

**Manual de Iniciación en el Uso y Aplicaciones Básicas de la Tarjeta ESP32.
Introduction Manual for the Use and Basic Applications of the ESP32 Card.**

Angela Patricia Punguil Eras, Jose Ismael Rojas Bueno, Mario David Guillen Gavilanes, Edison Fernando Herrera Núñez.

**CIENCIA E INNOVACIÓN EN
DIVERSAS DISCIPLINAS
CIENTÍFICAS.**

**Julio - Diciembre, V°5-N°2;
2024**

- ✓ **Recibido:** 05/07/2024
- ✓ **Aceptado:** 14/07/2024
- ✓ **Publicado:** 31/12/2024

PAIS

- Santo Domingo, Ecuador
- Santo Domingo, Ecuador
- Santo Domingo, Ecuador
- Santo Domingo, Ecuador

INSTITUCIÓN:

- Instituto Superior Tecnológico Tsachila
- Instituto Superior Tecnológico Tsachila
- Instituto Superior Tecnológico Tsachila
- Instituto Superior Tecnológico Tsachila

CORREO:

- ✉ angelapunguileras@tsachila.edu.ec
- ✉ joserobasbueno@tsachila.edu.ec
- ✉ marioguillen@tsachila.edu.ec
- ✉ edisonherrera@tsachila.edu.ec

ORCID:

- 🌐 <https://orcid.org/0009-0004-3683-1997>
- 🌐 <https://orcid.org/0009-0007-0812-9289>
- 🌐 <https://orcid.org/0000-0002-7942-1961>
- 🌐 <https://orcid.org/0000-0001-6095-0823>

FORMATO DE CITA APA.

Punguil, A. Rojas, J. Guillen, M. Herrera, E. (2024). Manual de Iniciación en el Uso y Aplicaciones Básicas de la Tarjeta ESP32. Revista G-ner@ndo, V°5 (N°2). 15-46.

Resumen

La tecnología actual ofrece múltiples beneficios tanto en el ámbito residencial como en el industrial. La Cuarta Revolución Industrial (I 4.0) ha impulsado la necesidad de dispositivos con microcontroladores avanzados y capacidad de conexión remota, lo que obliga a los desarrolladores a adoptar y dominar nuevas tecnologías. Este artículo tiene como objetivo proporcionar un manual de iniciación para el manejo, familiarización y aplicaciones básicas de la tarjeta de desarrollo ESP32. Se presenta una revisión del estado del arte, destacando las características del ESP32 en comparación con otras tarjetas de desarrollo. Se incluyen definiciones y conceptos básicos del entorno de desarrollo, así como una guía paso a paso para la implementación de proyectos básicos prácticos. Los proyectos ejemplos ilustran el uso del ESP32 en aplicaciones de control remoto y automatización. Los resultados demuestran la versatilidad y eficiencia de esta tarjeta en diversas aplicaciones, incluyendo la utilización del Arduino IoT Cloud, una de las funciones más destacadas del ESP32.

Palabras clave: Manual, ESP32, Aplicaciones, Arduino IoT Cloud

Abstract

Current technology offers multiple benefits in both residential and industrial settings. The Fourth Industrial Revolution (I 4.0) has driven the need for devices with advanced microcontrollers and remote connection capabilities, forcing developers to adopt and master new technologies. This article aims to provide a starter manual for the handling, familiarization and basic applications of the ESP32 development board. A review of the state of the art is presented, highlighting the features of the ESP32 in comparison to other development boards. Basic definitions and concepts of the development environment are included, as well as a step-by-step guide to implementing basic practical projects. The example projects illustrate the use of the ESP32 in remote control and automation applications. The results demonstrate the versatility and efficiency of this card in various applications, including the use of the Arduino IoT Cloud, one of the most notable functions of the ESP32.

Keywords: Manual, ESP32, Applications, Arduino IoT Cloud

Introducción

En un mundo cada vez más interconectado, la tecnología ha transformado numerosos aspectos de la vida cotidiana, desde el hogar hasta el entorno industrial. La Cuarta Revolución Industrial, conocida también como Industria 4.0, ha impulsado la integración de tecnologías avanzadas, lo que ha generado una mayor demanda de dispositivos inteligentes y conectados. Dentro de este contexto, los microcontroladores juegan un papel fundamental, actuando como el cerebro de una variedad de sistemas automatizados.

El ESP32, una de las más recientes y versátiles tarjetas de desarrollo, se ha destacado por sus capacidades avanzadas y su flexibilidad. Este microcontrolador de código abierto no solo incluye módulos de comunicación Wi-Fi y Bluetooth, sino que también posee un procesador de doble núcleo, pudiendo operar a frecuencias entre 80 MHz y 240 MHz de manera independiente (Systems, 2024). Estas características hacen del ESP32 una herramienta poderosa y adaptable tanto para aficionados como para profesionales en el campo de la tecnología.

Además, el ESP32 se basa en un diseño de hardware flexible que permite tanto su ensamblaje como su adquisición preensamblada. Su compatibilidad con entornos de desarrollo populares, como Arduino IDE, facilita su programación y configuración. La tarjeta ofrece una variedad de periféricos configurables, como puertos de entrada y salida de propósito general (GPIO), lo que amplía sus posibles aplicaciones. En este manual, se presenta una guía de iniciación para el uso del ESP32, dirigida a estudiantes y desarrolladores que buscan familiarizarse con esta herramienta. A través de este documento, se proporcionarán definiciones y conceptos básicos sobre el entorno de desarrollo, se compararán las características del ESP32 con otras tarjetas de desarrollo, y se presentarán proyectos prácticos que demuestran su aplicación en el control remoto y la automatización. También se destacará el uso del Arduino IoT Cloud, una funcionalidad esencial que potencia las capacidades del ESP32 en el ámbito del Internet de las Cosas (IoT).

La elección del tema "Manual de Iniciación en el Uso y Aplicaciones Básicas de la Tarjeta ESP32" responde a la necesidad creciente de formar a estudiantes, profesionales y entusiastas de la tecnología en el uso de dispositivos IoT (Internet de las Cosas). El ESP32 es una tarjeta de desarrollo de microcontroladores que ha ganado popularidad por su versatilidad, bajo costo y capacidad de conexión Wi-Fi y Bluetooth. Este manual tiene como objetivo proporcionar una guía completa y accesible para principiantes, facilitando la comprensión y aplicación de esta tecnología en proyectos diversos.

El ESP32 es una herramienta poderosa en el campo de la educación tecnológica. Su integración en el currículo de formación técnica y universitaria permitirá a los estudiantes desarrollar competencias clave en programación, electrónica y desarrollo de soluciones IoT. Según Pishva y Takeda (2020), la educación en tecnología y programación es esencial para preparar a los estudiantes para el futuro laboral en un mundo cada vez más digitalizado.

El mercado laboral actual demanda profesionales con habilidades en IoT, dado el crecimiento exponencial de dispositivos conectados y la necesidad de soluciones innovadoras en sectores como la domótica, la salud, la industria y el medio ambiente. De acuerdo con Statista (2021), se espera que el número de dispositivos IoT conectados alcance los 30.9 mil millones para 2025, lo que subraya la importancia de formar profesionales capacitados en esta área.

El ESP32 es una opción económica y accesible para la mayoría de los estudiantes y profesionales, permitiendo la realización de proyectos a bajo costo sin comprometer la calidad o funcionalidad. Además, promueve la sostenibilidad al incentivar el desarrollo de soluciones tecnológicas eficientes y responsables con el medio ambiente. Wang y Su (2020) destacan la importancia de soluciones tecnológicas sostenibles en la educación y la industria.

La facilidad de uso y las amplias capacidades del ESP32 estimulan la innovación y el emprendimiento. Los usuarios del manual estarán equipados para crear prototipos y desarrollar

productos innovadores, contribuyendo al crecimiento de startups y la creación de nuevas oportunidades de negocio. Según Christensen et al. (2019), la innovación tecnológica es un motor clave para el desarrollo económico y la creación de empleo.

y Materiales

La metodología aplicada en este artículo es de tipo investigativa-descriptiva con un enfoque práctico, basado en los siguientes componentes:

- 1.- Revisión del estado del arte
- 2.- Entorno de programación
- 3.- Programación e implementación de proyectos prácticos
- 4.-Evaluación de resultados

La metodología para la elaboración del "Manual de Iniciación en el Uso y Aplicaciones Básicas de la Tarjeta ESP32" se basará en un enfoque mixto, combinando la investigación documental y la experimentación práctica. En primer lugar, se realizará una revisión exhaustiva de la literatura existente sobre el ESP32, incluyendo libros, artículos académicos, y recursos en línea, para identificar los conceptos y aplicaciones más relevantes. Paralelamente, se llevarán a cabo una serie de experimentos y proyectos prácticos utilizando la tarjeta ESP32, documentando cada paso y resultado para garantizar que el manual sea comprensible y útil para principiantes. Además, se realizarán encuestas y entrevistas a usuarios novatos y expertos en la tecnología ESP32 para recoger sus experiencias y sugerencias, lo que permitirá ajustar y mejorar el contenido del manual. Finalmente, se validará el manual mediante pruebas piloto con un grupo de estudiantes y aficionados a la tecnología, recopilando retroalimentación para realizar ajustes finales antes de su publicación.

Revisión del estado del arte:

- Se realizó una investigación sobre las características y capacidades del ESP32, comparándolo con otras tarjetas de desarrollo populares.
- Se revisaron publicaciones académicas, documentación técnica y recursos en línea para obtener una comprensión detallada de las ventajas y desventajas del ESP32.

Para el desarrollo del artículo se requirieron los siguientes materiales:

1.- Tarjeta de desarrollo ESP32: Versión W-Room de 38 pines utilizada para la implementación y prueba de los proyectos prácticos.

2.- Computadora personal: Para la programación y configuración del ESP32.

3.-Software de programación:

Arduino IDE: Entorno de desarrollo integrado utilizado para escribir y cargar código en el ESP32.

Arduino IoT Cloud: Plataforma utilizada para la creación y gestión de proyectos de IoT con ESP32.

4.- Componentes electrónicos básicos:

Resistencias, Leds, cables y otros componentes auxiliares.

5.- Conexión a Internet: Necesaria para la configuración y uso del Arduino IoT Cloud.

6. Documentación y recursos en línea: Guías, tutoriales y foros de soporte relacionados con el uso del ESP32 y el Arduino IoT Cloud.

Tabla 1. *Técnicas y características.*

Características	ESP32	Raspberry Pi	Arduino Uno
-----------------	-------	--------------	-------------

Procesador	Doble núcleo Tensilica LX6 (80-240 MHz) (Systems, 2024)	ARM Cortex-A53/A72 (64 bits) (Pi, 2024)	ATmega328P (8 bits) (Arduino, 2020)
Memoria	520 KB SRAM, SPI flash externo	512 MB - 8 GB RAM, tarjeta microSD	2 KB SRAM, 32 KB flash, 1 KB EEPROM
Conectividad	Wi-Fi 802.11 b/g/n, Bluetooth 4.2	Ethernet, Wi-Fi, Bluetooth, USB	Requiere módulos adicionales
Entradas y Salidas	34 pines GPIO, ADCs 12 bits, DACs, SPI, I2C, UART	40 pines GPIO, HDMI, cámaras, audio, SPI, I2C, UART	14 pines digitales, 6 pines analógicos, SPI, I2C, UART
Consumo de Energía	Modos de bajo consumo	N/A	Bajo
Programación	Arduino IDE, Espressif IDF, Micropython	Python, C, C++, Java, varios otros	Arduino IDE
Uso	IoT, control remoto, automatización	Aplicaciones complejas, servidores, sistemas multimedia	Proyectos educativos, prototipos simples
Disponibilidad en el mercado nacional	Si	Si	Si
Precio (Se consideran precios del mercado nacional ecuatoriano)	\$12 -\$20	\$180 -\$250	\$10-\$25

Nota; Como se puede observar la ESP32 tiene varios periféricos cada uno con características particulares, en la siguiente tabla se muestra de manera resumida las especificaciones técnicas y características.

Análisis de Ventajas y Desventajas del ESP32

Esta revisión del estado del arte proporciona una base para comprender el contexto y las capacidades del ESP32, permitiendo tomar decisiones informadas en el desarrollo de proyectos específicos que utilicen esta tecnología, para lo cual se comparó con tarjetas de desarrollo Rasperry Pi y Arduino Uno.

Ventajas del ESP32:

Conectividad: Integración nativa de Wi-Fi y Bluetooth, ideal para proyectos de IoT.

Costo: Mucho más asequible en comparación con la Rasperry Pi, especialmente considerando sus capacidades de conectividad.

Flexibilidad en la programación: Compatible con varios entornos de desarrollo como Arduino IDE y Micropython.

Desventajas del ESP32:

Capacidad de procesamiento: Aunque adecuado para muchas aplicaciones de IoT y automatización, el procesador es menos potente en comparación con la Rasperry Pi, lo que lo hace menos adecuado para aplicaciones complejas.

Memoria: Menos memoria en comparación con la Rasperry Pi, limitando su uso en aplicaciones que requieren grandes cantidades de RAM.

Entradas y Salidas: Aunque tiene una amplia variedad de pines GPIO, no es tan versátil como la Rasperry Pi en términos de interfaces multimedia como HDMI y cámaras.

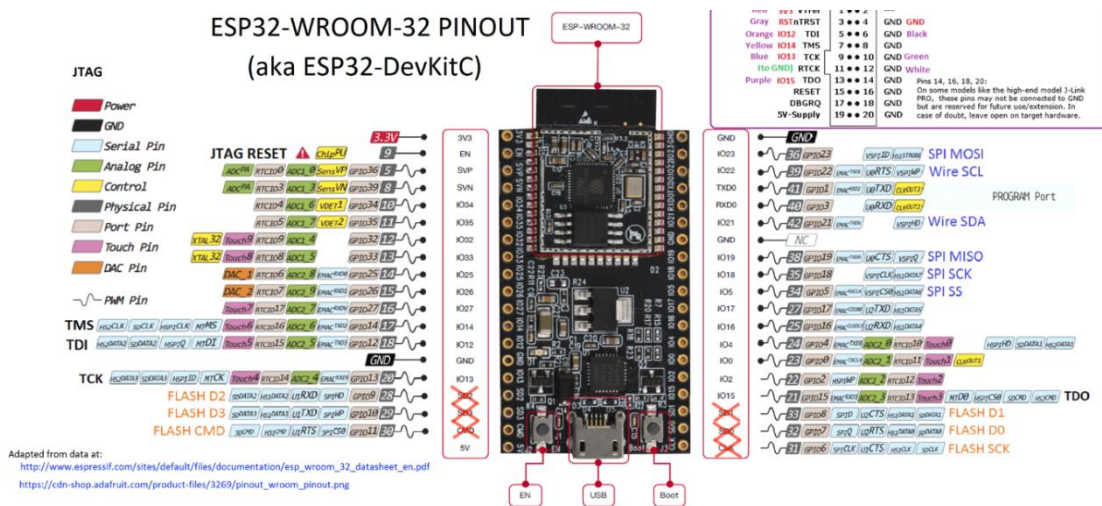
Especificaciones técnicas Esp 32

La placa de desarrollo ESP32 es ideal para prototipado de proyectos, con un conversor USB-serial CP2102, reguladores de voltaje y LEDs indicadores, soportando WiFi, Bluetooth y

BLE (Bluetooth Low Energy). Su CPU de 32 bits y dos núcleos de hasta 240MHz, junto con numerosos periféricos, permite la conexión con sensores táctiles, interfaces para tarjeta SD, Ethernet, SPI, UART, I2S e I2C. Es adecuado para mini servidores web, procesamiento digital, webcams, cámaras IP, robótica móvil y domótica, 32. Incluye memoria FLASH, cristal oscilador y antena WiFi en PCB.

ESP32 está diseñado para protoboards y puede alimentarse por micro-USB o fuentes externas de 5V o 3.3V, recomendándose una fuente de 5VDC/1A y un capacitor de 100uF para filtrar picos de corriente. Los pines GPIO trabajan a 3.3V, requiriendo conversores de nivel para sistemas de 5V. (Systems, 2024). La plataforma ESP32 permite desarrollo en varios lenguajes y frameworks: Arduino (C++), MicroPython, LUA, Esp-idf, Simba, RTOS (Zephyr, Mongoose OS, NuttX), Javascript (Espruino, Duktape, Mongoose JS) y Basic. El entorno Arduino ofrece un lenguaje conocido y un IDE sencillo, con amplia información y soporte comunitario. Entre las principales placas basadas en ESP32 están: ESP32-WROOM-32 -38, NodeMCU-32 ESP32 y ESP32-CAM, (Systems, 2024)

Figura 1. pin out (disposición de los pines) de la tarjeta ESP32 WRoom de 38 pines.



Como se puede observar la ESP32 tiene varios periféricos cada uno con características particulares, en la siguiente tabla se muestra de manera resumida las especificaciones técnicas y características.

Tabla 2. ESP32

Característica	Descripción
Tipo	Módulo Wifi + Bluetooth
Modelo	ESP32 38 Pines
Chip USB-Serial	CP2102
Voltaje de Alimentación (microUSB)	5V DC
Entradas/Salidas	3.3V DC
Consumo de energía	5 μ A en modo de suspensión
Microcontrolador	Dual Core Tensilica LX6 240 MHz
Frecuencia de Reloj	240Mhz
Procesador secundario	Permite hacer operaciones básicas en modo de ultra bajo consumo
SRAM	520 KB
Memoria flash	4MB
Módulo wifi incorporado	802.11 BGN HT40 (2.4 GHz hasta 150 Mbit/s)
Seguridad Wi-Fi	WEP, WPA/WPA2 PSK/Enterprise, AES / SHA2 / Criptografía de curva elíptica / RSA-4096
Bluetooth	4.2 BR/EDR BLE
Modo de control dual	si
Memoria	448 KByte ROM, 520 KByte SRAM, 6 KByte SRAM en RTC y QSPI admite múltiples chips flash / SRAM
Antena en PCB	si
Puertos GPIO	30 cables GPIO, que incluyen: 3x UART, 3x SPI, 2x I2C (2x I2S)
Convertidor ADC	12 canales

Convertidor DAC	2 canales
Salidas PWM	si
Interfaz de tarjeta SD	si
Dimensiones	55 x 28 x 8 mm (sin conectores)
Peso	11g

Entorno de programación:

Un entorno de programación es un conjunto de herramientas y software que permiten escribir, probar y depurar el código que se ejecutará en el microcontrolador. Este entorno facilita el proceso de desarrollo de software al proporcionar una interfaz amigable y diversas funcionalidades que simplifican las tareas del programador. (Red Hat, 2023)

En este apartado, se presentan diferentes entornos de programación compatibles con la tarjeta ESP32.

Arduino IDE

El Arduino IDE es uno de los entornos de programación más populares para el desarrollo de proyectos con microcontroladores, especialmente con placas Arduino y ESP32. Es conocido por su simplicidad y facilidad de uso, lo que lo hace ideal tanto para principiantes como para desarrolladores experimentados. (Xakata, 2022)

- Características Principales:
 - Editor de código con resaltado de sintaxis y autocompletado.
 - Compilador integrado para cargar código directamente en las placas compatibles.
 - Gran cantidad de bibliotecas y ejemplos de código disponibles.
 - Compatible con una amplia gama de placas de desarrollo, incluyendo Arduino y ESP32.
 - Plataforma multiplataforma (Windows, macOS, Linux).
 - Usos Comunes:
 - Proyectos de IoT, automatización, robótica, y prototipos educativos.
-

Visual Studio Code (VS Code)

Visual Studio Code es un editor de código fuente desarrollado por Microsoft, ampliamente utilizado por desarrolladores de software en diversas disciplinas. Su extensibilidad y personalización lo hacen adecuado para el desarrollo de proyectos con microcontroladores, incluyendo ESP32. (Webinars, 2022)

- Características Principales:

- Editor de código avanzado con soporte para múltiples lenguajes de programación.
- Integración con el terminal para ejecutar comandos y scripts.
- Extensiones y plugins para añadir funcionalidad adicional (por ejemplo, PlatformIO para desarrollo con microcontroladores).
- Depurador integrado y herramientas de control de versiones.
- Compatible con múltiples sistemas operativos (Windows, macOS, Linux).

- Usos Comunes:

- Desarrollo de software general, proyectos de IoT, aplicaciones web, y sistemas embebidos.

PlatformIO

PlatformIO es un entorno de desarrollo integral para sistemas embebidos y IoT que se integra como una extensión en editores de código como Visual Studio Code. Ofrece un conjunto de herramientas avanzadas para el desarrollo, compilación, y depuración de código en una amplia variedad de plataformas de hardware. (Gotoiot, 2021)

- Características Principales:

- Compatibilidad con más de 800 plataformas de desarrollo, incluyendo ESP32, Arduino, STM32, y más.
 - Gestión de bibliotecas y dependencias simplificada.
-

- Herramientas de compilación y depuración avanzadas.
- Integración con sistemas de control de versiones y plataformas de CI/CD.
- Soporte para múltiples entornos de desarrollo y lenguajes de programación.
 - Usos Comunes:
 - Proyectos de IoT, sistemas embebidos, desarrollo de firmware, y aplicaciones industriales.

Arduino Cloud

Arduino Cloud no es un entorno de programación tradicional, pero complementa y se integra con herramientas como el Arduino IDE para ofrecer una solución completa para el desarrollo y gestión de proyectos IoT, proporcionando una interfaz gráfica, dashboards (tableros) personalizados y funcionalidades de automatización y monitoreo. (Arduino Cloud, 2024)

Análisis de ventajas y desventajas del entorno de programación Arduino IDE.

Se hizo una comparativa entre tres entornos de desarrollo, Arduino IDE, Visual Studio Code y PlatformIO para tener una base sólida de las diferentes IDEs.

Ventajas de Arduino IDE

El Arduino IDE tiene varias ventajas sobre otras IDEs, especialmente para principiantes y proyectos simples. Su mayor ventaja es su facilidad de uso y su interfaz simple, que lo hace ideal para aquellos que recién comienzan en la programación de microcontroladores. Además, permite trabajar con placas Arduino y ESP32 (con la instalación de las librerías), lo que facilita la configuración y la programación de estas placas. Otra ventaja es la gran cantidad de bibliotecas y ejemplos de código disponibles, lo que facilita el desarrollo de proyectos sin tener que escribir todo el código desde cero.

Desventajas de Arduino IDE

Arduino IDE también tiene algunas desventajas en comparación con otras IDEs más avanzadas. Una de las principales es la falta de funcionalidades, como herramientas de depuración más avanzadas o soporte para lenguajes de programación más complejos. También

puede resultar menos eficiente para proyectos grandes que requieren una gestión más avanzada del código y de los recursos. Para trabajar con el entorno de programación arduino IDE y configuración del ESP32, se deben realizar los siguientes pasos:

Instalación del Arduino IDE:

Figura 2. *Arduino*

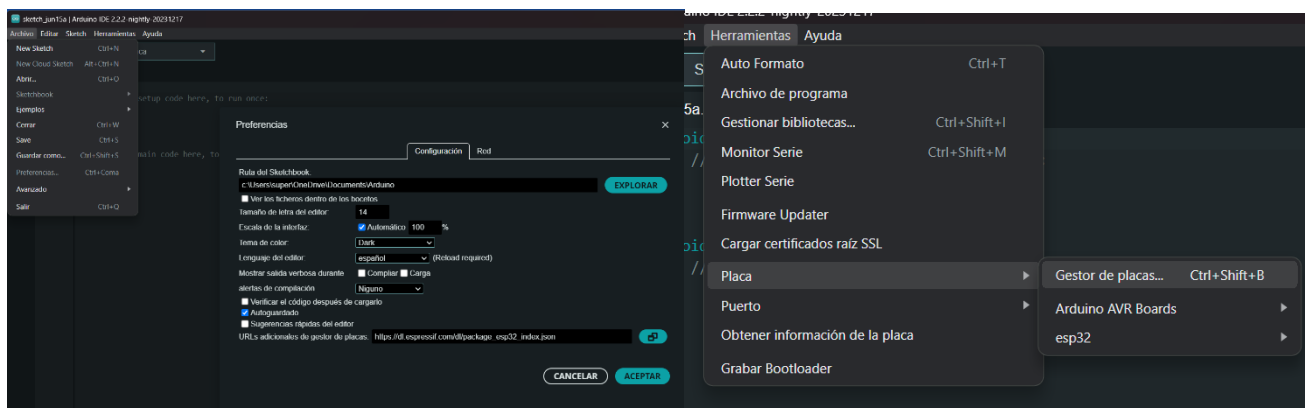


Funete: https://dl.espressif.com/dl/package_esp32_index.json

Configuración del entorno para soportar la placa ESP32:

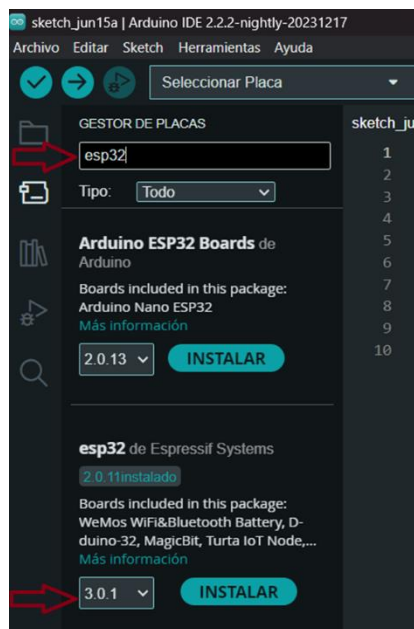
Una vez instalado y ejecutado el software arduino IDE, ingresar a la pestaña “Archivo” > “Preferencias” > “Configuración” > “URLs adicionales de gestor de placas”,

Figura 3. *Ir a “Herramientas” > “Placa” > “Gestor de placas”*



Fuente: https://dl.espressif.com/dl/package_esp32_index.json

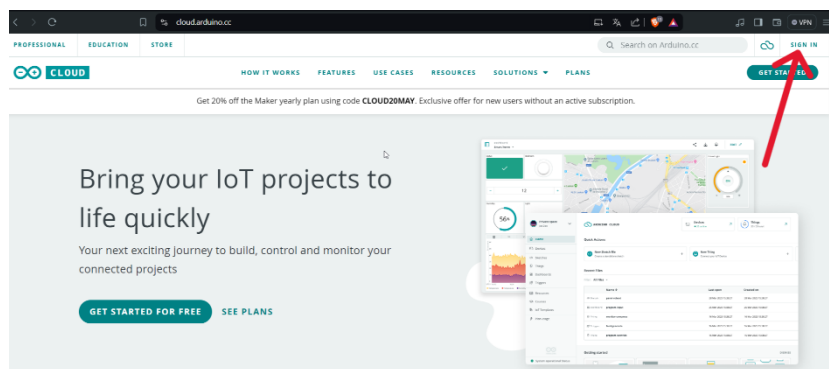
Figura 4. Buscar “esp32” e instalar el soporte para ESP32.



Registro y configuración en el Arduino Cloud:

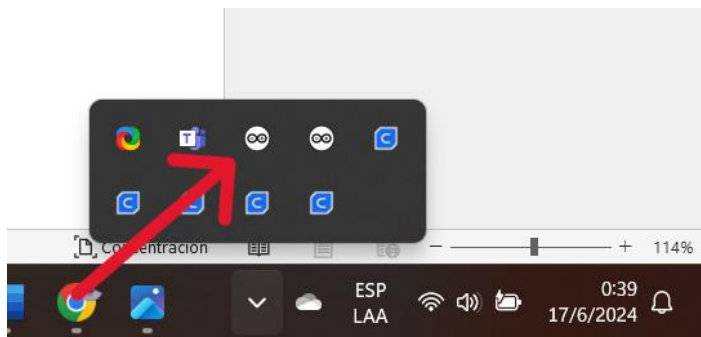
Una de las principales características de la ESP32 son los protocolos de comunicación de tipo inalámbrico bluetooth y wifi, siendo este último el que da la característica de tarjeta de desarrollo para el uso de internet de las cosas (IoT). A continuación, se presenta los pasos para el uso de esta característica mediante la plataforma Arduino Cloud: Se crea una cuenta en la plataforma Arduino Cloud ingresando al siguiente enlace:

Figura 5. Arduino Cloud



Fuente: <https://cloud.arduino.cc/>

Una vez ingresado a la plataforma se debe instalar un software (Arduino Create Agent) agente de comunicación entre la tarjeta esp32 y la computadora, el cual se puede descargar del siguiente enlace:



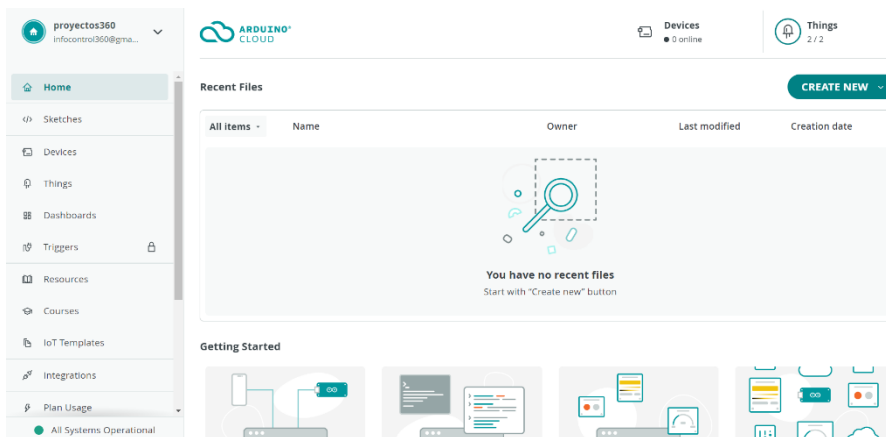
Fuente: <https://support.arduino.cc/hc/en-us/articles/360014869820-Install-the-Arduino-Create-Agent>

Se verifica una instalación exitosa al poder visualizar este icono en la barra de herramientas de Windows. Por consiguiente, se conecta la tarjeta ESP32 y se ingresa a la plataforma Arduino Cloud, en la interfaz de la plataforma, se observan 3 pestañas:

Sketches, Device, Things, Dashboards, entre otras a continuación se explica la función de las opciones mencionadas:

- Sketches: Aquí se gestionan y editan los códigos (sketches) para los dispositivos. Se pueden crear, modificar y guardar programas para ser cargados en las placas Arduino.
 - Devices: Esta pestaña permite gestionar y supervisar los dispositivos ESP32 conectados a la nube. Se pueden agregar, configurar y monitorear las placas Arduino y otros dispositivos compatibles.
 - Things: En esta sección se crean y administran los "Things" (cosas), que son representaciones virtuales de los dispositivos físicos. Aquí se configuran las variables y se enlazan con los dispositivos para recopilar datos y controlar funciones.
 - Dashboards: Aquí se diseñan y visualizan paneles de control personalizados para monitorizar y controlar los dispositivos en tiempo real. Los usuarios pueden crear gráficos,
-

botones y otras interfaces interactivas (widgets) para visualizar los datos de los dispositivos conectados.



Nota: En caso de no reconocer la tarjeta en la plataforma Arduino Cloud se sugiere instalar el driver CP210x USB to UART Bridge VCP Drivers, para la comunicación con el puerto USB serial, se adjunta el enlace de descarga. <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads>

3.- Programación e implementación de proyectos prácticos

En esta sección, se presentan ejemplos prácticos que demuestran el uso del ESP32 en diversas aplicaciones. Estos ejemplos están diseñados para introducir al usuario a las funciones básicas del ESP32, proporcionando una comprensión sólida de cómo programar y utilizar esta tarjeta de desarrollo en proyectos reales. A continuación, se explican paso a paso las funciones y las líneas de código ingresadas en el programa como comentario para lo cual se utiliza el símbolo //, además se simula el circuito utilizando la plataforma Wokwi:

Proyecto 1: Hola Mundo ISTT

Este proyecto es el primer paso para familiarizarse con el ESP32 y el entorno de desarrollo Arduino IDE. Consiste en imprimir un mensaje en el monitor serial.

Paso 1: Configuración del Entorno de Desarrollo

Abrir el Arduino IDE. Ir a Herramientas -> Placa -> Gestor de tarjetas y buscar "ESP32" para instalar el soporte de tarjetas ESP32.

Seleccionar la tarjeta ESP32 (por ejemplo, "ESP32 Dev Module") desde Herramientas -> Placa.

Paso 2: Escribir el Código

```
void setup() {  
    // Iniciar la comunicación serial a 115200 baudios  
    // Esta línea configura la velocidad de comunicación serial entre el ESP32 y el  
    monitor serial.  
    // El valor 115200 es una tasa de baudios comúnmente utilizada para transmisión  
    rápida de datos.  
    Serial.begin(115200);  
}  
  
void loop() {  
    // Imprimir "Hola Mundo" en el monitor serial  
    // Esta línea envía el mensaje "Hola Mundo" al monitor serial, que puede ser visto en  
    el Arduino IDE  
    // en la pestaña del Monitor Serial.  
    Serial.println("Hola Mundo ISTT");  
  
    // Esperar 1 segundo  
    // Esta línea pausa la ejecución del programa durante 1000 milisegundos (1  
    segundo).  
    // El uso de delay permite que el mensaje "Hola Mundo" se imprima cada segundo.  
    delay(1000);  
}
```

Explicación del código:

Serial.begin(115200): Esta línea inicializa la comunicación serial entre el ESP32 y la computadora. El valor 115200 especifica la velocidad de comunicación en baudios (bits por segundo).

Este ajuste es importante para asegurarse de que tanto el ESP32 como el monitor serial se comuniquen a la misma velocidad.

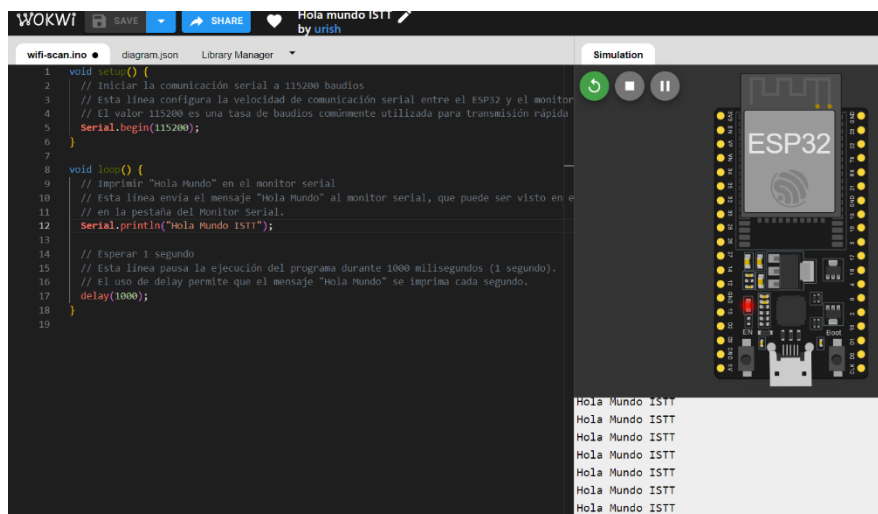
Serial.println("Hola Mundo");

Serial.println es una función que envía datos al puerto serial y añade un salto de línea al final del mensaje. "Hola Mundo" es la cadena de caracteres que se envía al monitor serial. Este mensaje se puede ver en el monitor serial del Arduino IDE, permitiendo verificar que el ESP32 está funcionando correctamente.

delay(1000);

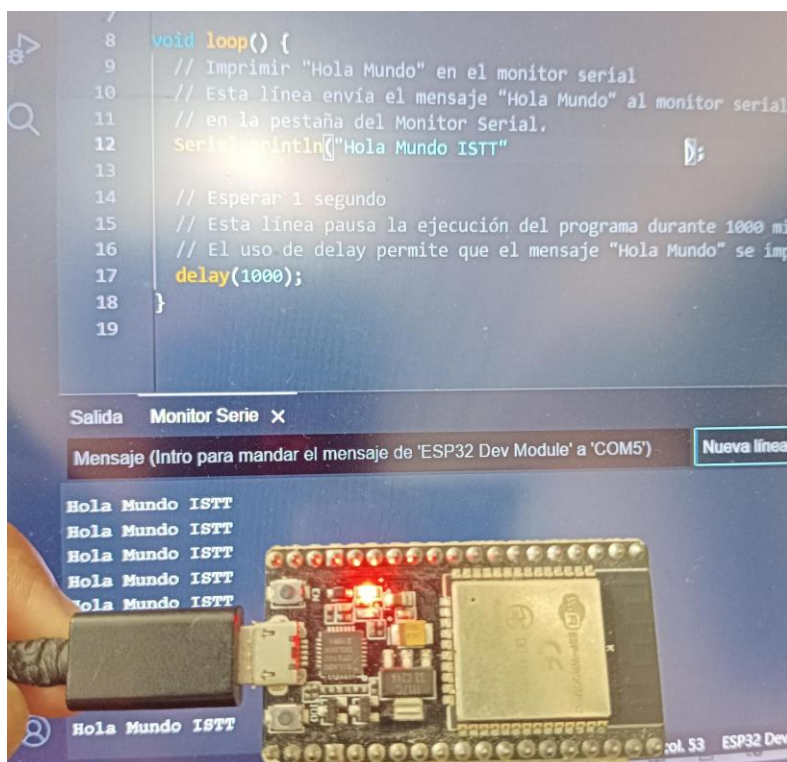
delay es una función que detiene la ejecución del programa durante un tiempo especificado en milisegundos. 1000 milisegundos equivalen a 1 segundo.

Este retraso hace que el mensaje "Hola Mundo" se imprima en el monitor serial una vez por segundo, proporcionando una cadencia regular a la salida de datos. A continuación, se muestra la simulación "Hola mundo ISTT", realizado en la plataforma WOKWI, se adjunta el enlace del software online para la simulación: <https://wokwi.com/esp32>



Paso 3: Subir el Código al ESP32

- Conectar el ESP32 a la computadora mediante un cable USB.
- Seleccionar el puerto correcto desde Herramientas -> Puerto.
- Subir el código haciendo clic en el botón Subir que en la tarjeta Esp32, está en la parte derecha del puerto USB, con las palabras boot o IOO.



Proyecto 2: Semáforo

Este proyecto simula un semáforo utilizando LEDs de tres colores (rojo, amarillo y verde) conectados al ESP32.

Paso 1: Conexión

Conectar el LED rojo al pin GPIO 15.

Conectar el LED amarillo al pin GPIO 16.

Conectar el LED verde al pin GPIO 17.

Conectar las resistencias de 220 ohm en serie con cada LED.

Conectar el cátodo de cada LED al GND del ESP32.

Paso 2: Escribir el Código

```
// Definir pines para los LEDs
// Estos son los números de pines en el ESP32 a los que están conectados los LEDs rojo, amarillo y verde.
const int ledRojo = 15;
const int ledAmarillo = 16;
const int ledVerde = 17;
void setup() {
  // Configurar pines como salida
  // Estos comandos configuran los pines ledRojo, ledAmarillo y ledVerde como salidas.
  // Esto significa que el ESP32 puede enviar señales de encendido o apagado a estos pines.
  pinMode(ledRojo, OUTPUT);
  pinMode(ledAmarillo, OUTPUT);
  pinMode(ledVerde, OUTPUT);
}
void loop() {
  // Encender LED rojo, apagar otros
  // Enciende el LED rojo estableciendo el pin ledRojo en HIGH (alta).
  // Apaga los LEDs amarillo y verde estableciendo sus pines en LOW (baja).
  digitalWrite(ledRojo, HIGH);
  digitalWrite(ledAmarillo, LOW);
  digitalWrite(ledVerde, LOW);
  // Esperar 3 segundos
  // Pausa la ejecución del programa durante 3000 milisegundos (3 segundos).
  delay(3000);
  // Encender LED amarillo, apagar otros
  // Enciende el LED amarillo estableciendo el pin ledAmarillo en HIGH (alta).
  // Apaga los LEDs rojo y verde estableciendo sus pines en LOW (baja).
  digitalWrite(ledRojo, LOW);
  digitalWrite(ledAmarillo, HIGH);
  digitalWrite(ledVerde, LOW);
  // Esperar 1 segundo
  // Pausa la ejecución del programa durante 1000 milisegundos (1 segundo).
  delay(1000);
  // Encender LED verde, apagar otros
  // Enciende el LED verde estableciendo el pin ledVerde en HIGH (alta).
  // Apaga los LEDs rojo y amarillo estableciendo sus pines en LOW (baja).
  digitalWrite(ledRojo, LOW);
  digitalWrite(ledAmarillo, LOW);
  digitalWrite(ledVerde, HIGH);
  // Esperar 3 segundos
  // Pausa la ejecución del programa durante 3000 milisegundos (3 segundos).
  delay(3000);
}
```

Explicación del código:

`const int ledRojo = 15; const int ledAmarillo = 16; const int ledVerde = 17;`

Estas líneas definen tres constantes enteras que representan los números de los pines del ESP32 a los que están conectados los LEDs rojo, amarillo y verde.

`const` indica que el valor de estas variables no cambiará durante la ejecución del programa.

`pinMode(ledRojo, OUTPUT); pinMode(ledAmarillo, OUTPUT); pinMode(ledVerde, OUTPUT);`

Estas líneas configuran los pines de los LEDs como salidas.

pinMode es una función que configura el modo de un pin específico, en este caso, como OUTPUT (salida).

```
digitalWrite(ledRojo, HIGH); digitalWrite(ledAmarillo, LOW); digitalWrite(ledVerde, LOW);
```

digitalWrite es una función que establece el estado de un pin digital a HIGH (alto) o LOW (bajo).

```
digitalWrite(ledRojo, HIGH);
```

 enciende el LED rojo.

```
digitalWrite(ledAmarillo, LOW);
```

 y

```
digitalWrite(ledVerde, LOW);
```

 apagan los LEDs amarillo y verde.

```
delay(3000); delay(1000);
```

delay es una función que pausa la ejecución del programa durante un número específico de milisegundos.

```
delay(3000);
```

 pausa el programa durante 3000 milisegundos (3 segundos).

```
delay(1000);
```

 pausa el programa durante 1000 milisegundos (1 segundo).

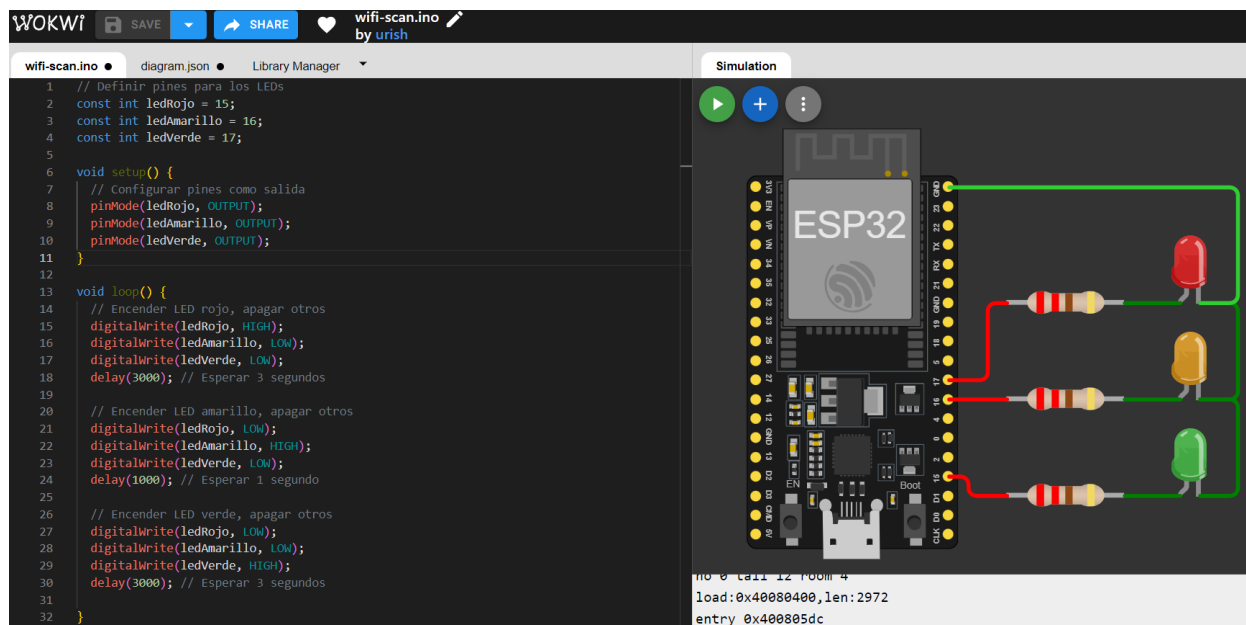
El código en loop se repite continuamente, encendiendo y apagando los LEDs en una secuencia que simula el funcionamiento de un semáforo:

LED rojo encendido por 3 segundos.

LED amarillo encendido por 1 segundo.

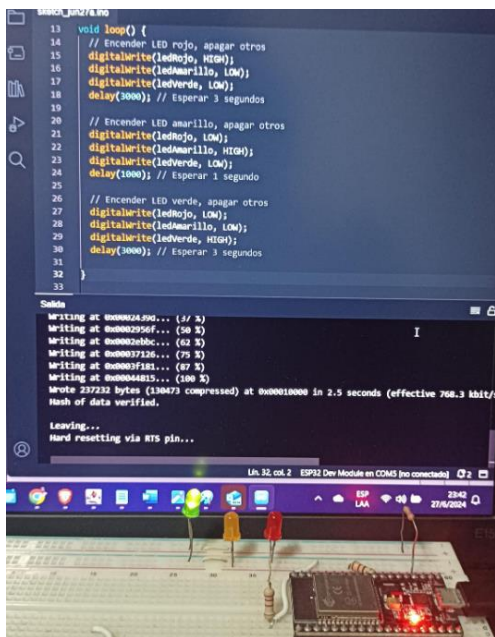
LED verde encendido por 3 segundos.

A continuación, se muestra la simulación “semáforo”, realizado en la plataforma WOKWI, se adjunta el enlace del software online para la simulación: <https://wokwi.com/esp32>



Paso 3: Subir el Código al ESP32

- Conectar el ESP32 a la computadora mediante un cable USB.
- Seleccionar el puerto correcto desde Herramientas -> Puerto.
- Subir el código haciendo clic en el botón Subir.



Proyecto 3: Activación remota led (Introducción al internet de las cosas)

En este proyecto, se explora cómo controlar un LED de manera remota utilizando la plataforma Arduino IoT Cloud, aprovechando las capacidades del ESP32 para la automatización y el control desde cualquier ubicación con conexión a internet.

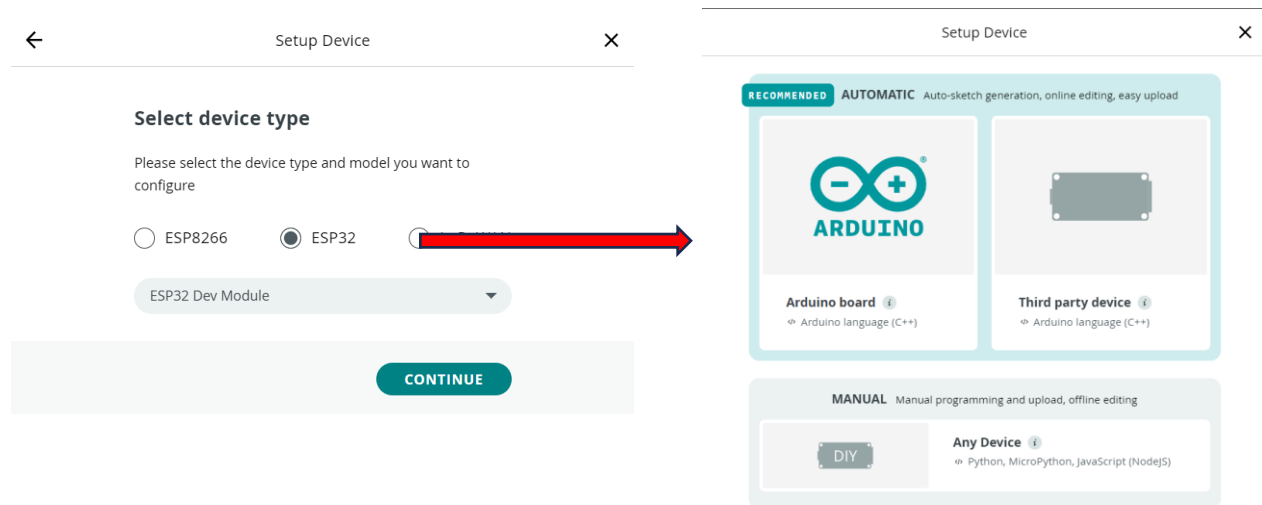
Paso 1: Configuración del Entorno

Configuración Inicial:

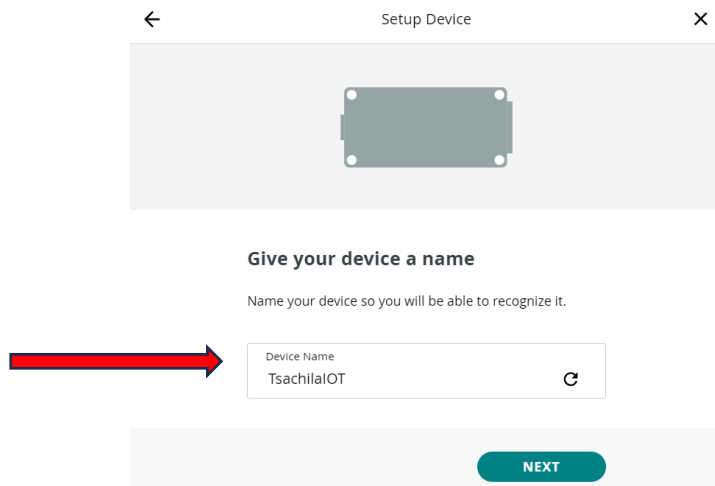
Acceder a la plataforma Arduino IoT Cloud, asegurarse de haber creado una cuenta y haber instalado el Arduino Create Agent en la computadora además dependiendo del integrado de la comunicación del puerto serial, instalar el driver respectivo mencionado en el apartado Arduino Cloud y siguiendo los pasos previamente mencionados en el artículo.

Se configura el dispositivo (Devices) que permitirá controlar las cosas (Things) de manera remota. En este caso el dispositivo que dará la característica IoT a la cosa (led) será la esp32, siendo esta un dispositivo de otro fabricante se debe crear c

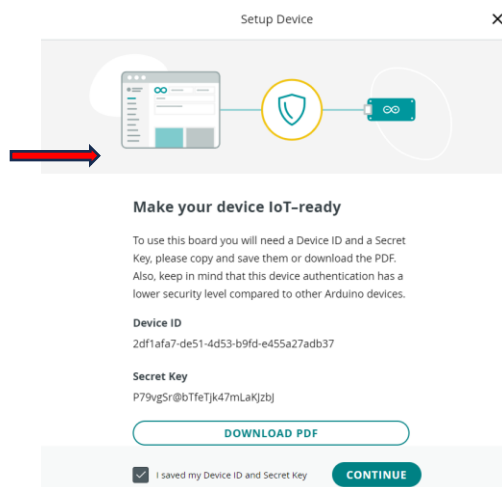
omo un tercer fabricante (third device) y seleccionar (Dev Module) .



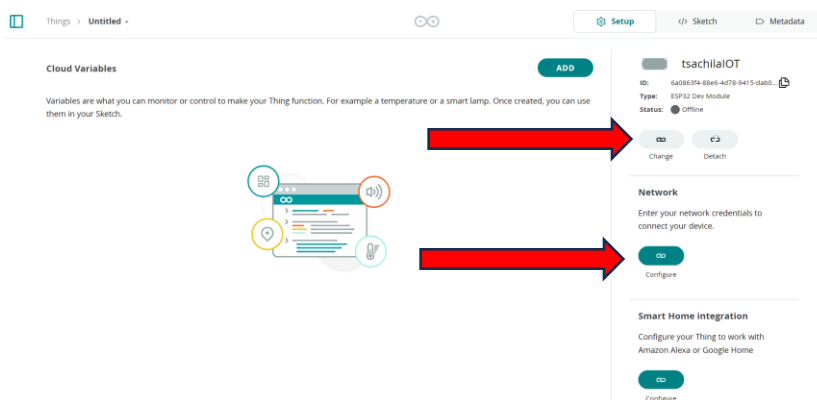
El software asigna un nombre aleatorio al dispositivo, el mismo que puede ser editado, tal como se muestra en la imagen.



Se necesitará una ID de dispositivo y una clave secreta; los cuales se recomienda copiar y guardar o descargar el PDF. Además, se debe tener en cuenta que la autenticación de ESP32 tiene un nivel de seguridad más bajo en comparación con otros dispositivos Arduino.



En la pestaña Things (cosas) se vincula el dispositivo previamente creado, además se configura la red de internet con la que se va a trabajar y se ingresa el Secret Key antes proporcionado.



Things > Untitled

Cloud Variables

Variables are what you can monitor or control to make your Thing function. For example a temperature or a smart lamp. Once created, you can use them in your Sketch.

ADD

tsachilaIoT

ID: 6a0863f4-88e6-4d7b-9415-dab0-1b

Type: ESP32 Dev Module

Status: Offline

Change Detach

Network

Enter your network credentials to connect your device.

Configure

Smart Home integration

Configure your Thing to work with Amazon Alexa or Google Home

Configure

Configure network

Enter your network credentials to allow your device to connect to the Cloud.

Wi-Fi Name *

redinternet

Password

.....

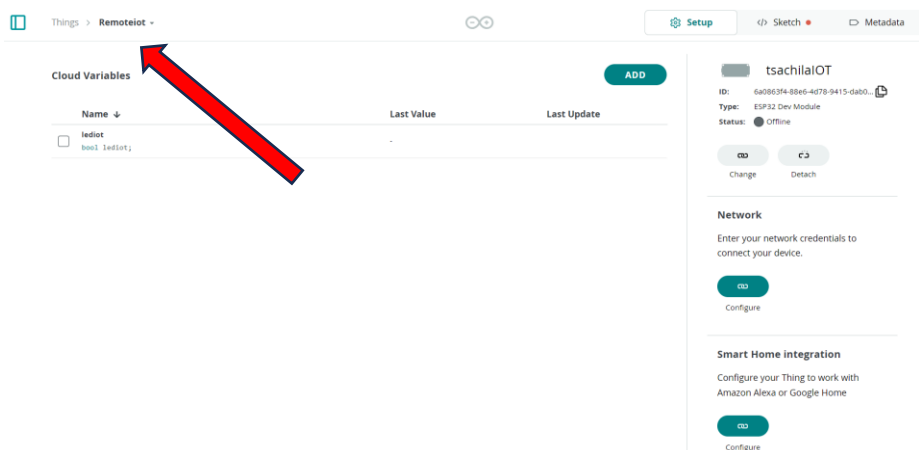
Secret Key *

.....

IMPORTANT: Remember to go to the "Sketch" tab and upload the sketch to load the credentials on the board.

SAVE

Se asigna un nombre al proyecto y se agrega la variable de la nube que va a interactuar con el dispositivo actuador o la variable física.



Things > Remoteiot

Cloud Variables

ADD

Name	Last Value	Last Update
lediot		
bool lediot1		

tsachilaIoT

ID: 6a0863f4-88e6-4d7b-9415-dab0-1b

Type: ESP32 Dev Module

Status: Offline

Change Detach

Network

Enter your network credentials to connect your device.

Configure

Smart Home integration

Configure your Thing to work with Amazon Alexa or Google Home

Configure

Add variable

×

Name

lediot

Sync with other Things

Boolean eg. true

Declaration

`bool lediot;`

Variable Permission

☒ Read & Write

☐ Read Only

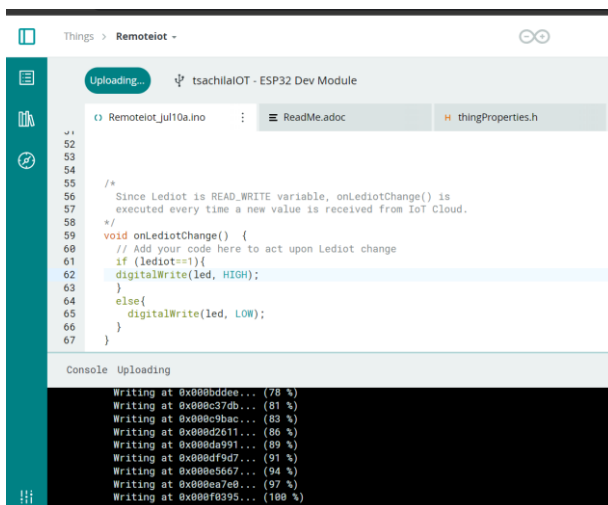
Variable Update Policy

CANCEL

ADD VARIABLE

Paso 1: Programación

Una vez creada la variable se ingresa al apartado Sketch, donde se realiza la programación para el control de encendido del elemento actuador (led). Al igual que en la plataforma Arduino IDE, se procede a compilar y cargar el programa.

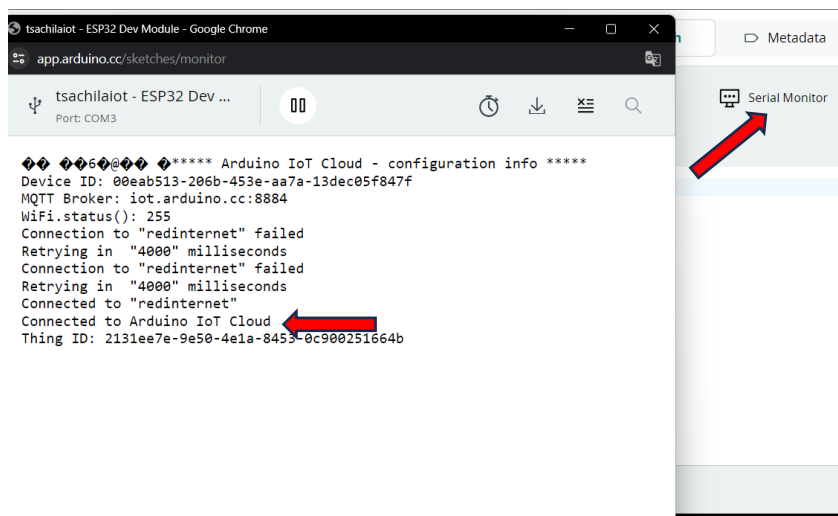


```
52
53
54
55
56  /*
57   Since lediot is READ_WRITE variable, onLedioteChange() is
58   executed every time a new value is received from IoT Cloud.
59  */
60  void onLedioteChange() {
61    // Add your code here to act upon lediot change
62    if (lediot==1){
63      digitalWrite(led, HIGH);
64    }
65    else{
66      digitalWrite(led, LOW);
67    }
68  }
```

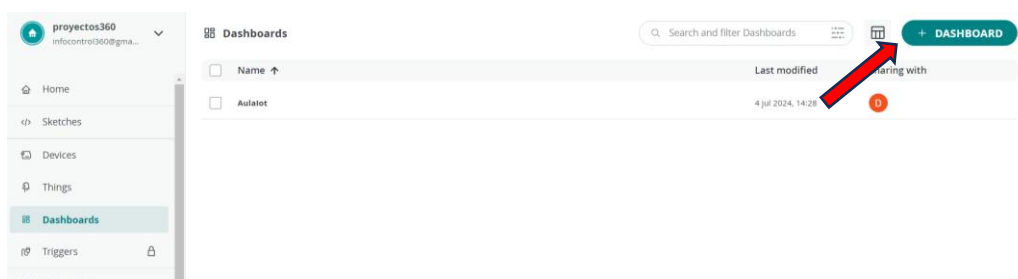
Console Uploading

```
Writing at 0x000b0d0e... (78 %)
Writing at 0x000c37db... (81 %)
Writing at 0x000c9bac... (83 %)
Writing at 0x000d2611... (86 %)
Writing at 0x000da991... (89 %)
Writing at 0x000df9d7... (91 %)
Writing at 0x000e5667... (94 %)
Writing at 0x000ea7e0... (97 %)
Writing at 0x000f0395... (100 %)
```

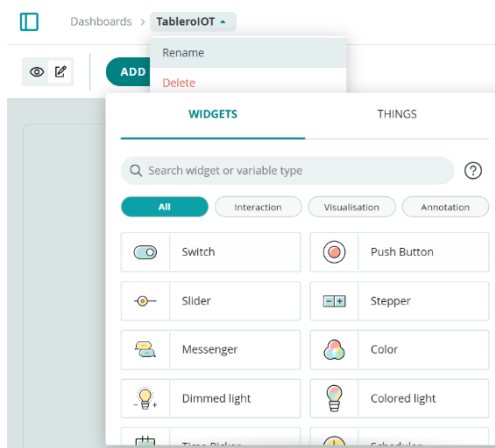
Una vez cargado se verifica la conexión de la tarjeta a la red en Serial Monitor.



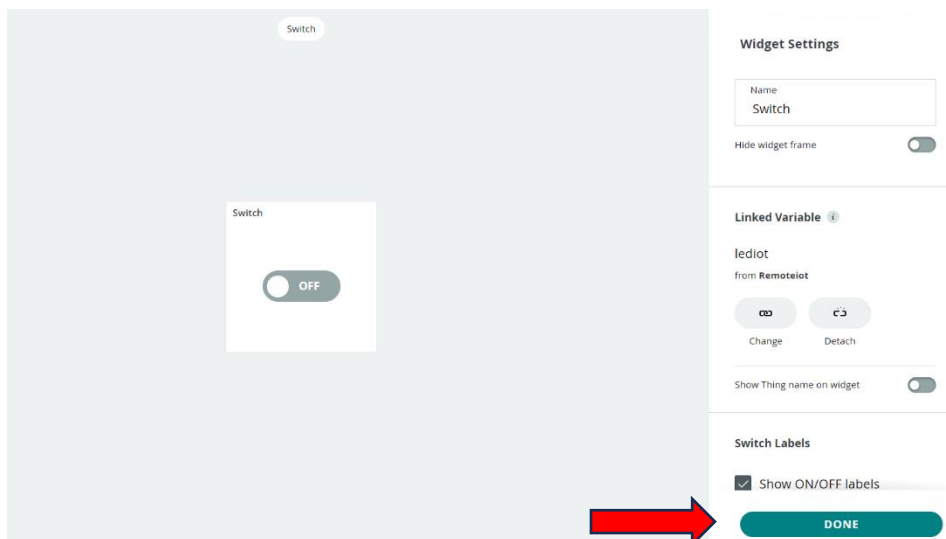
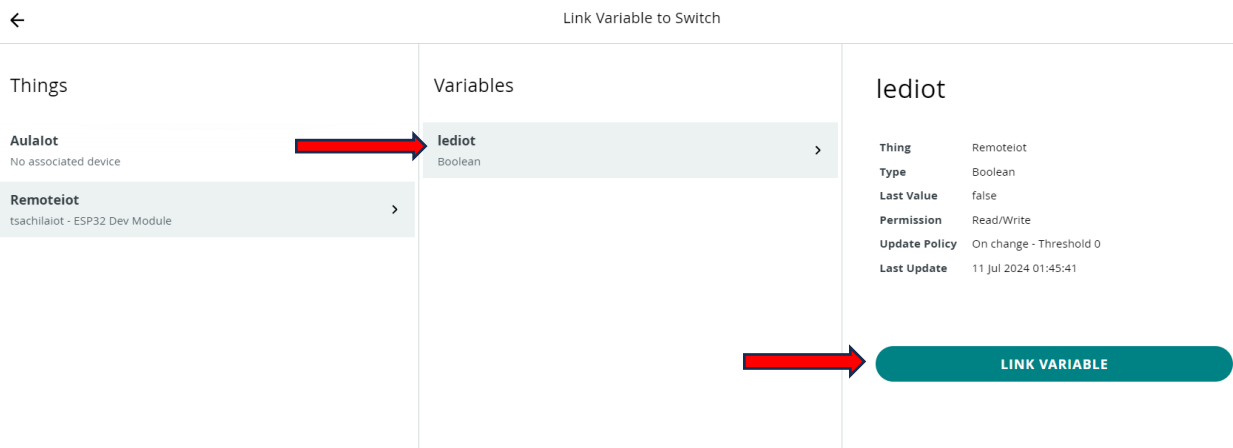
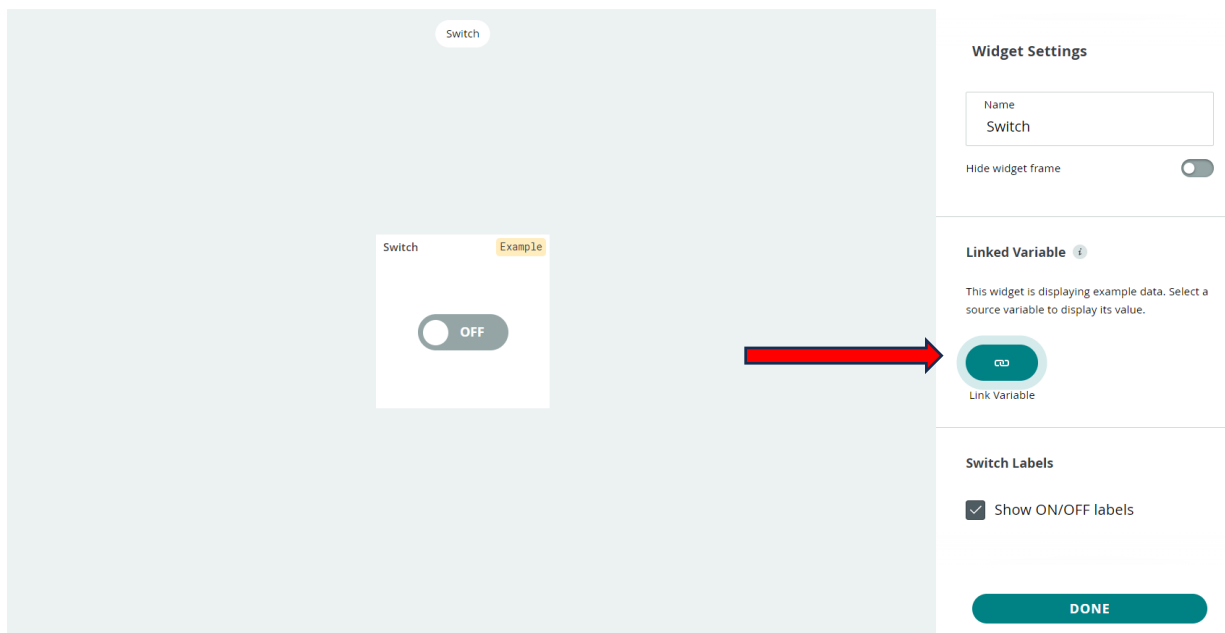
Para finalizar se procede a crear el dashboard para dentro poner un widget, al seleccionar DASHBOARD se puede asignar un nombre referente al tablero digital que corresponda.



Un widget es un elemento digital que interactúa con la variable física, se puede utilizar varios widgets para la misma variable. En este caso se pretende encender el led, vinculado a la variable cloud (lediot) y la variable vinculada al pin de salida de la ESP32 (led).

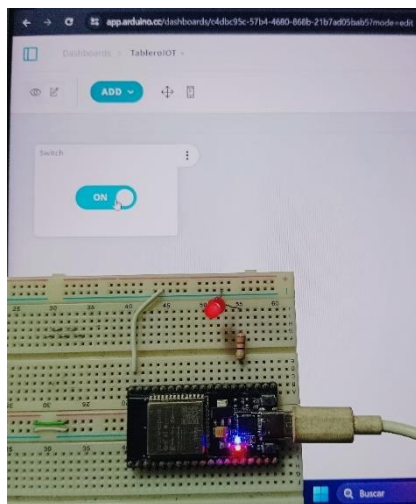


Se vincula el widget a la variable IoT tal como se muestra a continuación.



Paso 3: Implementación en protoboard del circuito.

Finalmente se verifica la comunicación entre la plataforma Arduino cloud y ESP32 al activar o desactivar el elemento Widget Switch, tal como se muestra en la siguiente figura.



Análisis de Resultados

Se proporciona una evaluación crítica de los proyectos prácticos implementados con la tarjeta ESP32, así como una comparación de sus características y desempeño con otras plataformas de desarrollo populares como Raspberry Pi y Arduino Uno. A continuación, se detallan los puntos clave de este análisis:

Evaluación de Proyectos Prácticos

Proyecto 1: Hola Mundo ISTT

Implementación: Este proyecto básico fue exitosamente implementado utilizando el entorno de desarrollo Arduino IDE con soporte para ESP32. El código se ejecutó correctamente y se logró la comunicación serial para imprimir "Hola Mundo" en el monitor. Resultados: La tarjeta ESP32 demostró su facilidad de uso y compatibilidad con el entorno de desarrollo estándar de Arduino, proporcionando una buena introducción para principiantes.

Proyecto 2: Semáforo

Implementación: Se diseñó y simuló un semáforo utilizando LEDs de tres colores conectados al ESP32. El código escrito en Arduino IDE permitió controlar los LEDs en secuencias adecuadas para simular el funcionamiento de un semáforo real.

Resultados: El ESP32 mostró su capacidad para manejar múltiples salidas digitales de manera efectiva, demostrando su versatilidad y adecuación para proyectos de automatización y control.

Proyecto 3: Control de Dispositivos

Implementación: En este proyecto se implementó el control remoto de un dispositivo mediante WiFi utilizando el ESP32. El entorno de desarrollo utilizado fue el Arduino IDE, y el proyecto consistió en encender y apagar un LED de manera remota a través de una interfaz web.

Resultados: El ESP32 demostró una rápida respuesta y una conectividad confiable, destacándose en aplicaciones de control remoto y domótica. Comparado con el Arduino Uno, el ESP32 ofrece capacidades de conectividad WiFi integradas, lo que lo hace más adecuado para aplicaciones IoT.

Comparativa General

ESP32 vs. Raspberry Pi: Aunque el ESP32 ofrece una excelente conectividad inalámbrica y es más económico, la Raspberry Pi supera al ESP32 en capacidad de procesamiento y memoria, lo que la hace más adecuada para proyectos que requieren mayor potencia de cálculo y capacidades multimedia.

ESP32 vs. Arduino Uno: El ESP32 ofrece capacidades superiores en términos de conectividad inalámbrica y manejo de múltiples entradas y salidas, siendo más adecuado para proyectos IoT y de automatización. El Arduino Uno, por otro lado, sigue siendo una excelente

opción para proyectos más simples y educativos debido a su simplicidad y amplia documentación.

En resumen, el ESP32 es una herramienta versátil y eficiente para una amplia gama de aplicaciones, desde proyectos educativos hasta prototipos industriales, especialmente en el campo del Internet de las Cosas (IoT).

Conclusiones

El ESP32 se ha consolidado como una opción potente y económica para desarrolladores que buscan integrar capacidades de conectividad inalámbrica en sus proyectos. Su compatibilidad con entornos de desarrollo populares y su flexibilidad en la programación facilitan su adopción tanto por aficionados como por profesionales. Los proyectos prácticos realizados demostraron la capacidad del ESP32 para controlar dispositivos de manera efectiva, destacando su potencial en aplicaciones de domótica, monitoreo ambiental y automatización. La integración con la plataforma Arduino IoT Cloud potencia aún más estas capacidades, proporcionando un entorno de desarrollo robusto y accesible para la creación de soluciones IoT. Las limitaciones del ESP32 en términos de capacidad de procesamiento y memoria no deben desanimar su uso, ya que sus ventajas en conectividad y costo lo hacen ideal para una gran variedad de aplicaciones. El ESP32 es especialmente adecuado para proyectos que requieren conectividad WiFi y Bluetooth, y donde el consumo energético es una consideración importante. El futuro del desarrollo con ESP32 promete aplicaciones aún más innovadoras y sofisticadas, extendiéndose desde la educación hasta la industria, y facilitando la creación de soluciones tecnológicas avanzadas que contribuyen a la evolución de la Industria 4.0 y el Internet de las Cosas.

Referencia bibliográfica

Arduino. (2020). Obtenido de <https://arduino.cl/arduino-uno/>

Arduino Cloud. (2024). Obtenido de <https://cloud.arduino.cc/>

Gotoiot. (02 de 2021). Obtenido de https://www.gotoiot.com/pages/articles/platformio_vscode_usage/index.html

Pi, R. (2024). Raspberry Pi. Obtenido de <https://www.raspberrypi.com/documentation/computers/getting-started.html>

Red Hat. (31 de 07 de 2023). Obtenido de <https://www.redhat.com/es/topics/middleware/what-is-ide>

Systems, E. (2024). Espressif. Obtenido de https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Tostatronic. (2022). Obtenido de <https://www.tostatronic.com/product/esp32-19-pines-bluetooth-wifi/>

Webinars, O. (julio de 2022). Obtenido de <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>

Xakata. (09 de 2022). Obtenido de <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

Christensen, C. M., Raynor, M. E., & McDonald, R. (2019). What is disruptive innovation? Harvard Business Review. Recuperado de <https://hbr.org/2019/12/what-is-disruptive-innovation>

Pishva, D., & Takeda, K. (2020). Teaching coding in higher education: Enhancing students' problem-solving skills. Journal of Educational Technology & Society, 23(4), 44-56.

Statista. (2021). Number of connected IoT devices worldwide 2019-2025. Recuperado de <https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/>

Wang, H., & Su, Y. (2020). Sustainable development in education and technology: A roadmap. Journal of Cleaner Production, 250, 119-139.
