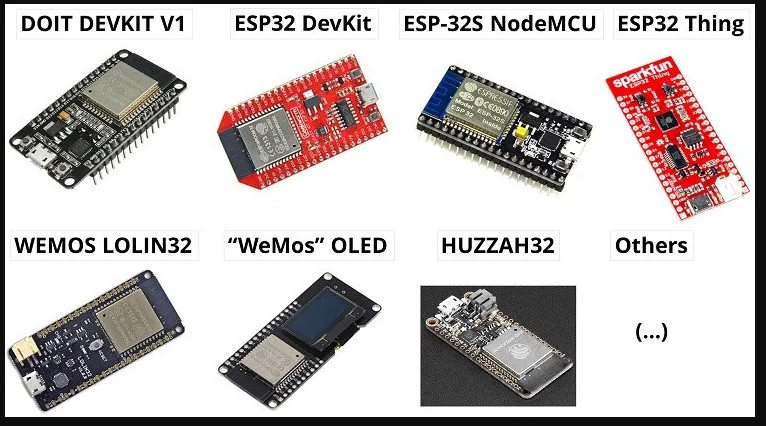
**INFORME SOBRE SOC ESP32**

**Especificación del modulo**

* ‎El ESP32 es de doble núcleo, esto significa que tiene 2 procesadores.‎
* ‎Tiene Wi-Fi y bluetooth incorporados.‎
* ‎Ejecuta programas de 32 bits.‎
* ‎La frecuencia de reloj puede ir hasta 240MHz y tiene una RAM de 512 kB.‎
* ‎Esta placa en particular tiene 30 o 36 pines, 15 en cada fila.‎
* ‎También tiene una amplia variedad de periféricos disponibles, como: táctil capacitiva, ADC, DAC, UART, SPI, I2C y mucho más.‎
* ‎Viene con sensor de efecto hall incorporado y sensor de temperatura incorporado

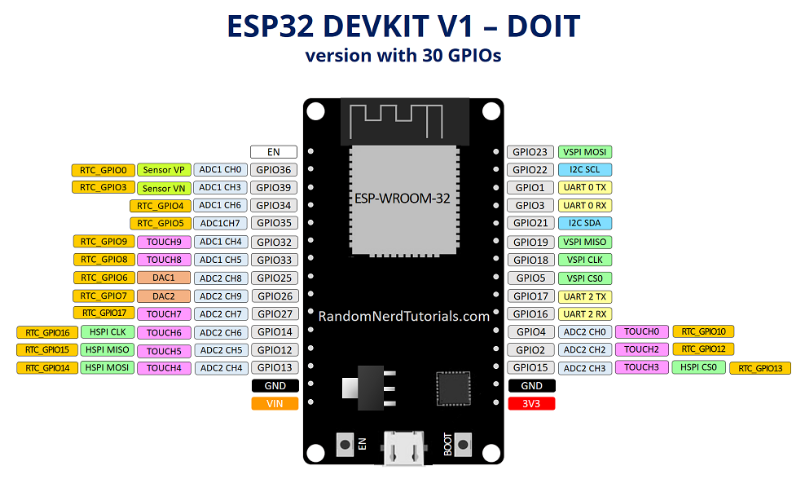
**Tipos de versiones**

****

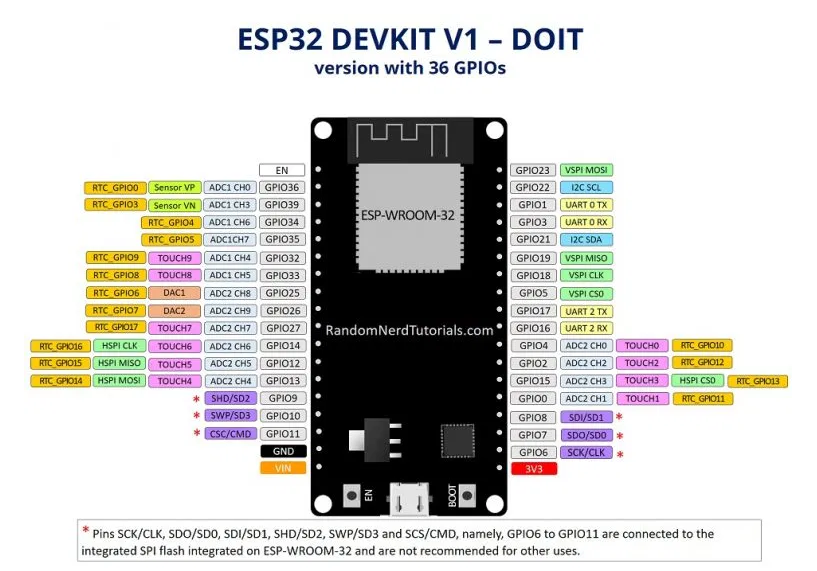
**Distribución de pines de módulos ESP32**

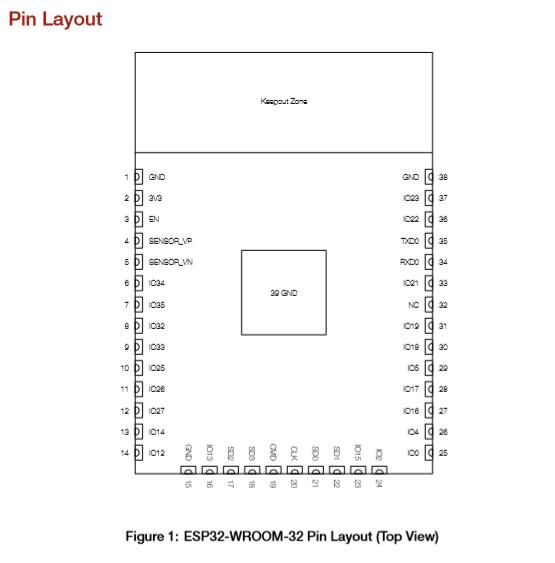
‎Con el ESP32 puede decidir qué pines son UART, I2C o SPI, solo necesita configurarlo en el código. Esto es posible gracias a la función de multiplexación del chip ESP32 que permite asignar múltiples funciones al mismo pin. Si no los configura en el código, los pines se usarán de forma predeterminada, como se muestra en la figura a continuación (la ubicación del pin puede cambiar según el fabricante)

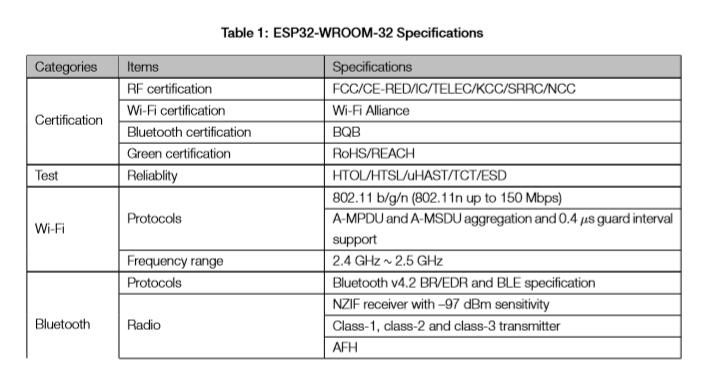
**‎Versión con 30 GPIOs‎**

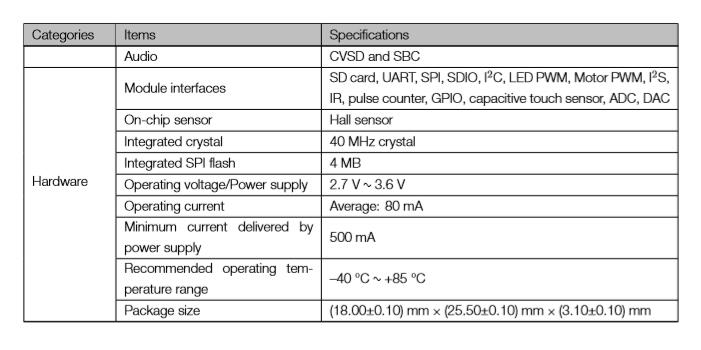
****

**Version with 36 GPIOs**

****







**IDE y Lenguajes:**

El ESP32 se puede programar en diferentes entornos de programación. Puede utilizar:‎

* [Arduino IDE](https://randomnerdtutorials.com/projects-esp32/)
* Espressif IDF (IoT Development Framework)
* [Micropython](https://randomnerdtutorials.com/getting-started-micropython-esp32-esp8266/)
* JavaScript
* LUA

**Cargar código en el ESP32 usando Arduino IDE‎ ejemplo:**

‎Copie el siguiente código en su IDE de Arduino:‎

/\*

Blink

\*/

// ledPin refers to ESP32 GPIO 23

const int ledPin = 23;

// the setup function runs once when you press reset or power the board

void setup() {

// initialize digital pin ledPin as an output.

pinMode(ledPin, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(ledPin, LOW); // turn the LED off by making the voltage LOW

delay(1000); // wait for a second

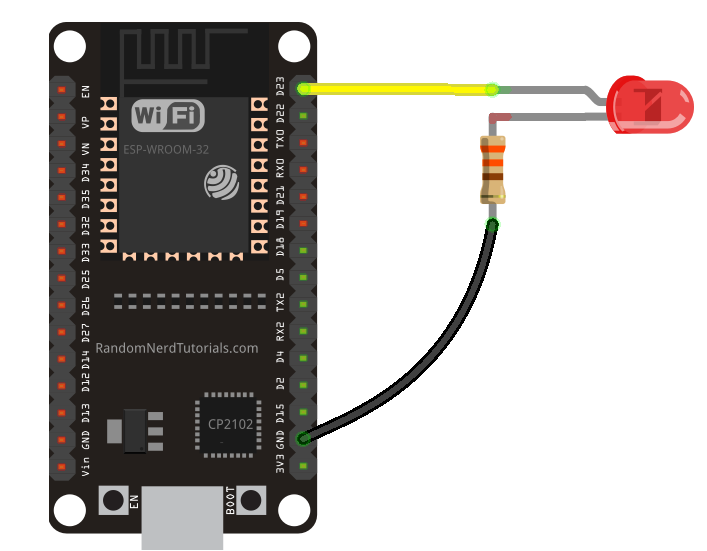
}

En este código, estamos controlando un LED conectado a GPIO 23.‎

const int ledPin = 23;

‎Por lo tanto, conecte un LED a su ESP32 siguiendo el siguiente diagrama esquemático.‎

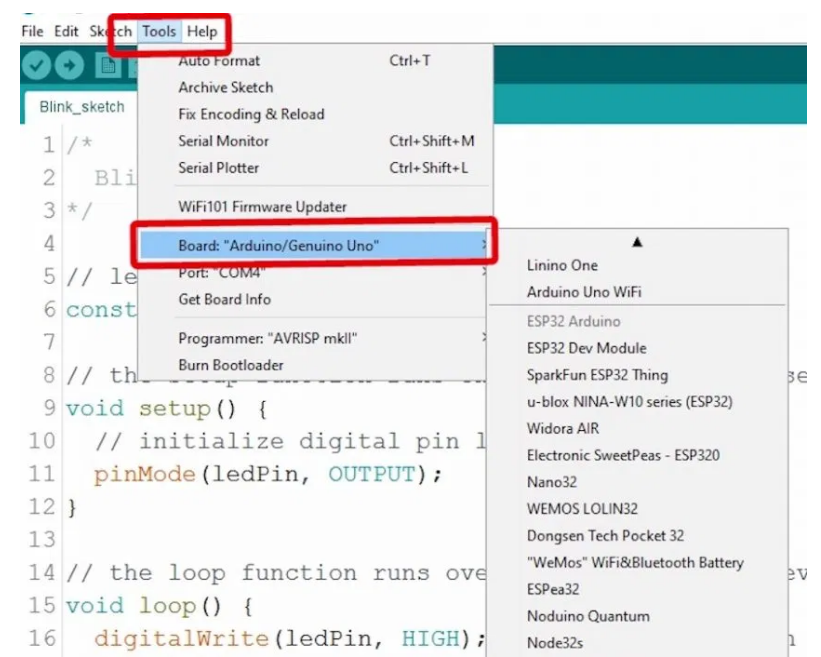
**‎Importante:‎**‎ siempre revise el pinout de su placa específica antes de construir cualquier circuito



‎Aquí hay una lista de las piezas que necesita para construir este circuito anterior:‎

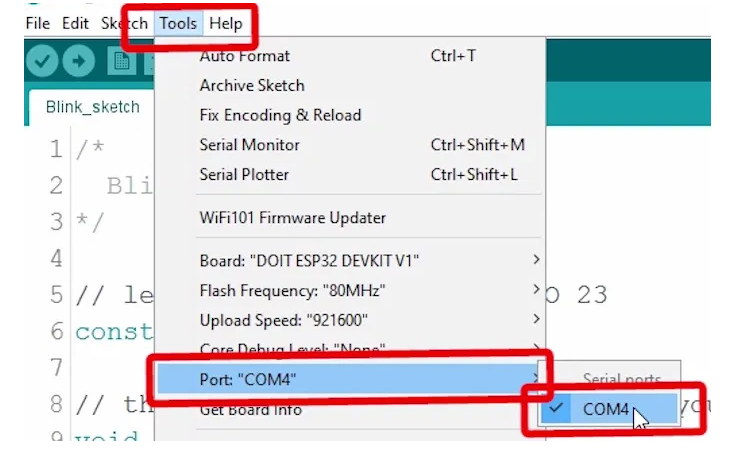
* [‎ESP32 DEBE DEVKIT V1 Placa‎](https://makeradvisor.com/tools/esp32-dev-board-wi-fi-bluetooth/)
* [‎LED de 5 mm‎](https://makeradvisor.com/tools/3mm-5mm-leds-kit-storage-box/)
* [‎Resistencia de 330 Ohm‎](https://makeradvisor.com/tools/resistors-kits/)
* [‎Cables de puente‎](https://makeradvisor.com/tools/jumper-wires-kit-120-pieces/)
* [‎Placa de pruebas‎](https://makeradvisor.com/tools/mb-102-solderless-breadboard-830-points/)‎ (opcional)

‎Conecte la placa de desarrollo ESP32 al ordenador y siga las siguientes instrucciones:

‎

‎1) Vaya a **Tools** > **Board**‎**‎ ,‎**‎ desplácese hacia abajo hasta la sección ESP32 y seleccione el nombre de su placa ESP32. En mi caso, es la placa DOIT ESP32 DEVKIT V1.‎

‎2) Vaya a ‎ **Tools > Port‎**‎ y seleccione un puerto COM disponible.‎



‎3) Presione el botón de carga.‎

‎¡Eso es todo!‎