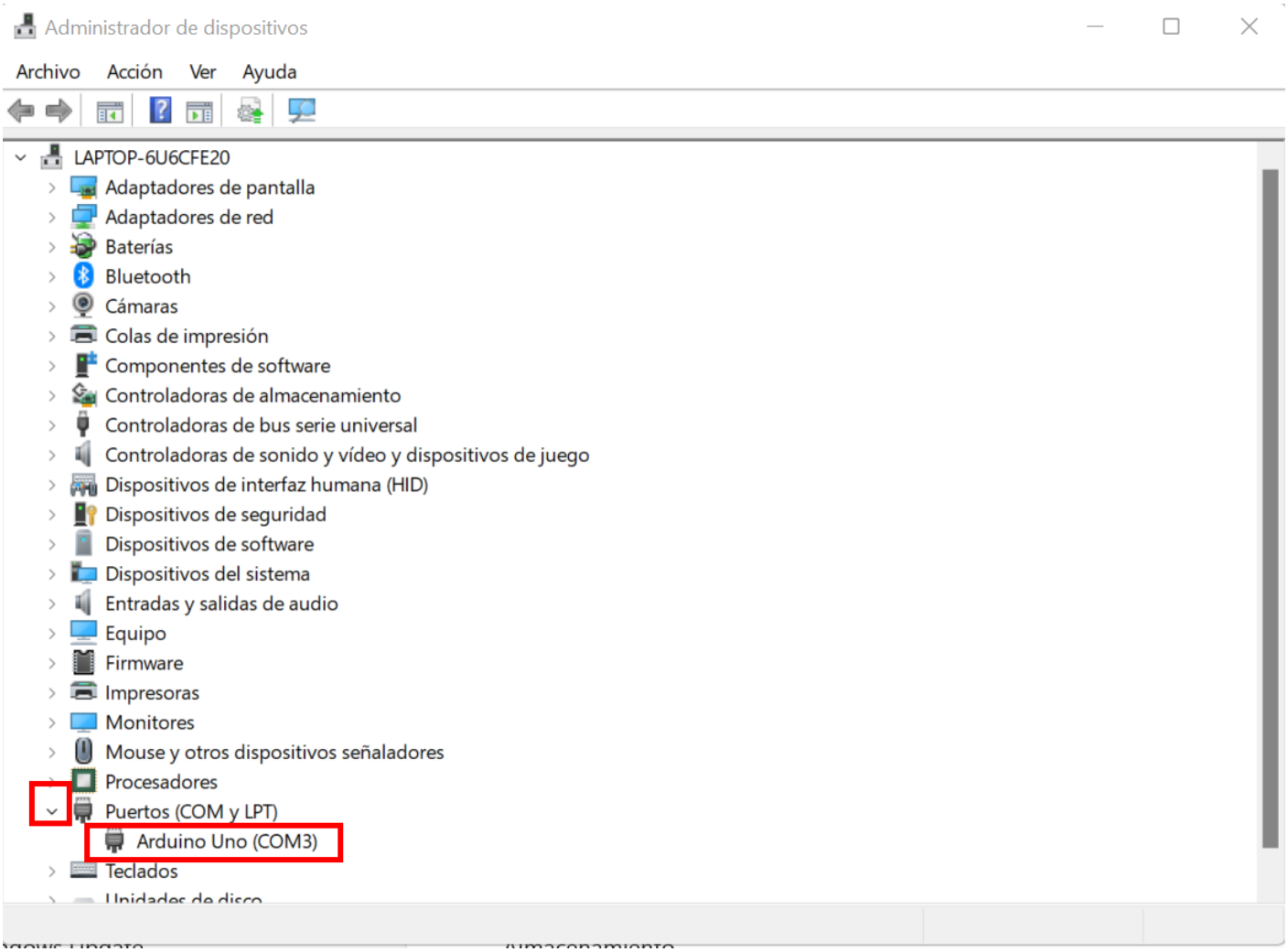
COMPROBAR QUE DETECTA LA PLACA

Ahora **conectamos la placa** (ESP32, Arduino, NodeMCU, KeyStudio TDR STEAM...) a nuestro ordenador, y observamos si lo detecta, en Windows entramos en Administrador de dispositivos:

Copyright 2026



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

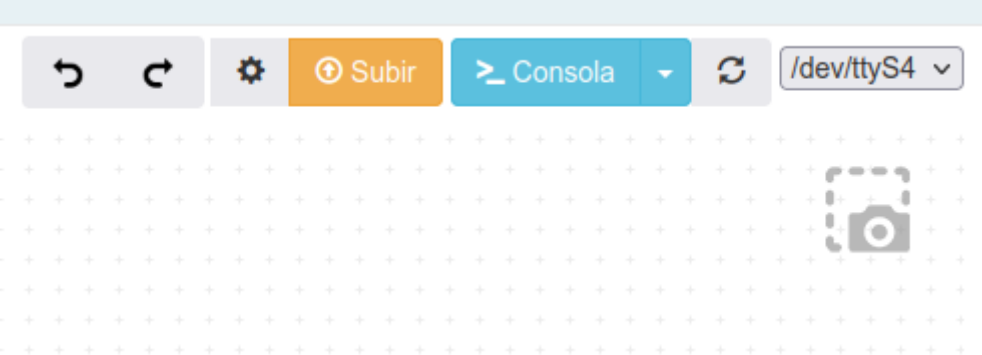
Entramos en la web ARDUINOBLOCKS http://www.arduinoblocks.com/ nos logueamos e iniciciamos un proyecto, Vemos que en el editor que aparece ya los puertos COM (si no te aparece, dale a la rueda actualizar)

> 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

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 que tipo de proyecto vamos a realizar:

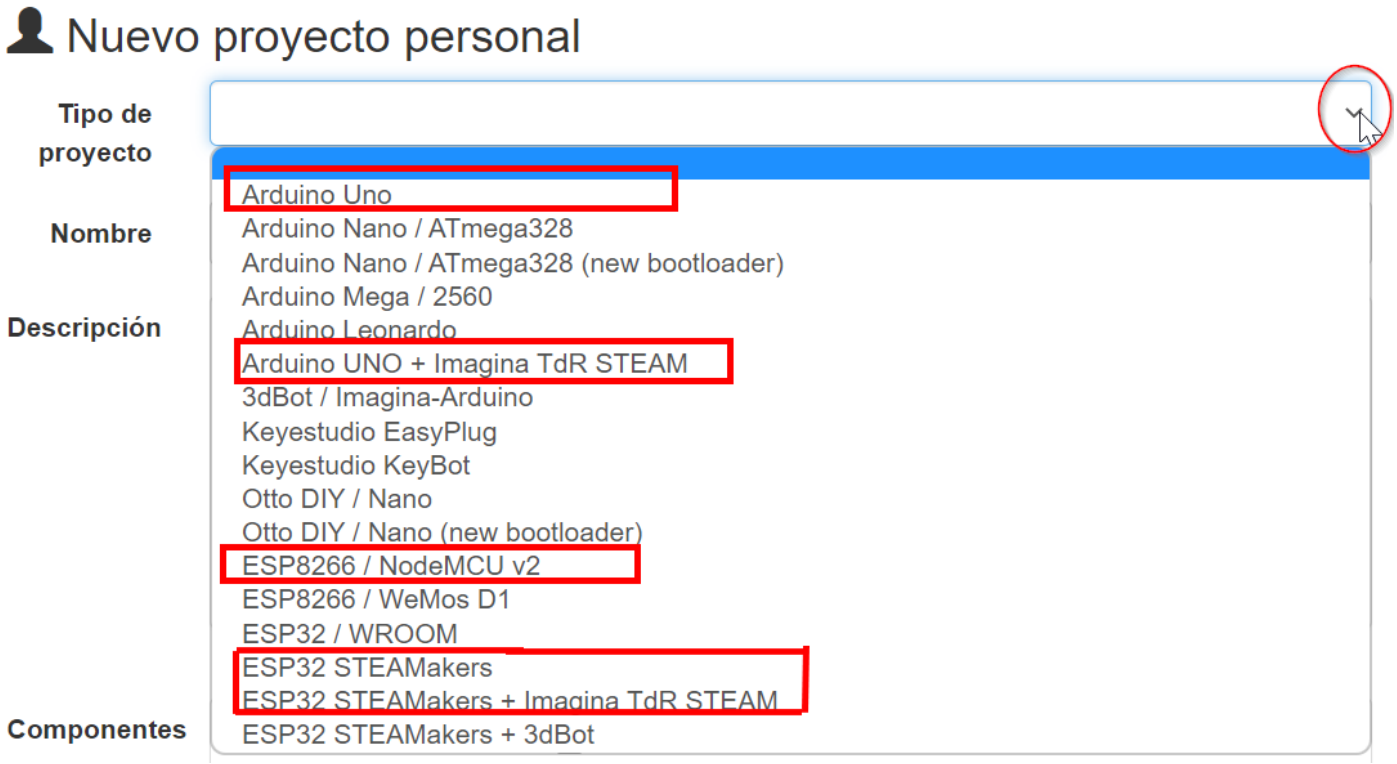
- **Proyecto Personal**: Iniciar un nuevo proyecto que sólo será accesible para el usuario. Posteriormente se puede compartir al resto de la comunidad si se desea.
- **Proyecto Profesor**: Iniciar un proyecto como profesor. De esta forma no se inicia un proyecto como tal, sino que se especifican los 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.

Copyright 2026

> *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 **STEAMAKERSBLOCKS 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 STEMakers + Imagina TdR STEAM**
- En el caso de que estés con el kit de CATEDU **SMARTHOME ESP32** tienes que elegir:
  - **ESP32 STEAMakers**
-



> **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 simpe: las instrucciones cambian)

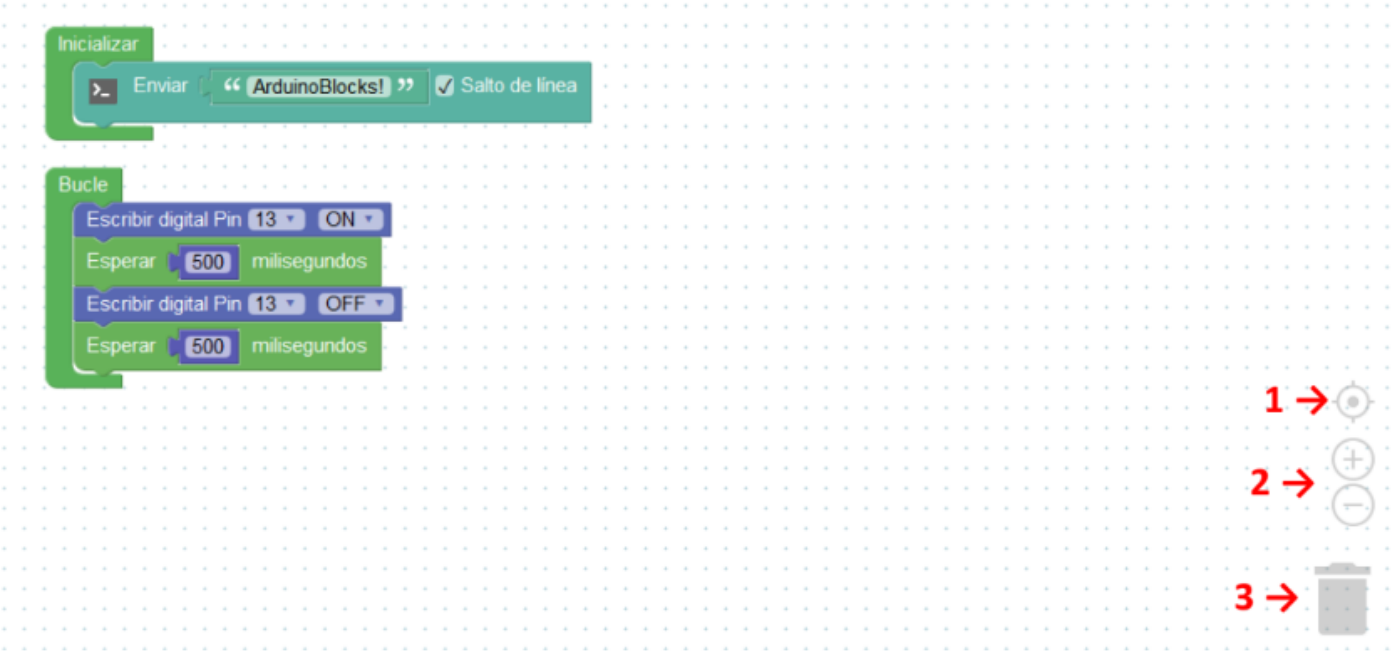
Luego el nombre y el resto de campos es optativo pero **importante y buena costumbre** rellenarlos, sobre todo si el proyecto lo **compartimos**:

- 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*

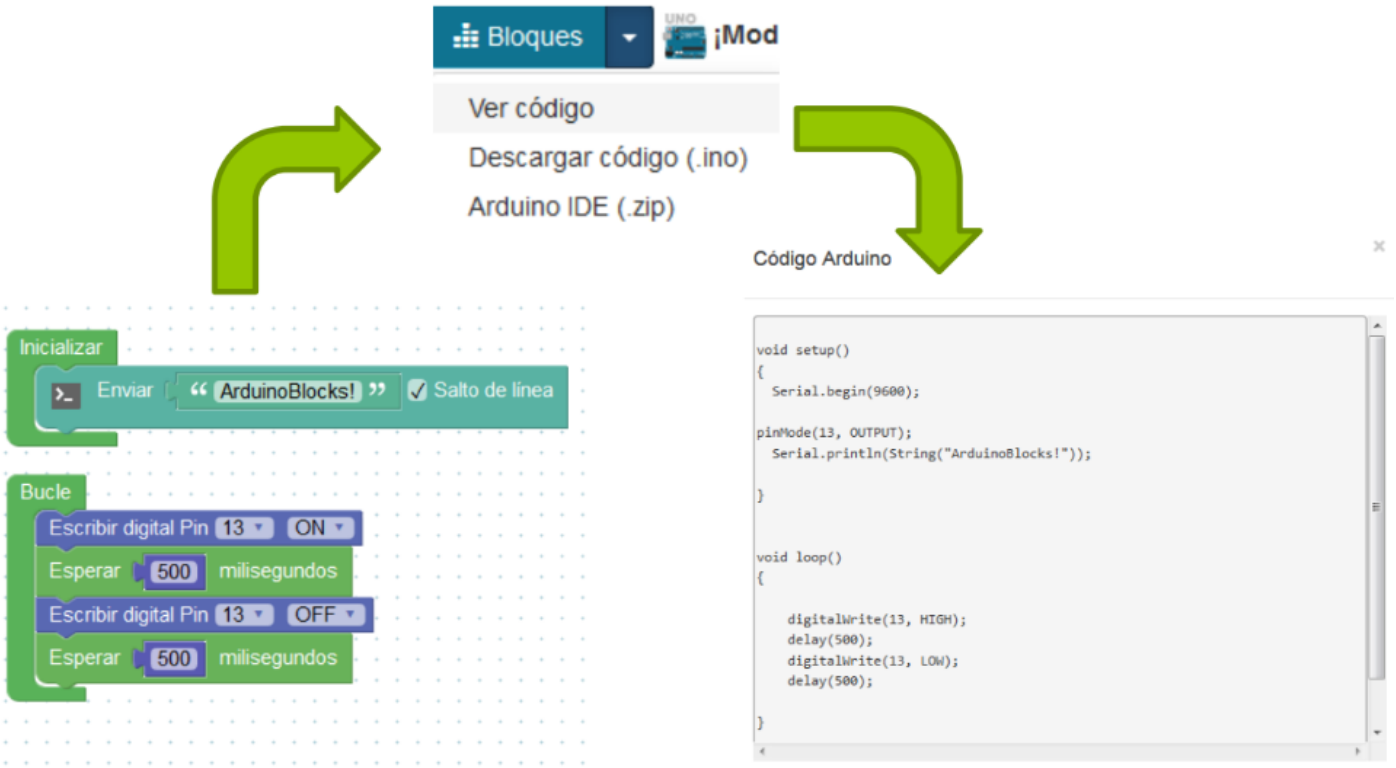
Las principales secciones del área de programación son las siguientes :



> *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

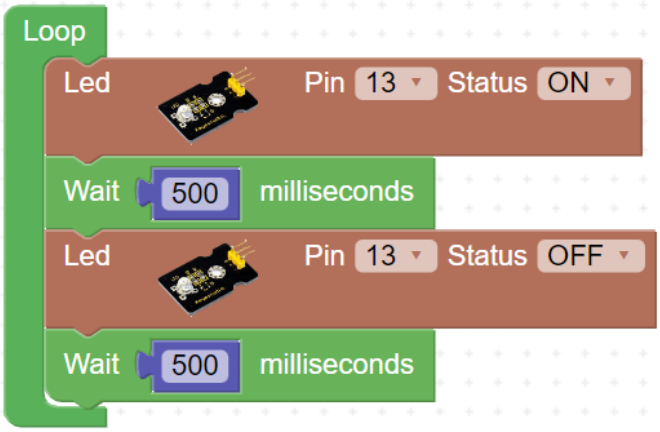
Con Imagina TdR STEAM ¿Qué tipo de proyecto elijo?

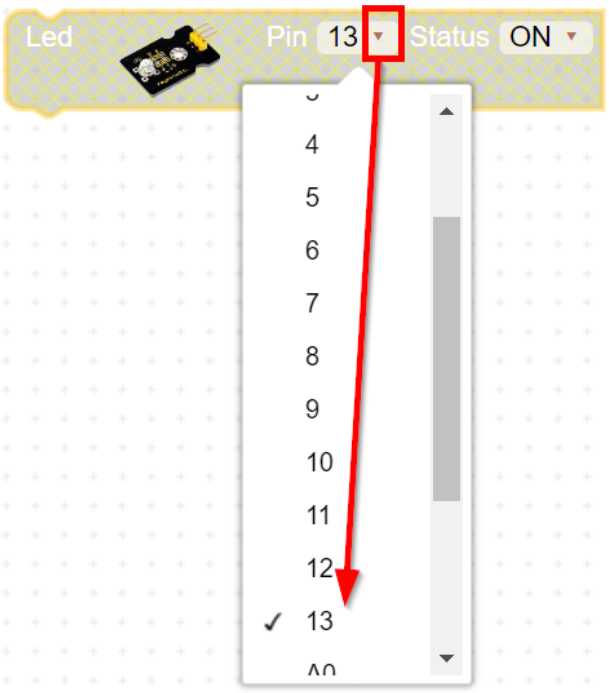
Para aclarar, las dos opciones son válidas para nuestros proyectos, pero la específica con Imagina TdR STEAM es más cómoda:

- **La placa sola (ESP32 STEAMakers)** donde te aparecerán todos los sensores y salidas con todas las posibilidades de entradas y salidas (E/S) de la placa (Arduino o ESP32)
- **La placa (ESP32 STEAMakers) + Imagina TdR STEAM** donde aparecen:
  - Los sensores y salidas específicas de la placa TdR STEAM ya conectadas en sus respectivos pines E/S del ESP32.
  - En contra, algunos sensores ya no aparecen todas las E/S del Arduino que están ocupadas por los elementos de la placa TdR STEAM.

Vamos a poner un ejemplo: Hacer una intermitencia con el LED AZUL del TdR STEAM

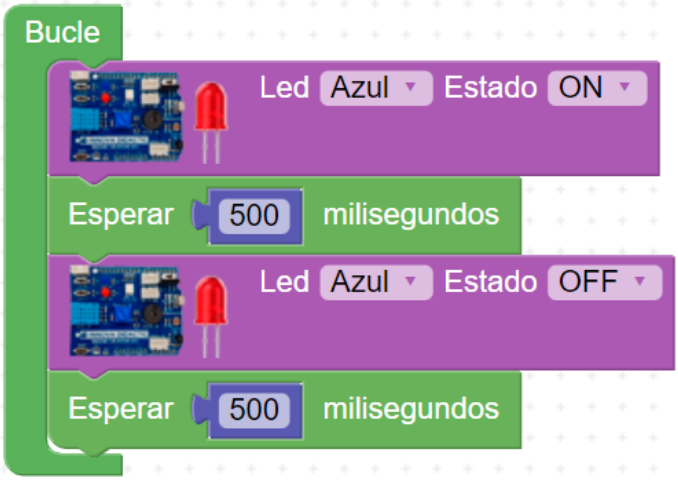
Si lo hacemos con el tipo de proyecto **ESP32 STEAMaker** tenemos que poner el Actuador LED y en el pin elegir el 13 porque el led Azul está conectado en el 13. Fíjate que en el desplegable del Pin están todas las E/S del ESP32

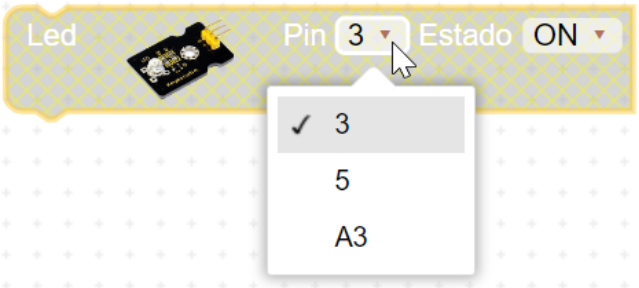




En cambio si elegimos un proyecto tipo **ESP32 STEAMaker+ Imagina TdR STEAM** veremos que tenemos añadidos unos actuadores específicos para esta placa TDR STEAM que es más fácil usarlos, donde en el desplegable sólo aparece ROJO o AZUL y el alumno **no necesita recordar** que ROJO es el 12 y AZUL es el 13

> Pero si elegimos un led del desplegable de actuadores, veremos que en su desplegable de pines sólo aparece los números 3, 5 y A3 **porque son los que tiene libres, el resto están ocupados por la placa TdR STEAM**





Está claro que el tipo de proyecto **ESP32STAMaker+ Imagina TdR STEAM** es más cómodo por que los bloques ya tienen fijados qué pines tienen cada sensor y si eliges un sensor no incluido en la placa, ya no muestra los pines que están ocupados.

> Si eliges **ESP32STEAMaker + Imagina TdR STEAM** es muy cómoda la programación y se evitan muchos errores.

> Si eliges **ESP32STEAMaker** utilizas un entorno de programación como si la placa Imagina TdR STEAM no existiera y los alumnos se acostumbran a usar los elementos propios de ESP32, los pines, etc.... pero tienen que saber en cada momento qué pin utilizar.
>
> Aconsejamos en este caso que los alumnos tengan impresa la página Componentes de la placa TdR STEAM.

En los apuntes de este libro vamos a utilizar el proyecto indistintamente las dos formas de trabajar.