

## INSTALACIÓN Y CONFIGURACIÓN

### Arduino IDE y SimulIDE

Para dar nuestros primeros pasos en el mundo de la **programación embebida**, realizaremos ejemplos de programación simple en el entorno de desarrollo **Arduino IDE**, sobre placas de desarrollo Arduino.



Fig. 1 - Arduino IDE

Además, usaremos un programa denominado **SimulIDE**, el cual tiene funcionalidad tanto online como offline, y nos permitirá simular el comportamiento programado en estas mismas placas de desarrollo.



### SimulIDE Circuit Simulator

Fig. 2 - SimulIDE

En ambos casos, programaremos sobre placas Arduino UNO o Arduino NANO. También puedes usar otra placa Arduino de tu preferencia.

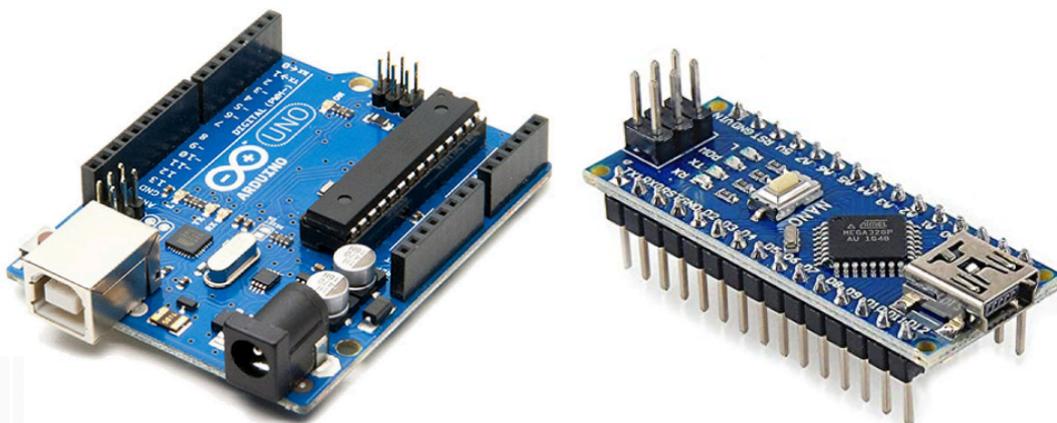


Fig. 3 - Arduino UNO (izquierda) y Arduino NANO (derecha)

Ambas placas tienen un microcontrolador llamado Atmega328p, en distinto formato.

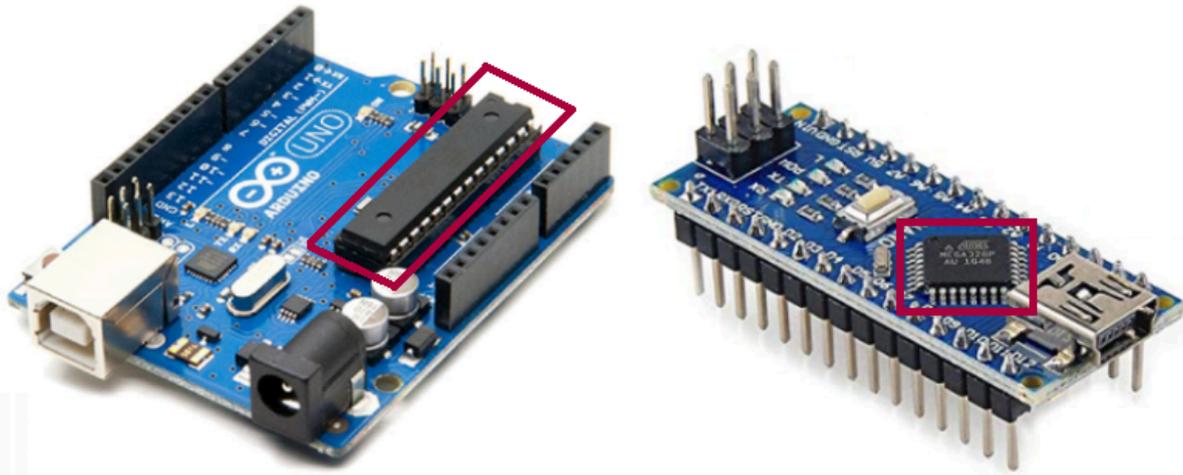


Fig. 4 - Arduino UNO (izquierda) y Arduino NANO (derecha), con Atmega328p.

El microcontrolador puede venir en dos formatos o encapsulados:

- **THT** (Through-Hole Thecnology), extraíble, encapsulado DIP28 (dos tiras de 14 pines)
- **SMD** (superficial), soldado a la placa, 28 pines distribuidos en sus cuatro lados

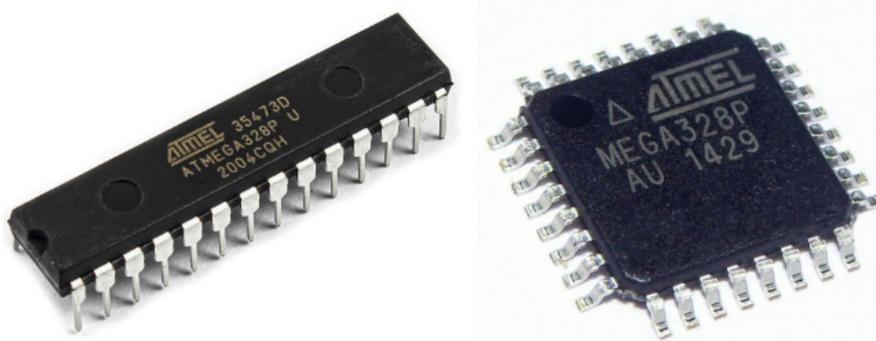


Fig. 5 - Atmega328p THT (izquierda) y SMD (derecha)

Este microcontrolador, tiene 28 “pines”, cada uno con su funcionalidad específica, los cuales veremos con más detalle más adelante.

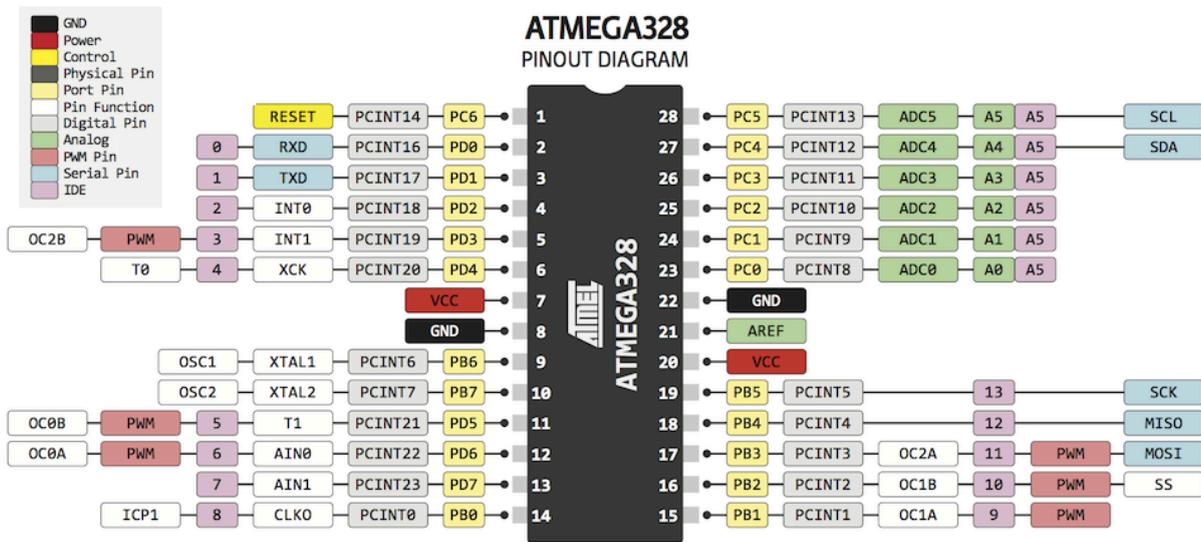


Fig. 6 - Atmega328 pinout diagram

A su vez, dentro de cada placa Arduino, los pines del microcontrolador tienen determinada conexión con los pines de la placa.

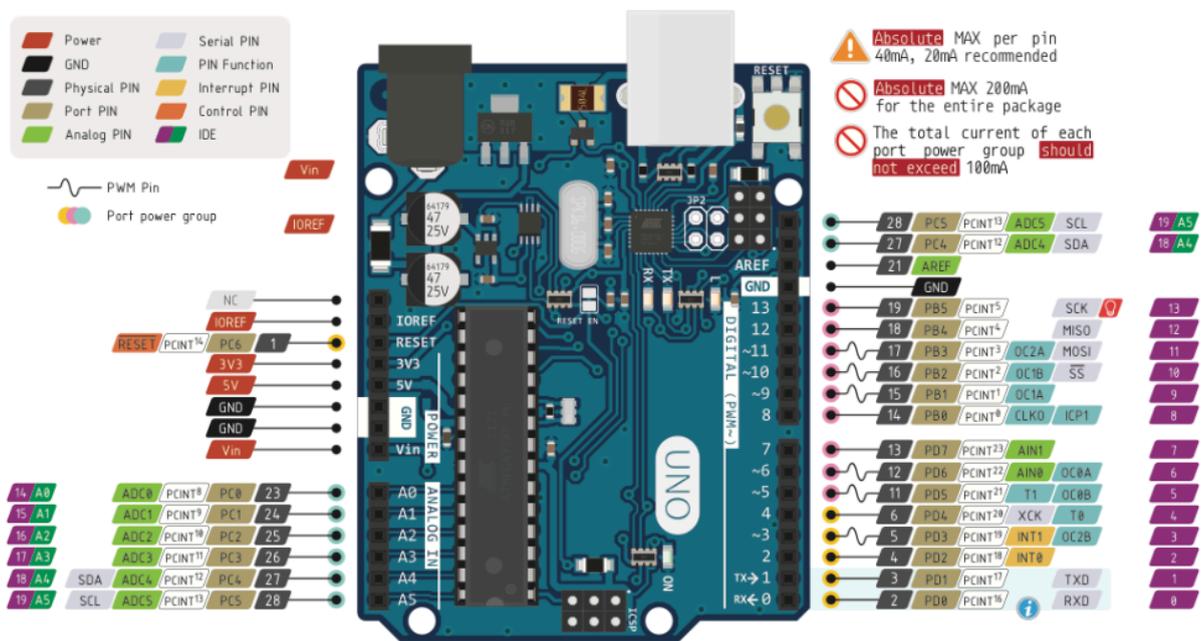


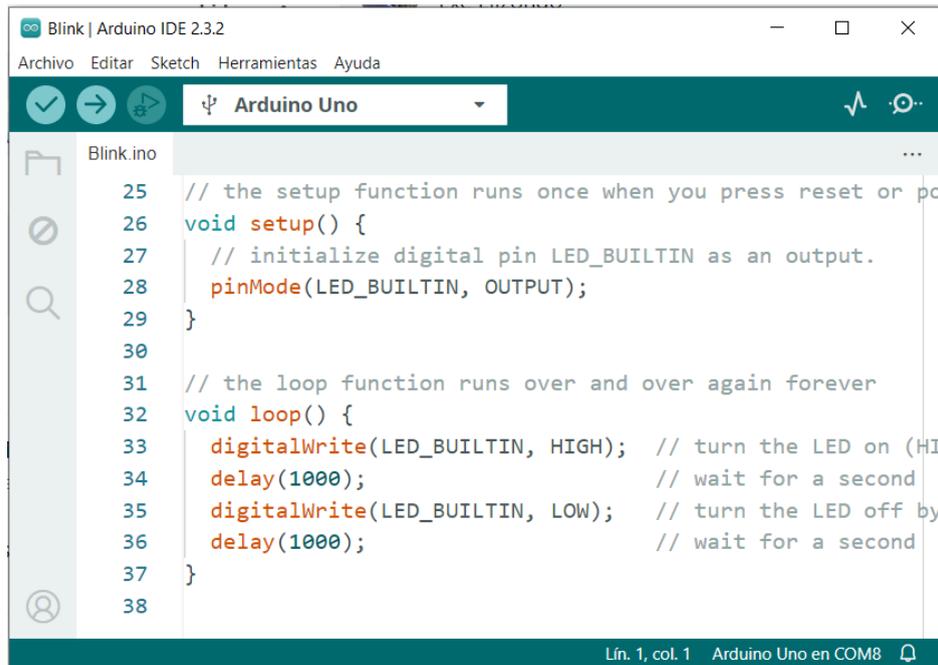
Fig. 7 - Arduino UNO pinout diagram



## ¿Cómo será el procedimiento?

Nosotros escribiremos líneas de código en el entorno Arduino IDE, es decir, desarrollaremos lo que se llama el “**código fuente**”, comenzando por la modificación de ejemplos simples existentes en el mismo programa.

Este proceso es similar a escribir texto en un editor como un bloc de notas o Microsoft Word, pero más amigable para el desarrollo de firmware.



```
Blink | Arduino IDE 2.3.2
Archivo  Editar  Sketch  Herramientas  Ayuda
Arduino Uno
Blink.ino
25 // the setup function runs once when you press reset or po
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HI
34   delay(1000); // wait for a second
35   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by
36   delay(1000); // wait for a second
37 }
38
Ln. 1, col. 1  Arduino Uno en COM8
```

Fig. 9 - Arduino IDE, código fuente.

Luego, este código se compilará, es decir, se traducirá a “**código máquina**”, mediante herramientas internas del programa.

Se pasa entonces del código fuente que nosotros escribimos y entendemos, a un “archivo ejecutable” que entiende el microcontrolador.

Por último, viene el proceso de grabación. El archivo ejecutable se guarda en la memoria de programa de nuestro microcontrolador, para luego ser “**ejecutado**” cada vez que se lo alimenta con tensión (o se conecta por USB la placa Arduino).

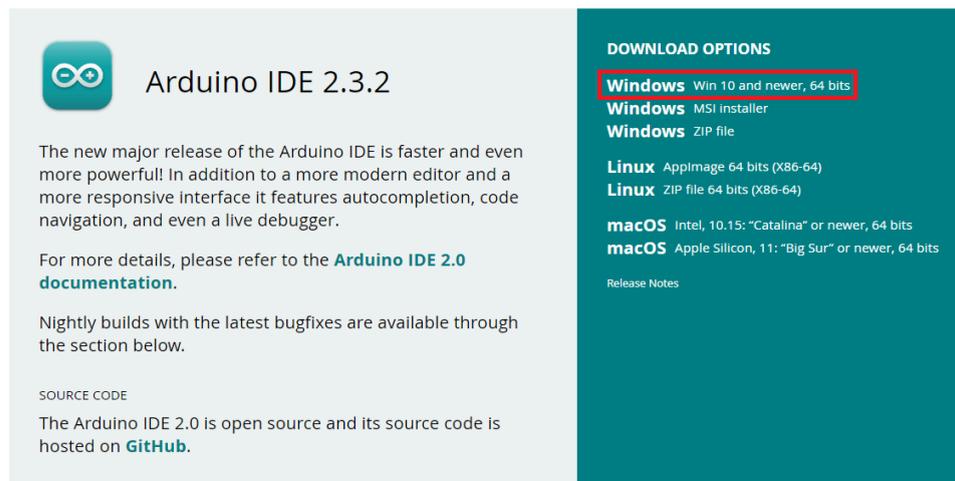
## Descarga e instalación de Arduino IDE

En primer lugar, descarga el Entorno de Desarrollo Integrado Arduino, desde su página oficial.

<https://www.arduino.cc/en/software>

Elige la versión que se adapte a tu sistema operativo, la puedes encontrar en *Información del Sistema*. Se muestra ejemplo para Windows 10 de 64 bits, eligiendo la versión más reciente en este momento.

### Downloads



**Arduino IDE 2.3.2**

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

**SOURCE CODE**

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

**DOWNLOAD OPTIONS**

- Windows** Win 10 and newer, 64 bits
- Windows** MSI installer
- Windows** ZIP file
- Linux** AppImage 64 bits (X86-64)
- Linux** ZIP file 64 bits (X86-64)
- macOS** Intel, 10.15: "Catalina" or newer, 64 bits
- macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

Fig. 10 - Descarga de Arduino IDE según tu sistema operativo

Acepta los términos y condiciones y descarga a tu PC.

### Stay in the Loop: Join Our Newsletter!

As a beginner or advanced user, you can find inspiring projects and learn about cutting-edge Arduino products through our **weekly newsletter!**

email \*  
tuemail@gmail.com

I confirm to have read the [Privacy Policy](#) and to accept the [Terms of Service](#) \*

I would like to receive emails about special deals and commercial offers from Arduino.

**SUBSCRIBE & DOWNLOAD**

or

**JUST DOWNLOAD**

Fig. 11 - Acepta términos y condiciones, descarga a tu PC

Descarga el archivo (ejemplo: *arduino-ide\_2.3.2\_Windows\_64bit.exe*) en alguna ubicación de tu PC. Luego haz click derecho en él y ejecútalo como Administrador.



Fig. 12 - Ejecuta como Administrador el archivo ejecutable descargado

Sigue los pasos de la instalación hasta finalizar.

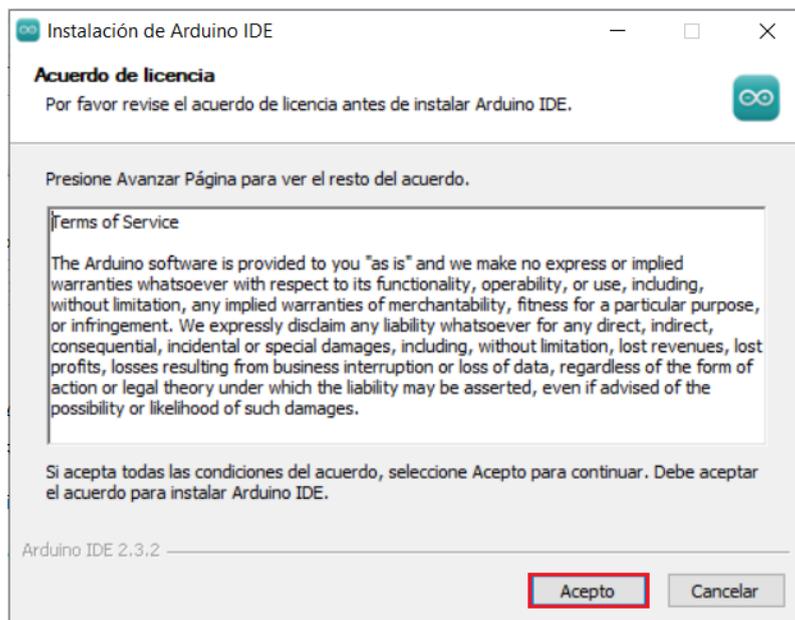


Fig. 13 - Acepta términos y condiciones

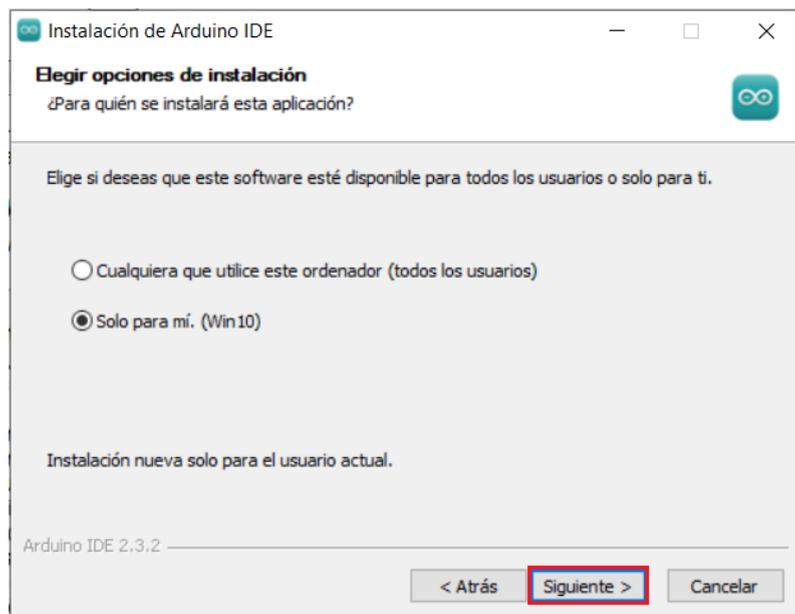


Fig. 14 - Software disponible sólo para mí

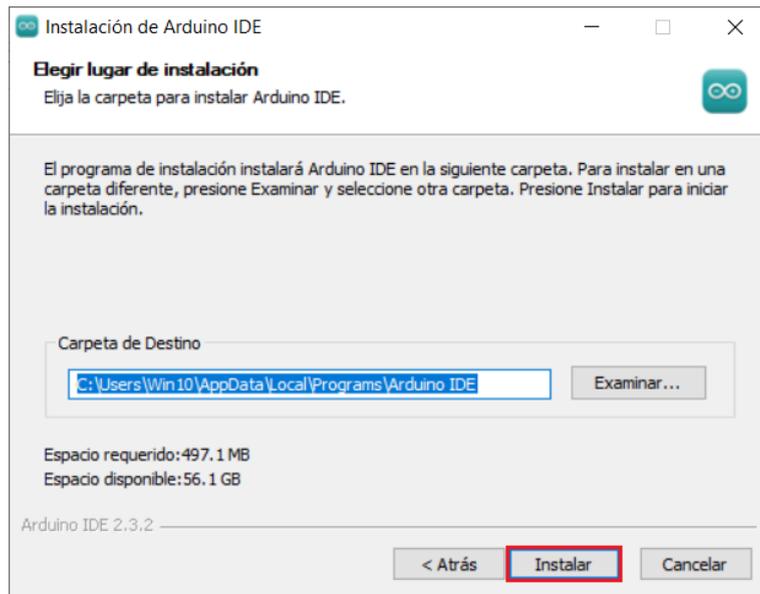


Fig. 15 - Elige ubicación predeterminada para la instalación

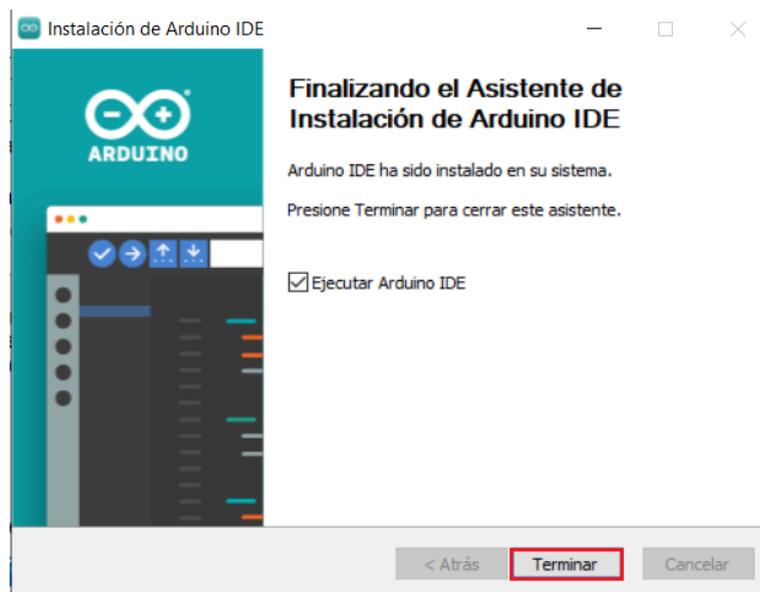


Fig. 16 - Finaliza la instalación

Abre el IDE de Arduino.

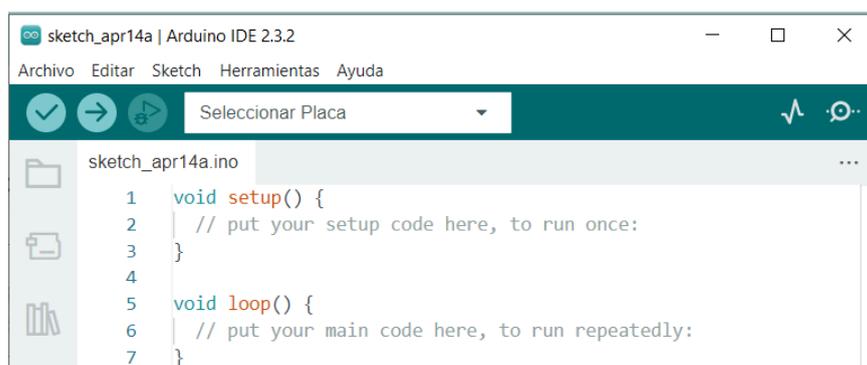


Fig. 17 - Abre Arduino IDE

## Instalación de Drivers

Enchufa la placa Arduino a tu PC, a través de un cable compatible, desde el extremo de la placa hasta el puerto USB de tu PC.

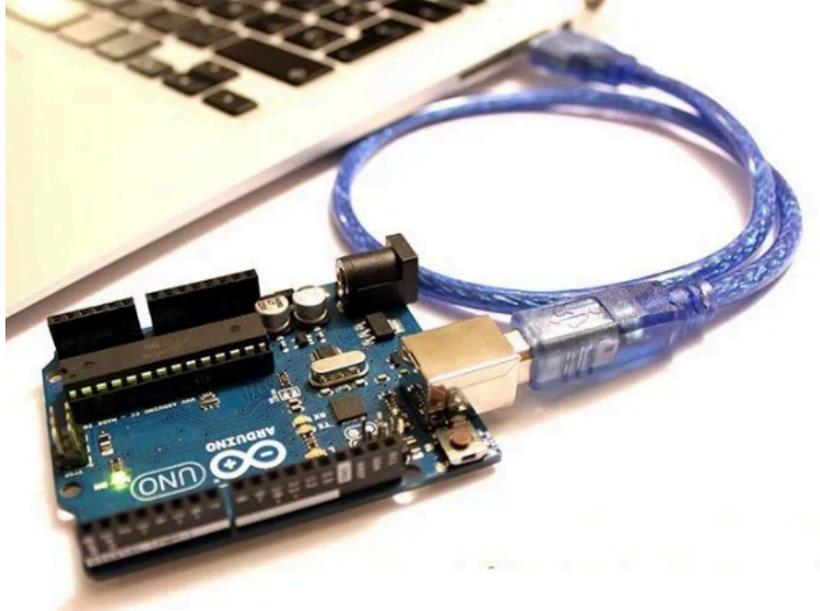


Fig. 18 - Conexión de Arduino UNO a tu PC

Nota: el cable debe ser de DATOS y CARGA (no solo de CARGA).

Ve al Administrador de Dispositivos de tu PC. Si el Driver está bien instalado, tu PC reconocerá automáticamente el nuevo dispositivo conectado y le asignará un puerto COM con un número. En el ejemplo mostrado, se asigna el **COM8**, pero en tu caso puede variar. También puede variar conectando en distintos puertos USB de una misma PC.

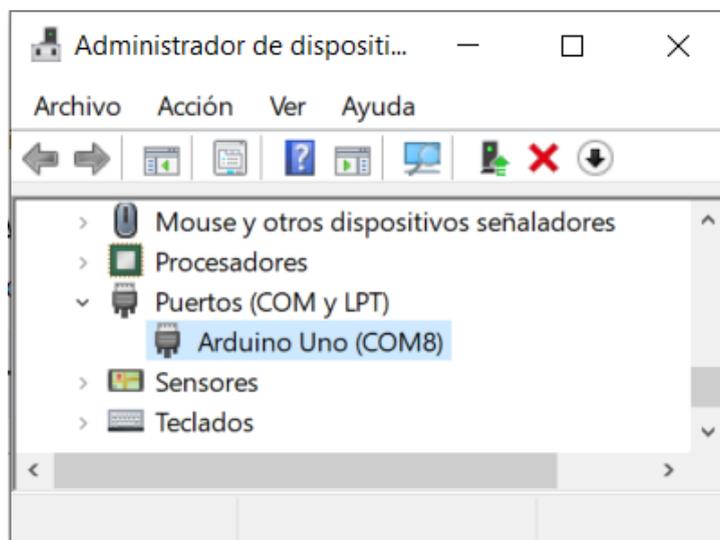


Fig. 19 - Administrador de Dispositivos al enchufar tu placa a la PC

Si no se asigna un puerto COM correcto o presenta un problema de detección, se deberá instalar el Driver adecuado. Puede ser el CP2102, CH340 u otros.

Si el problema persiste, se puede forzar el cambio de hardware con el programa Zadig. ¡Pídenos ayuda para estos pasos!

De lo contrario, si el Driver se instaló de forma correcta, omitir este paso.

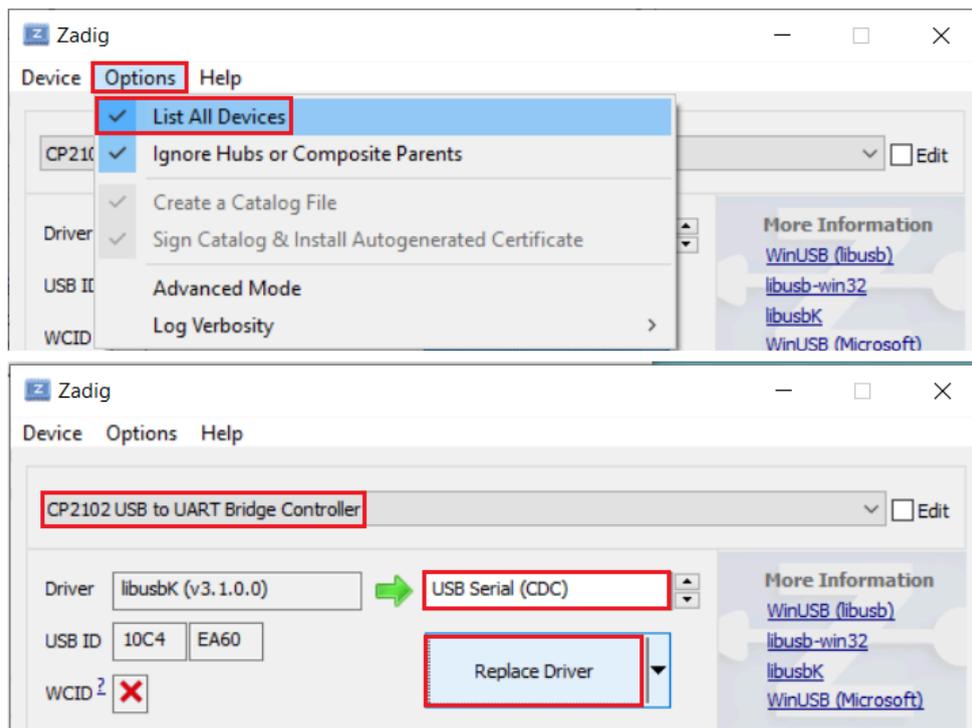


Fig. 20 - Ejemplo de cambio de hardware Driver CP2102 (solo en caso de error de Driver)

¡Ya está todo listo para comenzar a crear tus primeros proyectos!

## Primera prueba

Abre ArduinoIDE y elige el ejemplo básico Blink.

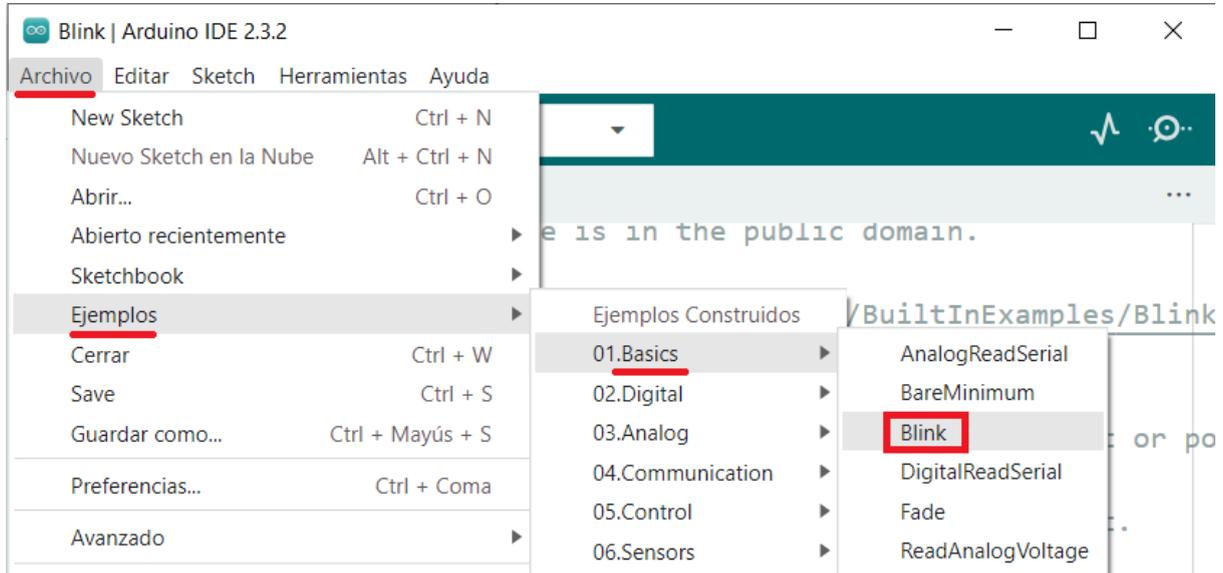


Fig. 21 - Abre ejemplo básico Blink

Se te abrirá este código fuente:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Fig. 22 - Ejemplo básico Blink

Elige tu placa en Herramientas:

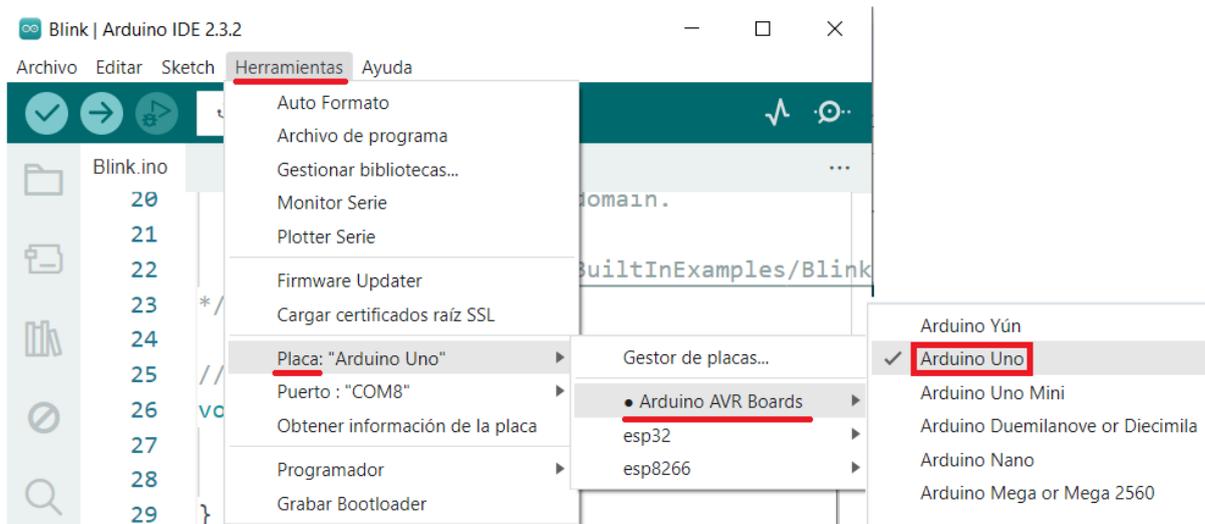


Fig. 23 - Elección de placa Arduino UNO

Elige el puerto COM adecuado (ver Administrador de dispositivos).

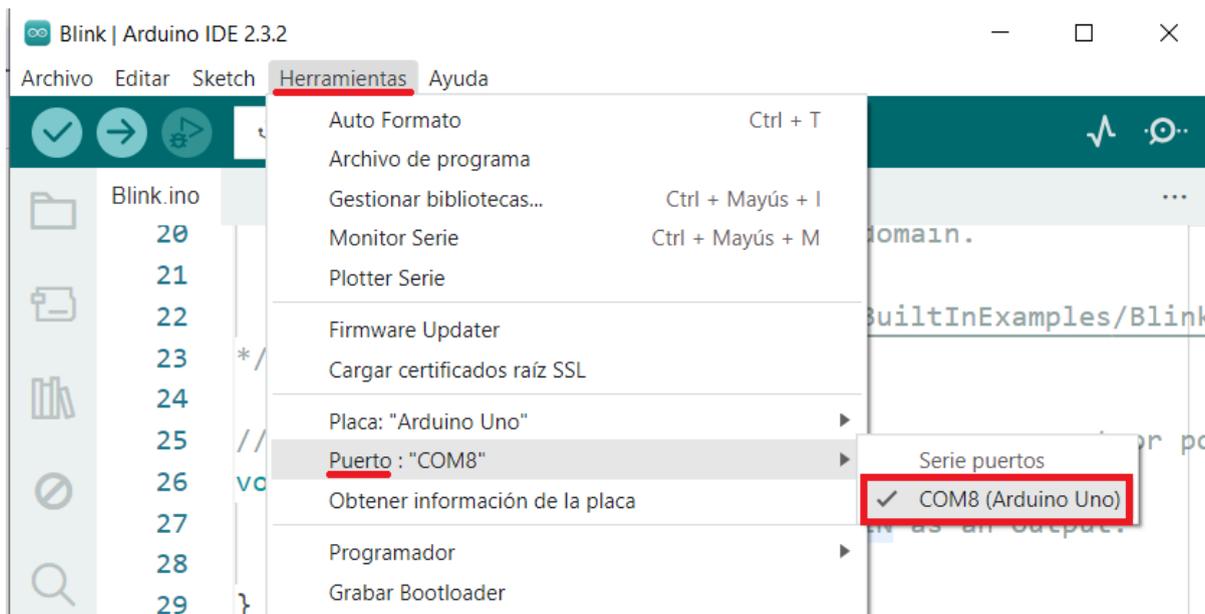


Fig. 24 - Elección de puerto COM

Carga tu programa:

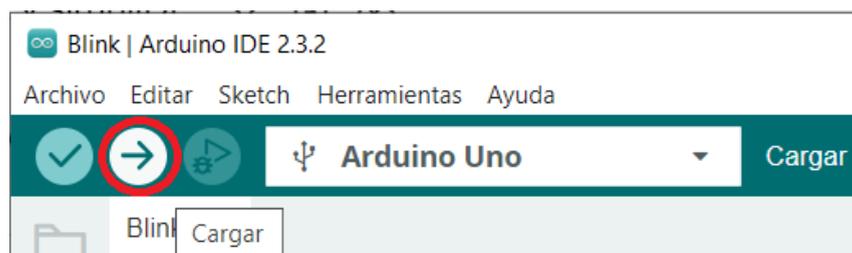


Fig. 25 - Carga tu programa

Al finalizar el procedimiento de compilación y carga, te mostrará un mensaje de Carga finalizada.

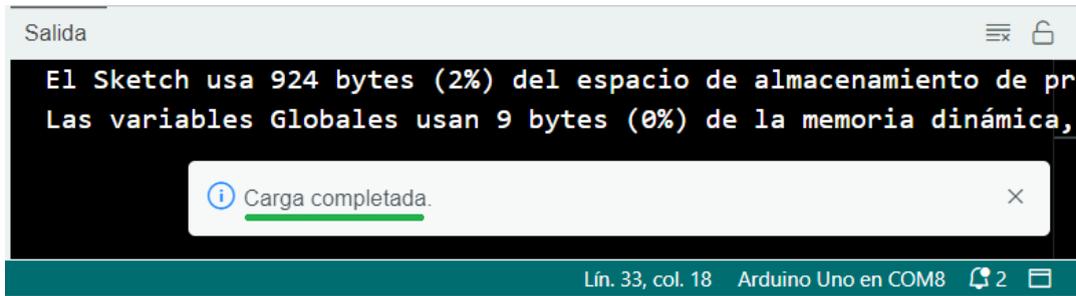


Fig. 26 - Verifica que se haya cargado correctamente el programa

Por último, verifica el parpadeo del led interno de tu placa Arduino, con cambios de estado cada 1 segundo.

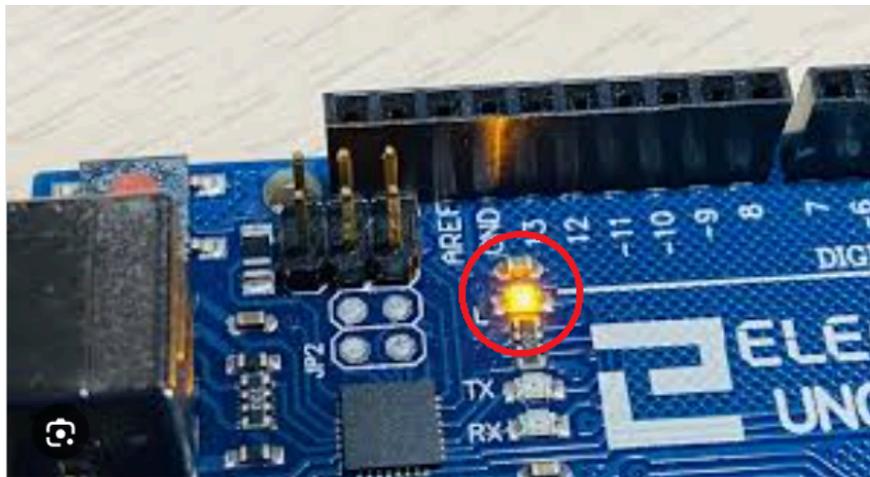
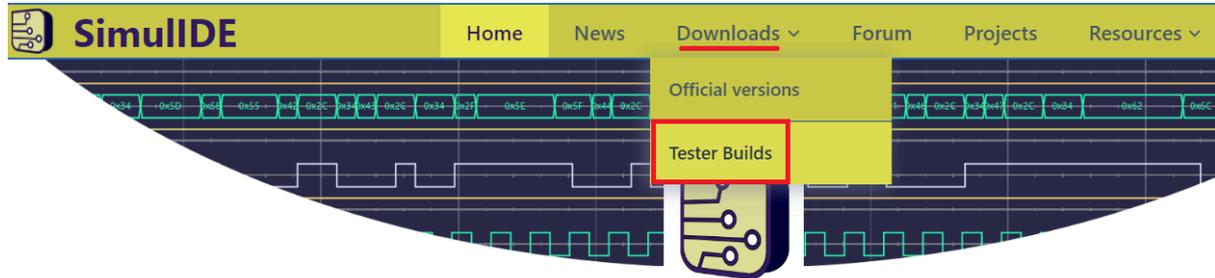


Fig. 27 - Prueba parpadeo led interno

## Descarga e instalación de SimulIDE

En primer lugar, descarga SimulIDE, desde su página oficial.

<https://simulide.com/p/>



## SimulIDE Circuit Simulator

SimulIDE is a simple real time electronic circuit simulator, intended for hobbyist or students to learn and experiment with analog and digital electronic circuits and microcontrollers.

It supports PIC, AVR, Arduino and other MCUs and MPUs.

Fig. 28 - SimulIDE

Elige la versión que se adapte a tu sistema operativo, la puedes encontrar en *Información del Sistema*. Se muestra ejemplo para Windows de 64 bits.

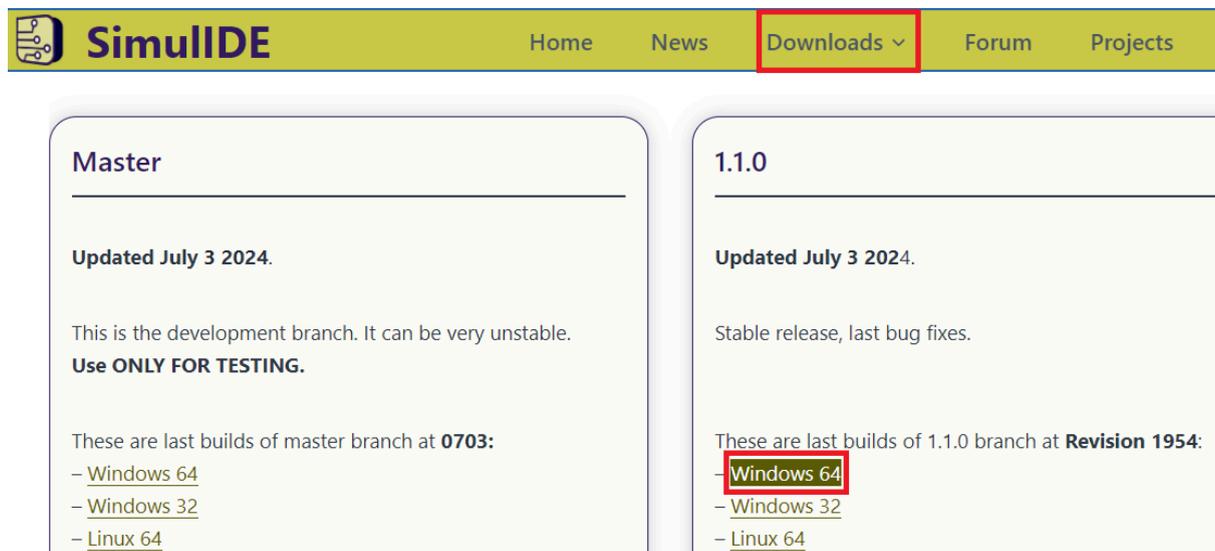


Fig. 29 - SimulIDE para Windows de 64 bits

Ejecuta el archivo .exe descargado como Administrador.

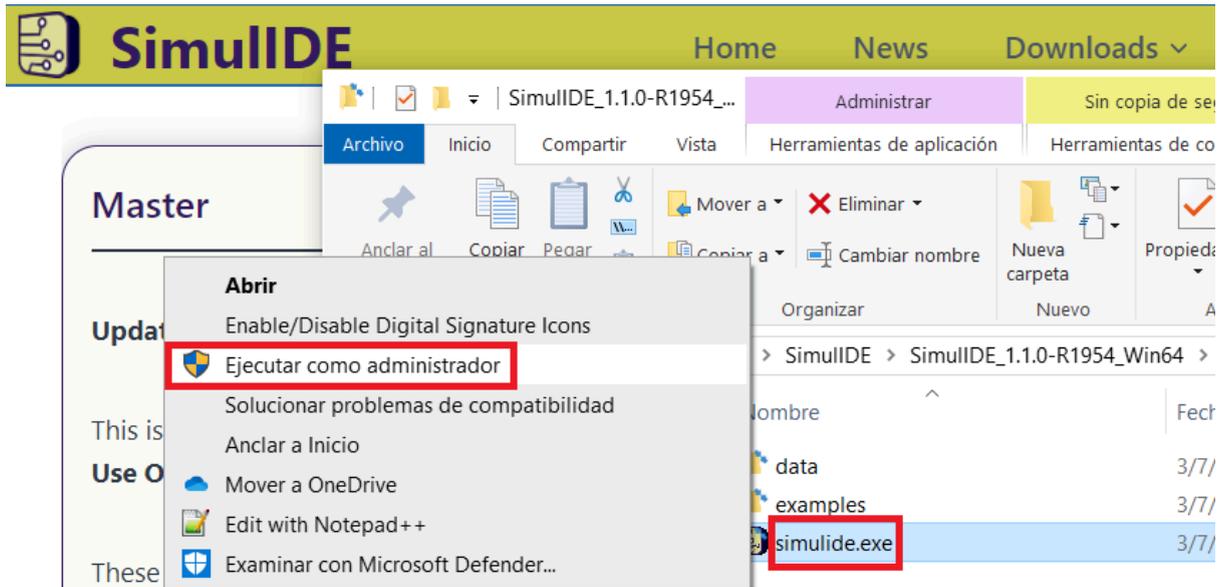


Fig. 30 - Instalación SimulIDE

Abre SimulIDE y arrastra uno a uno los componentes que quieres simular en tu circuito.

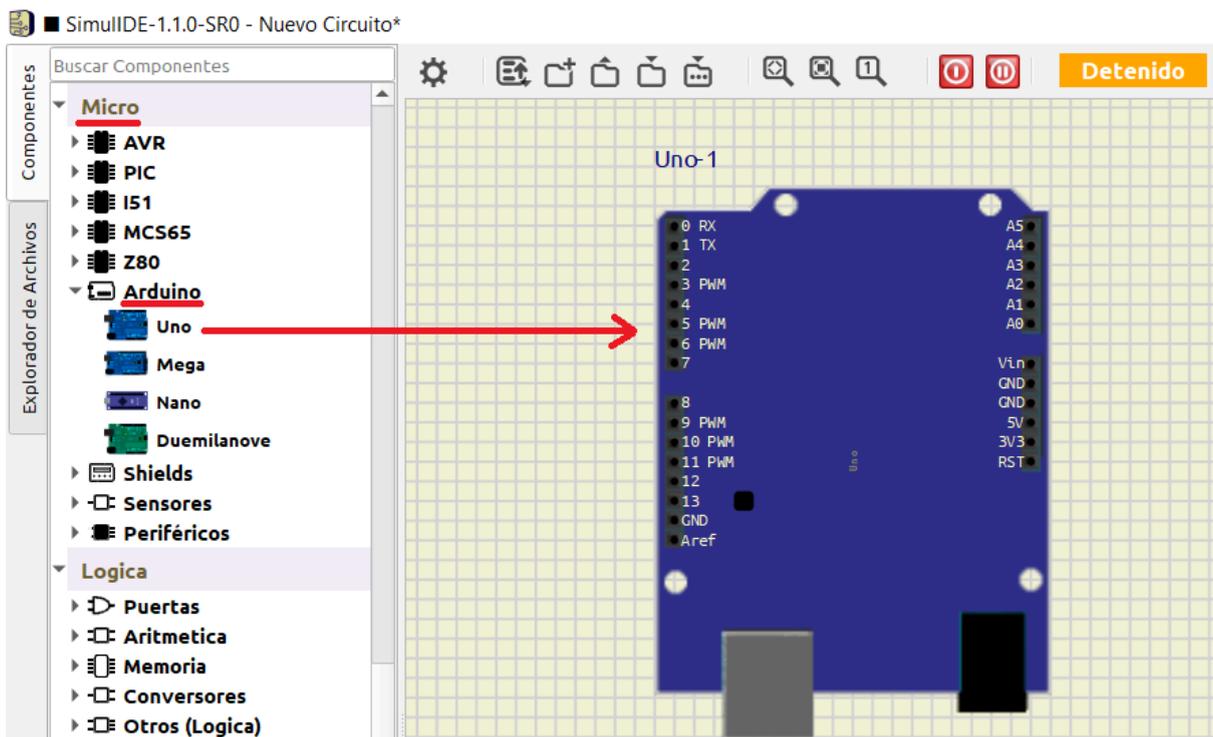


Fig. 31 - Agrega Arduino UNO a SimulIDE

¡Ya está todo listo para comenzar a simular tus proyectos!