



IEEE CIRCUIT & SYSTEM SOCIETY UNAM

MANUAL DE INSTALACIÓN DE ESP-IDF PARA EL SISTEMA OPERATIVO WINDOWS

CURSO

Internet de las Cosas (IoT's) y Fundamentos de RTOS con el ESP32WROOM

ELABORADO POR:

SALVADOR SARABIA

DANIEL ORTIZ

1. ANTES DE EMPEZAR

1.1. Uso básico de la consola de Windows y PowerShell

La consola de Windows es una opción recomendable para que usuarios medios/avanzados y administradores realicen tareas en el sistema operativo de forma más flexible y rápida. PowerShell y el símbolo del sistema se ejecutan en modo texto y no son tan intuitivos como una interfaz gráfica, pero muestran su potencial a la hora de ejecutar tareas repetitivas, para gestionar determinados componentes o acceder a cierta información del sistema que no está disponible de ninguna otra manera.

Su funcionamiento es simple a la vez que potente. Escribimos el comando y la aplicación CMD hace de intérprete para su ejecución.

PowerShell es una interfaz de consola que además de las posibilidades del símbolo del sistema, ofrece un mayor número de posibilidades ya que permite crear sus propios comandos y secuencias de comandos utilizando el lenguaje de programación C#. Tanto PowerShell como C# están integrados en Microsoft .NET Framework, lo que significa que tiene acceso a muchas funciones y herramientas preexistentes para ayudarlo a crear mejores comandos y scripts con menos esfuerzo.

Comandos populares en símbolos de sistema

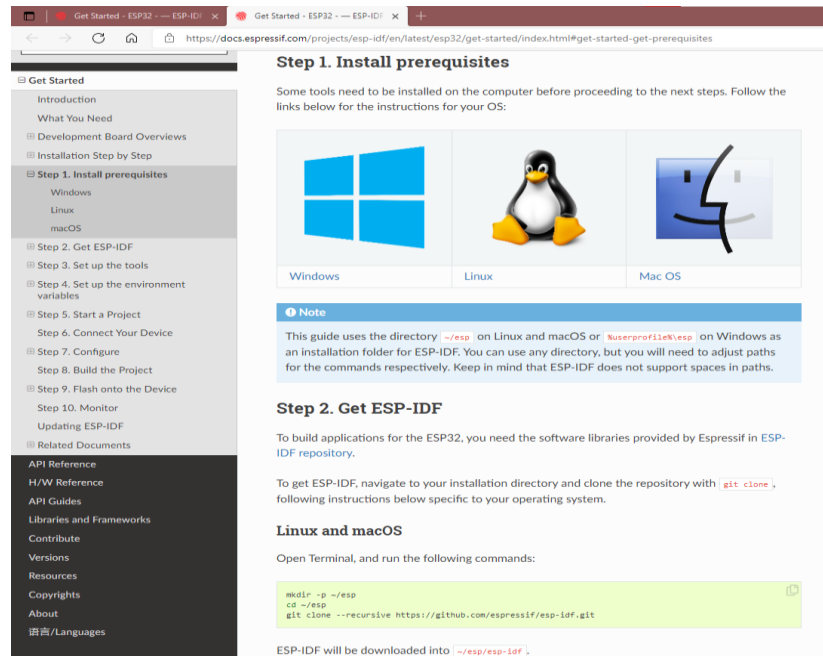
- **cd:** Muestra el directorio actual y permite cambiar a otros directorios. Con el parámetro /D más la unidad y la ruta también puede cambiarse la unidad. Mediante cd puede cambiarse al directorio superior (tiene la misma función que chdir).
- **cls:** Elimina el contenido de la pantalla.
- **date:** Muestra la fecha actual y permite modificarla. Con el parámetro /T solo se muestra la fecha sin opción de modificarla.
- **dir:** Muestra todas las carpetas y archivos dentro del directorio actual. Puede restringirse la salida mediante atributos (/A), simplificar la lista (/B), o mostrar todos los subdirectorios y sus archivos (/S).
- **echo:** Muestra un mensaje y es utilizado, sobre todo, en scripts y archivos batch.
- **mkdir:** Crea un nuevo directorio en la ruta especificada. Si los directorios ya no existen en la ruta, mkdir los genera automáticamente. Alternativamente también puede utilizarse el comando md.
- **more:** Muestra el contenido de un archivo (por ejemplo, de un archivo de texto) por páginas, pero también puede dividir la salida de otro comando en páginas.
- **rmdir:** Elimina un directorio que no puede ni incluir ni ocultar ningún archivo. Con el parámetro /S pueden eliminarse árboles de directorios completos. Alternativamente también puede utilizarse el comando rd.
- **copy:** Copia uno o varios archivos en otro lugar. Además, también es posible unir varios archivos en uno. Como wildcard puede utilizarse el asterisco.
- **del:** Elimina uno o varios archivos. Para eliminar todos los archivos de las subcarpetas puede utilizarse el parámetro /S. Los archivos de solo lectura se eliminan con /F. Como alternativa también puede utilizarse el comando erase.
- **move:** Desplaza uno o varios archivos de un directorio a otro y también puede modificar el nombre de los directorios. Por defecto, el comando sobrescribe otros

archivos con el mismo nombre al mover archivos a su destino. Para evitarlo se utiliza el parámetro /-Y.

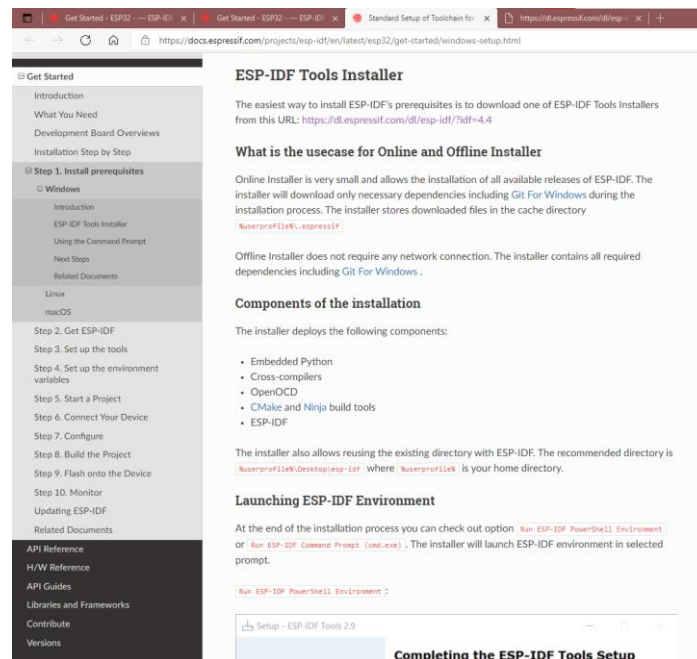
- **rename:** Tiene la misma función que ren.
- **xcopy:** Copia archivos y estructuras de directorio completas, para lo que el comando ofrece numerosas opciones adicionales. Así puede especificarse, por ejemplo, que solo se copien archivos anteriores a una fecha determinada (/D) y, asimismo, que también se sobrescriban archivos protegidos contra escritura (/R).
- **ipconfig:** Facilita información sobre la IP de cualquier adaptador de red utilizado. Con este comando también pueden liberarse direcciones (/release) o renovarse (/renew). /flushdns permite vaciar la caché DNS.
- **ping:** Envía una solicitud de eco mediante el protocolo de mensajes de control de Internet (Internet Control Message Protocol, ICMP) a un host específico para comprobar que está disponible y ofrece, además, la duración del eco. Con -t pueden enviarse señales ping de forma ininterrumpida. Para poder visualizar una estadística sobre esta acción presiona las teclas Ctrl + Pausa/Inter. Puedes cancelar el proceso con Ctrl+C.

2. INSTALACIÓN DEL ENTORNO DE DESARROLLO PARA TRABAJAR CON FREERTOS Y ESP-IDF

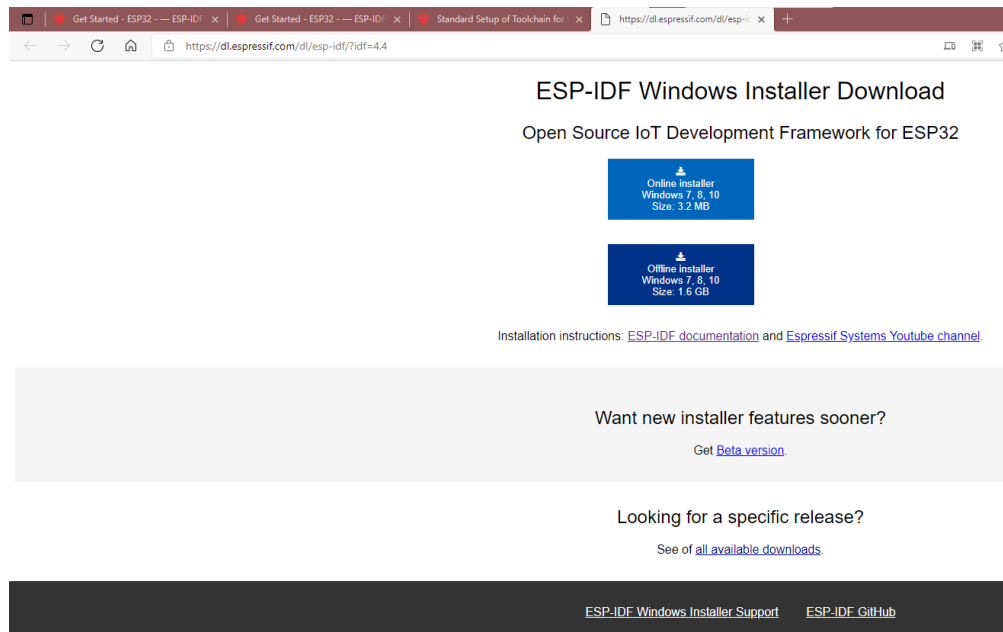
2.1. Dirigirse al sitio oficial de ESP32-IDF y seleccionar Get Started. El paso 1: Instalar prerequisites para Windows.



2.2. Una vez seleccionado el icono de Windows se mostrará el sitio que dirige a una descarga del instalador online y offline. La ventaja del offline con respecto al online es que el archivo descargado lo puedes llevar a donde sea con una sola descarga, pero ocupa 1.5 Gb de espacio. La instalación online descarga el archivo en tiempo real sin requerir un espacio adicional. Enlace del instalador: <https://dl.espressif.com/dl/esp-idf/?idf=4.4>



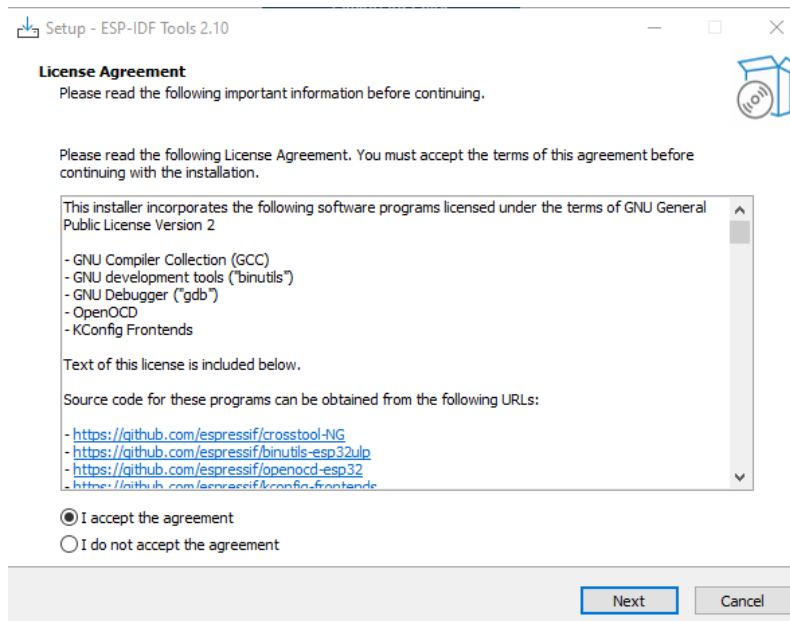
2.3. Seleccionar el deseado (Offline Installer altamente recomendado)



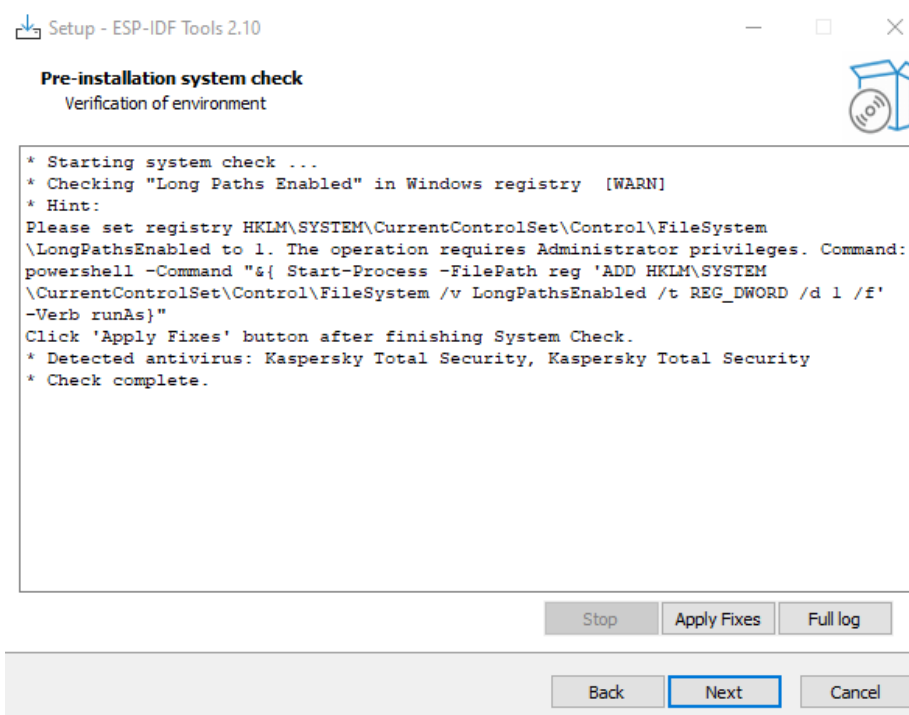
2.4. Componentes de la instalación: El ejecutable instalará los siguientes componentes:

- Embedded Python
- Cross_compilers
- OpenOCD
- CMake and Ninja build tools
- ESP-IDF

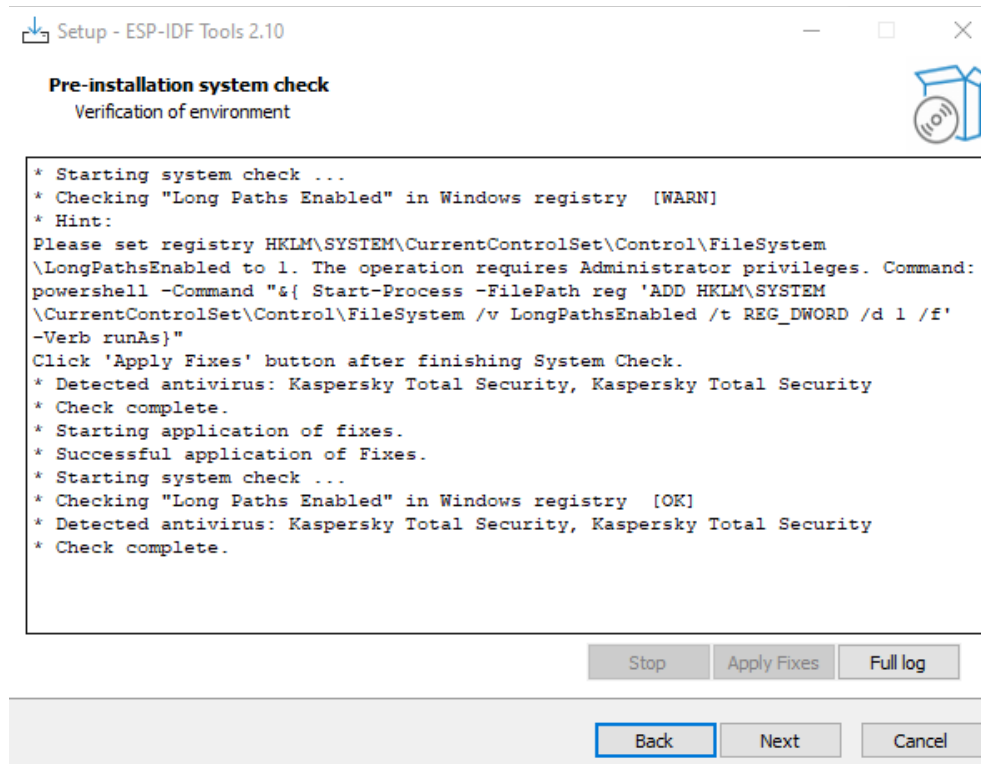
Dar clic derecho al instalador descargo y seleccionar ejecutar como administrador, al abrir el instalador seleccione “Acepto el acuerdo”



2.5 Si existe detalles con el registro de Windows, dar clic en “Aplicar correcciones”



2.6. Una vez realizado las correcciones, dar clic en siguiente



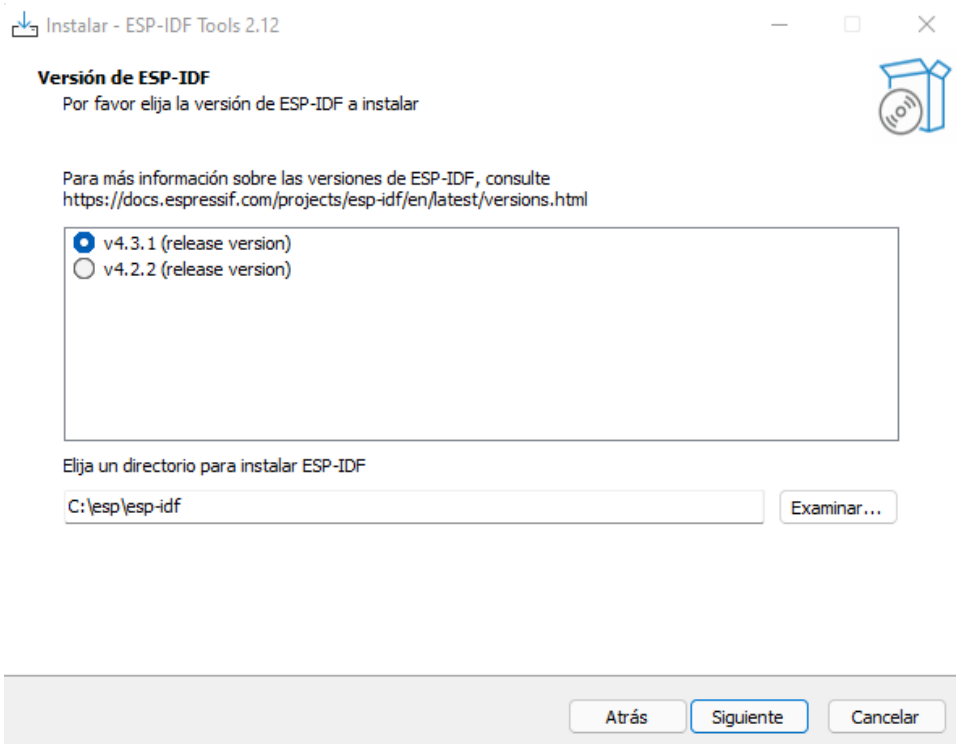
2.7. Elegir la versión 4.3 de ESP-IDF

2.8. [Importante] Elegir la ruta de instalación de ESP-IDF como:

C:\esp\esp-idf

Dar clic en “siguiente”.

➔ Es de vital importancia seguir con esta regla ya que, de no cumplirse, se podrá tener errores en variables de entorno y requerirá una nueva instalación

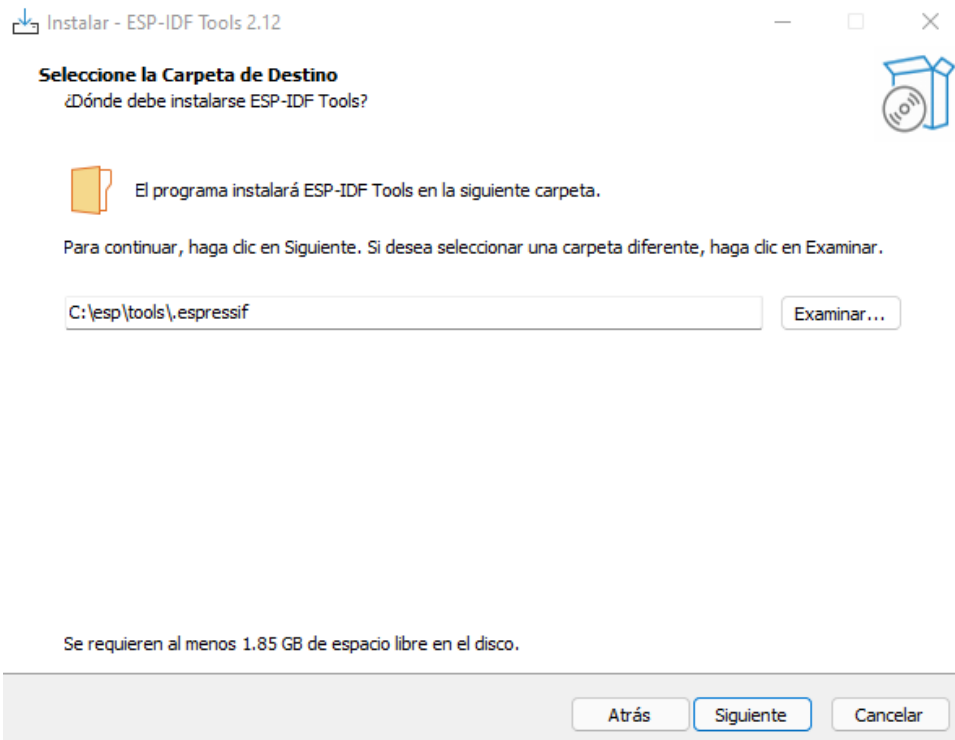


2.9. [Importante] Elegir la ruta de instalación de ESP-IDF Tools como:

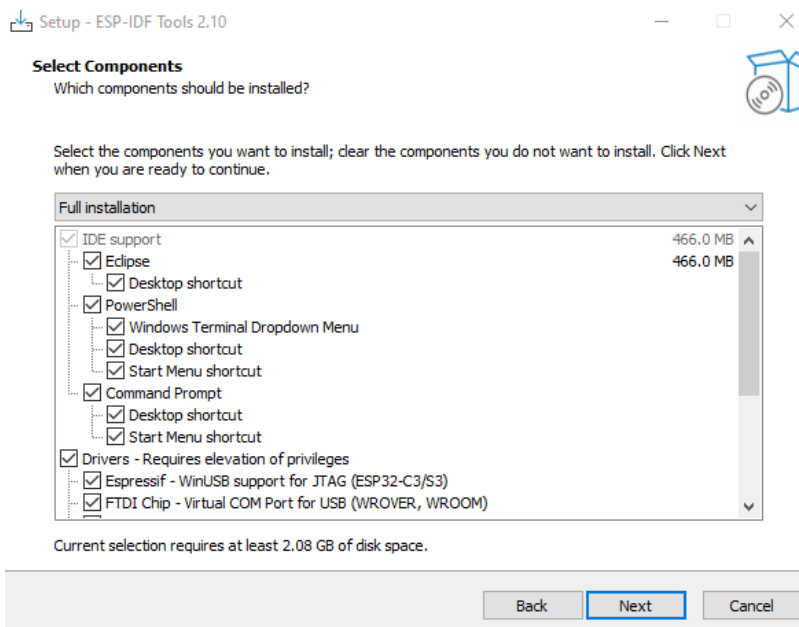
C:\esp\tools\espressif

Dar clic en “siguiente”.

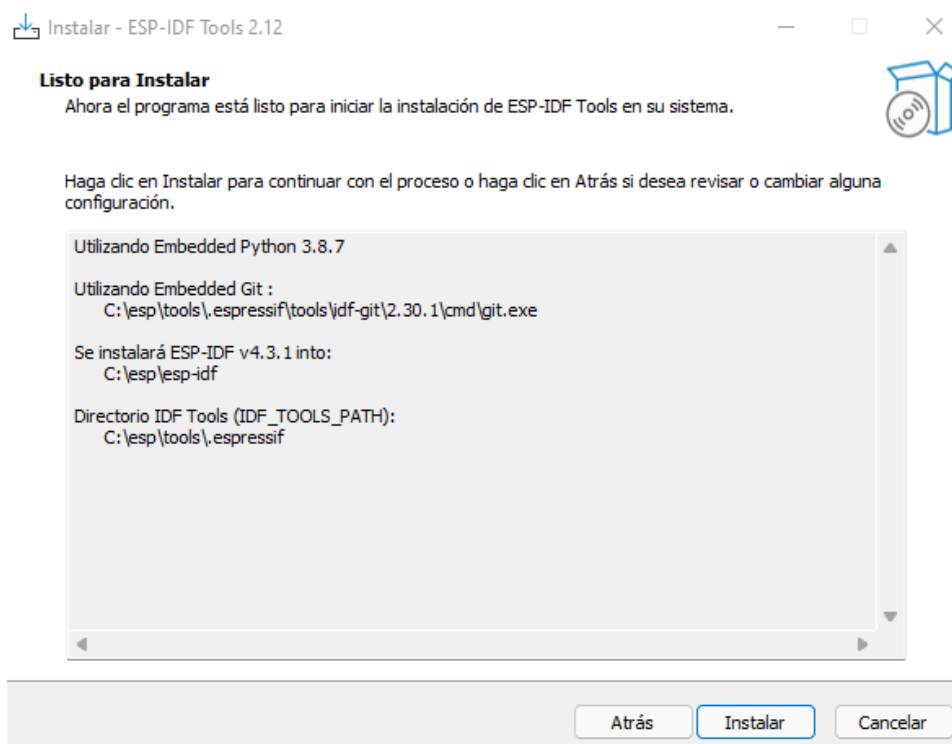
➔ Es de vital importancia seguir con esta regla ya que, de no cumplirse, se podrá tener errores en variables de entorno y requerirá una nueva instalación



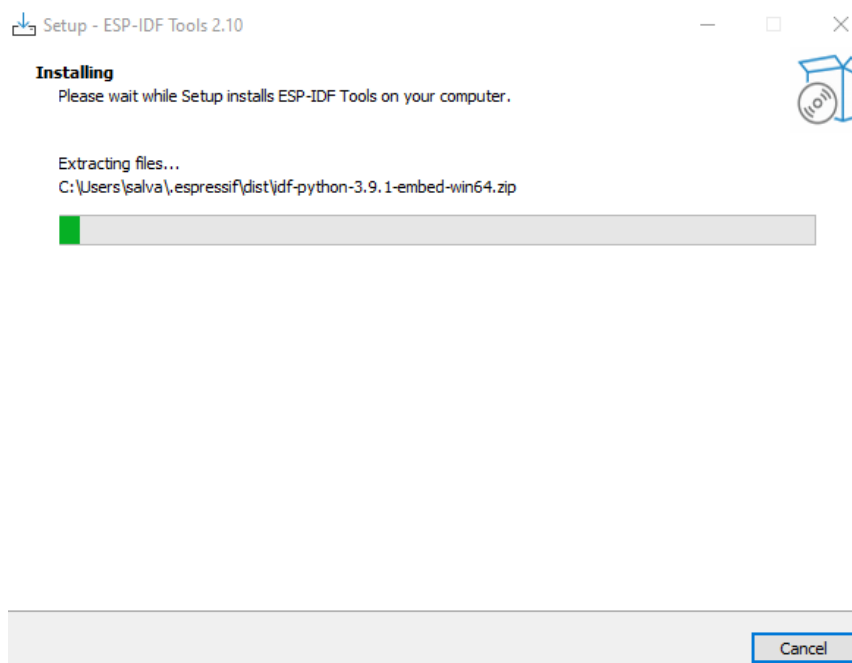
- 2.10. En la ventana de “seleccionar componentes”, elegir “Instalación completa”, dar clic en “siguiente”



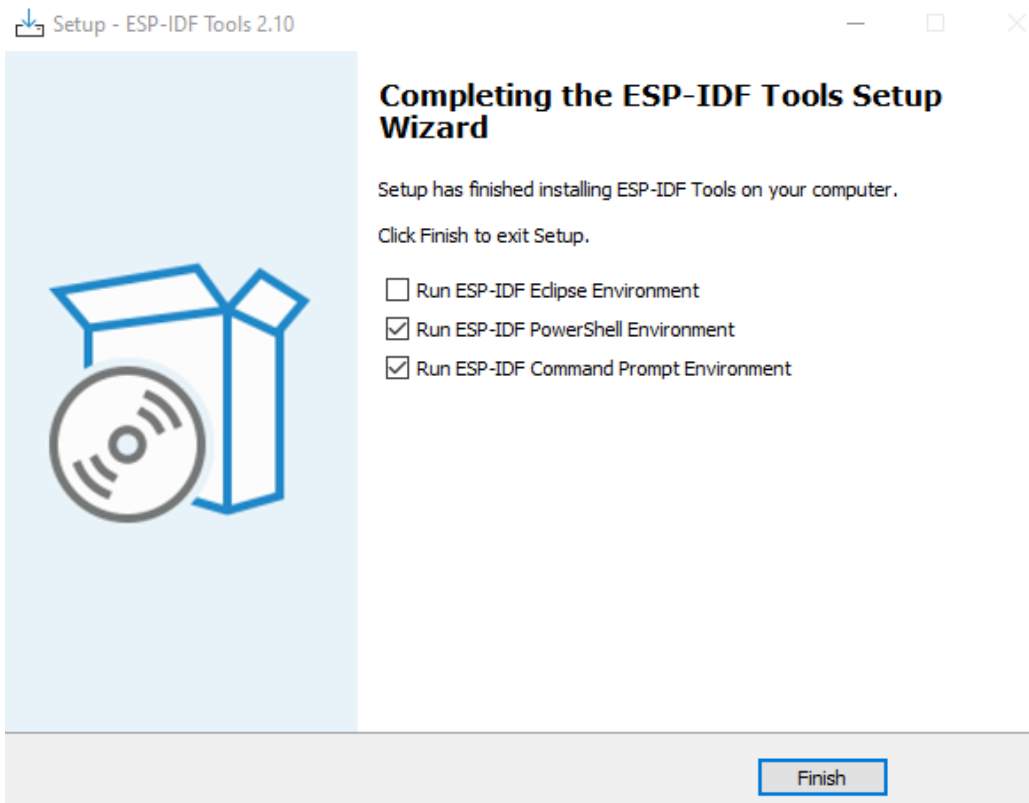
- 2.11. Para finalizar con la configuración de instalación se mostrará un resumen de las actividades que se realizarán durante la instalación, dar clic en “instalar”



2.12. Inicia el proceso de instalación



- 2.13. Al término del proceso, se mostrará la pantalla de asistente completado, mantener las casillas palomeadas para probar que la instalación es correcta.



3. PRUEBA DE FUNCIONAMIENTO DE LA INSTALACIÓN

- 3.1. Una vez abierto tanto la terminal de símbolos de sistema como la de PowerShell se mostrará un mensaje de la ejecución del entorno de ESPIDF con la palabra “Hecho”, si es de su preferencia cierre alguna de las ventanas, preferentemente PowerShell

```
ESP-IDF 4.3 PowerShell
C:\Users\salsa\Desktop\esp-idf\components\esptool_py\esptool
C:\Users\salsa\Desktop\esp-idf\components\app_update
C:\Users\salsa\Desktop\esp-idf\components\espcoredump
C:\Users\salsa\Desktop\esp-idf\components\partition_table
C:\Users\salsa\espessif\tools\xtensa-esp32-elf\esp-2020r3-8.4.0\xtensa-esp32-elf\bin
C:\Users\salsa\espessif\tools\xtensa-esp32s2-elf\esp-2020r3-8.4.0\xtensa-esp32s2-elf\bin
C:\Users\salsa\espessif\tools\xtensa-esp32s3-elf\esp-2020r3-8.4.0\xtensa-esp32s3-elf\bin
C:\Users\salsa\espessif\tools\riscv32-esp-elf\1.24.0.123_64eb9ff-8.4.0\riscv32-esp-elf\bin
C:\Users\salsa\espessif\tools\esp32ulp-elf\2.28.51-esp-20191205\esp32ulp-elf-binutils\bin
C:\Users\salsa\espessif\tools\esp32s2ulp-elf\2.28.51-esp-20191205\esp32s2ulp-elf-binutils\bin
C:\Users\salsa\espessif\tools\cmake\3.16.4\bin
C:\Users\salsa\espessif\tools\openocd-esp32\v0.10.0-esp32-20210401\openocd-esp32\bin
C:\Users\salsa\espessif\tools\ninja\1.10.2\
C:\Users\salsa\espessif\tools\idf-exe\1.0.1\
C:\Users\salsa\espessif\tools\ccache\3.7\
C:\Users\salsa\espessif\tools\dfu-util\0.9\dfu-util-0.9-win64
C:\Users\salsa\Desktop\esp-idf\tools
ZPATH%
Checking if Python packages are up to date...
Python requirements from C:\Users\salsa\Desktop\esp-idf\requirements.txt are satisfied.
Done! You can now compile ESP-IDF projects.
Go to the project directory and run:
idf.py build
PS C:\Users\salsa\Desktop\esp-idf>
```

```
ESP-IDF 4.3 CMD - "C:\Users\salsa\esp-idf\idf_cmd_init.bat" esp-idf-9b1cbe0e7972bc08ea08e3a6588be481
Python 3.8.7
Using Git in C:\Users\salsa\esp-idf\tools\idf-git\2.30.1\cmd\
git version 2.30.1.windows.1
Setting IDF_PATH: C:\Users\salsa\Desktop\esp-idf

Adding ESP-IDF tools to PATH...
C:\Users\salsa\esp-idf\tools\xtensa-esp32-elf\esp-2020r3-8.4.0\xtensa-esp32-elf\bin
C:\Users\salsa\esp-idf\tools\xtensa-esp32s2-elf\esp-2020r3-8.4.0\xtensa-esp32s2-elf\bin
C:\Users\salsa\esp-idf\tools\xtensa-esp32s3-elf\esp-2020r3-8.4.0\xtensa-esp32s3-elf\bin
C:\Users\salsa\esp-idf\tools\riscv32-esp-elf\1.24.0.123_64eb9ff-8.4.0\riscv32-esp-elf\bin
C:\Users\salsa\esp-idf\tools\esp32ulp-elf\2.28.51-esp-20191205\esp32ulp-elf-binutils\bin
C:\Users\salsa\esp-idf\tools\esp32s2ulp-elf\2.28.51-esp-20191205\esp32s2ulp-elf-binutils\bin
C:\Users\salsa\esp-idf\tools\cmake\3.16.4\bin
C:\Users\salsa\esp-idf\tools\openocd-esp32\v0.10.0-esp32-20210401\openocd-esp32\bin
C:\Users\salsa\esp-idf\tools\ninja\1.10.2\
C:\Users\salsa\esp-idf\tools\idf-exe\1.0.1\
C:\Users\salsa\esp-idf\tools\ccache\3.7\
C:\Users\salsa\esp-idf\tools\dfu-util\0.9\dfu-util-0.9-win64
C:\Users\salsa\Desktop\esp-idf\tools

Checking if Python packages are up to date...
Python requirements from C:\Users\salsa\Desktop\esp-idf\requirements.txt are satisfied.

Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

idf.py build
```

Cada vez que se desee trabajar con el entorno, hay que ejecutar el ejecutable correspondiente a ESP-IDF. Al trabajar con Node.js y Visual Studio, este proceso se hace en automático

Durante la instalación se pedirán permisos de instalación del driver, una vez realizado este paso, podrá conectar la ESP32-RWROOM sin complicaciones



4. HOLA MUNDO DEL ESP32-WROOM (PRIMERA VEZ O USO SIN NODE.JS)

A continuación, se probará que la instalación realiza las tareas de compilación y escritura en el ESP32-WROOM correctamente.

- 4.1. Para empezar a trabajar, ejecute una nueva terminal usando el icono de ESP-IDF 4.3 CMD o el respectivo de PowerShell

```
ESP-IDF 4.3 CMD - "C:\esp\tools\esp-idf\idf_cmd_init.bat" esp-idf-4b04d6d91c7fb55e09ae84fcd8d2de
Python 3.8.7
Using Git in C:\esp\tools\esp-idf\tools\idf-git\2.30.1\cmd\
git version 2.30.1.windows.1
Setting IDF_PATH: C:\esp\esp-idf

Adding ESP-IDF tools to PATH...
C:\esp\tools\esp-idf\tools\xtensa-esp32-elf\esp-2020r3-8.4.0\xtensa-esp32-elf\bin
C:\esp\tools\esp-idf\tools\xtensa-esp32s2-elf\esp-2020r3-8.4.0\xtensa-esp32s2-elf\bin
C:\esp\tools\esp-idf\tools\xtensa-esp32s3-elf\esp-2020r3-8.4.0\xtensa-esp32s3-elf\bin
C:\esp\tools\esp-idf\tools\riscv32-esp-elf\1.24.0.123_64eb9ff-8.4.0\riscv32-esp-elf\bin
C:\esp\tools\esp-idf\tools\esp32ulp-elf\2.28.51-esp-20191205\esp32ulp-elf-binutils\bin
C:\esp\tools\esp-idf\tools\esp32s2ulp-elf\2.28.51-esp-20191205\esp32s2ulp-elf-binutils\bin
C:\esp\tools\esp-idf\tools\cmake\3.16.4\bin
C:\esp\tools\esp-idf\tools\openocd-esp32\v0.10.0-esp32-20210401\openocd-esp32\bin
C:\esp\tools\esp-idf\tools\ninja\1.10.2\
C:\esp\tools\esp-idf\tools\idf-exe\1.0.1\
C:\esp\tools\esp-idf\tools\ccache\3.7\
C:\esp\tools\esp-idf\tools\dfu-util\0.9\dfu-util-0.9-win64
C:\esp\esp-idf\tools

Checking if Python packages are up to date...
Python requirements from C:\esp\esp-idf\requirements.txt are satisfied.

Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

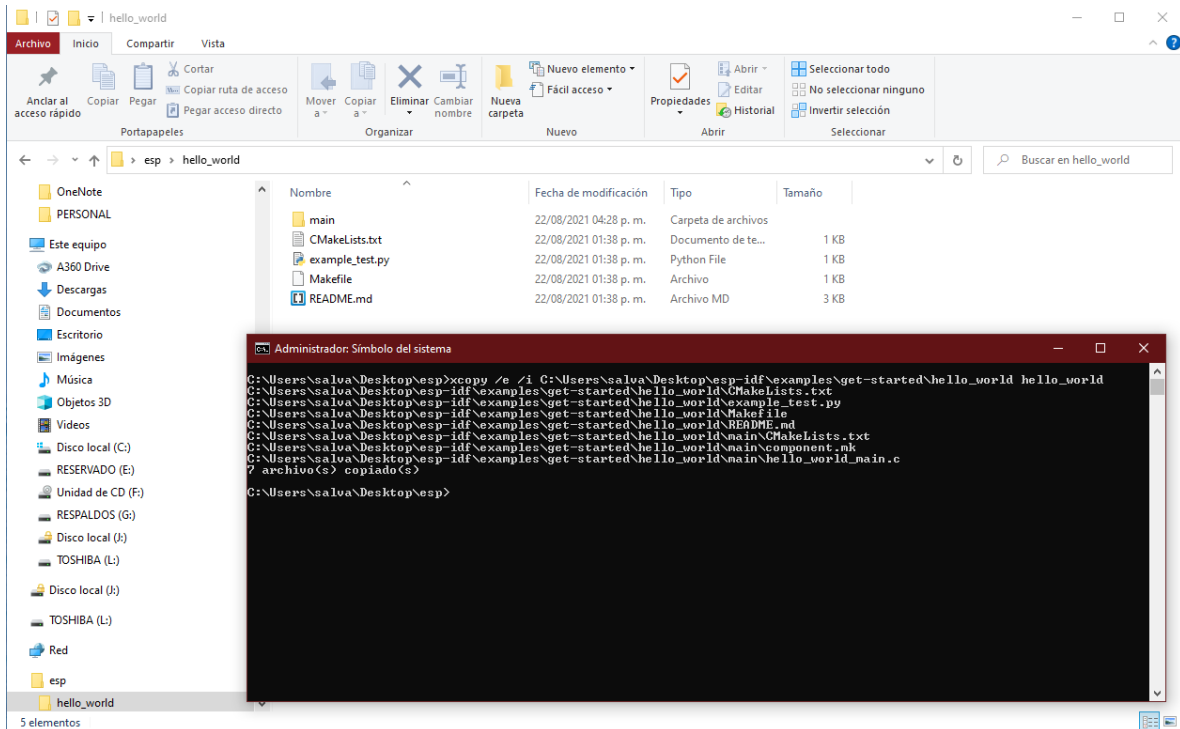
idf.py build

C:\esp\esp-idf>
```

4.2. Use el comando xcopy con los parámetros /e /i para copiar el ejemplo de hola mundo.

El comando completo es:

```
Xcopy /e /i C:\esp\esp-idf\examples\get-started\hello_world  
C:\esp\projects\hello_world
```



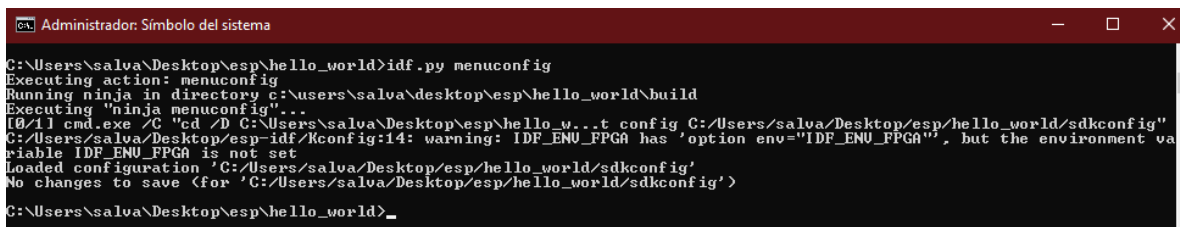
4.3. Dirijase a la ruta de la carpeta copiada y ejecute lo siguiente:

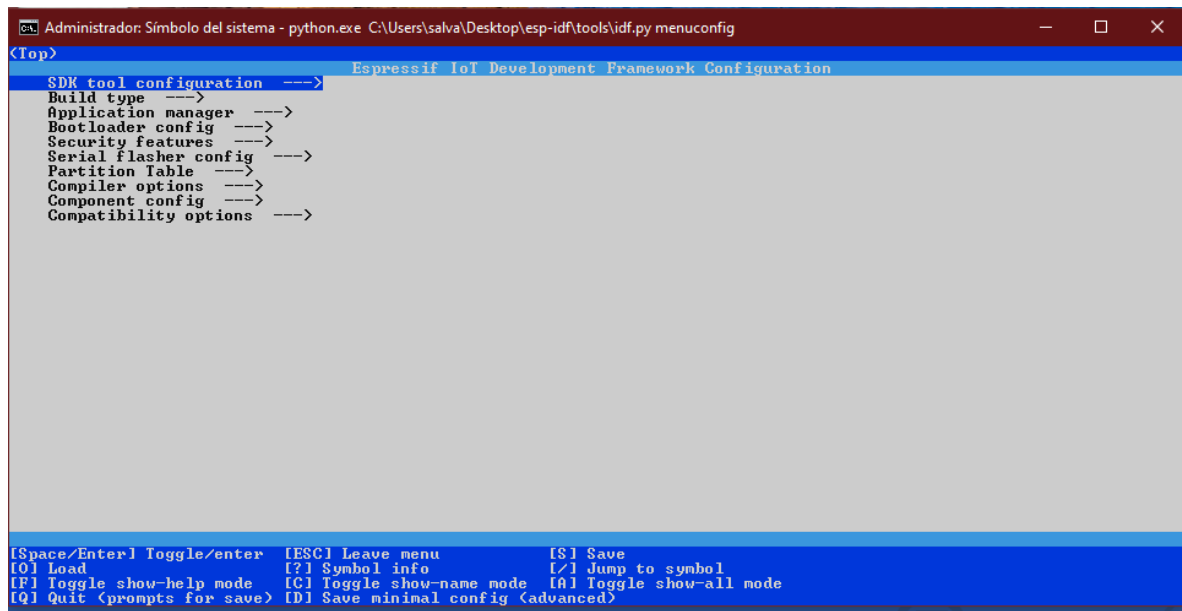
→ `cd C:\esp\projects\hello_world`

→ `idf.py set-target esp32`

```
C:\Users\salsa\Desktop\esp\hello_world>idf.py set-target esp32  
Adding "set-target"'s dependency "fullclean" to list of commands with default set of options.  
Executing action: fullclean  
Build directory 'c:\users\salsa\desktop\esp\hello_world\build' not found. Nothing to clean.  
Executing action: set-target  
Set Target to: esp32, new sdkconfig created. Existing sdkconfig renamed to sdkconfig.old.  
Running cmake in directory c:\users\salsa\desktop\esp\hello_world\build  
Executing 'cmake -G Ninja -DPYTHON_DEPS_CHECKED=1 -DESP_PLATFORM=1 -DIDF_TARGET=esp32 -DCCACHE_ENABLE=1 c:\users\salsa\desktop\esp\hello_world\..'..  
-- Found Git: C:/Program Files/Git/cmd/git.exe (found version "2.33.0.windows.1")
```

→ `idf.py menuconfig` (solo para configuración adicional, no es necesario ejecutarla siempre)





La ventana de menuconfig permite realizar información relacionada con la ESP32-WROOM. Si no se desea realizar algún cambio, se puede omitir este paso.

4.4. Cree el Proyecto con el comando idf.py build

```
idf.py build
```

```

C:\Administrador: Símbolo del sistema
C:\Users\salsa\Desktop>esp\hello_world>idf.py build
Executing action: all (aliases: build)
Running ninja in directory c:\users\salsa\desktop\esp\hello_world\build
Executing "ninja all"...
[0/1] Re-running CMake...
-- cache will be used for faster recompilation
-- Project is not inside a git repository, or git repository has no commits; will not use 'git describe' to determine PROJECT_VERSION
-- Building ESP-IDF components for target esp32
-- Project sdkconfig file C:/Users/salsa/Desktop/esp/hello_world/sdkconfig
-- Could NOT find Perl (missing: PERL_EXECUTABLE)
CMake Warning (dev) at C:/Users/salsa/Desktop/esp-idf/components/mbedtls/CMakeLists.txt:114 (target_sources):
  Policy CMP0076 is not set: target_sources() command converts relative paths
  to absolute. Run "cmake --help-policy CMP0076" for policy details. Use
  the cmake_policy command to set the policy and suppress this warning.

  A private source from a directory other than that of target "mbedcrypto"
  has a relative path.
This warning is for project developers. Use -Wno-dev to suppress it.

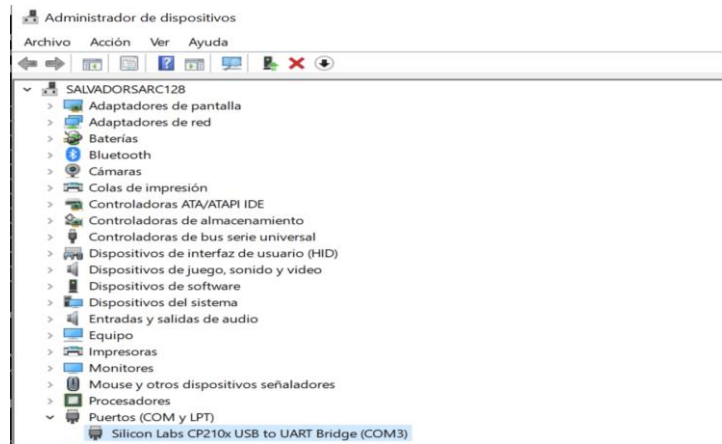
-- App "hello-world" version: 1
-- Adding linker script C:/Users/salsa/Desktop/esp-idf/components/esp_rom/esp32/ld/esp32.rom.ld
-- Adding linker script C:/Users/salsa/Desktop/esp-idf/components/esp_rom/esp32/ld/esp32.rom.api.ld
-- Adding linker script C:/Users/salsa/Desktop/esp-idf/components/esp_rom/esp32/ld/esp32.rom.libgcc.ld
-- Adding linker script C:/Users/salsa/Desktop/esp-idf/components/esp_rom/esp32/ld/esp32.rom.newlib-data.ld
-- Adding linker script C:/Users/salsa/Desktop/esp-idf/components/esp_rom/esp32/ld/esp32.rom.syscalls.ld
-- Adding linker script C:/Users/salsa/Desktop/esp-idf/components/esp_rom/esp32/ld/esp32.rom.newlib-funcs.ld
-- Adding linker script C:/Users/salsa/Desktop/esp-idf/components/esp_rom/esp32/ld/esp32.rom.newlib-time.ld
-- Adding linker script C:/Users/salsa/Desktop/esp/hello_world/build/esp-idf/esp32/esp32_out.ld
-- Adding linker script C:/Users/salsa/Desktop/esp-idf/components/esp32/ld/esp32.project.ld.in
-- Adding linker script C:/Users/salsa/Desktop/esp-idf/components/esp32/ld/esp32.peripherals.ld
-- Components: app_trace app_update asio bootloader bootloader_support bt cbor cmock coap console cxx driver efuse esp-tls esp32 esp32_adc esp32_adc_cal esp32_common esp32_eth esp32_event esp32_gdbstub esp32_hid esp32_http_client esp32_http_server esp32_https_ota esp32_https_server esp_hw_support esp_ipc esp_local_ctrl esp_netif esp_pm esp_ringbuf esp_rom esp_serial_slave_link esp_system esp_timer esp_websocket_client esp_wifi espcoredump esptool.py expat fatfs freemodbus freertos hal heap idf_test jsmn json log lwip main mbedtls mdns mqtt newlib nghttp nvs flash openssl partition_table perfmon protobuf-c protocol_pwm pthread sdmmc soc spi_flash spiffs tcp_transport tcpip_adapter tinyusb ulp unity vfs wear_levelling wifi_provisioning wpa_supplicant xtensa
-- Component paths: C:/Users/salsa/Desktop/esp-idf/components/app_trace C:/Users/salsa/Desktop/esp-idf/components/app_update C:/Users/salsa/Desktop/esp-idf/components/asio C:/Users/salsa/Desktop/esp-idf/components/bootloader C:/Users/salsa/Desktop/esp-idf/components/bootloader_support C:/Users/salsa/Desktop/esp-idf/components/bt C:/Users/salsa/Desktop/esp-idf/components/cbor C:/Users/salsa/Desktop/esp-idf/components/cmock C:/Users/salsa/Desktop/esp-idf/components/coap C:/Users/salsa/Desktop/esp-idf/components/console C:/Users/salsa/Desktop/esp-idf/components/cxx C:/Users/salsa/Desktop/esp-idf/components/driver C:/Users/salsa/Desktop/esp-idf/components/efuse C:/Users/salsa/Desktop/esp-idf/components/esp-tls C:/Users/salsa/Desktop/esp-idf/components/esp32 C:/Users/salsa/Desktop/esp-idf/components/esp32_adc_cal C:/Users/salsa/Desktop/esp-idf/components/esp32_common C:/Users/salsa/Desktop/esp-idf/components/esp32_eth C:/Users/salsa/Desktop/esp-idf/components/esp32_event C:/Users/salsa/Desktop/esp-idf/components/esp32_gdbstub C:/Users/salsa/Desktop/esp-idf/components/esp32_hid C:/Users/salsa/Desktop/esp-idf/components/esp32_http_client C:/Users/salsa/Desktop/esp-idf/components/esp32_http_server C:/Users/salsa/Desktop/esp-idf/components/esp32_https_ota C:/Users/salsa/Desktop/esp-idf/components/esp32_https_server C:/Users/salsa/Desktop/esp-idf/components/esp_hw_support C:/Users/salsa/Desktop/esp-idf/components/esp_ipc C:/Users/salsa/Desktop/esp-idf/components/esp_local_ctrl C:/Users/salsa/Desktop/esp-idf/components/esp_netif C:/Users/salsa/Desktop/esp-idf/components/esp_pm C:/Users/salsa/Desktop/esp-idf/components/esp_ringbuf C:/Users/salsa/Desktop/esp-idf/components/esp_rom C:/Users/salsa/Desktop/esp-idf/components/esp_serial_slave_link C:/Users/salsa/Desktop/esp-idf/components/esp_system C:/Users/salsa/Desktop/esp-idf/components/esp_timer C:/Users/salsa/Desktop/esp-idf/components/esp_websocket_client C:/Users/salsa/Desktop/esp-idf/components/esp_wifi C:/Users/salsa/Desktop/esp-idf/components/espcoredump C:/Users/salsa/Desktop/esp-idf/components/esptool.py
[186/186] Generating binary image from built executable
esptool.py v3.1-dev
Merged 1 ELF section
Generated C:/Users/salsa/Desktop/esp/hello_world/build/bootloader/bootloader.bin
[990/990] Generating binary image from built executable
esptool.py v3.1-dev
Merged 1 ELF section
Generated C:/Users/salsa/Desktop/esp/hello_world/build/hello-world.bin

Project build complete. To flash, run this command:
C:\Users\salsa>esptool.py python .\env\idf4.3_py3.8\env\Scripts\python.exe .\..\esp-idf\components\esptool.py\esptool.py
esptool.py -p (PORT) -b 460800 --before default_reset --after hard_reset --chip esp32 write_flash --flash_mode dio --flash_size detect --flash_freq 40m 0x1000 build\bootloader\bootloader.bin 0x8000 build\partition_table\partition-table.bin 0x10000 build\hello-world.bin
or run 'idf.py -p (PORT) flash'

```

Al finalizar el proceso, aparecerá un mensaje que indica la ejecución del siguiente comando

- 4.5. Para grabar el programa en el ESP32-WROOM se usa el siguiente comando (Ver puerto COM del dispositivo en el administrador de dispositivos)



`idf.py -p PORT [-b BAUD] flash`

Se puede configurar el “rate baud” reemplazando la palabra BAUD por el valor deseado. Por defecto es 460800.

```

C:\Users\salsa\Desktop\esp\hello_world>idf.py -p COM3 flash
Executing action: flash
Running ninja in directory c:\users\salsa\Desktop\esp\hello_world\build
Executing "ninja flash"...
[1/4] Performing build step for 'bootloader'
ninja: no work to do.
[1/2] cmd.exe /C "cd /D C:\Users\salsa\Desktop\esp-idf\com...sktop/esp-idf/components/esptool_py/run_serial_tool.cmake"
esptool.py esp32 -p COM3 -b 460800 --before=default_reset --after=hard_reset write_flash --flash_mode dio --flash_freq 4
0m --flash_size 2MB 0x8000 partition_table/partition-table.bin 0x1000 bootloader/bootloader.bin 0x10000 hello-world.bin
esptool.py v3.1-dev
Serial port COM3
Connecting...
Chip is ESP32-D0WD06 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: ac:67:b2:36:f1:28
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Flash will be erased from 0x00000000 to 0x00000fff...
Flash will be erased from 0x00001000 to 0x00007fff...
Flash will be erased from 0x00001000 to 0x00007fff...
Compressed 3072 bytes to 103...
Writing at 0x00000000... (100 %)
Wrote 3072 bytes (103 compressed) at 0x00000000 in 0.2 seconds (effective 115.4 kbit/s)...
Hash of data verified.
Compressed 25072 bytes to 15383...
Writing at 0x00001000... (100 %)
Wrote 25072 bytes (15383 compressed) at 0x00001000 in 0.8 seconds (effective 238.2 kbit/s)...
Hash of data verified.
Compressed 161488 bytes to 84535...
Writing at 0x00010000... (16 %)
Writing at 0x0001ab6a... (33 %)
Writing at 0x0002031e... (50 %)
Writing at 0x00025b7c... (66 %)
Writing at 0x0002f2d9... (83 %)
Writing at 0x00036602... (100 %)
Wrote 161488 bytes (84535 compressed) at 0x00010000 in 2.7 seconds (effective 478.2 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
Done

```

- 4.6. Para monitorear la salida del comando, se puede hacer uso del comando `idf.py monitor`, especificando el puerto con la opción `-p PORT`

`$ idf.py -p COM3 monitor`

- 4.7. Para terminar el proceso del monitor, se puede usar la combinación de teclas `CTRL+C` o `Windows+C`

```
Hello world!
This is esp32 chip with 2 CPU core(s), WiFi/BT/BLE, silicon revision 1, 2MB external flash
Minimum free heap size: 291544 bytes
Restarting in 10 seconds...
Restarting in 9 seconds...
Restarting in 8 seconds...
Restarting in 7 seconds...

Traceback (most recent call last):
  File "C:\Users\salsa\Desktop\esp-idf\tools\idf_monitor.py", line 569, in main_loop
    item = self.cmd_queue.get_nowait()
  File "queue.py", line 198, in get_nowait
  File "queue.py", line 167, in get
_queue.Empty

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\Users\salsa\Desktop\esp-idf\tools\idf_monitor.py", line 1318, in <module>
    main()
  File "C:\Users\salsa\Desktop\esp-idf\tools\idf_monitor.py", line 1131, in main
    monitor.main_loop()
  File "C:\Users\salsa\Desktop\esp-idf\tools\idf_monitor.py", line 572, in main_loop
    item = self.event_queue.get(True, 0.03)
  File "queue.py", line 179, in get
  File "threading.py", line 306, in wait
KeyboardInterrupt

Aborted!
C:\Users\salsa\Desktop\esp\hello_world>
```

El resultado del monitor sería como la siguiente:


```
Administrator: Símbolo del sistema
C:\Users\salsa\Desktop\esp\hello_world>idf.py -p COM3 monitor
Executing action: monitor
Running idf_monitor in directory c:\users\salsa\Desktop\esp\hello_world
Executing "C:\Users\salsa\esp\python_env\idf4.3_py3.8_env\Scripts\python.exe C:\Users\salsa\Desktop\esp-idf\tools\idf_monitor.py -p COM3 -b 115200 --toolchain-prefix xtensa-esp32-elf- c:\users\salsa\Desktop\esp\hello_world\build\hello-world.elf -m 'C:\Users\salsa\esp\python_env\idf4.3_py3.8_env\Scripts\python.exe' 'C:\Users\salsa\Desktop\esp-idf\tools\idf.py' -p 'COM3'..."
<I0:33n--- WARNING: GDB cannot open serial ports accessed as COMx+{0m
<I0:33n--- Using \\.\COM3 instead. <{0m
--- idf_monitor on \\.\COM3 115200 ---
--- Quit: Ctrl+I ! Menu: Ctrl+T ! Help: Ctrl+T followed by Ctrl+H ---
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
flash read err, 1000
ets_main.c 371
ets Jun  8 2016 00:22:57

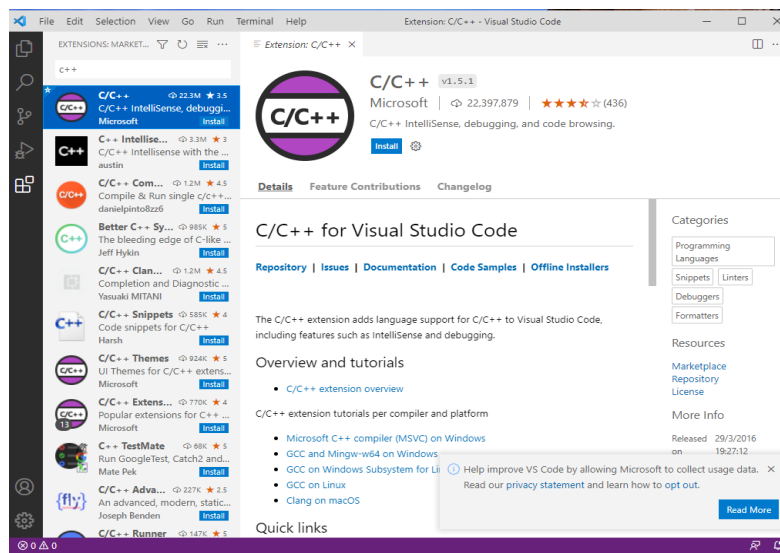
rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config:ip: 0, SPIWP:0xee
clk_drv:0x00, d_drv:0x00, d_drv:0x00, cs0_drv:0x00, hd_drv:0x00, wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030, len:7008
load:0x40078000, len:14292
ho 0 tail 12 room 4
load:0x40080400, len:3688
0x40080400: _init at ???:
entry 0x40080678
I (29) boot: ESP-IDF v4.3 2nd stage bootloader
I (29) boot: compile time 10:07:34
I (29) boot: chip revision: 1
I (32) boot_comm: chip revision: 1, min. bootloader chip revision: 0
I (39) boot.esp32: SPI Speed      : 40MHz
I (43) boot.esp32: SPI Mode       : DIO
I (48) boot.esp32: SPI Flash Size : 2MB
I (52) boot: Enabling RNG early entropy source...
I (58) boot: Partition Table:
I (61) boot: ## Label                Usage            Type ST Offset   Length
I (69) boot: 0 nvs                   WiFi data        01 02 00009000  00006000
I (76) boot: 1 phy_init              RF data          01 01 0000f000  00001000
I (84) boot: 2 factory               factory app      00 00 00010000  00100000
I (91) boot: End of partition table
I (95) boot_comm: chip revision: 1, min. application chip revision: 0
I (102) esp_image: segment 0: paddr=00010020 vaddr=3f400020 size=06a14h ( 27156) map
I (121) esp_image: segment 1: paddr=00016a3c vaddr=3ffb0000 size=02900h ( 10496) load
I (126) esp_image: segment 2: paddr=00019344 vaddr=40080000 size=06cd4h ( 27860) load
I (148) esp_image: segment 3: paddr=00020020 vaddr=40040020 size=138c4h ( 80060) map
I (170) esp_image: segment 4: paddr=000338ec vaddr=40086cd4 size=03da8h ( 15784) load
I (177) esp_image: segment 5: paddr=0003769c vaddr=50000000 size=00010h ( 16) load
I (184) boot: Loaded app from partition at offset 0x10000
I (184) boot: Disabling RNG early entropy source...
I (199) cpu_start: Pro cpu up.
I (199) cpu_start: Starting app cpu, entry point is 0x40080fd8
0x40080fd8: call_start_cpu1 at C:\Users\salsa\Desktop\esp-idf/components/esp_system/port/cpu_start.c:141
I (0) cpu_start: App cpu up.
I (213) cpu_start: Pro cpu start user code
I (213) cpu_start: cpu freq: 160000000
I (213) cpu_start: Application information:
I (218) cpu_start: Project name:      hello-world
I (223) cpu_start: App version:      1
I (227) cpu_start: Compile time:      Aug 23 2021 10:06:54
I (234) cpu_start: ELF file SHA256:   d7179620b500fc49...
I (240) cpu_start: ESP-IDF:          v4.3
I (245) heap_init: Initializing. RAM available for dynamic allocation:
I (252) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (258) heap_init: At 3FFB3128 len 0002CED8 (179 KiB): DRAM
I (264) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (270) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (277) heap_init: At 4008AA7C len 00015584 (85 KiB): IRAM
I (284) spi_flash: detected chip: generic
I (287) spi_flash: flash io: dio
W (291) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the size in the
binary image header.
I (306) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
Hello world!
This is esp32 chip with 2 CPU core(s), WiFi/BT/BLE, silicon revision 1, 2MB external flash
Minimum free heap size: 291544 bytes
Restarting in 10 seconds...
Restarting in 9 seconds...
Restarting in 8 seconds...
Restarting in 7 seconds...
```

5. INSTALACIÓN DEL COMPLEMENTO ESP32 A VISUAL STUDIO CODE

El uso de Visual Studio Code permite el trabajar en una interfaz especializada en el desarrollo de código. El uso de IntelligentSense permite identificar fácilmente errores de sintaxis, errores de identificación de rutas y un fácil acceso a herramientas de depuración, compilación y/o ejecución de código.

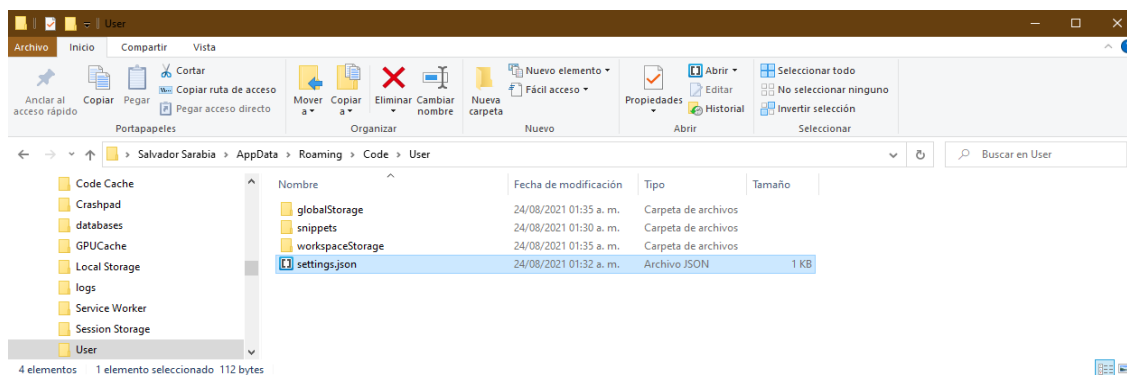
5.1. Descargue el instalador de Visual Studio Code en: <https://code.visualstudio.com/>
Seguir las instrucciones del asistente para instalarlo.

5.2. Una vez instalado y ejecutado el programa, ir a la sección de extensiones, “el icono de un conjunto de bloques y uno separado del resto”, en el cuadro de texto con el símbolo de una lupa, escribir C/C++. Instalar el primer complemento tal y como se muestra en la imagen

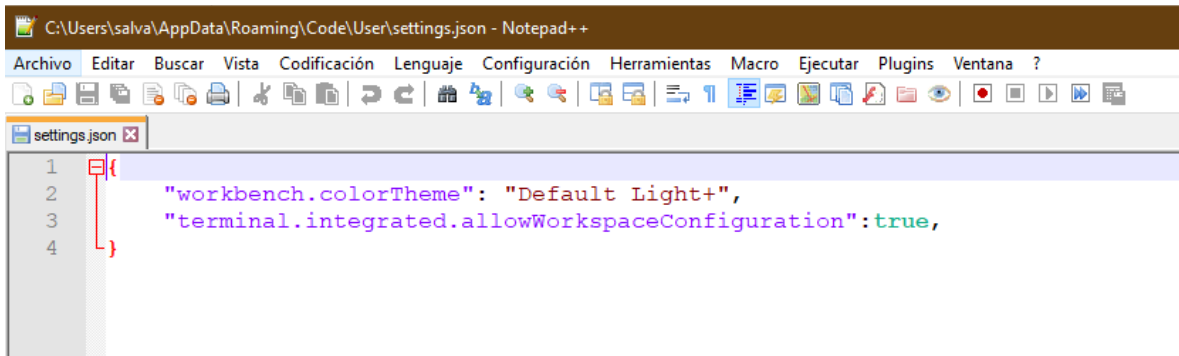


5.3. Se procederá a hacer una modificación en el archivo setting.json de la instalación de Visual Studio del usuario. Abrir el archivo con Bloc de Notas (o Notepad++) para editar.

La ruta usualmente es: C:\Users\nombre_usuario\AppData\Roaming\Code\User
Usualmente la carpeta AppData esta oculta. Compruebe la ruta del archivo según: https://code.visualstudio.com/docs/getstarted/settings#_settings-file-locations



5.4. Editar el archivo tal y como se muestra en la imagen de este punto.
Agregar la línea:
"terminal.integrated.allowWorkspaceConfiguration":true,

A screenshot of the Notepad++ application window. The title bar shows the file path: C:\Users\salsa\AppData\Roaming\Code\User\settings.json - Notepad++. The menu bar includes Archivo, Editar, Buscar, Vista, Codificación, Lenguaje, Configuración, Herramientas, Macro, Ejecutar, Plugins, Ventana, and ?. The toolbar contains various icons for file operations and editing. The main text area shows the contents of settings.json, which is a JSON object with two properties: "workbench.colorTheme" and "terminal.integrated.allowWorkspaceConfiguration". The file is named settings.json, as shown in the tab. Line numbers 1 through 4 are visible on the left margin. A red bracket highlights the opening and closing curly braces of the JSON object.

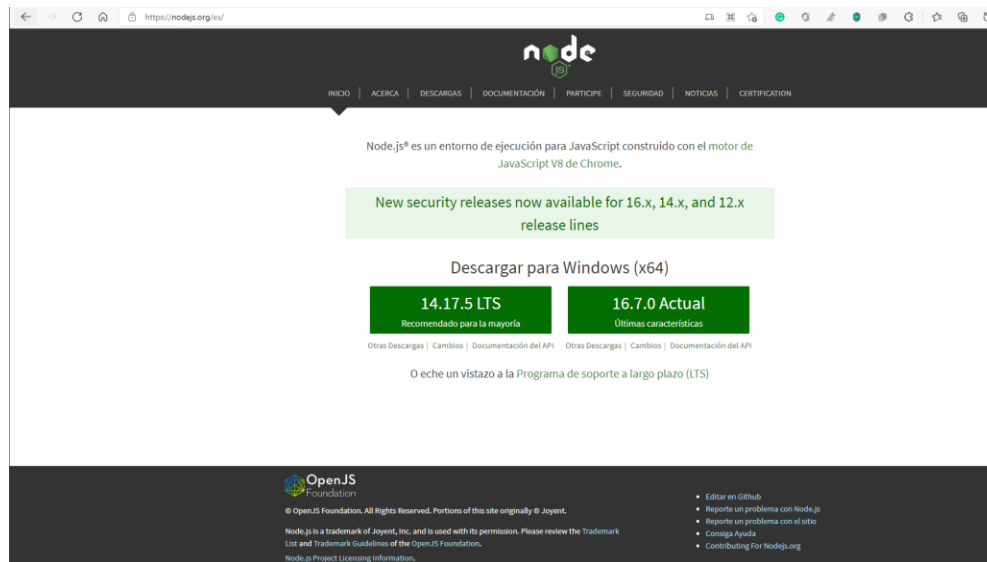
```
1 {  
2   "workbench.colorTheme": "Default Light+",  
3   "terminal.integrated.allowWorkspaceConfiguration":true,  
4 }
```

6. INSTALACIÓN DE NODE JS

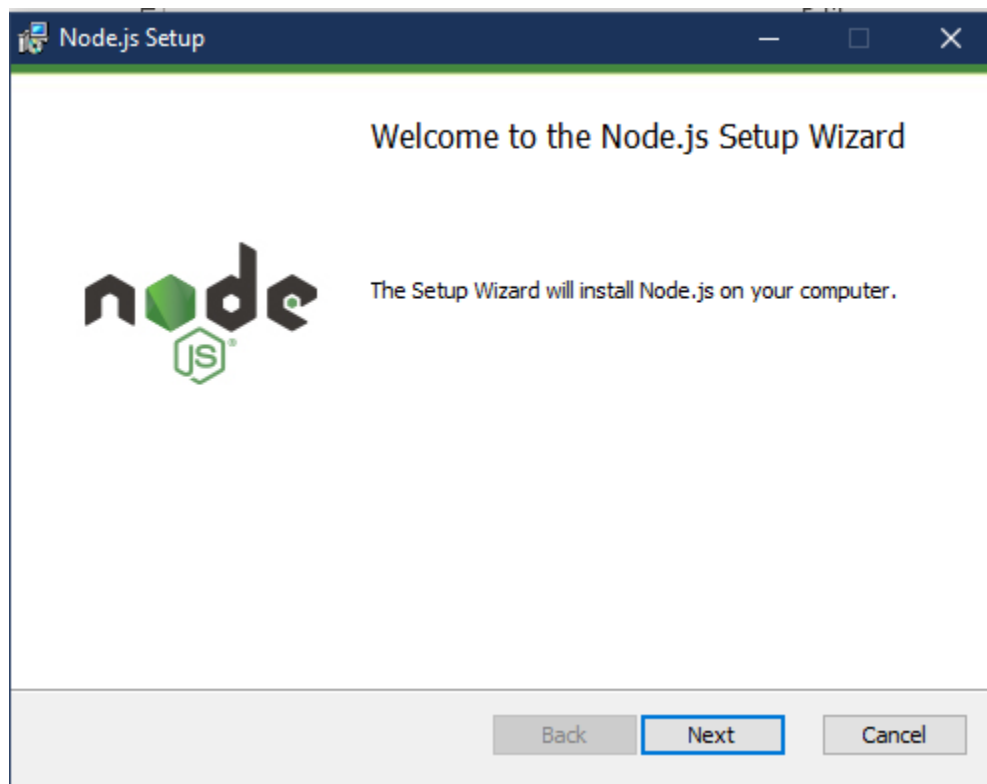
El soporte de NODEJS es para la creación de proyectos nuevos para ESP-IDF así como proporcionar las variables de entorno necesarios para la integración con Visual Studio Code.

- 6.1. Dirijase al sitio de descarga de nodejs y elegir la versión que diga Actual (Para esta fecha 04/09/2021 16.7.0)

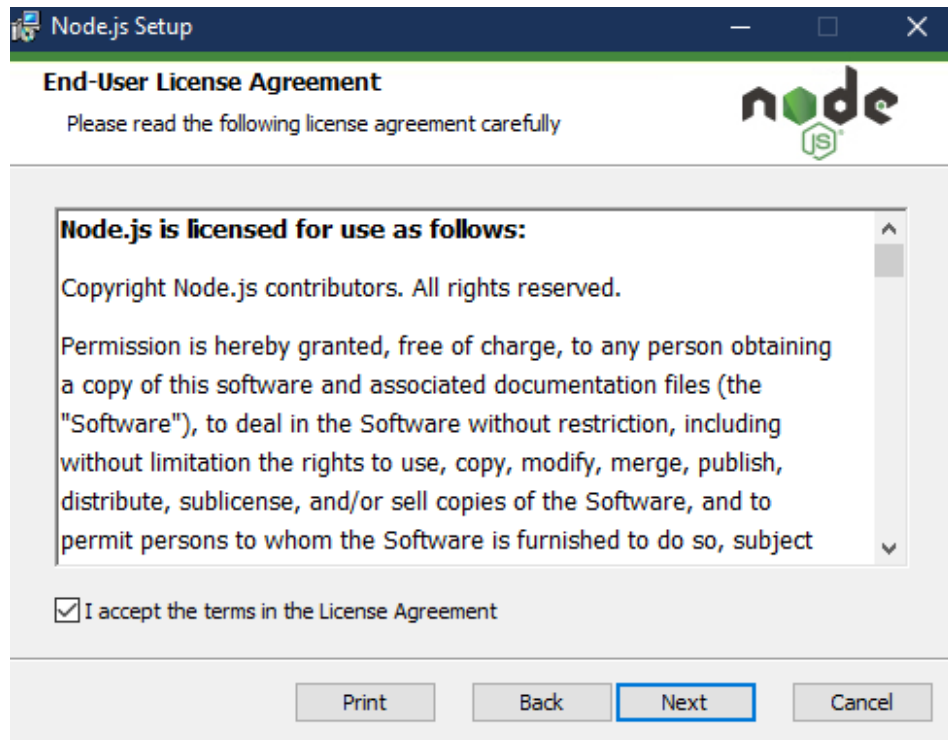
<https://nodejs.org/es/>



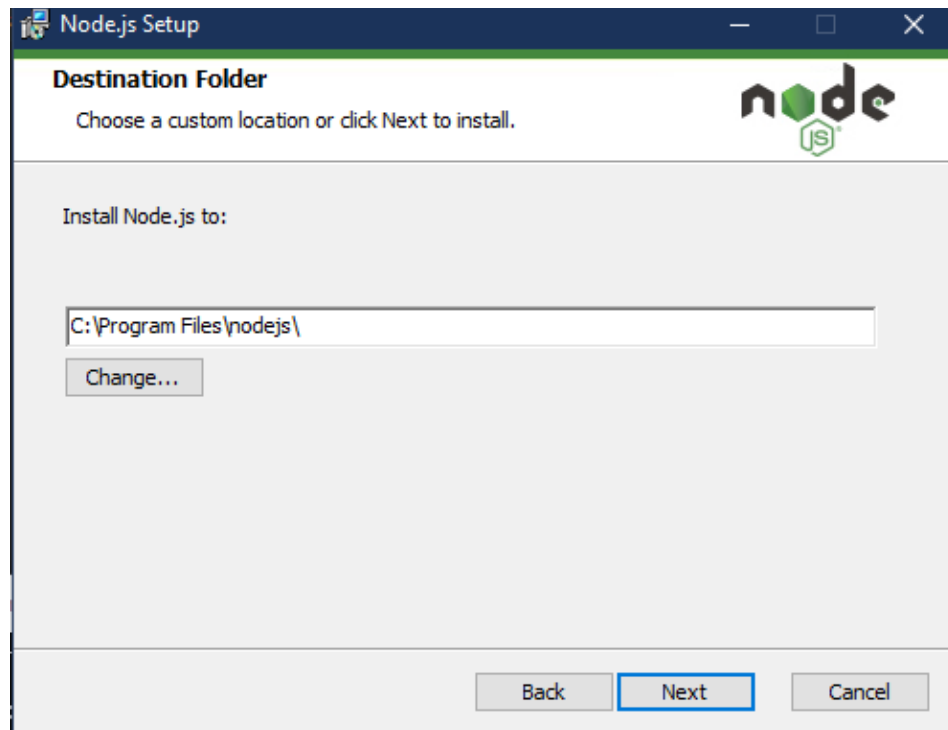
- 6.2. Una vez descargado, dar clic derecho y elegir ejecutar



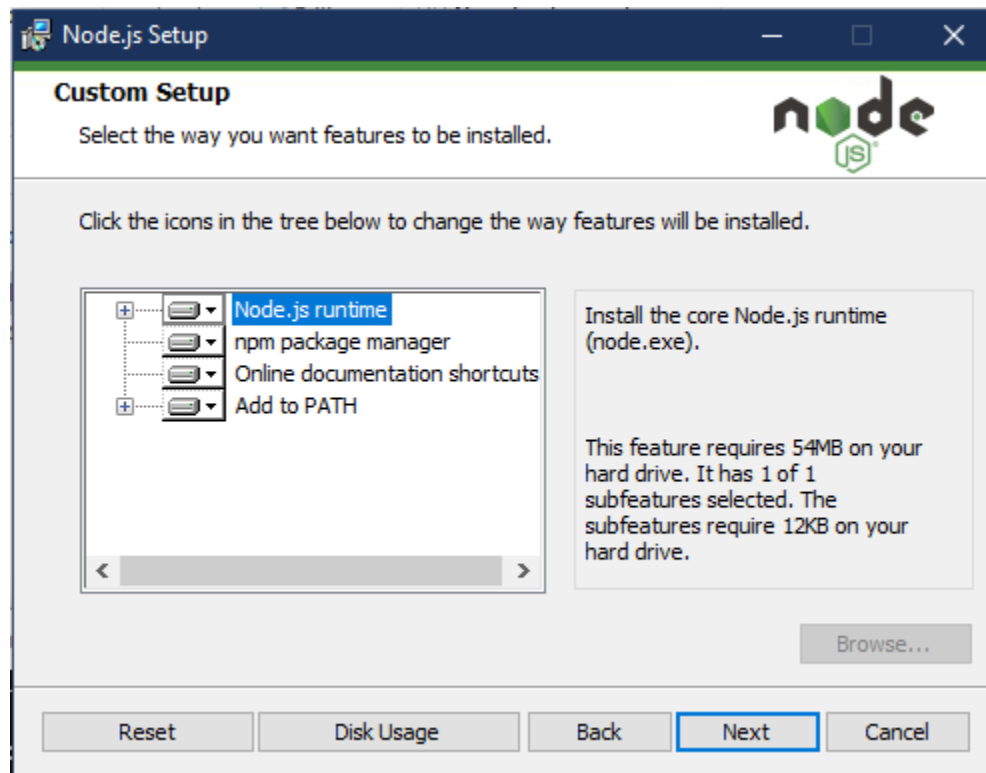
6.3. Aceptar el acuerdo de licencia y dar clic en continuar



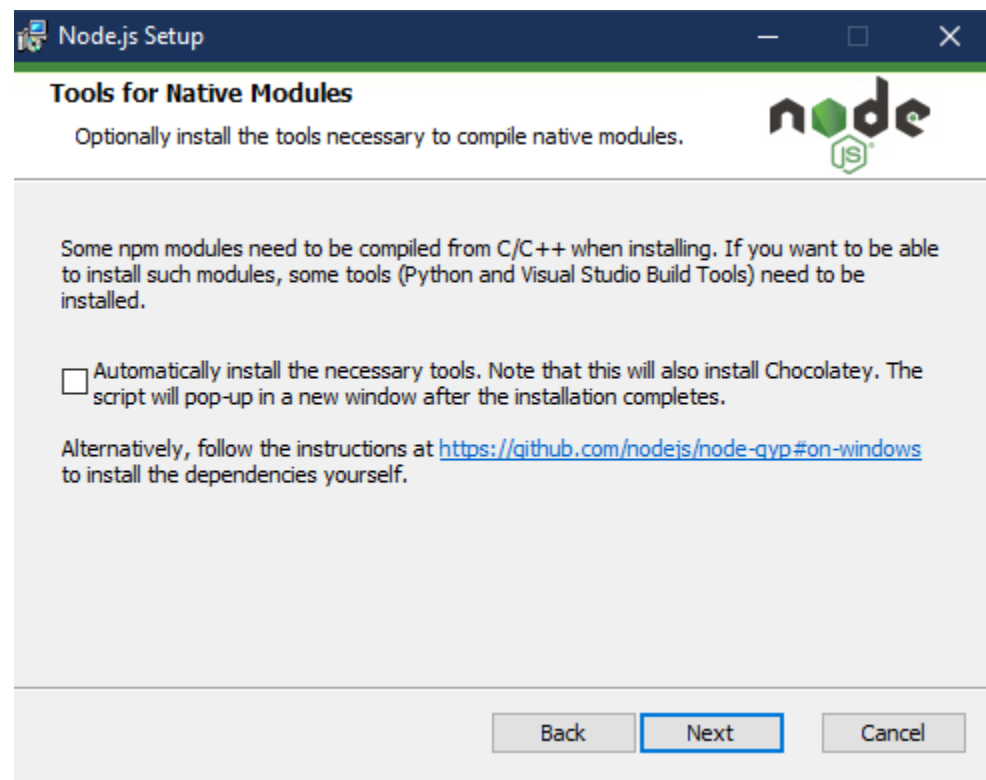
6.4. Elegir la ruta predeterminada para instalar el programa, dar clic en "siguiente".



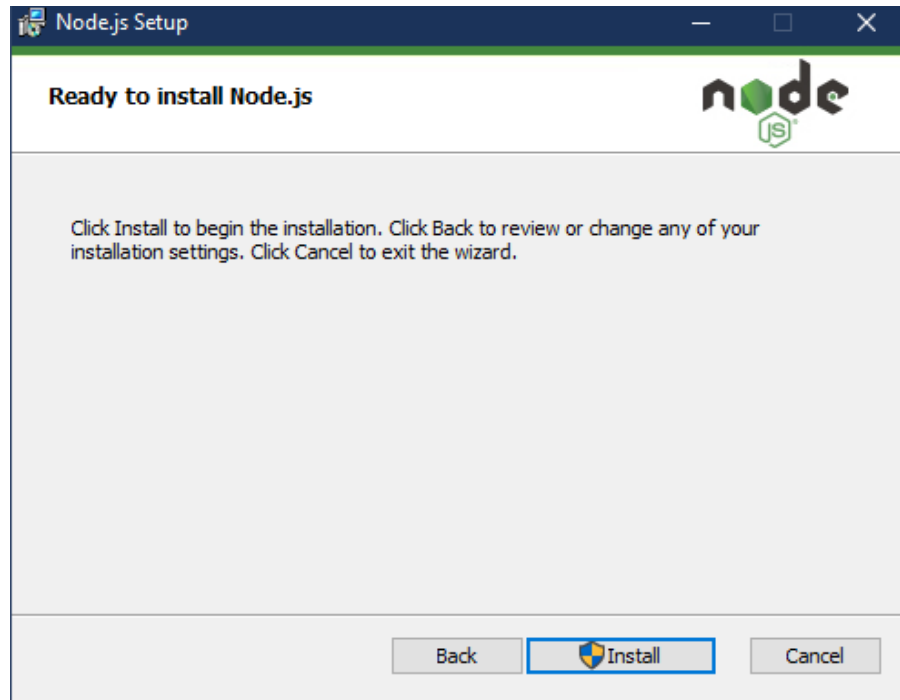
6.5. Elegir la configuración predeterminada y dar clic en siguiente



6.6. Si así lo desea marcar la casilla “Instalar automáticamente las herramientas necesarias”, dar clic en siguiente.



- 6.7. Finalmente se mostrará un mensaje de “preparado para instalar”. Dar clic en “instalar”

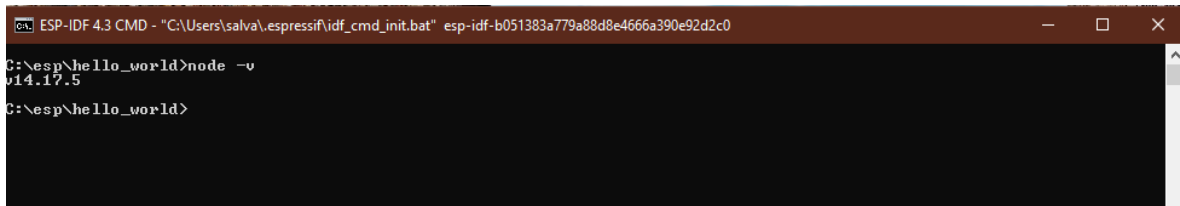


- 6.8. Cuando se terminen de instalar todos los componentes, se verán los siguientes iconos indicando los programas necesarios para empezar a trabajar.



7. PRUEBA DE TRABAJO DE TODA LA INSTALACIÓN

- 7.1. Ejecutar el entorno de trabajo de Node.js del icono de Node.js command
 - 7.1.1. Para probar que se está ejecutando Node.js, ejecutar el comando:
node -v



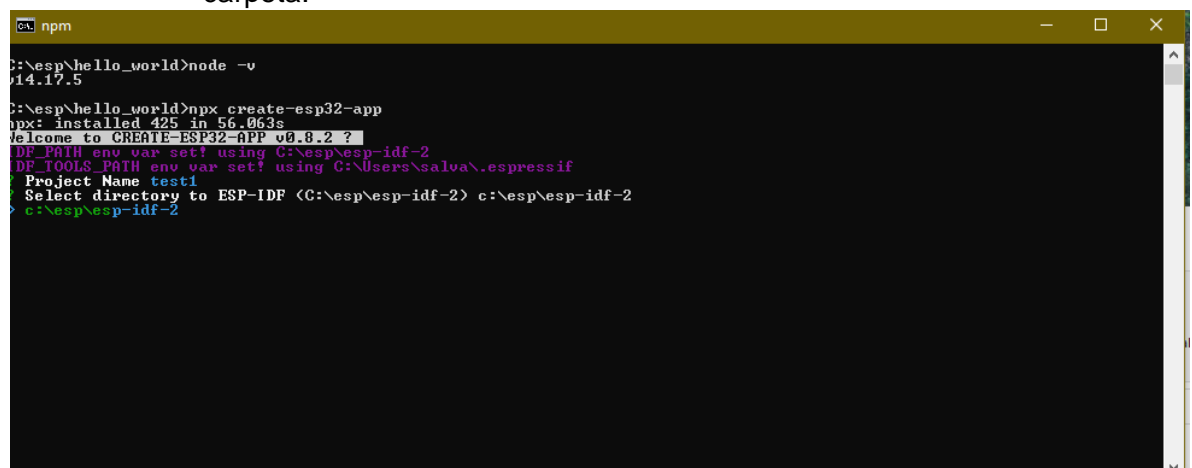
```
ESP-IDF 4.3 CMD - "C:\Users\salsa\esp\idf_cmd_init.bat" esp-idf-b051383a779a88d8e4666a390e92d2c0
C:\esp\hello_world>node -v
v14.17.5
C:\esp\hello_world>
```

- 7.2. Escriba la ruta de la carpeta donde desee trabajar el nuevo proyecto con el comando cd. Preferentemente usar C:\esp\projects
 - 7.2.1. Una vez en la carpeta de proyectos escribir el comando:
npx create-esp32-app
Se ejecutará un script.



```
npm
C:\esp\hello_world>node -v
v14.17.5
C:\esp\hello_world>npx create-esp32-app
npx: installed 425 in 56.063s
Welcome to CREATE-ESP32-APP v0.8.2 ?
IDF_PATH env var set! using C:\esp\esp-idf-2
IDF_TOOLS_PATH env var set! using C:\Users\salsa\esp\idf
? Project Name
```

- 7.3. Para crear un Proyecto aparecerán los siguientes puntos:
 - 7.3.1. Project Name: Escriba el nombre que desee
 - 7.3.2. Elegir la carpeta de ESP-IDF. Si se da enter se elige el que está por defecto, si no escriba la ruta o navegue hasta que aparezca la carpeta.



```
npm
C:\esp\hello_world>node -v
v14.17.5
C:\esp\hello_world>npx create-esp32-app
npx: installed 425 in 56.063s
Welcome to CREATE-ESP32-APP v0.8.2 ?
IDF_PATH env var set! using C:\esp\esp-idf-2
IDF_TOOLS_PATH env var set! using C:\Users\salsa\esp\idf
? Project Name test1
Select directory to ESP-IDF <C:\esp\esp-idf-2> c:\esp\esp-idf-2
> c:\esp\esp-idf-2
```

- 7.3.3. Elegir la ruta de Espressif Tools. Si se da enter se elige el que está por defecto, si no escriba la ruta o navegue hasta que aparezca la carpeta.


```
npm
C:\esp\hello_world>node -v
v14.17.5
C:\esp\hello_world>npx create-esp32-app
npx: installed 425 in 56.063s
Welcome to CREATE-ESP32-APP v0.8.2 ?
IDF_PATH env var set! using C:\esp\esp-idf-2
IDF_TOOLS_PATH env var set! using C:\Users\salva\.espressif
? Project Name test1
? Select directory to ESP-IDF c:\esp\esp-idf-2
? Select directory to espressif Tools (Xtensa tools\espressif\ directory) C:\Users\salva\.espressif
c:\Windows\WinSxS
> C:\Users\salva\.espressif
c:\
c:\$Recycle.Bin
c:\$Recycle.Bin\S-1-5-21-2535402163-1783148562-3686427778-1001
c:\$WINDOWS.BT
c:\$WINDOWS.BT\Sources
(Move up and down to reveal more choices)
```

7.4. Cuando le pregunte el usar C++, escriba No.

```
npm
C:\esp\hello_world>node -v
v14.17.5
C:\esp\hello_world>npx create-esp32-app
npx: installed 425 in 56.063s
Welcome to CREATE-ESP32-APP v0.8.2 ?
IDF_PATH env var set! using C:\esp\esp-idf-2
IDF_TOOLS_PATH env var set! using C:\Users\salva\.espressif
? Project Name test1
? Select directory to ESP-IDF c:\esp\esp-idf-2
? Select directory to espressif Tools (Xtensa tools\espressif\ directory) C:\Users\salva\.espressif
? Use C++ (default is c [press Enter]) Yes
? Select additional sample code (Press <space> to select, <a> to toggle all, <i> to invert selection)
< > blinky [example: blink led]
< > c++ [example: c++ starter]
< > debug [debug cfg files]
< > example connect [example: connect to internet]
< > menuconfig [example: config menu with idf.py menuconfig]
```

7.5. En la sección “elegir elementos adicionales” elija la información que desee, para este ejemplo se puede usar “LED Blink”. Dar enter.

```
ESP-IDF 4.3 CMD - "C:\Users\salva\.espressif\idf_cmd_init.bat" esp-idf-b051383a779a88d8e4666a390e92d2c0
C:\esp\hello_world>npx create-esp32-app
npx: installed 425 in 56.063s
Welcome to CREATE-ESP32-APP v0.8.2 ?
IDF_PATH env var set! using C:\esp\esp-idf-2
IDF_TOOLS_PATH env var set! using C:\Users\salva\.espressif
? Project Name test1
? Select directory to ESP-IDF c:\esp\esp-idf-2
? Select directory to espressif Tools (Xtensa tools\espressif\ directory) C:\Users\salva\.espressif
? Use C++ (default is c [press Enter]) Yes
? Select additional sample code
answers {
  projectName: 'test1',
  idfPath: 'c:\esp\esp-idf-2',
  toolsPath: 'C:\Users\salva\.espressif',
  isCpp: true,
  pythonDir: 'C:\Users\salva\.espressif\tools\idf-python\3.8.7\Scripts',
  additions: []
}
Generating Template with name"test1"
Success
see read me for more information or visit us on
https://learnesp32.com
Warning! As of vs code v1.56.1 you must add
"terminal.integrated.allowWorkspaceConfiguration":true,
to your user settings for the integrated terminal to work. see https://github.com/Mair/create-esp32-app/issues/10
please navigate to your new project and open it in vscode
cd test1
code .
C:\esp\hello_world>
```

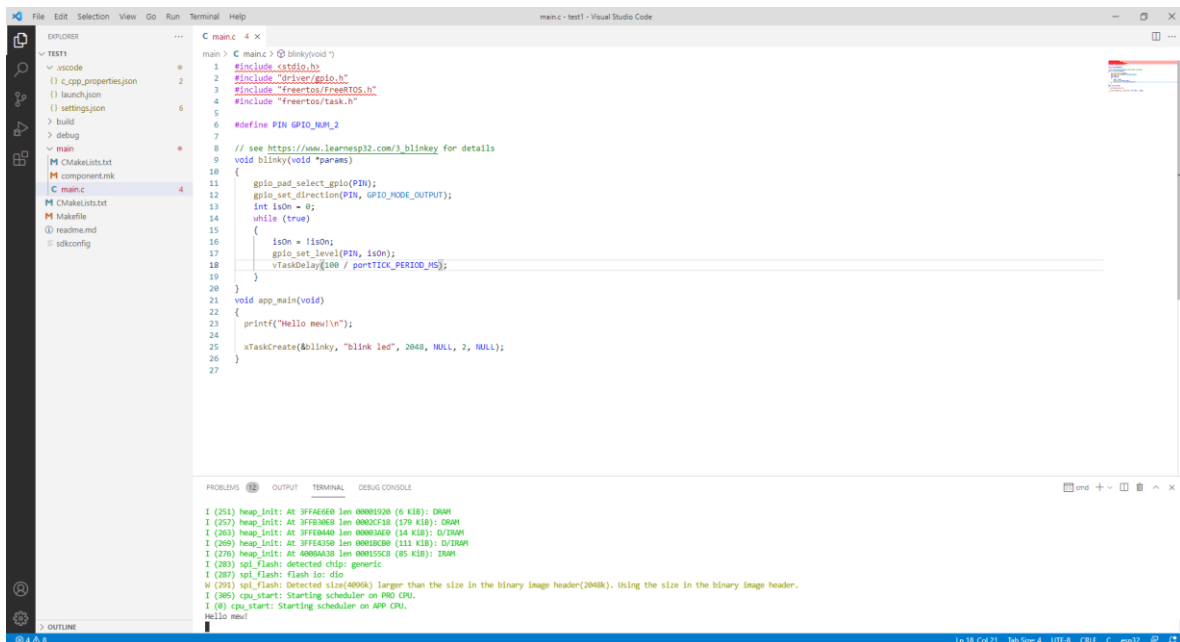
7.6. Una vez terminado el proceso de creación, diríjase a la carpeta del nuevo proyecto, usualmente cd nombre_proyecto

7.7. Una vez dentro del directorio, escribir el comando code .

Se inicia la ventana de Visual Studio configurada para ese proyecto.

```
Simbolo del sistema
idf_tools_path env var set? using C:\Users\salsa\.espressif
? Project Name test1
? Select directory to ESP-IDF C:\esp
? Select directory to espressif Tools (Xtensa tools[.espressif] directory) C:\Users\salsa\.espressif
? Use C++ (default is c [press Enter]) No
? Select additional sample code
answers {
  projectName: 'test1',
  idfPath: 'C:\esp',
  toolsPath: 'C:\Users\salsa\.espressif',
  isCpp: false,
  pythonDir: 'C:\Users\salsa\.espressif\tools\idf-python\3.8.7\Scripts',
  additions: []
}
Generating Template with name"test1"
Success
see read me for more information or visit us on
https://learnesp32.com
Warning! As of vs-code v1.56.1 you must add
'terminal.integrated.allowWorkspaceConfiguration':true,
to your user settings for the integrated terminal to work. see https://github.com/Mair/create-esp32-app/issues/10
please navigate to your new project and open it in vscode
cd test1
code .
C:\esp\hello_world>cd test1
C:\esp\hello_world\test1>code .
C:\esp\hello_world\test1>
```

7.8. Si todo sale correctamente y tiene conectado el ESP32 al programa, puede seleccionar la opción “Ejecutar” de la barra de menú para realizar la compilación automática del proyecto y ver el comportamiento del código en la terminal.



```
main.c 4 X
main > C main.c > @ blinky(void)
1 #include <stdio.h>
2 #include "driver/gpio.h"
3 #include "freertos/FreeRTOS.h"
4 #include "freertos/task.h"
5 #define PIN GPIO_NUM_2
6
7
8 // see https://www.learnesp32.com/3_blinky for details
9 void blinky(void *params)
10 {
11     gpio_pad_select_gpio(PIN);
12     gpio_set_direction(PIN, GPIO_MODE_OUTPUT);
13     int isOn = 0;
14     while (true)
15     {
16         isOn = !isOn;
17         gpio_set_level(PIN, isOn);
18         vTaskDelay(100 / portTICK_PERIOD_MS);
19     }
20 }
21 void app_main(void)
22 {
23     printf("Hello me!\n");
24
25     xTaskCreate(&blinky, "blink led", 2048, NULL, 2, NULL);
26 }
27

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
I (251) heap_init: At 3FFA0500 len 00001000 (4 KiB): DRAM
I (257) heap_init: At 3FFB0600 len 0000CF10 (179 KiB): DRAM
I (263) heap_init: At 3FFC0400 len 00003400 (14 KiB): D/2RAM
I (269) heap_init: At 3FFC4500 len 00003C00 (11 KiB): D/2RAM
I (276) heap_init: At 40000A38 len 000155C8 (85 KiB): IRAM
I (283) spi_flash: detected chip: generic
I (287) spi_flash: flash io: dio
W (291) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the size in the binary image header.
I (305) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
Hello me!
```

REFERENCIAS

- <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html#get-started-get-prerequisites>
- <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/vscode-setup.html>
- <https://github.com/espressif/vscode-esp-idf-extension/blob/master/docs/tutorial/toc.md>
- <https://nodejs.org/es/>