
ESP32

STEAMMAKERS

Abacus es una organización de referencia en la educación, la cultura y el ocio que crea y distribuye contenidos y productos de calidad para contribuir a un país y un mundo mejores desde los valores del cooperativismo.

Tanto Habilis, la plataforma de creación de contenidos educativos, como RO-BOTICA, un referente en la distribución de soluciones tecnológicas, se engloban dentro de Abacus Educación.

HABILIS

En Habilis nos dedicamos a la creación de contenido educativo innovador, a la formación docente y al acompañamiento de las escuelas en la transformación digital, con más de 80 asesores pedagógicos.

RO-BOTICA

RO-BOTICA abrió en 2007 el primer espacio de Europa dedicado a robótica educativa. Desde entonces, distribuye e implementa soluciones tecnológicas de robótica y programación en centros de todas las etapas educativas. Contribuye al desarrollo competencial digital y tecnológico de la sociedad, facilitando e impulsando vocaciones científico-tecnológicas, la alfabetización STEAM, la educación inclusiva, equitativa y de calidad.

Consulta nuestro catálogo:



ÍNDICE

1.	Descripción	4
2.	¿Qué hay en la caja?	5
2.1.	¿Cuáles son los componentes de la placa?	6
2.2.	¿Cuál es la función de cada componente?	9
3.	Programa desde el ordenador	10
3.1.	Instala los controladores necesarios	10
3.2.	Conoce el entorno de programación	11
3.2.1.	ArduinoBlocks	11
3.2.2.	Arduino IDE	12
3.3.	Comienza a programar	13
3.3.1.	Programando con ArduinoBlocks	13
3.3.2.	Programando con Arduino IDE	18
4.	¿Te atreves con los retos?	20
4.1.	Posibles soluciones	25
5.	Consejos de seguridad	27
6.	Anexo	28

DESCRIPCIÓN



La placa **ESP32 STEAMakers** es una placa electrónica programable basada en el microcontrolador ESP32 que, gracias a una disposición versátil de conectores, facilita el acceso a sus numerosas entradas y salidas, a las que conectar toda clase de sensores y actuadores. Incorpora comunicación wifi y *bluetooth* integrados, así como más capacidad de procesamiento que la popular placa electrónica Arduino UNO.

Esta placa puede programarse mediante diversas plataformas y lenguajes, como la Arduino IDE con código basado en C/C++ o los entornos de programación visual ArduinoBlocks y MicroBlocks.

Todo ello convierte a la ESP32 STEAMakers en un recurso muy versátil y accesible para iniciarse en la electrónica, la programación, el prototipaje de proyectos de robótica, el internet de las cosas, la domótica, etc. Es especialmente adecuada para docentes y estudiantes gracias a su fácil programación y conectividad inalámbrica integrada.

INFORMACIÓN

ETAPA EDUCATIVA

ESO

Bachillerato

CONECTIVIDAD



BLE

ENTORNO DE PROGRAMACIÓN

ArduinoBlocks



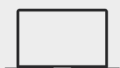
MicroBlocks



Arduino



DISPOSITIVOS



SISTEMAS OPERATIVOS



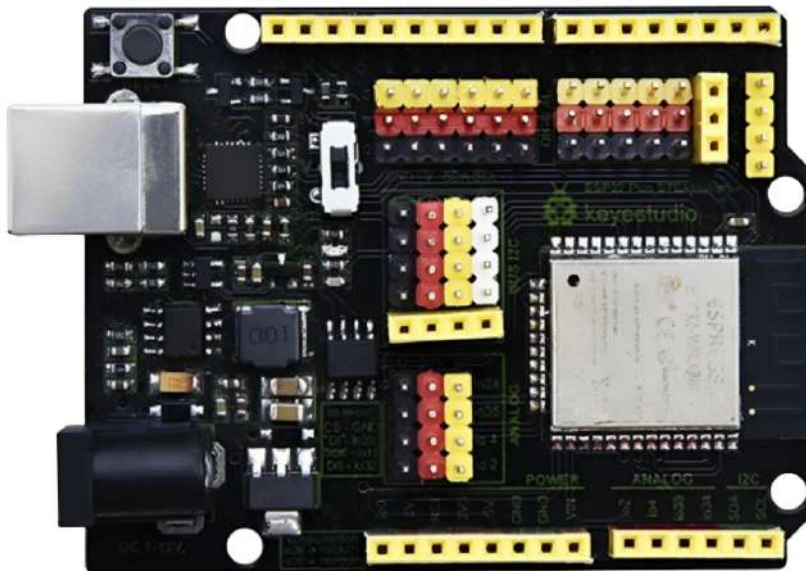
linkatedu®



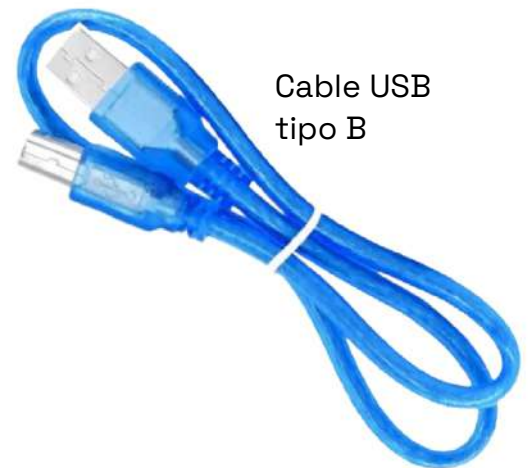
2

ESP32 STEAMAKERS

¿QUÉ HAY EN LA CAJA?



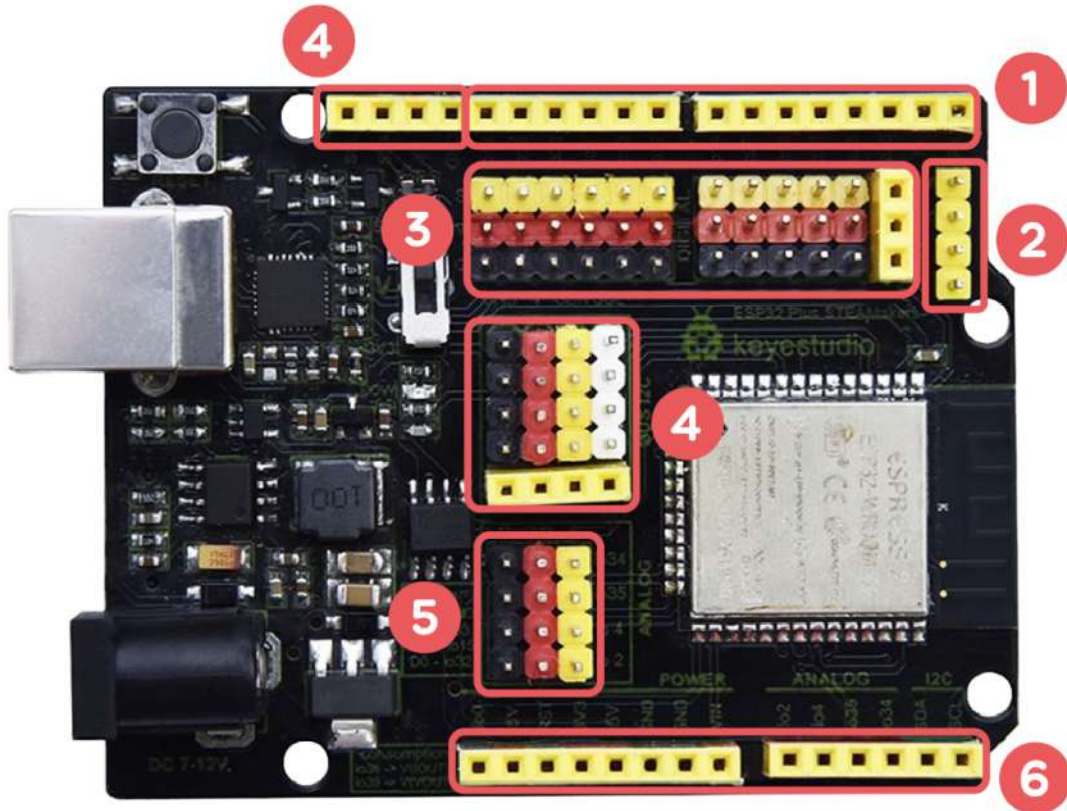
Placa ESP32 STEAMakers



Cable USB
tipo B

2.1. ESP32 STEAMMAKERS

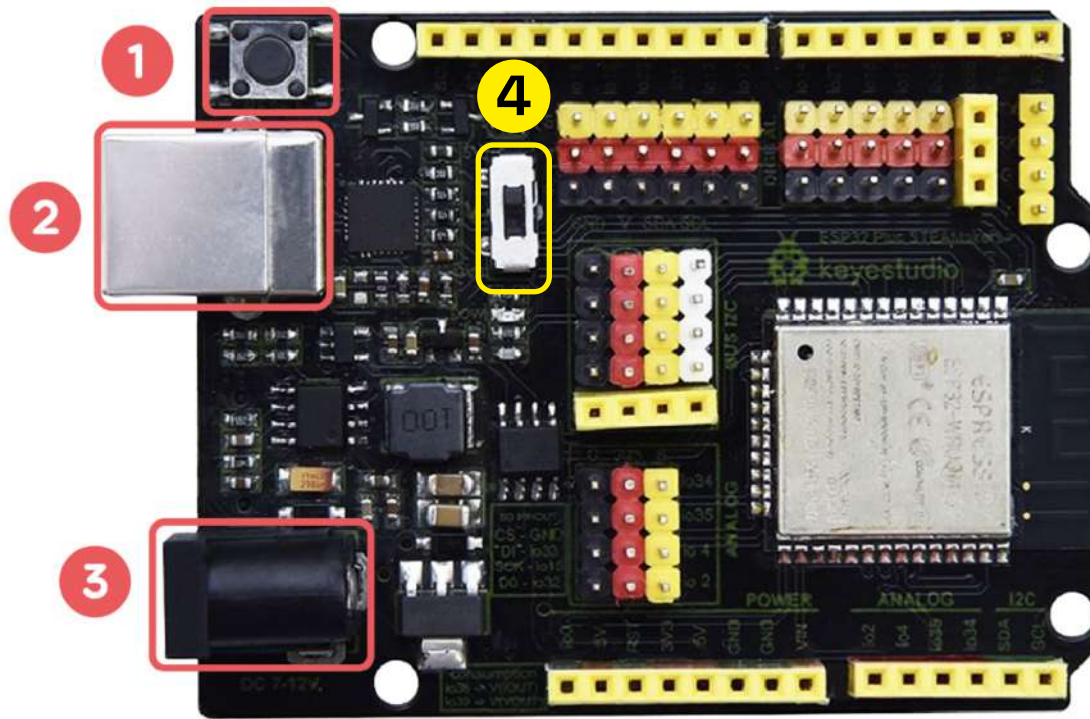
¿CUÁLES SON LOS COMPONENTES DE LA PLACA?



Conexiones

- 1 Pines de entradas y salidas digitales
- 2 Pines RX/TX (comunicación en serie)
- 3 Pines GVS de entradas y salidas digitales
- 4 Pines I2C
- 5 Pines GVS de entradas y salidas analógicas
- 6 Pines de entradas y salidas analógicas

2.1. ESP32 STEAMAKERS



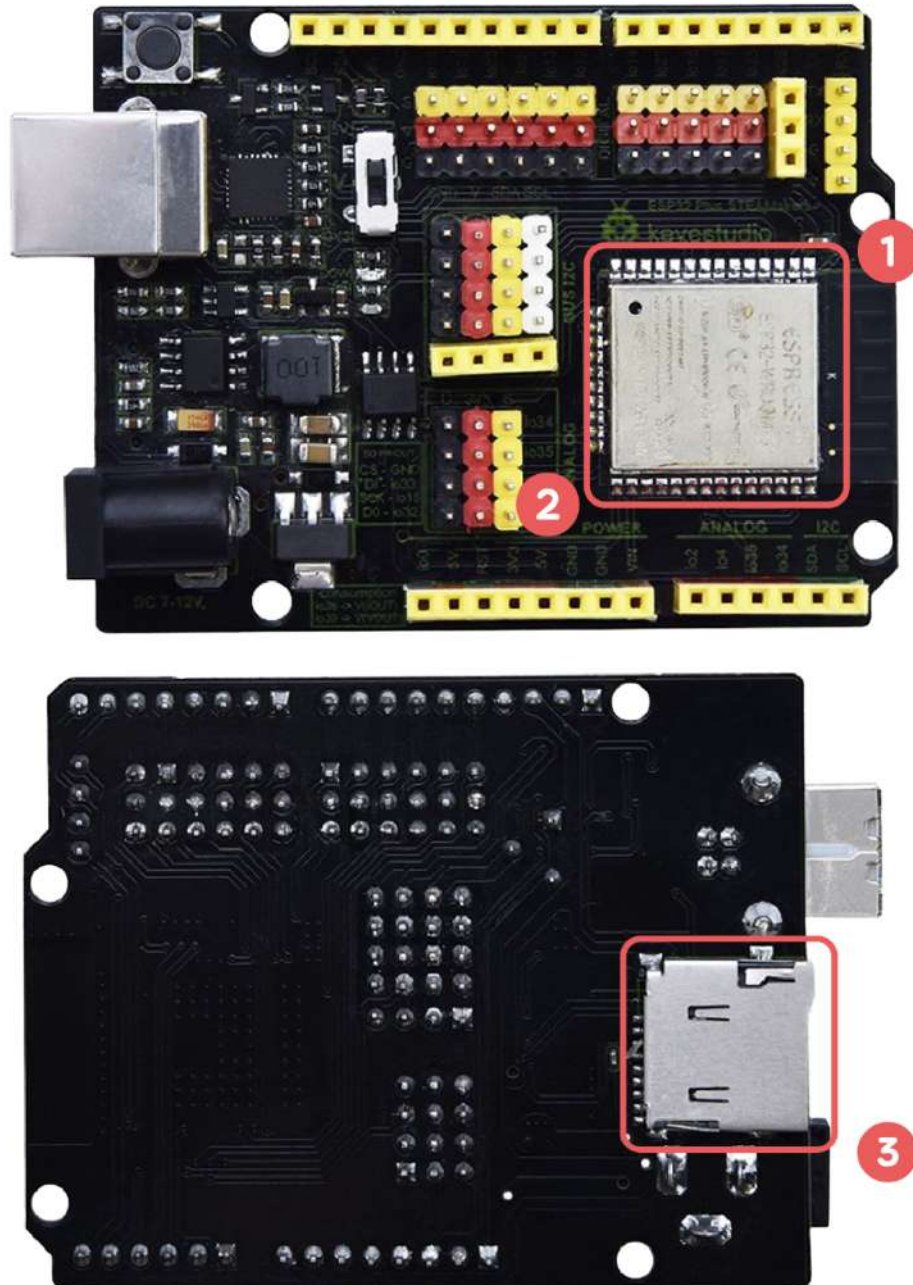
Fuente de alimentación

- 1 Botón de arranque
- 2 Puerto USB tipo B
- 3 Conector jack de alimentación

Interruptores

- 4 Selector de voltaje 3,3 V o 5 V

2.1. ESP32 STEAMMAKERS



Componentes integrados

- 1 Sensor de temperatura interno
- 2 Sensor Hall interno (magnético)
- 3 Lector de tarjetas micro-SD

¿CUÁL ES LA FUNCIÓN DE CADA COMPONENTE?

La placa ESP32 STEAMakers facilita el acceso a las potentes capacidades del microcontrolador ESP32 mediante una disposición versátil de conectores y una amplia variedad de entradas y salidas. Eso permite crear de manera sencilla toda clase de proyectos interactivos con sensores, actuadores y comunicaciones inalámbricas.

Algunos ejemplos de proyectos que se pueden realizar en el aula:

DETECTAR EL NIVEL DE LUZ

A través de un sensor de luz conectado a una de las entradas analógicas de la placa, se puede medir la intensidad lumínica de un espacio y activar actuadores en función de si hay más o menos claridad.

MEDIR LA TEMPERATURA

El sensor de temperatura integrado permite obtener y monitorizar la temperatura. Se puede mostrar a través de una pantalla o enviar a un servidor.

DETECTAR PROXIMIDAD O MOVIMIENTO

Mediante el sensor de tipo Hall (magnético) interno en la placa o un sensor de ultrasonidos conectado a una entrada del ESP32 STEAMakers se puede detectar cuándo un objeto o persona se acerca o se aleja de la placa.

REPRODUCIR AUDIO

Empleando un zumbador se pueden generar tonos y melodías que puedan formar parte de una experiencia interactiva.

CONTROLAR REMOTAMENTE POR WIFI

Aprovechando la conectividad wifi integrada, se pueden crear interfaces web o apps móviles para controlar de forma remota los componentes conectados a la placa.

COMUNICAR CON OTROS DISPOSITIVOS

La conectividad *bluetooth* permite que la placa intercambie datos en tiempo real con tabletas, teléfonos inteligentes u otros sistemas externos con *bluetooth* integrado.

PROGRAMA DESDE EL ORDENADOR

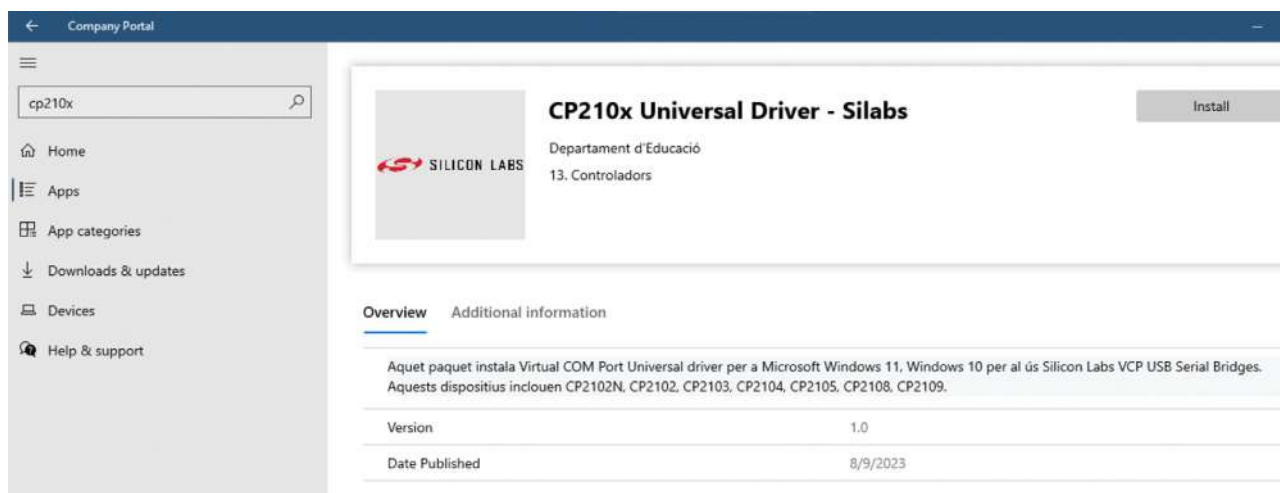
3.1. INSTALA LOS CONTROLADORES NECESARIOS

La placa ESP32 STEAMakers utiliza un controlador CP2102 para la conexión USB. Dependiendo del sistema operativo que se utilice, habrá que instalar el controlador de una manera o de otra.

Lo podemos descargar desde el Portal d'aplicacions del Pla d'Educació Digital de Catalunya o el Company Portal (<https://projectes.xtec.cat/edc/aplicacions/>) escribiendo "CP210" en el buscador.

A Windows

1. Abre el Portal d'aplicacions o el Company Portal haciendo doble clic en el icono que encontrarás en el escritorio.
2. En el buscador de la parte superior izquierda escribe el nombre de la aplicación o controlador que quieras instalar. En este caso, **CP210X**.



3. Para instalarlo, habrá que hacer clic en el botón "Install" que se muestra en la ventana.
4. Una vez se haya instalado el controlador, se mostrará un mensaje de "Installed" y ya se habrá acabado el proceso de instalación.

B Linkat y Chromebook

En el caso de utilizar Linkat, no hace falta instalar ningún controlador, el entorno de programación detectará automáticamente la placa.

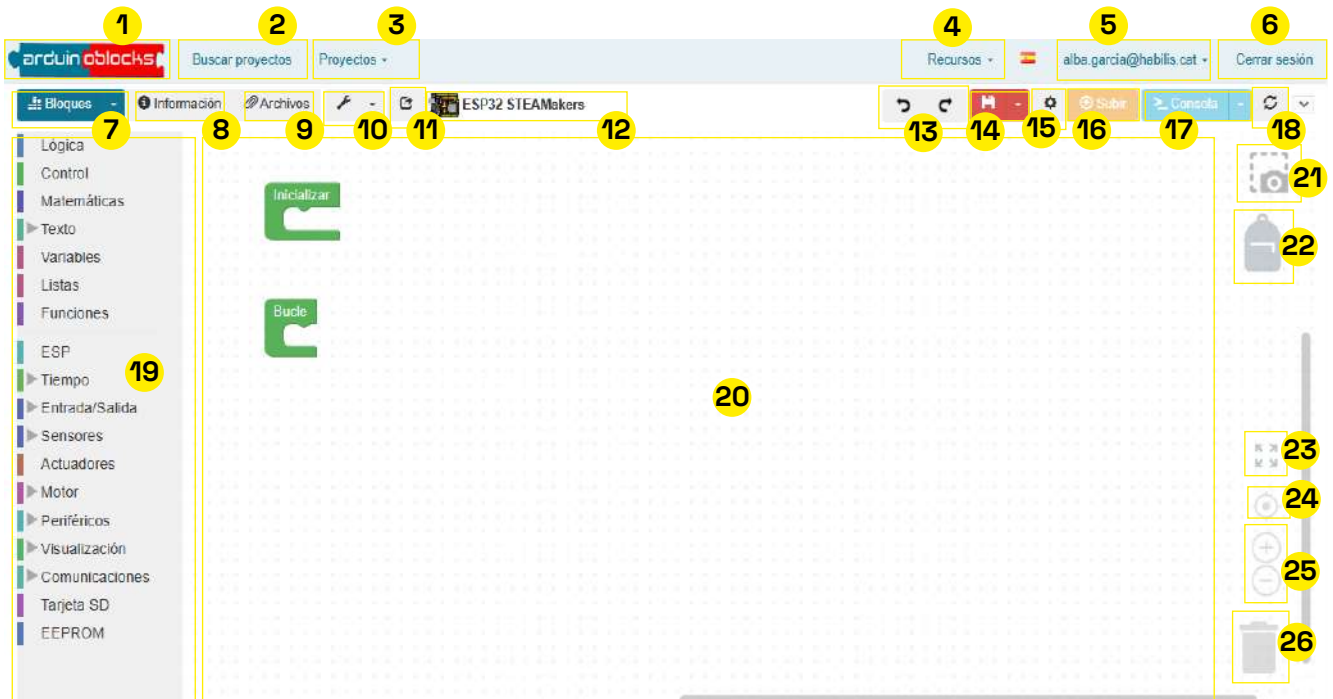
3.2. ESP32 STEAMAKERS

CONOCE EL ENTORNO DE PROGRAMACIÓN

3.2.1. ARDUINOBLOCKS

ArduinoBlocks (<http://www.arduinoblocks.com/>) es un entorno de programación visual que facilita la programación de la placa. Utiliza bloques de programación similares a los de Scratch.

ArduinoBlocks es compatible con diversas placas electrónicas, como la placa Keystudio UNO y la placa ESP32 STEAMakers. En su entorno de programación se pueden distinguir las siguientes partes:



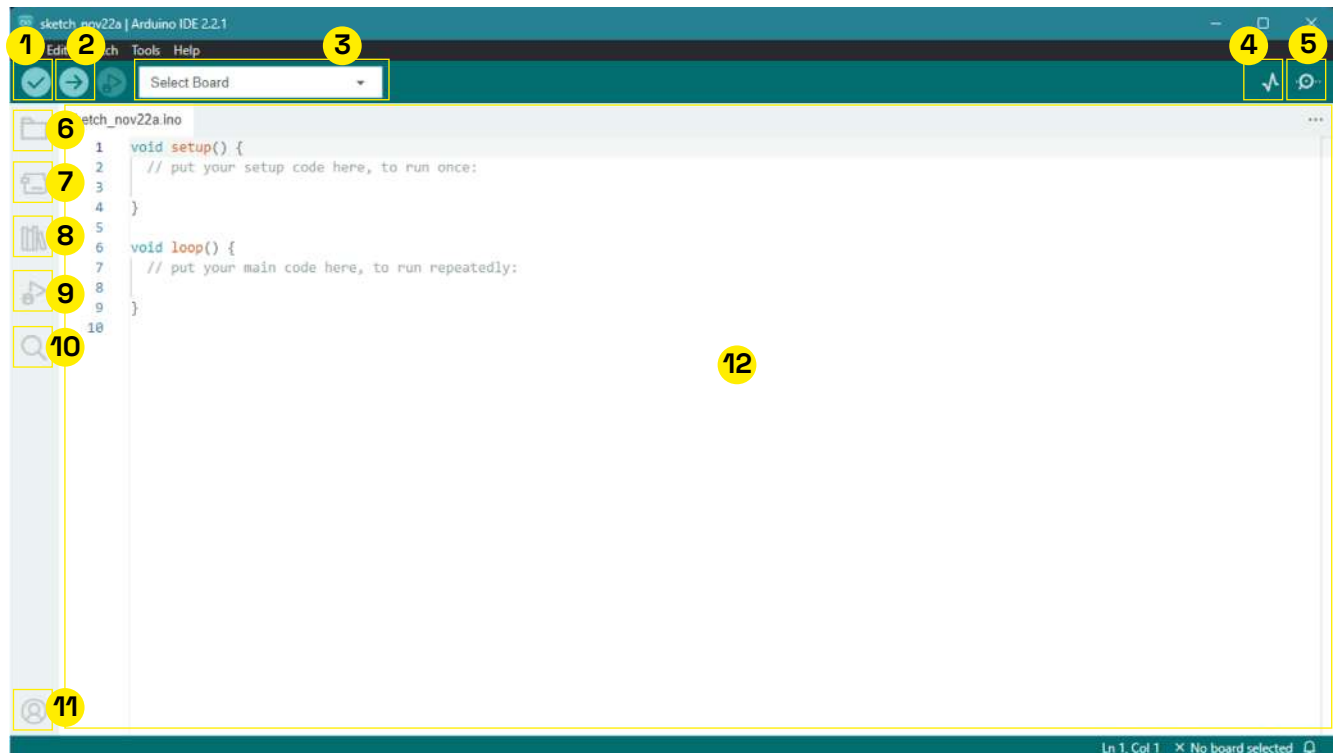
1. Página principal ArduinoBlocks.
2. Repositorio de proyectos.
3. Desplegable para crear un nuevo proyecto y para acceder a proyectos propios.
4. Desplegable para instalar un controlador y para acceder a documentación diversa.
5. Acceso a la cuenta personal.
6. Cerrar sesión.
7. Desplegable para cambiar de bloques a código y descargar el programa.
8. Acceso a la información del proyecto (nombre, descripción, etc.).
9. Acceso a archivos adjuntos al proyecto.
10. Acceso a herramientas de edición de matrices de ledes y pantallas LCD y OLED.
11. Botón para compartir proyectos.
12. Nombre del proyecto con el icono de la placa seleccionada para programar.
13. Botones para deshacer y rehacer.
14. Botón con desplegable para guardar y exportar el proyecto.
15. Configuración del programa AB-Connector.
16. Botón para subir el programa a la placa una vez conectada.
17. Botón para abrir la consola serie.
18. Actualización.
19. Bloques de programación organizados por categorías.
20. Espacio de programación.
21. Captura de pantalla del programa sin fondo.
22. Espacio donde guardar las instrucciones y estructuras de programación más utilizadas.
23. Botón para maximizar la medida de los bloques.
24. Botón para centrar los bloques.
25. Botones para ampliar o reducir la medida de los bloques.
26. Papelera donde descartar los bloques.

3.2.2. ARDUINO IDE

Arduino IDE es el entorno de desarrollo integrado multiplataforma para Windows, macOS o Linux para programar placas compatibles con Arduino, como es el caso de la placa ESP32 STEAMakers, con un lenguaje de programación basado en C/C++.

IDE (Entorno Integrado de Desarrollo) incorpora un editor de código que facilita la escritura de programas. También dispone de numerosas bibliotecas y ejemplos que ayudan en la programación. Permite compilar y subir los programas a la placa con tan solo un clic, visualizando el resultado a través del monitor serie, por ejemplo.

En el entorno de programación se pueden distinguir las siguientes partes:



1. Verificación del programa.
2. Carga del programa.
3. Selección de la placa de control que programar.
4. Botón para abrir el *plotter*, herramienta para visualizar los datos enviados por el puerto en serie en forma de gráfico.
5. Botón para abrir el monitor en serie.
6. Acceso a otros programas propios, en local o en la nube.
7. Acceso al gestor de placas de control.
8. Acceso al gestor de bibliotecas.
9. Acceso a los datos de depuración del programa para analizar su funcionamiento.
10. Acceso a la herramienta "Search and Replace" para buscar cadenas de texto dentro de código de categorías.
11. Perfil de usuario.
12. Área de programación.

3.3. ESP32 STEAMAKERS

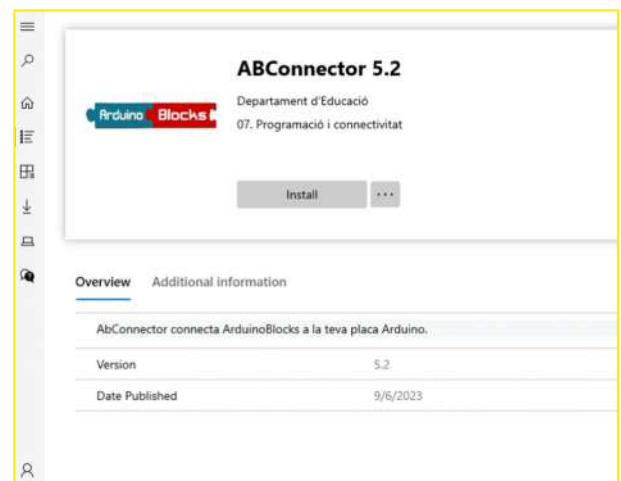
COMIENZA A PROGRAMAR

3.3.1. PROGRAMANDO CON ARDUINOBLOCKS

Para conectar y programar la placa ESP32 STEAMakers con ArduinoBlocks, hay que instalar "AB-Connector" para permitir la comunicación entre el entorno ArduinoBlocks y a la placa electrónica.

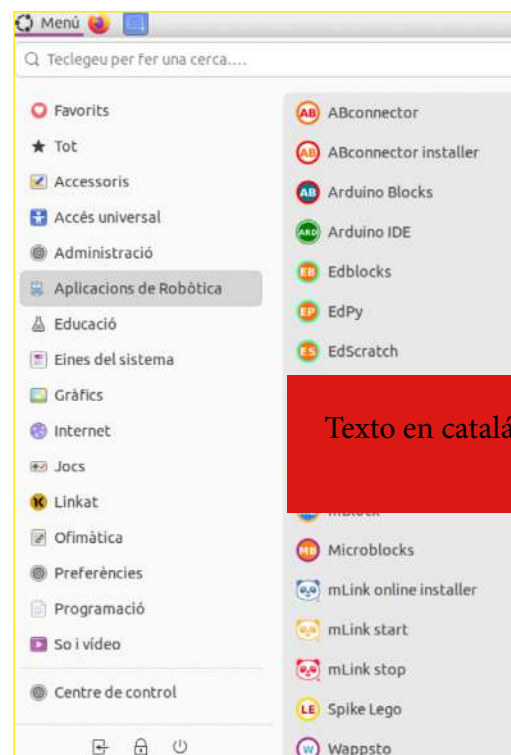
Windows

1. Abre el Portal d'aplicacions o el Company Portal haciendo doble clic en el icono que encontrarás en el escritorio.
2. En el buscador de la parte superior izquierda escribe el nombre de la aplicación o controlador que quieres instalar. En este caso, AB-Connector.
3. Para instalarlo habrá que hacer clic en el botón "Install" que aparece en la ventana.
4. Una vez se ha instalado el controlador, aparecerá el mensaje "Installed", y ya se habrá acabado el proceso de instalación.



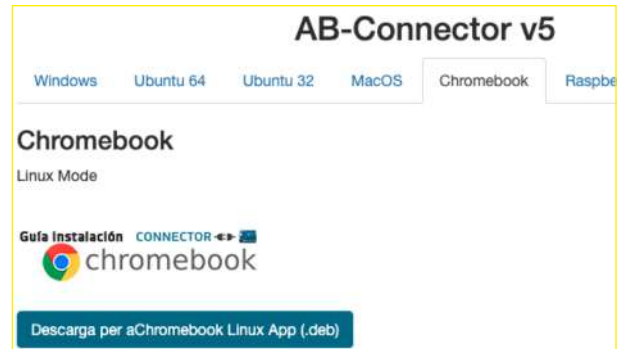
Linkat

1. En la carpeta Aplicaciones de Robòtica busca AB-Connector.
2. Selecciónalo y haz clic sobre él.
3. En el caso de que sea necesaria una actualización del controlador, se hará automáticamente.

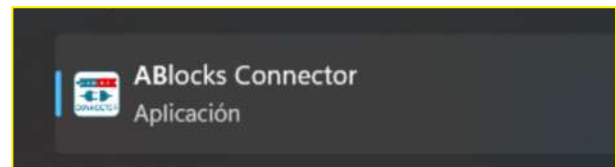


Chromebook

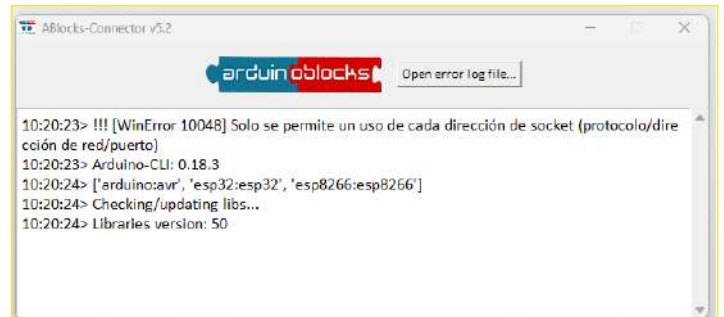
1. • Accede a la página web:
<http://www.arduinooblocks.com/web/site/abconnector5>
- Descarga la versión de AB-Connector para Chromebook (.deb)



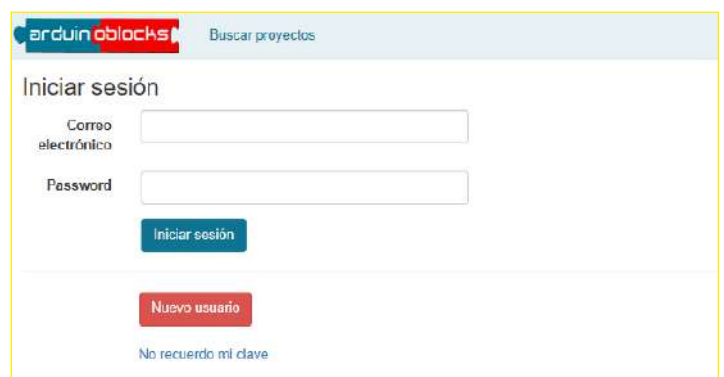
2. Una vez instalado el programa, lo buscaremos en la barra de navegación y lo ejecutaremos.



Al abrir AB-Connector, aparece este cuadro de diálogo que va mostrando la información sobre la placa ArduinoBlocks:



- 1 En ArduinoBlocks, crea una cuenta haciendo clic en "Iniciar sesión" y, posteriormente, en "Nuevo usuario":



3.3. ESP32 STEAMAKERS

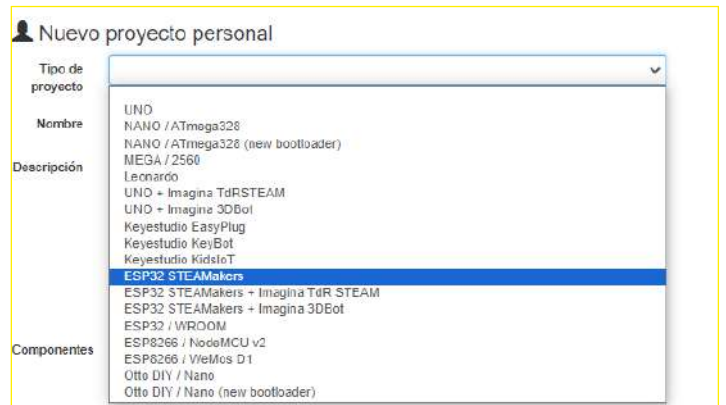
2 Al entrar en la plataforma, encontrarás esta pantalla:



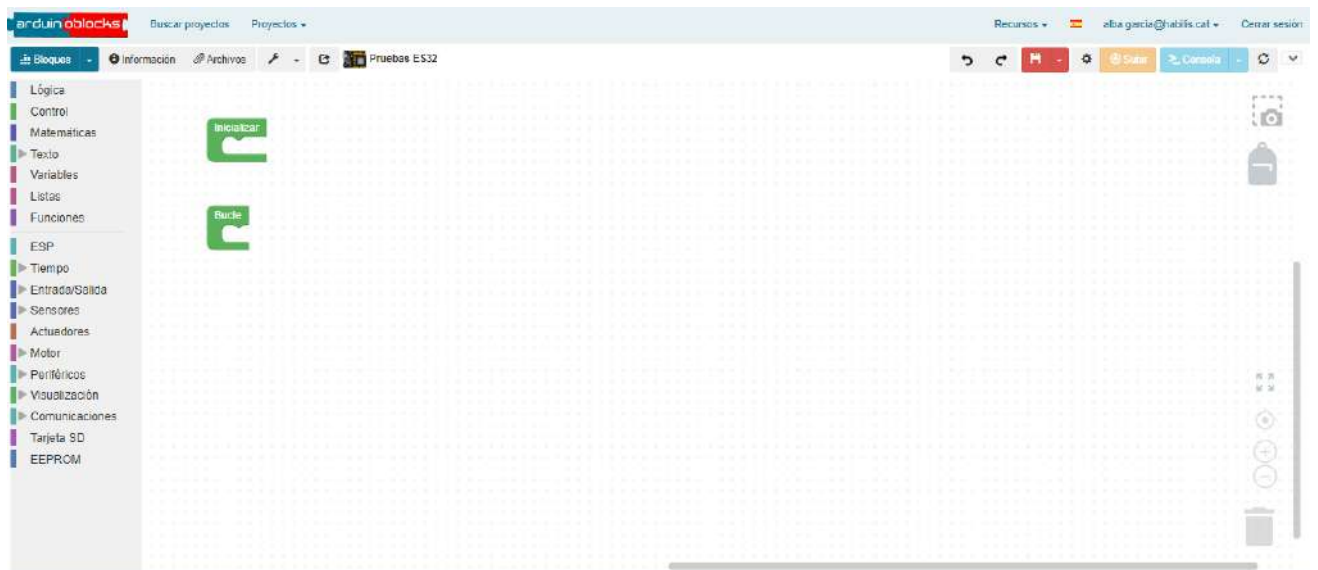
3 Al clicar en “¡Empezar un nuevo proyecto!”, aparece la pantalla siguiente para elegir el tipo de proyecto:



4 Para crear un nuevo proyecto personal, tienes que rellenar un formulario. Si quieres utilizar la placa ESP32 STEAMakers, la has de seleccionar en “Tipo de proyecto”.



5 Una vez creado el proyecto, después de haber completado el formulario, se abre el entorno de programación:



3.3.

ESP32 STEAMAKERS

6 En la parte izquierda de esta pantalla encontrarás los bloques disponibles clasificados por diferentes categorías. Por ejemplo, en la siguiente imagen puedes ver los bloques de lectura de los sensores integrados en la placa ESP32 STEAMakers.



7 Arrastra los bloques al espacio de programación para programar tu placa. Podemos programar, por ejemplo, el envío en bucle de un mensaje en concreto por el puerto serie. El puerto serie, también conocido como puerto de comunicaciones en serie o interfaz en serie, es un tipo de conexión utilizada en ordenadores y dispositivos periféricos para la transferencia de datos. La característica principal del puerto serie es que envía los datos en serie, es decir, bit a bit, a través de un solo canal o cable.



Los bloques más importantes para utilizar el puerto en serie son:

8 El programa de este ejemplo sería así:

Dentro de la estructura "Inicializar" colocamos el bloque de configuración de la velocidad de transferencia de datos ("baudios") del puerto serie a 9.600. Y en el bucle principal colocamos el bloque para enviar un mensaje en concreto, seleccionando que se realice un salto de línea.



3.3.

ESP32 STEAMAKERS

9 Una vez finalizado el programa, tienes que transferirlo a la placa. Para hacerlo, sigue los siguientes pasos:

- Comprueba que ABlocks-Connector está abierto.
 - Conecta la placa al ordenador por USB.
 - Selecciona el puerto de comunicación. Puedes conectar y desconectar el cable USB del ordenador para diferenciar cuál es el puerto de comunicación.
 - Si no aparece el "COM" directamente, haz clic en el icono de actualización.
-
- Haciendo clic en el botón "Subir", carga el programa en la placa.

Para visualizar el monitor serie y comprobar qué mensajes está enviando la placa al ordenador, ábrelo en tu entorno de programación haciendo clic en "Consola":



10 Se abrirá la siguiente ventana:



11 Selecciona la tasa de baudios (velocidad de transmisión de datos) con la que hayas iniciado el puerto serie y después conéctalo.

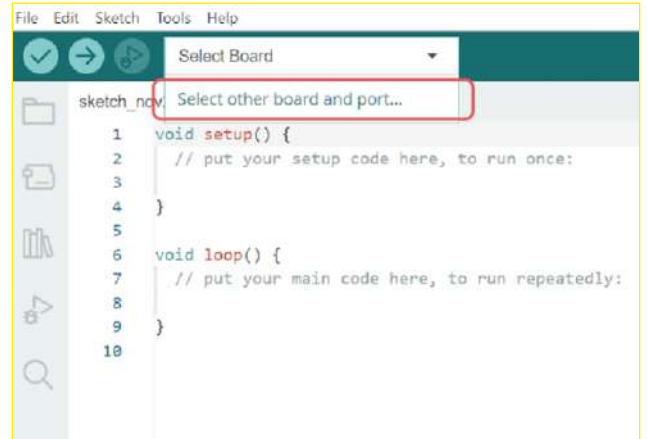


3.3.

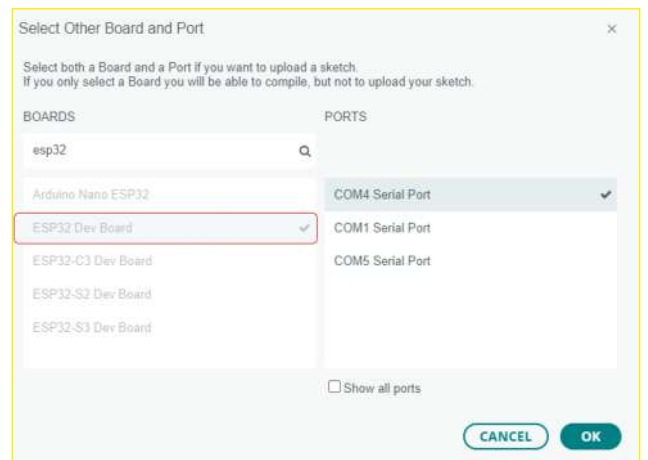
ESP32 STEAMAKERS

3.3.2. PROGRAMANDO CON ARDUINO IDE

- 1 Para programar la placa ESP32 STEAMakers al entorno Arduino IDE, hay que conectar la placa al ordenador con el cable USB y seleccionar el desplegable "Select Board".



- 2 Selecciona la placa "ESP32 Dev Board" en la ventana que se abre:



- 3 Una vez configurado el entorno, podemos probar a programar el envío en bucle de un mensaje en concreto por puerto serie.

El programa sería así:



- 4 Dentro de la estructura “setup()” añadimos la instrucción “Serial.begin(9600);” para configurar la velocidad de transferencia de dato (“baudios”) del puerto serie a 9.600. Y en el bucle principal, el “loop()”, añadimos la instrucción “Serial.println()”, para enviar un mensaje en concreto.

Cargamos el programa en la placa haciendo clic en el botón de carga:



- 5 Abrimos la consola serie para visualizar los datos que recibe el ordenador desde la placa mediante la comunicación serie:



¿TE ATREVES CON LOS RETOS?

Nivel I

¡MUESTRA UN MENSAJE!

Crea un programa que muestre para la consola serie de ArduinoBlocks un mensaje con la frase “Hola, soy ESP32” cada 2 segundos siempre. Cada mensaje debe aparecer en una línea diferente.

- Comienza el programa configurando el puerto serie a una tasa de baudios (velocidad de transferencia de datos) de 9.600. Coloca el bloque “Iniciar Baudios” de la categoría “Comunicaciones” → “Puerto serie” dentro de la estructura “Inicializar” para que se ejecute esta instrucción al inicio del programa y solo una vez.

La programación del bloque “Inicializar” te debería quedar así:



- Seguidamente, coloca dentro del “bucle” principal la estructura de programación “Ejecutar cada X ms” de la categoría “Tiempo” para indicar al programa cada cuánto quieres ejecutar una acción. Escribe “2000” en el parámetro de la estructura “Ejecutar cada X ms” para que se ejecute cada 2 segundos.

- Finalmente, dentro de esta estructura coloca la acción para llevarla a cabo en la frecuencia de tiempo especificada. En este caso consiste en enviar un mensaje por puerto serie. Seguidamente, selecciona el bloque “Enviar” de la categoría “Comunicaciones” → “Puerto serie”. Escribe dentro del bloque de texto el mensaje que quieres que se muestre y asegúrate de que esté marcada la opción “Salto de línea” para que cada vez que se muestre el mensaje por el monitor serie, sea una nueva línea.



4 Este es el programa completo:



5 Finalmente, descarga el programa en la placa ESP32 STEAMakers pulsando el botón “Sube” situado en la parte superior del editor.

Asegúrate de que la placa está conectada con cable USB al ordenador y abre la “Consola” serie para visualizar el envío del mensaje de forma periódica.



Nivel II

¿A QUÉ TEMPERATURA ESTAMOS?

En este segundo reto crea un programa con ArduinoBlocks que lea la temperatura ambiente mediante el sensor interno de la placa y muestre este valor por la consola serie cada 3 segundos.

- De nuevo, comienza el programa configurando el puerto serie en una tasa de baudios (velocidad de transferencia de datos) de 9.600. Coloca el bloque "Iniciar Baudios" de la categoría "Comunicaciones" → "Puerto serie" dentro de la estructura "Inicializar" para que se ejecute esta instrucción al inicio del programa solo una vez.

La programación del bloque "Inicializar" te debería quedar así:



- Seguidamente, coloca dentro del "bucle" principal la estructura de programación "Ejecutar cada X ms" de la categoría "Tiempo" para indicar al programa cada cuánto quieres ejecutar una acción. Escribe "3000" en el parámetro de la estructura "Ejecutar cada X ms" para que se ejecute cada 3 segundos.

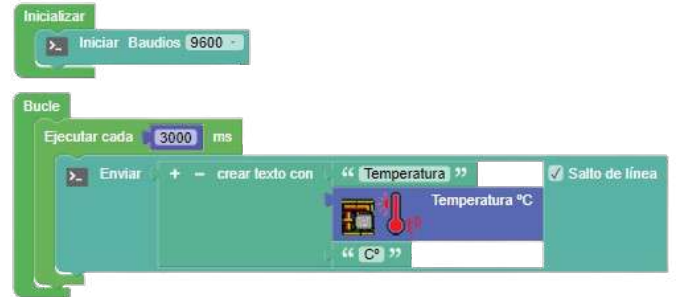
- Finalmente, dentro de esta estructura, coloca la acción para llevarla a cabo en la frecuencia de tiempo especificada. En este caso es enviar un mensaje por puerto serie. De nuevo, selecciona un bloque de la categoría "Comunicaciones" → "Puerto serie", esta vez es el bloque "Enviar".



- El mensaje a enviar puede constar de tres elementos: un texto que sea "Temperatura: ", el valor numérico que lea el sensor de temperatura y otro texto que sea " °C". Para poder colocar los tres bloques a la vez en la instrucción de enviar, necesitas el bloque "crear texto con" del apartado "Texto". En este mismo apartado encontrarás el bloque para escribir texto. El bloque para hacer la lectura del sensor de temperatura lo encontrarás en "Sensores" → "Integrados". Para acabar, asegúrate de que esté marcada la opción "Salto de línea", para que cada vez que se muestre el mensaje por el monitor serie sea en una línea nueva.



5 Este es el programa completo:



6 Finalmente, descarga el programa en la placa ESP32 STEAMakers pulsando el botón “Subir” situado en la parte superior del editor.

Asegúrate de que la placa esté conectada con cable USB al ordenador y abre la “Consola” serie para visualizar el envío de datos de forma periódica.

Nivel III

MONITOREA EL CONSUMO

En este último reto, crea un programa con ArduinoBlocks que lea y muestre por el valor de la intensidad, el voltaje y la energía consumida por la placa ESP32 STEAMakers cada 3 segundos por la consola serie.

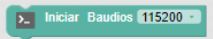

- 1 Utiliza los siguientes bloques de programación para realizar este reto:



POSIBLES SOLUCIONES


Nivel III


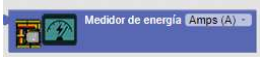







MONITOREA EL CONSUMO

- 1 De nuevo, comienza el programa configurando el puerto serie en una tasa de baudios (velocidad de transferencia de datos) de 9.600. Coloca el bloque "Iniciar Baudios"  de la categoría "Comunicaciones" → "Puerto serie" dentro de la estructura "Inicializar" , para que se ejecute esta instrucción al inicio del programa solo una vez.

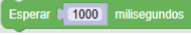
La programación del bloque "Inicializar" te debería quedar así:



- 2 Seguidamente, coloca dentro del "bucle" principal un conjunto de instrucciones de forma secuencial para mostrar un conjunto de datos a través de la consola serie. Para ello, utiliza el bloque "Enviar"  para puerto serie de la categoría "Comunicaciones" → "Puerto serie". Plantea este conjunto de datos que enviar y en este orden:

- Texto: "Intensidad: " (sin salto de línea) .
- Lectura Medidor de energía: Amps (A) (sin salto de línea) .
- Texto: "A" (con salto de línea) .
- Texto: "Voltaje: " (sin salto de línea) .
- Lectura medidor de energía: Volts (V) (sin salto de línea) .
- Texto: "V" (con salto de línea) .
- Texto: "Energía: " (sin salto de línea) .
- Lectura Medidor de energía: Energy (Wh) (sin salto de línea) .
- Texto: "Wh" (con salto de línea) .

- 3 Puedes escribir el texto directamente en el bloque "Enviar" del puerto serie, y el bloque para hacer la lectura del medidor de energía interno de la placa ESP32 STEAMakers lo encontrarás en "Sensores" → "Integrados", simplemente tienes que seleccionar qué magnitud quieres que se mida en cada caso.

- 4 Finalmente, coloca un último bloque bajo estas 9 instrucciones para que se pare el programa 3 segundos. Utiliza el bloque “Esperar X milisegundos”  de la categoría “Tiempos”.

Este es el programa completo:



- 5 Deberías ver una cosa similar a la imagen siguiente para la consola serie:



CONSEJOS DE SEGURIDAD

En el momento de utilizar la ESP32 STEAMakers, es fundamental seguir unas pautas de seguridad para garantizar tanto el buen funcionamiento de la placa como la seguridad de los usuarios. A continuación, se exponen diversos consejos clave:

- **Evita cortocircuitos:** Hay que asegurarse de que la ESP32 STEAMakers no esté alimentada cuando se manipulen materiales y herramientas conductores de electricidad. Eso ayuda a prevenir cortocircuitos inesperados que podrían dañar la placa o causar accidentes.
- **Manipula con cuidado la placa:** Los pines donde se conectan componentes diversos son delicados y pueden estropearse o doblarse si se aplica mucha fuerza o palanca. Hay que ser especialmente prudentes en la manipulación y evitar caídas o golpes que puedan afectar a la placa.
- **Controla la temperatura:** Es necesario tener en cuenta que la temperatura de la placa aumenta considerablemente durante un uso prolongado. Algunos componentes pueden calentarse y es importante evitar el contacto directo para prevenir quemaduras o daños en la placa.
- **Revisa regularmente los componentes:** Es conveniente comprobar con regularidad el estado de los componentes electrónicos y las conexiones. Los conectores sueltos o desgastados pueden causar un mal funcionamiento de la placa o riesgos de seguridad.
- **Usa adecuadamente la alimentación:** Utiliza solo las fuentes de alimentación recomendadas. Un voltaje incorrecto puede dañar la placa o incluso ser peligroso, ya que puede producir sobrecalentamiento de componentes o cortocircuitos.
- **Mantén la placa en un entorno seco y limpio:** La presencia de líquidos o polvo puede causar cortocircuitos o dañar los componentes de la placa.

Siguiendo estos consejos, el alumnado podrá disfrutar de una experiencia segura y enriquecedora con la placa programable.

CATEGORÍAS DE BLOQUES DEL ENTORNO DE PROGRAMACIÓN ARDUINOBLOCKS:

- **Lógica:** instrucciones para operaciones lógicas como SI, SI NO, y operaciones booleanas.
- **Control:** bloques que permiten controlar la ejecución del programa, como bucles e interrupciones.
- **Matemática:** incluye bloques para realizar operaciones matemáticas como sumas, restas, multiplicaciones y divisiones.
- **Texto:** funciones para la manipulación de cadenas de texto y caracteres.
 - **JSON:** instrucciones para la manipulación de datos en formato JSON.
- **Variables:** permite la creación y gestión de variables.
- **Listas:** instrucciones para la manipulación de listas y vectores.
- **Funciones:** incluye bloques para la creación, la llamada y el uso de funciones personalizadas.
- **ESP:** funciones específicas para la comunicación con módulos ESP8266/ESP32.
- **Tiempo:** permite la gestión del tiempo y los temporizadores.
 - **Multi-Tarea:** funcionalidades para la implementación de tareas simultáneas.
- **Entrada/Salida:** instrucciones para la manipulación de entradas y salidas.
 - **I2C:** instrucciones para la comunicación a través del protocolo I2C.
- **Sensores:** incluye diversos bloques para la lectura de diferentes sensores.
 - **Integrados:** funciones para la utilización de sensores integrados en placas controladoras.
 - **Receptor IR:** permite la lectura de señales infrarrojas.
 - **Acelerómetros:** incluye funciones relacionadas con la lectura de aceleración.
 - **Color:** proporciona instrucciones para la lectura de sensores de color.
 - **Gestos:** permite la interpretación de gestos utilizando sensores específicos.
- **Actuadores:** bloques para el control de dispositivos que realizan acciones.
- **Motor:** funciones para la gestión de motores en general.
 - **Servo:** instrucciones específicas para la manipulación de servomotores.

Paso a paso: permite controlar motores paso a paso.

DC motor: funciones para el control de motores de corriente continua.

- **Periféricos:** funciones para la comunicación con periféricos externos.

GPS: instrucciones para la lectura de datos de sistemas de posicionamiento global.

Reloj RTC: permite el acceso y la manipulación de un reloj de tiempo real.

RFID: funciones para la lectura de tarjetas RFID.

RFID (I2C): instrucciones específicas para la comunicación I2C con dispositivos RFID.

MP3: permite la reproducción y el control de ficheros de audio MP3.

Keypad: incluye bloques para la lectura de teclados numéricos.

Keypad-TTP229: funciones específicas para el teclado capacitivo TTP229.

- **Visualización:** instrucciones para la presentación de datos visualmente.

Pantalla LCD: permite la interacción con pantallas de LCD.

Pantalla OLED: funciones para el uso de pantallas OLED.

LedMatrix 8x8: incluye instrucciones para la manipulación de matrices LED 8x8.

NeoPixel: permite el control de ledes RGB NeoPixel.

- **Comunicaciones:** funciones para la transmisión de datos entre dispositivos.

Puerto serie: permite la comunicación serie con otros dispositivos.

Bluetooth: incluye bloques para la comunicación *bluetooth*.

Teclado BLE: funciones para la conexión con teclados *bluetooth*.

Ratón BLE: permite la interacción con ratones *bluetooth*.

WiFi / IoT: instrucciones específicas para la comunicación wifi y IoT.

MQTT Cliente: permite la comunicación mediante el protocolo MQTT.

HTTP Cliente: instrucciones para la realización de peticiones HTTP.

HTTP Servidor: funciones para la creación de servidores HTTP.

HTML: incluye instrucciones para la generación de código HTML.

Alexa (Philips Hue): permite la integración con asistentes de voz como Alexa.

Telegram Bot: funciones para la interacción con Telegram a través de bots.

NTP Client: instrucciones para la sincronización de tiempo con servidores NTP.

Blynk IoT: proporciona funcionalidades para la integración con la plataforma Blynk IoT.

Blynk (legacy): incluye funciones para la compatibilidad con versiones anteriores de Blynk.

WiFi-Mesh: funciones para la creación de redes wifi mesh.

DMX: permite la comunicación con dispositivos DMX para control de iluminación.

- **LoRa / IoT:** funciones para la comunicación a través de la tecnología LoRa.

LoRa: instrucciones para la comunicación mediante LoRa.

LoRa-Wan: funciones específicas para la implementación de redes LoRa-Wan.

- **Tarjeta SD:** instrucciones para la lectura y escritura de datos en tarjetas SD.
- **EEPROM:** permite el acceso a la memoria EEPROM para el almacenaje persistente de datos.