



Tutorial 1: Como integrar Wowki en Visual Studio Code

Guia oficial:

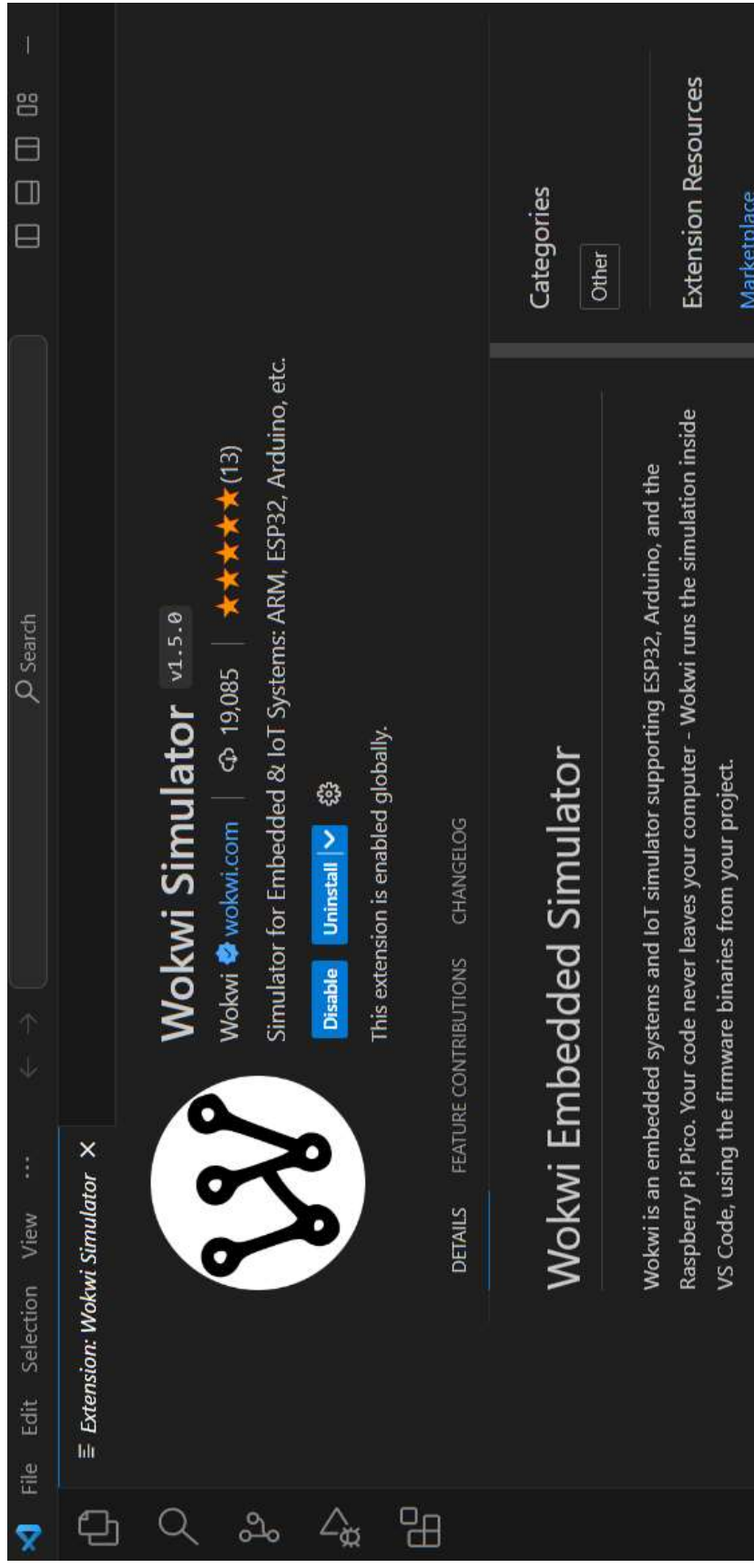
<https://docs.wokwi.com/vscode/getting-started#arduino-extension-examples>

Wowki con VS Code

1. Instalar la extension de Wokwi para VS Code

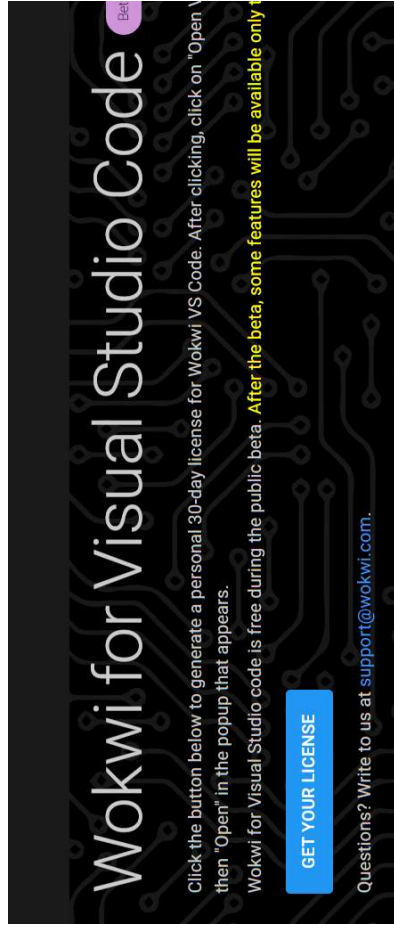
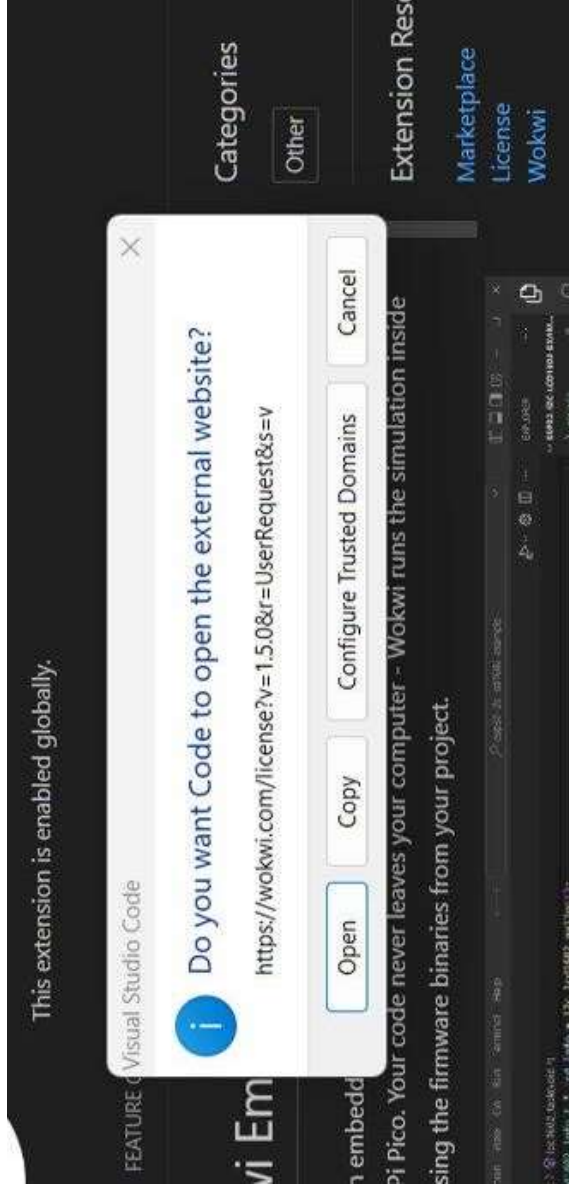
Link de la extension:

<https://marketplace.visualstudio.com/items?itemName=wokwi-vscode>



Wowki con VS Code

2. En Visual Code: presionar F1 en su teclado y escribir: 'Wokwi: request a new license'
3. Hacer click en 'get your license', les redirijira a la pagina web de Wokwi donde tendran iniciar sesion para obtener su licencia



Wowki con VS Code



4. Copiar su licencia.
5. En Visual Code: presionar F1 y escribir 'Wowki: manually enter license key'
6. Pegar su licencia y presionar Enter
7. Si todo va bien, en la esquina inferior derecha de su Visual Code aparecera una notificación de confirmación

Su licencia tiene una duración de un mes, y para renovarla simplemente tienen que volver a una nueva licencia como en el paso 2 de este tutorial.

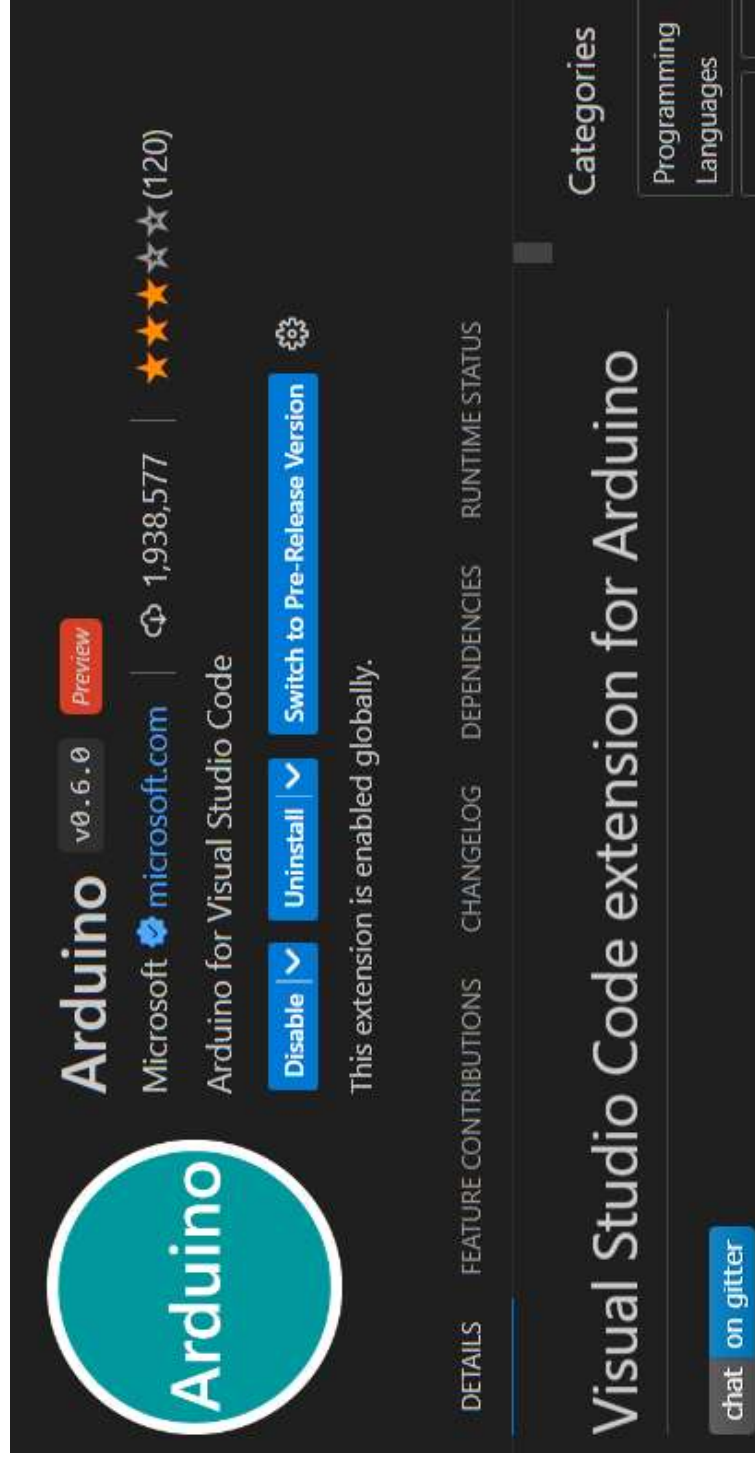


Wowki con VS Code

8. Instalar la extension de Arduino para Visual Code

Link de la extension:

<https://marketplace.visualstudio.com/items?itemName=vsciot-vscode.vscode-arduino>

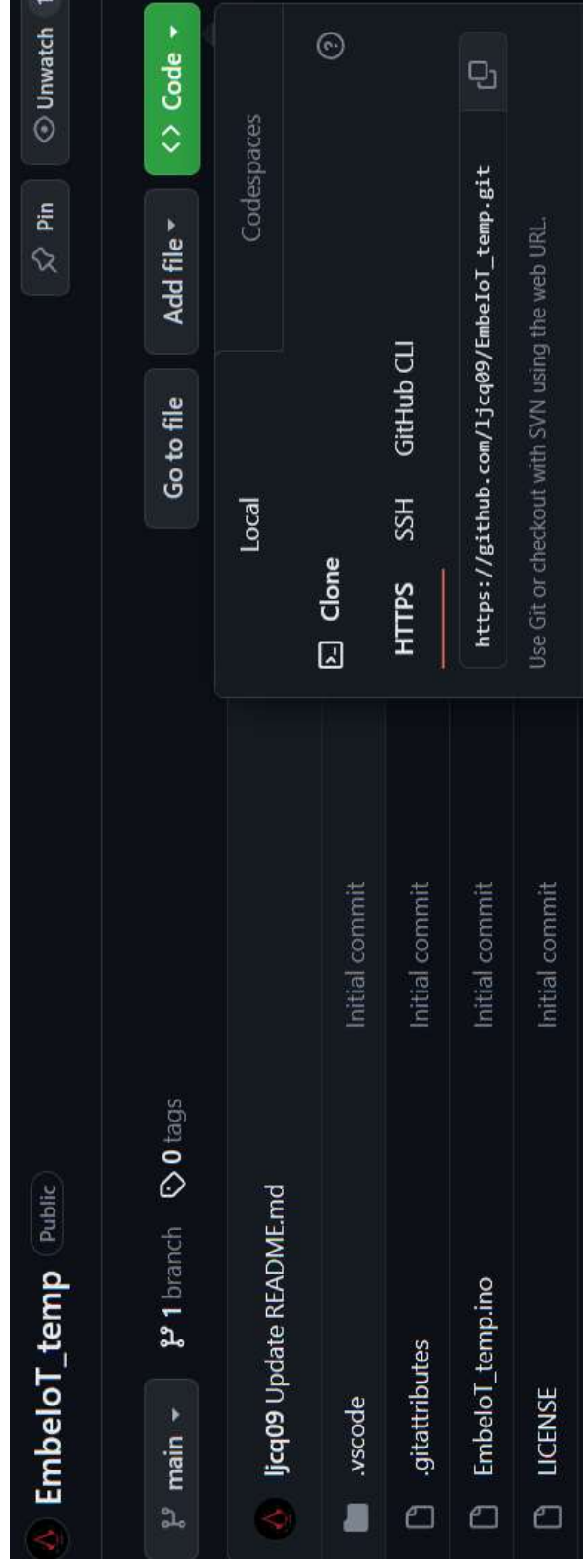


Wowki con VS Code

9. Clonar el template preparado por nosotros de Github

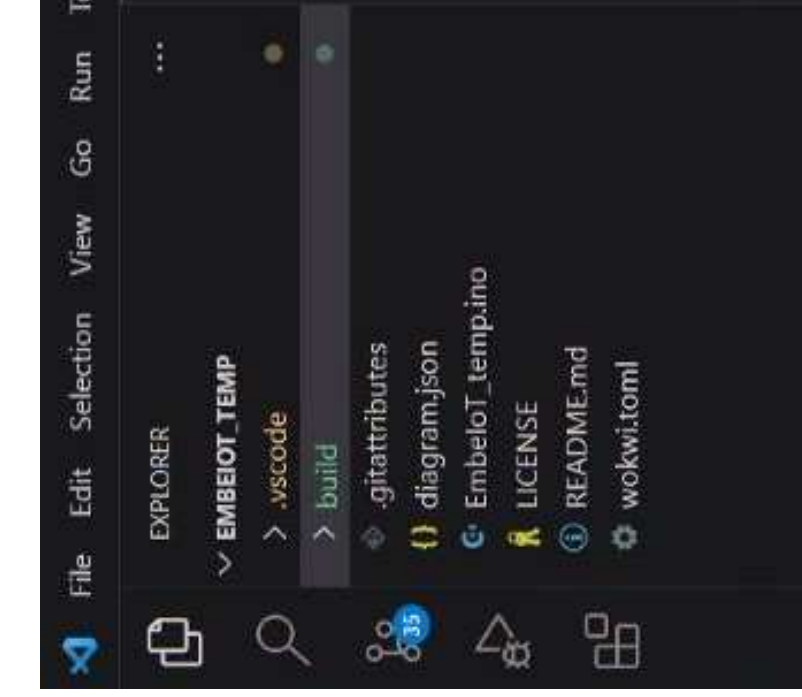
Link del repositorio:

[ljcq09/EmbeloT_temp \(github.com\)](https://github.com/ljcq09/EmbeloT_temp)



Wowki con VS Code

10. Estructura del Template



Los unicos archivos que se deberian modificar para el pr
son:

- Diagram.json
 - Contiene las propiedades de la simulacion Wokwi
 - Aqui se definen los componentes fisicos como el Ardu sensores, motores, etc.
 - Tambien se define en codigo las conexiones entre com ('el cableado')
- EmbeloT_temp.ino
 - Contiene el codigo de su programa
 - Codigo en lenguaje C/Arduino



Wowki con VS Code

10. Estructura del Template: *EmbeloT_temp.ino*



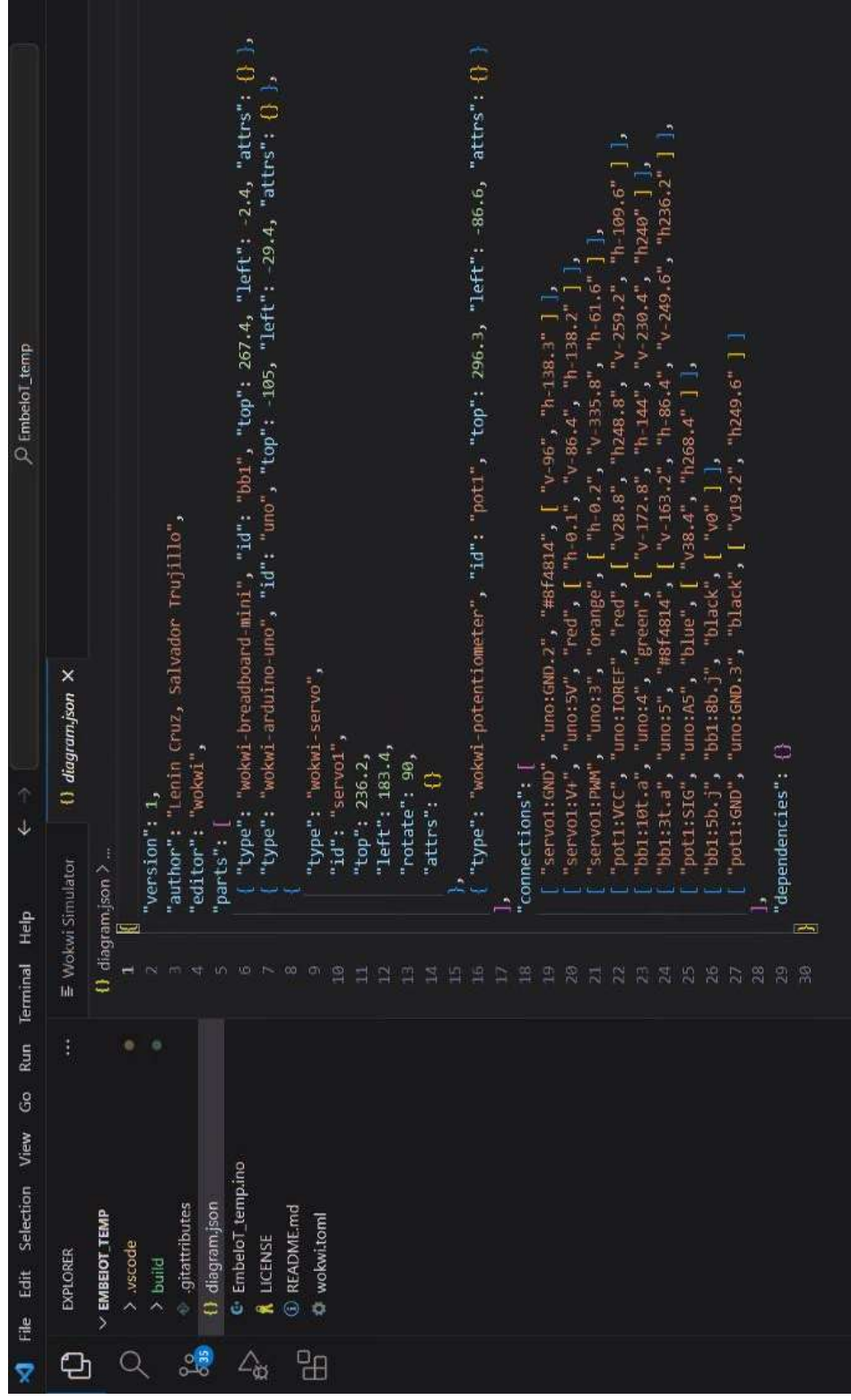
The screenshot shows the VS Code editor interface with the file `EmbeloT_temp.ino` open. The Explorer sidebar on the left shows the project structure, including the `EMBEIOT_TEMP` folder and its subfolders `.vscode`, `build`, `.gitattributes`, `diagram.json`, `EmbeloT_temp.ino` (selected), `LICENSE`, `README.md`, and `wokwi.toml`. The main editor area displays the content of `EmbeloT_temp.ino`, which is an Arduino sketch template. The code includes a comment, preprocessor directives for `Arduino.h` and `Servo.h`, variable declarations for pins and a servo motor, and the `setup` and `loop` functions.

```
1  1
2  2
3  3  /* Hello Wowki! */
4  4  #include <Arduino.h>
5  5  #include <Servo.h>
6  6
7  7  int servopwmPin = 3;
8  8  int Button1 = 4;
9  9  int Button2 = 5;
10 10 int Potenciometro = A5;
11 11
12 12 Servo servomotor;
13 13
14 14 int input = 0;
15 15 int angulo = 0;
16 16
17 17 void setup() {
18 18
19 19     pinMode(Potenciometro, INPUT);
20 20     servomotor.attach(servopwmPin);
21 21     servomotor.write(0);
22 22 }
23 23
24 24 void loop() {
```



Wowki con VS Code

10. Estructura del Template: *diagram.json*

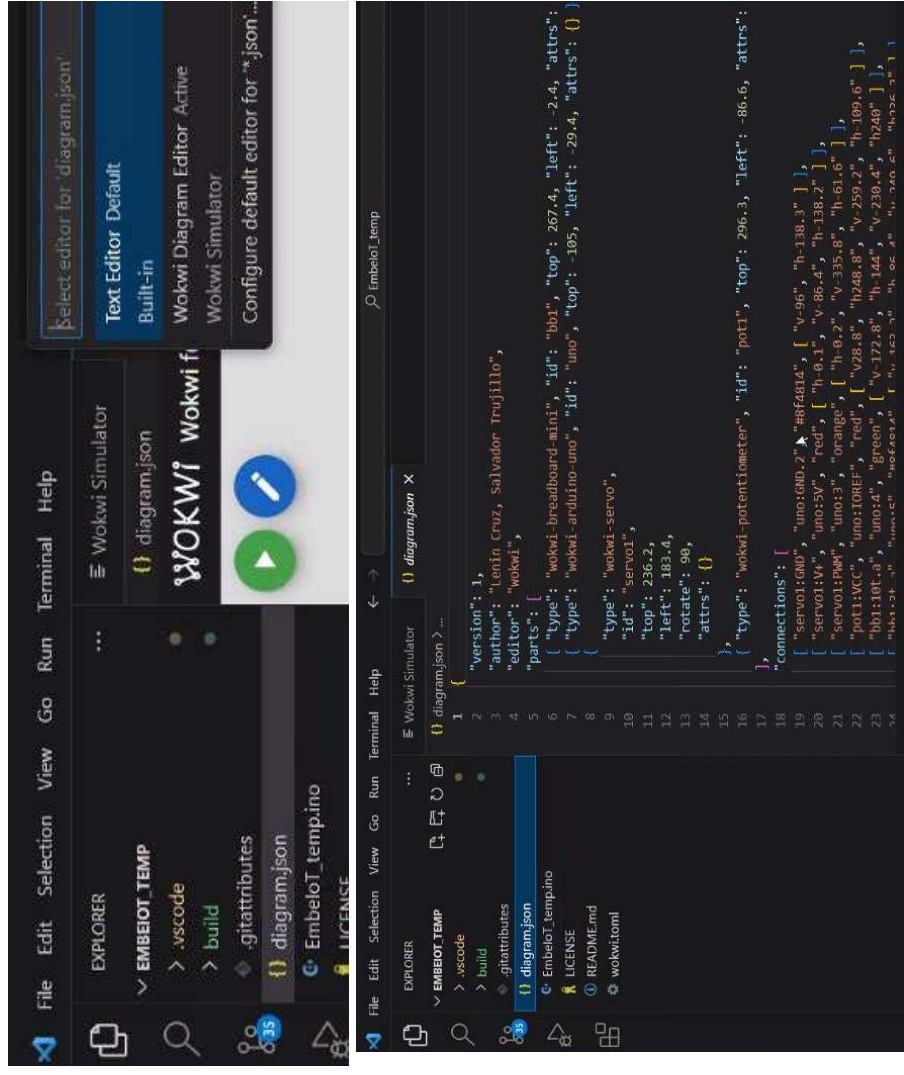
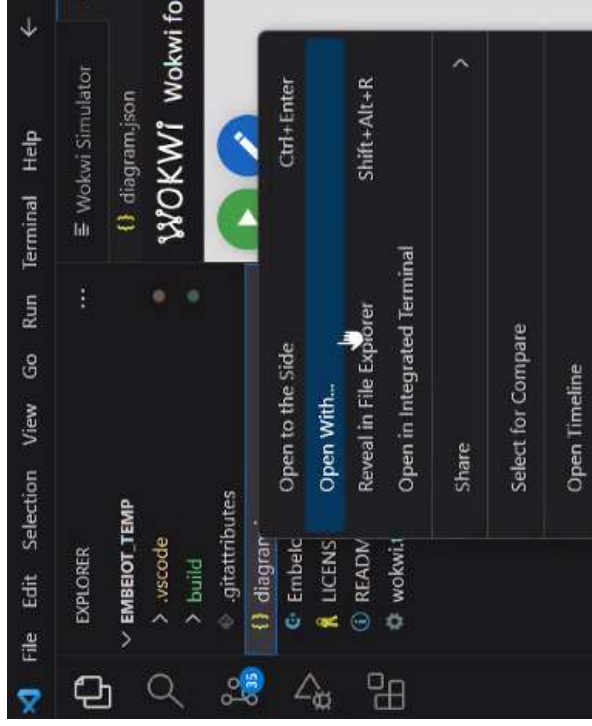


```
1 {
2   "version": 1,
3   "author": "Lenin Cruz, Salvador Trujillo",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "wokwi-breadboard-mini", "id": "bb1", "top": 267.4, "left": -2.4, "attrs": {} },
7     { "type": "wokwi-arduino-uno", "id": "uno", "top": -105, "left": -29.4, "attrs": {} },
8     {
9       "type": "wokwi-servo",
10      "id": "servo1",
11      "top": 236.2,
12      "left": 183.4,
13      "rotate": 90,
14      "attrs": {}
15    },
16     { "type": "wokwi-potentiometer", "id": "pot1", "top": 296.3, "left": -86.6, "attrs": {} }
17   ],
18   "connections": [
19     [ "servo1:GND", "uno:GND.2", "#8f4814", [ "v-96", "h-138.3" ] ],
20     [ "servo1:V+", "uno:5V", "red", [ "h-0.1", "v-86.4", "h-138.2" ] ],
21     [ "servo1:PWM", "uno:3", "orange", [ "h-0.2", "v-335.8", "h-61.6" ] ],
22     [ "pot1:VCC", "uno:IOREF", "red", [ "v28.8", "h248.8", "v-259.2", "h-109.6" ] ],
23     [ "bb1:10t.a", "uno:4", "green", [ "v-172.8", "h-144", "v-230.4", "h240" ] ],
24     [ "bb1:3t.a", "uno:5", "#8f4814", [ "v-163.2", "h-86.4", "v-249.6", "h236.2" ] ],
25     [ "pot1:SIG", "uno:A5", "blue", [ "v38.4", "h268.4" ] ],
26     [ "bb1:5b.j", "bb1:8b.j", "black", [ "v0" ] ],
27     [ "pot1:GND", "uno:GND.3", "black", [ "v19.2", "h249.6" ] ]
28   ],
29   "dependencies": {}
30 }
```

Wowki con VS Code

Nota: cuando se abre el archivo *diagram.json* aveces nos redirije a la simulacion sin dejarnos acceder al codigo. Para solucionar esto:

- Click derecho en el archivo y
- Abrir con editor de texto



Wowki con VS Code



10.5 Dependiendo de las configuraciones de su VS Code, se puede necesitar unicamente
pasos

- Presionar 'F1' en su teclado y escribir: *'Arduino: Verify'*
- Presionar 'F1' en su teclado y escribir: *'Wokwi: Start Simulator'*

Si esto va bien, su simulacion deberia funcionar.

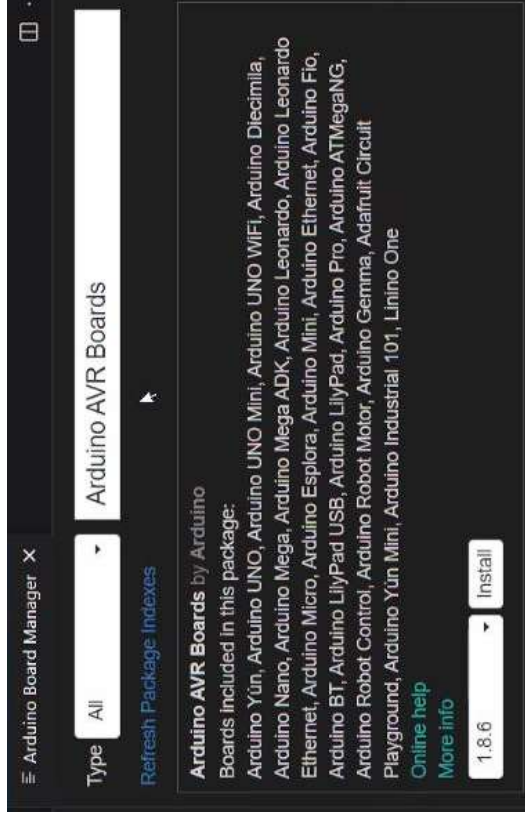
Wowki con VS Code

11. Instalacion de paquetes de Plataformas de Desarrollo Arduino

- Presionar 'F1' en su teclado y escribir: *'Arduino: Board Manager'*



- Buscar: *'Arduino AVR Boards'* e instalar la version que se muestra abajo a la izquierda.
- La imagen de la derecha es el output de la terminal mostrando el mensaje de instalacion exitosa

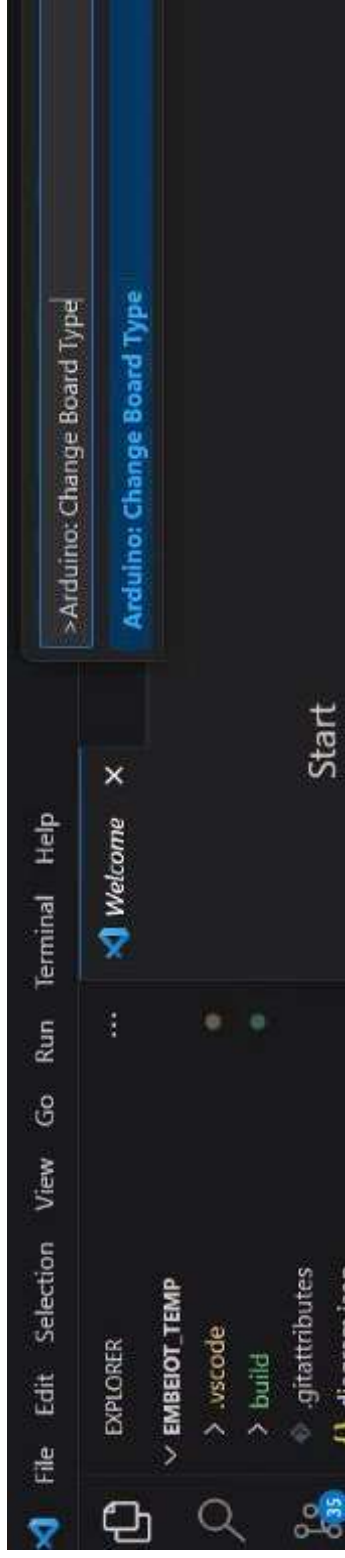


```
arduino:avr:1.8.6
Tool arduino:avr-gcc@7.3.0-atmel3.6.1-arduino7 already installed
Tool arduino:avr-libc@6.3.0-arduino17 already installed
Tool arduino:arduinoOTA@1.3.0 already installed
Downloading packages...
arduino:avr@1.8.6 arduino:avr@1.8.6 already downloaded
Installing platform arduino:avr@1.8.6...
Skipping platform configuration...
Platform arduino:avr@1.8.6 installed
[Done] Installed board package - arduino
```

Wowki con VS Code

12. Seleccionar el tipo de Plataforma de Desarrollo

- Presionar 'F1' en su teclado y escribir: *'Arduino: Change Board Type'*

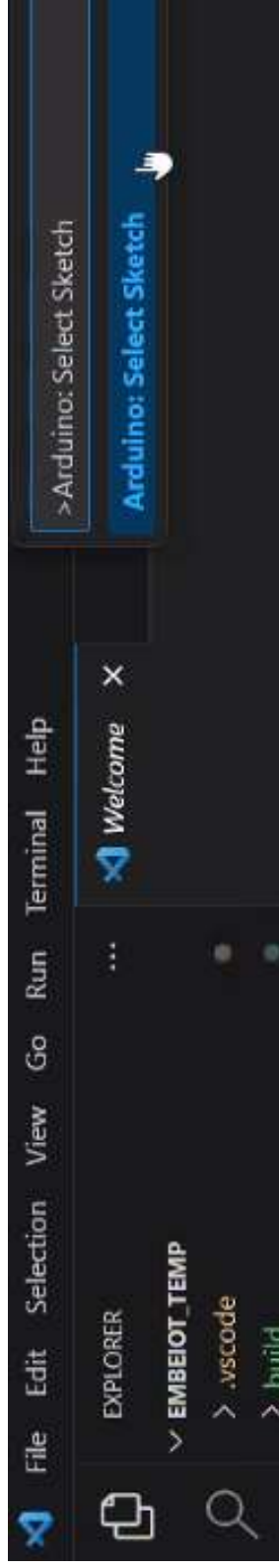


- Seleccionar: *'Arduino Uno'*



Wowki con VS Code

13. Presionar F1 en su teclado y escribir: 'Arduino: Select Sketch'



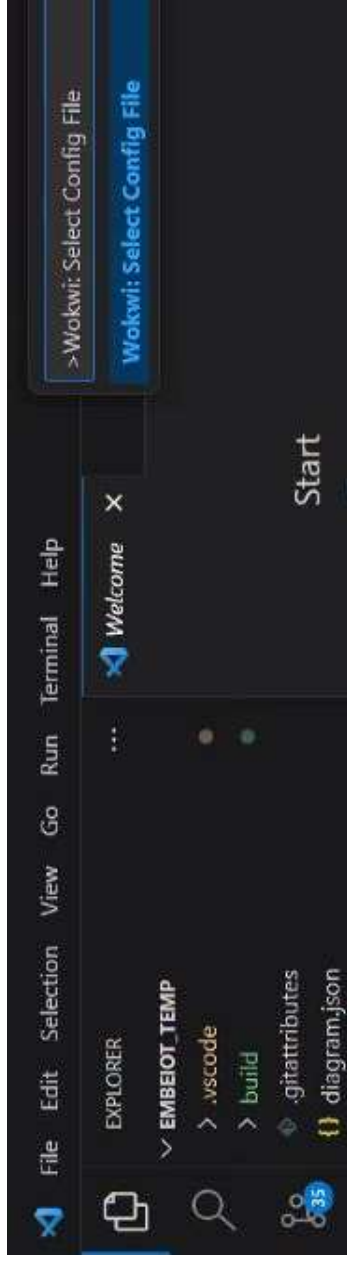
14. Seleccionar el archivo: 'EmbeloT_temp.ino'



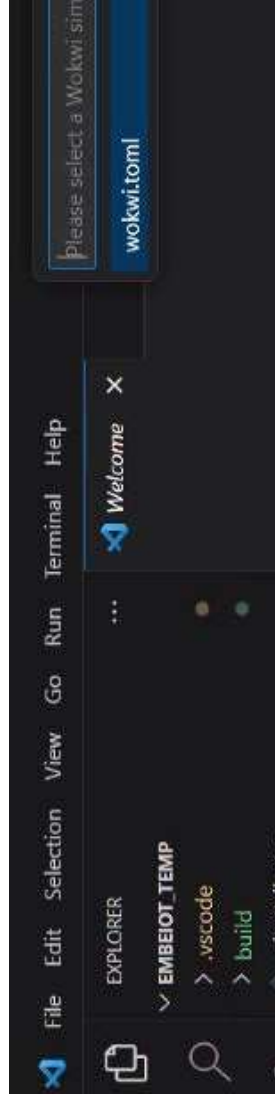
Wowki con VS Code

15. Seleccionar el archivo de simulacion

- Presionar F1 en su teclado y escribir: 'Wokwi: Select Config File'



- Seleccionar el archivo: 'wokwi.toml'



Wowki con VS Code



16. Compilar su código en lenguaje C/Arduino

- Presionar F1 en su teclado y escribir: 'Arduino: Verify'



- El mensaje de confirmación debe lucir como el de abajo

```
[Starting] verifying sketch 'EmbeIoT_temp.ino'
Please see the build logs in output path: c:\Users\Testing\Documents\GitHub\EmbeIoT_temp\build
Sketch uses 2152 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 50 bytes (2%) of dynamic memory, leaving 1998 bytes for local variables. Maximum is 2048 bytes.
IntelliSense configuration already up to date. To manually rebuild your IntelliSense configuration run "Ctrl+Alt+I"
[Done] verifying sketch 'EmbeIoT_temp.ino'
```


Wowki con VS Code

17. Finalmente, para empezar su simulacion:

- Presionar F1 en su teclado y escribir: 'Wokwi: Start Simulator'

