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

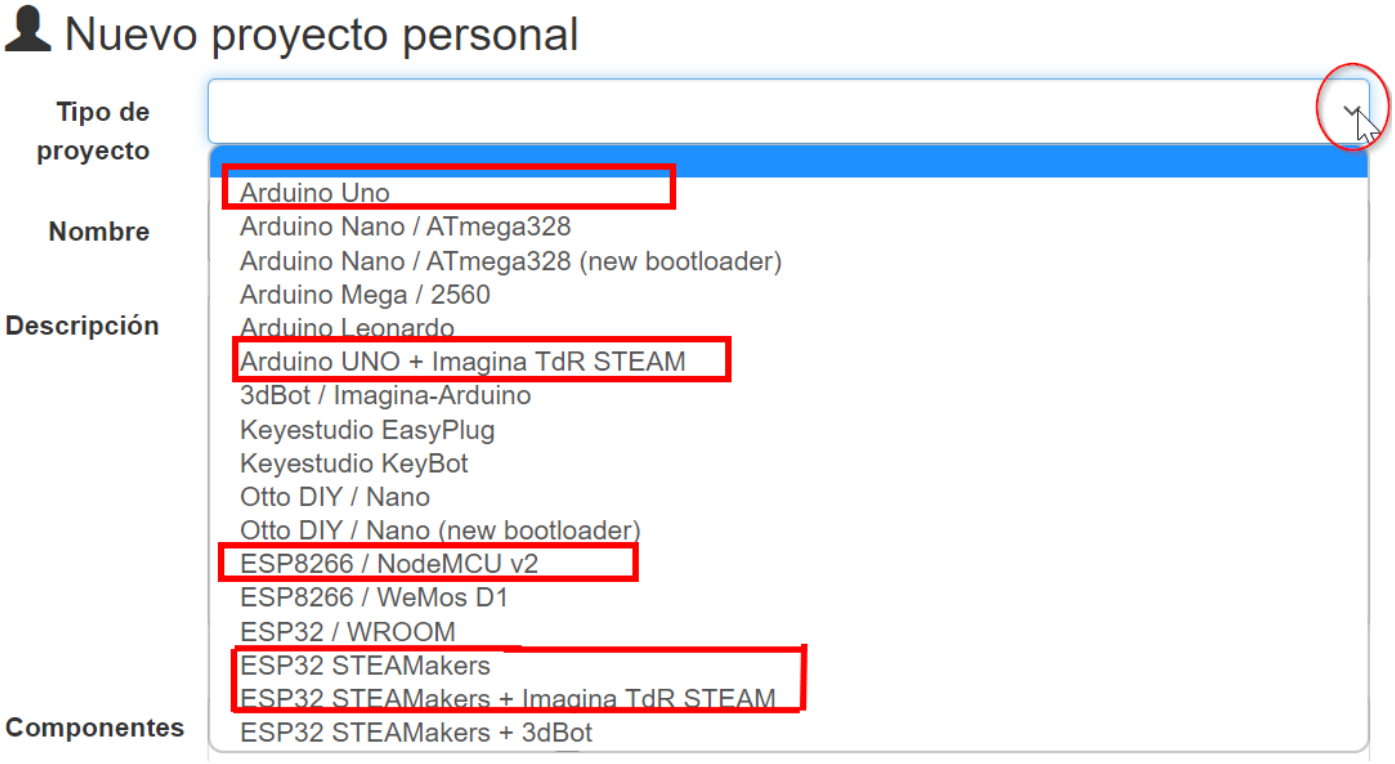
Copyright 2026

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 **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)

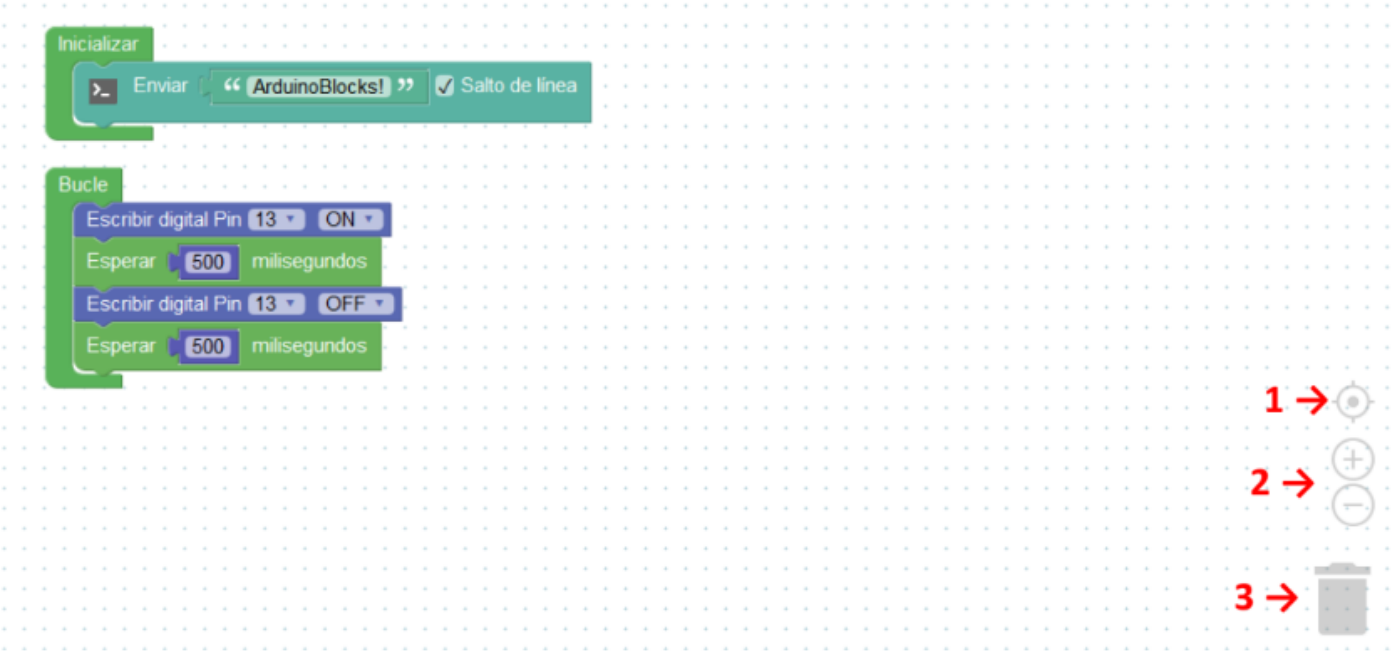
Luego el nombre y el resto de campos es optativo pero **importante y buena costumbre** rellenarlos, sobre todo si el proyecto lo **compartimos**:

- 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*

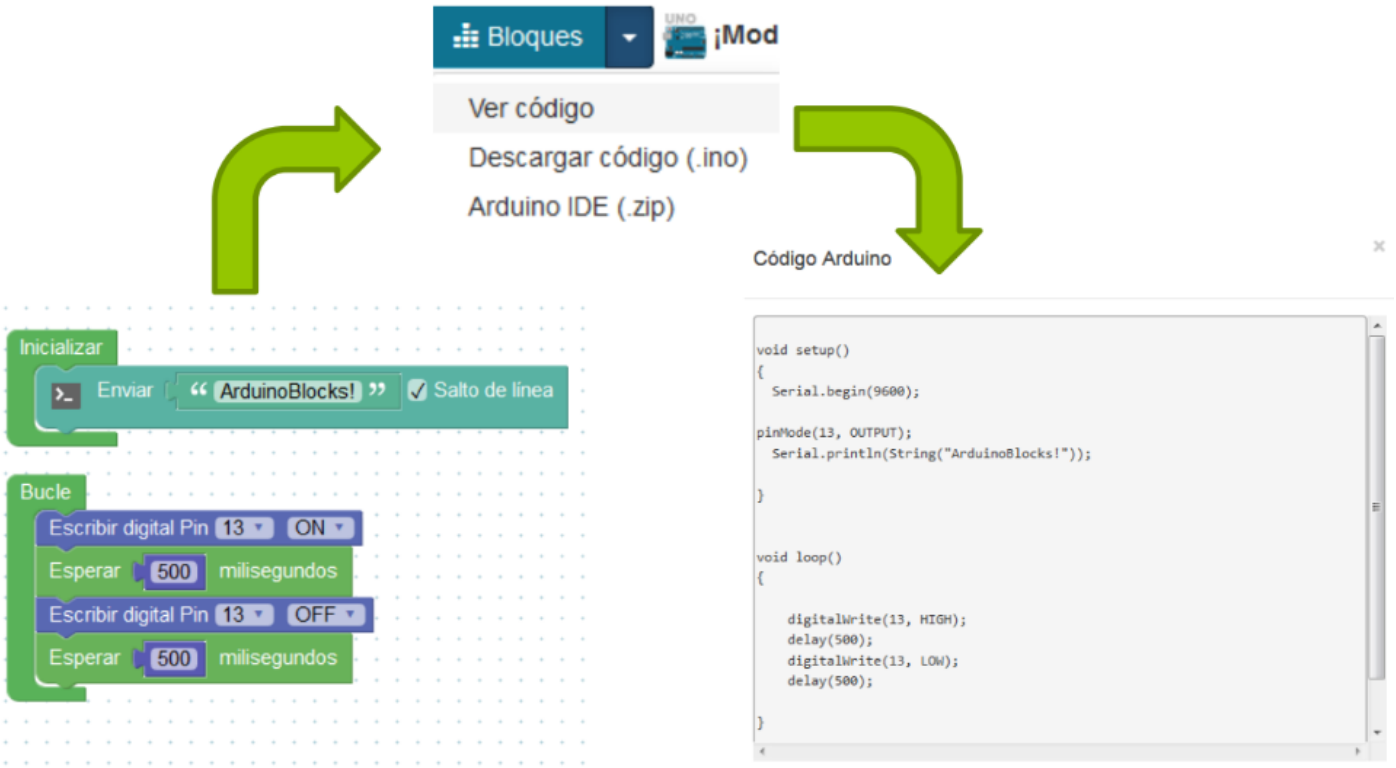
Las principales secciones del área de programación son las siguientes :



> *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?

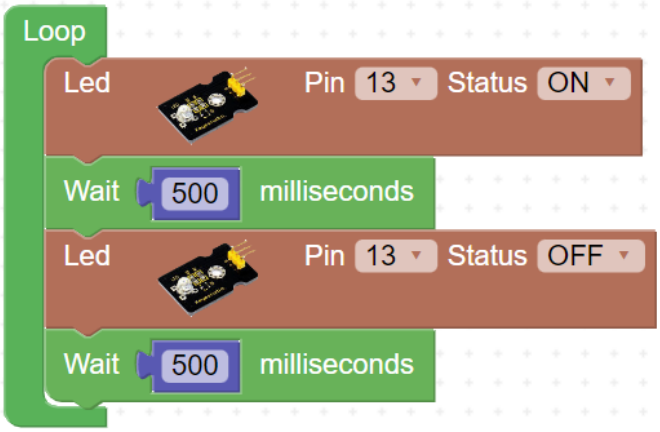
Para aclarar, las dos opciones son válidas para nuestros proyectos, pero la específica con Imagina TdR STEAM es más cómoda:

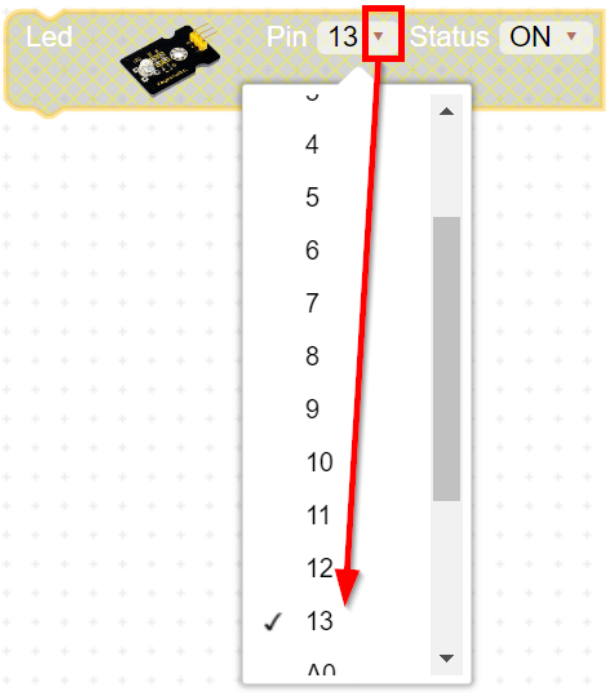
- **La placa sola (ESP32 STEAMakers)** donde te aparecerán todos los sensores y salidas con todas las posibilidades de entradas y salidas (E/S) de la placa (Arduino o ESP32)
- **La placa (ESP32 STEAMakers) + Imagina TdR STEAM** donde aparecen:
  - Los sensores y salidas específicas de la placa TdR STEAM 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

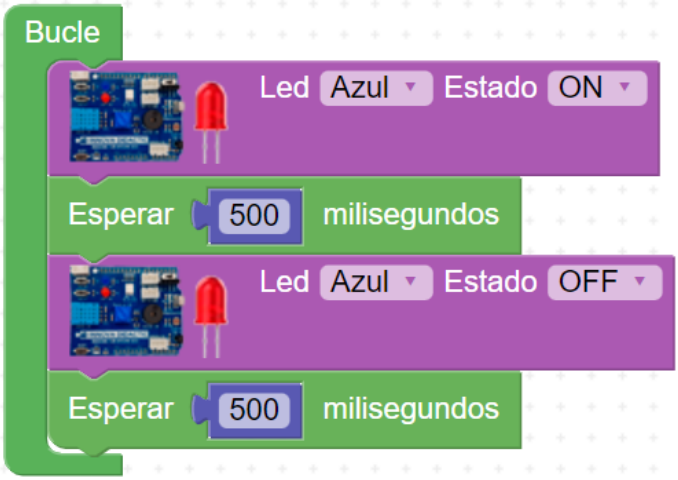
Copyright 2026

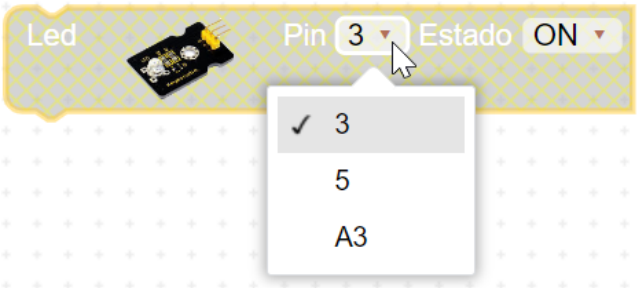




En cambio si elegimos un proyecto tipo **ESP32 STEAMaker+ Imagina TdR STEAM** veremos que tenemos añadidos unos actuadores específicos para esta placa TDR STEAM que es más fácil usarlos, donde en el desplegable sólo aparece ROJO o AZUL y el alumno **no necesita recordar** que ROJO es el 12 y AZUL es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





Copyright 2026

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.

Revision #1
Created 2026-09-18 13:18:43 CEST by Silvia Coscolin Sanchez
Updated 2026-09-18 13:19:14 CEST by Silvia Coscolin Sanchez