

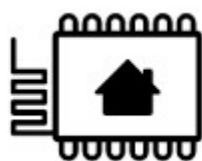


POLITÉCNICO
DE LEIRIA

iModDom



Tutorial - Instalação *ESPHome*



ESPHome

Elaborado por:

Marco Pereira - 2190516

Orientado por:

Luís Bento

Carlos Neves

Índice

1. INTRODUÇÃO.....	1
1.1. OBJETIVOS.....	1
2. INSTALAÇÃO ESPHOME.....	2
2.1. INSTALAÇÃO PYTHON.....	2
2.2. INSTALAÇÃO DO ESPHOME	5
3. PRIMEIRO PROJETO	6
3.1. CRIAÇÃO DE UM PROJETO.....	6
3.2. EDITAR O PROJETO	8
3.3. UPLOAD DO PROJETO	9

1. Introdução

O *ESPHome* é um sistema que permite controlar o *ESP8266/ESP32* através de arquivos de configuração simples, no entanto muito poderosos, e controlá-los remotamente através de sistemas de automação residencial.

1.1. Objetivos

Pretende-se com este tutorial demonstrar como instalar o *ESPHome* bem como criar e configurar o primeiro projeto.

2. Instalação *ESPHome*

2.1. Instalação *Python*

Caso não tenha instalado o *Python* irá necessitar de o instalar, para isso basta aceder ao seguinte *link*: <https://www.python.org/>.

1. Após ter acedido ao *link* fornecido anteriormente, no menu “*Downloads*”, clique em “*Python 3.9.2*” (Figura 1).

Nota: À data que estiver a seguir o tutorial a versão do *Python* poderá ser mais recente.

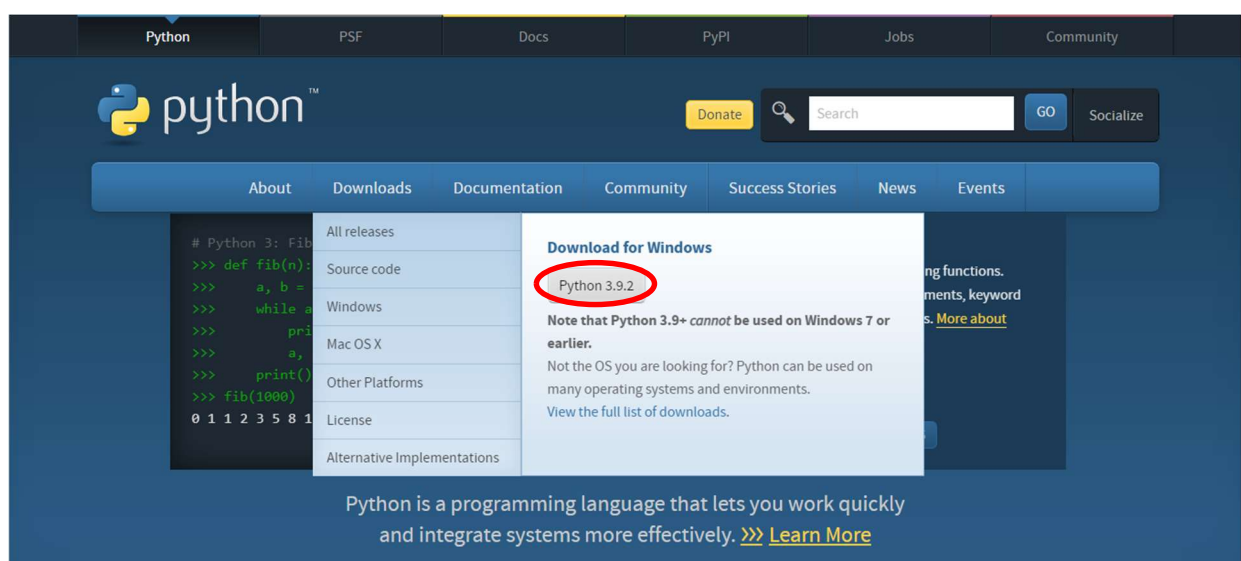


Figura 1 – Download *Python*

2. Após o *download* estar concluído com sucesso, aceda à pasta “*Transferências*” e execute o ficheiro descarregado (Figura 2).

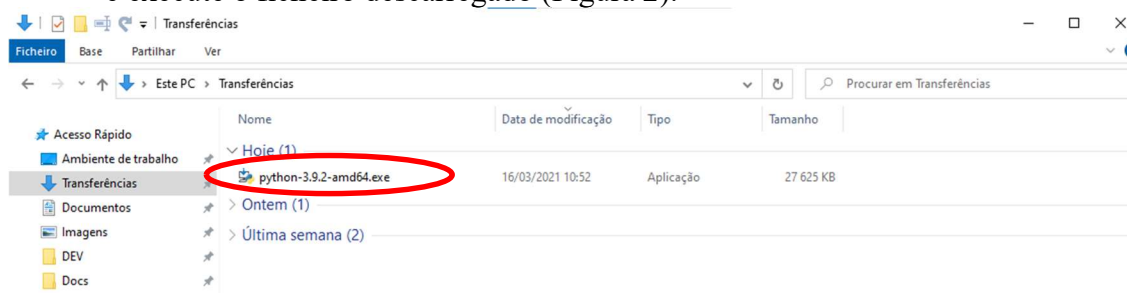


Figura 2 – Executar o Ficheiro

3. Na janela que irá surgir selecione a opção “*Add Python 3.8 to PATH*” e de seguida clique em “*Install Now*” (Figura 3).

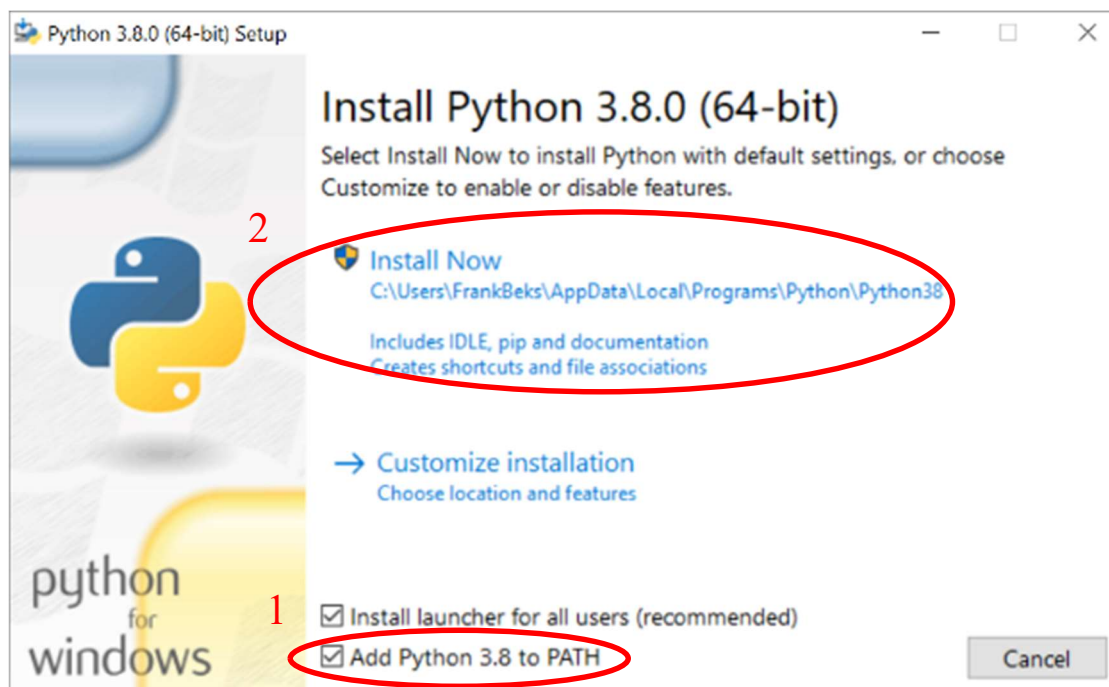


Figura 3 – Instalação do *Python*

4. Quando a instalação terminar, surgir-lhe-á a seguinte janela onde vai clicar em “*Disable path length limit*” e de seguida em “*Close*” (Figura 4).

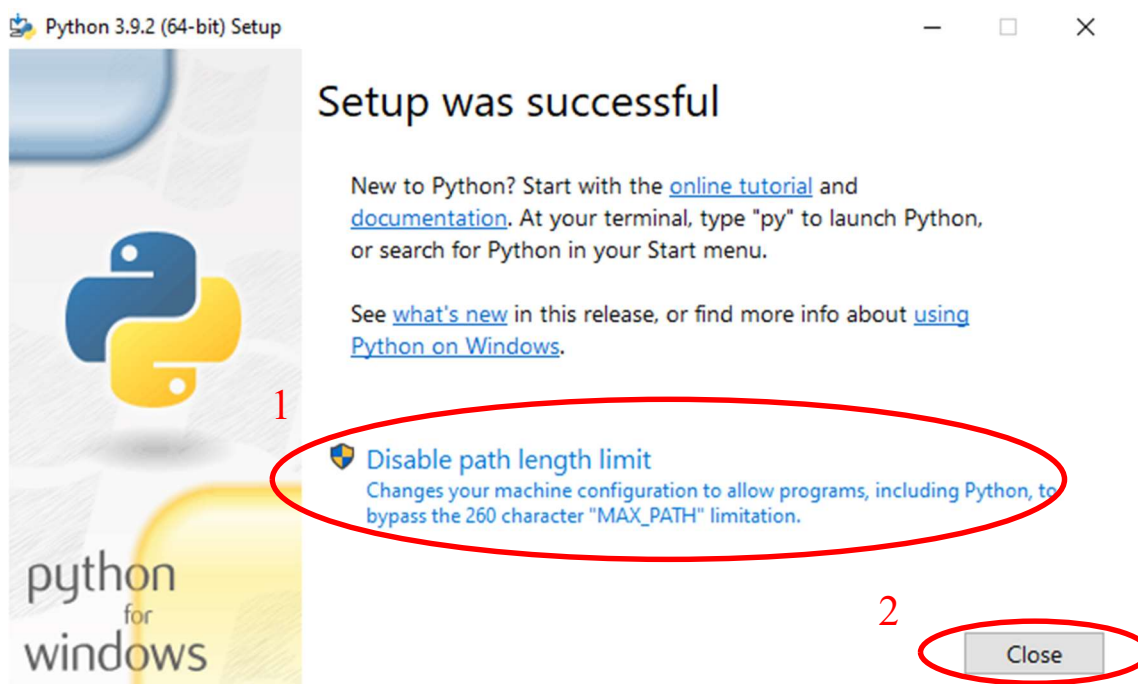
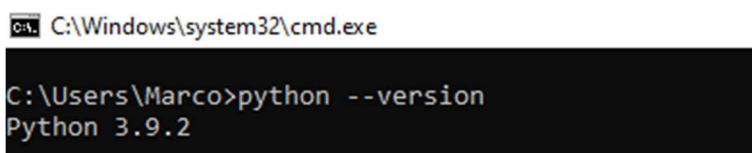


Figura 4 – Instalação Terminada

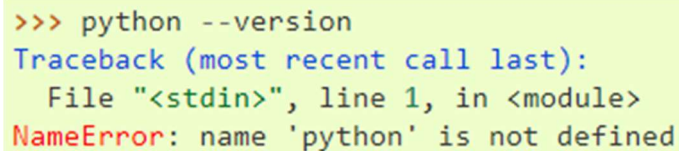
5. Pode confirmar que o *Python* foi instalado corretamente abrindo a linha de comandos do seu computador e executar o seguinte comando “*python --version*”. Deverá surgir uma mensagem com a indicação da versão do *Python* instalada (Figura 5). Se ao executar o comando indicado surgir um erro semelhante ao da Figura 6 , aceda a <https://packaging.python.org/tutorials/installing-packages/> de forma a solucionar o mesmo e conseguir prosseguir com o tutorial.



```
C:\Windows\system32\cmd.exe

C:\Users\Marco>python --version
Python 3.9.2
```

Figura 5 – Versão do *Python* Instalada



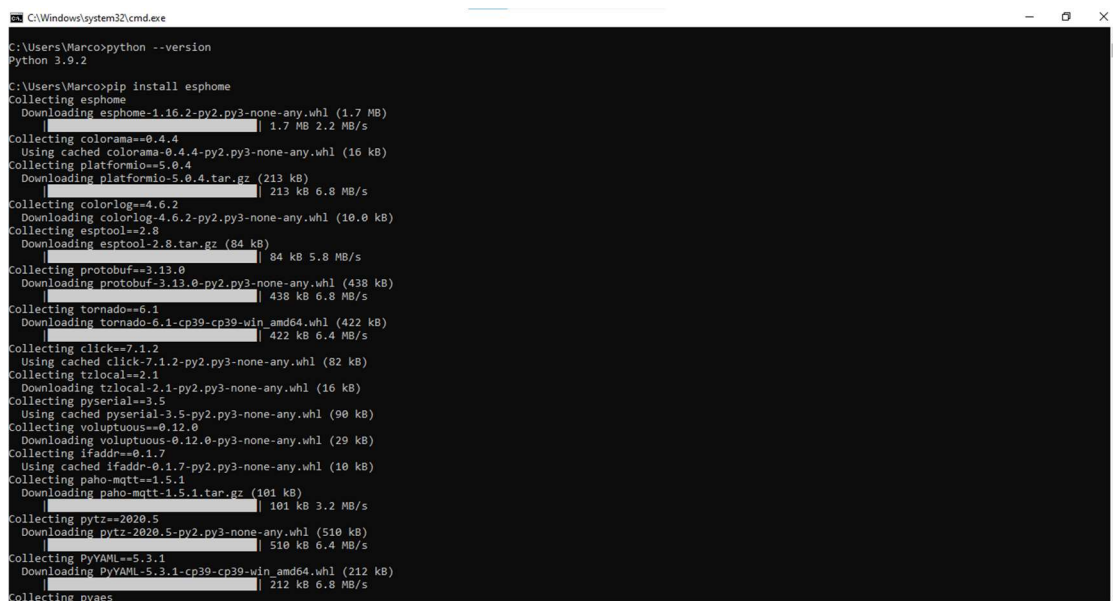
```
>>> python --version
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'python' is not defined
```

Figura 6 – Possível Erro ao Executar o Comando

2.2. Instalação do *ESPHome*

Agora que já instalou o *Python* com sucesso, passemos à instalação do *ESPHome*.

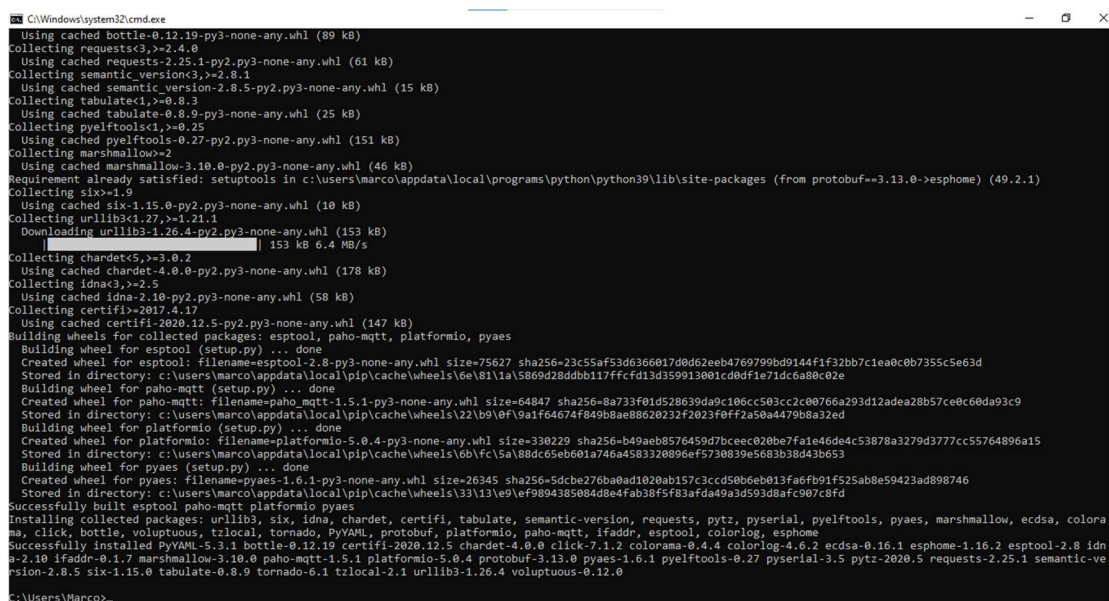
1. Comece por executar o comando “*pip install esphome*” (Figura 7), aguarde uns segundos até que a instalação seja terminada com sucesso, ficando com o aspeto da Figura 8.



```
C:\Windows\system32\cmd.exe
C:\Users\Marco>python --version
Python 3.9.2

C:\Users\Marco>pip install esphome
Collecting esphome
  Downloading esphome-1.16.2-py2.py3-none-any.whl (1.7 MB)
    | 1.7 MB 2.2 MB/s
Collecting colorama==0.4.4
  Using cached colorama-0.4.4-py2.py3-none-any.whl (16 kB)
Collecting platformio==5.0.4
  Downloading platformio-5.0.4.tar.gz (213 kB)
    | 213 kB 6.8 MB/s
Collecting colorlog==4.6.2
  Downloading colorlog-4.6.2-py2.py3-none-any.whl (10.0 kB)
Collecting esptool==2.8
  Downloading esptool-2.8.tar.gz (84 kB)
    | 84 kB 5.8 MB/s
Collecting protobuf==3.13.0
  Downloading protobuf-3.13.0-py2.py3-none-any.whl (438 kB)
    | 438 kB 6.8 MB/s
Collecting tornado==6.1
  Downloading tornado-6.1-cp39-cp39-win_amd64.whl (422 kB)
    | 422 kB 6.4 MB/s
Collecting click==7.1.2
  Using cached click-7.1.2-py2.py3-none-any.whl (82 kB)
Collecting tzlocal==2.1
  Downloading tzlocal-2.1-py2.py3-none-any.whl (16 kB)
Collecting pyserial==3.5
  Using cached pyserial-3.5-py2.py3-none-any.whl (90 kB)
Collecting voluptuous==0.12.0
  Downloading voluptuous-0.12.0-py3-none-any.whl (29 kB)
Collecting ifaddr==0.1.7
  Using cached ifaddr-0.1.7-py2.py3-none-any.whl (10 kB)
Collecting paho-mqtt==1.5.1
  Downloading paho-mqtt-1.5.1.tar.gz (101 kB)
    | 101 kB 3.2 MB/s
Collecting pytz==2020.5
  Downloading pytz-2020.5-py2.py3-none-any.whl (510 kB)
    | 510 kB 6.4 MB/s
Collecting PyYAML==5.3.1
  Downloading PyYAML-5.3.1-cp39-cp39-win_amd64.whl (212 kB)
    | 212 kB 6.8 MB/s
Collecting pyaes
```

Figura 7 – Comando para Instalar o *ESPHome*



```
C:\Windows\system32\cmd.exe
Using cached bottle-0.12.19-py3-none-any.whl (89 kB)
Collecting requests<3,>=2.4.0
  Using cached requests-2.25.1-py2.py3-none-any.whl (61 kB)
Collecting semantic-version<3,>=2.8.1
  Using cached semantic-version-2.8.5-py2.py3-none-any.whl (15 kB)
Collecting tabulate<1,>=0.8.3
  Using cached tabulate-0.8.9-py3-none-any.whl (25 kB)
Collecting pyelftools<1,>=0.25
  Using cached pyelftools-0.27-py2.py3-none-any.whl (151 kB)
Collecting marshmallow>=2
  Using cached marshmallow-3.10.0-py2.py3-none-any.whl (46 kB)
Requirement already satisfied: setuptools in c:\users\marco\appdata\local\programs\python\python39\lib\site-packages (from protobuf==3.13.0->esphome) (49.2.1)
Collecting six>=1.9
  Using cached six-1.15.0-py2.py3-none-any.whl (10 kB)
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.4-py2.py3-none-any.whl (153 kB)
    | 153 kB 6.4 MB/s
Collecting chardet<5,>=3.0.2
  Using cached chardet-4.0.0-py2.py3-none-any.whl (178 kB)
Collecting idna<3,>=2.5
  Using cached idna-2.10-py2.py3-none-any.whl (58 kB)
Collecting certifi>=2017.4.17
  Using cached certifi-2020.12.5-py2.py3-none-any.whl (147 kB)
Building wheels for collected packages: esptool, paho-mqtt, platformio, pyaes
  Building wheel for esptool (setup.py) ... done
  Created wheel for esptool: filename=esptool-2.8-py3-none-any.whl size=75627 sha256=23c55af53d6366017d0d62eeb4769799bd9144f1f32bb7c1ea0c0b7355c5e63d
  Stored in directory: c:\users\marco\appdata\local\pip\cache\wheels\6e\81\1a\5869d28ddb117frcfd13d3599130e1c0df1e71dc6a80c0e
  Building wheel for paho-mqtt (setup.py) ... done
  Created wheel for paho-mqtt: filename=paho-mqtt-1.5.1-py3-none-any.whl size=64847 sha256=8a733f01d528639da9c106cc503cc2c00766a293d12adea28b57ce0c60da93c9
  Stored in directory: c:\users\marco\appdata\local\pip\cache\wheels\22\b9\0f\9a1f64674f849b8ae88620232f2023f0ff2a50a4479b8a32ed
  Building wheel for platformio (setup.py) ... done
  Created wheel for platformio: filename=platformio-5.0.4-py3-none-any.whl size=380229 sha256=b49aeb8576459d7bceec020be7fa1e46de4c53878a3279d3777cc55764896a15
  Stored in directory: c:\users\marco\appdata\local\pip\cache\wheels\0b\fc\5a\88dc65eb6e1a746a4583320896ef5730839e5683b38d43b653
  Building wheel for pyaes (setup.py) ... done
  Created wheel for pyaes: filename=pyaes-1.6.1-py3-none-any.whl size=26345 sha256=5dcbe276ba0ad1020ab157c3cc5d50bbe013fa6fb91f525ab8e59423ad898746
  Stored in directory: c:\users\marco\appdata\local\pip\cache\wheels\33\13\1e\9ef9894385084d8e4fab38f5f83afda9a3d593d8af907c8fd
Successfully built esptool paho-mqtt platformio pyaes
Installing collected packages: urllib3, six, idna, chardet, certifi, tabulate, semantic-version, requests, pytz, pyserial, pyelftools, pyaes, marshmallow, ecdsa, colorama, click, bottle, voluptuous, tzlocal, tornado, PyYAML, protobuf, platformio, paho-mqtt, ifaddr, esptool, colorlog, esphome
Successfully installed PyYAML-5.3.1 bottle-0.12.19 certifi-2020.12.5 chardet-4.0.0 click-7.1.2 colorama-0.4.4 colorlog-4.6.2 ecdsa-0.16.1 esphome-1.16.2 esptool-2.8 idna-2.10 ifaddr-0.1.7 marshmallow-3.10.0 paho-mqtt-1.5.1 platformio-5.0.4 protobuf-3.13.0 pyaes-1.6.1 pyelftools-0.27 pyserial-3.5 pytz-2020.5 requests-2.25.1 semantic-version-2.8.5 six-1.15.0 tabulate-0.8.9 tornado-6.1 tzlocal-2.1 urllib3-1.26.4 voluptuous-0.12.0
C:\Users\Marco>
```

Figura 8 – Instalação Terminada com Sucesso

3. Primeiro Projeto

3.1. Criação de um Projeto

1. Agora que já instalou o *ESPHome*, crie uma pasta onde pretende guardar os seus projetos, de seguida, na linha de comandos, selecione a diretoria da pasta criada através do comando “*cd*” seguido do caminho da pasta que criou (Figura 9).

```
C:\Users\Marco>cd C:\Users\Marco\DEV\ESPHome_workspace
```

Figura 9 – Pasta Criada

2. Execute o comando “*esphome myfirst.yaml wizard*” de modo a executar o assistente de configuração do *ESPHome*. O primeiro passo é atribuir um nome para o *node*, pode atribuir o nome que desejar (Figura 14).

```
C:\Users\Marco\DEV\ESPHome_workspace>esphome myfirst.yaml wizard
Hi there!
I'm the wizard of ESPHome :)
And I'm here to help you get started with ESPHome.
In 4 steps I'm going to guide you through creating a basic configuration file for your custom ESP8266/ESP32 firmware. Yay!

===== STEP 1 =====

CORE

=====
First up, please choose a name for your node.
It should be a unique name that can be used to identify the device later.
For example, I like calling the node in my living room livingroom.

+ [01;37m(name): + [0m
```

Figura 10 - Nome do *Node*

3. Depois de escolher o nome do seu *node*, é-lhe pedido para escolher entre o *ESP32* e o *ESP8266* dependendo daquele que esteja a utilizar, escolha um (Figura 11).

```
===== STEP 2 =====

ESP

=====
Now I'd like to know what microcontroller you're using so that I can compile firmwares for it.
Are you using an ESP32 or ESP8266 platform? (Choose ESP8266 for Sonoff devices)

Please enter either ESP32 or ESP8266.
+ [01;37m(ESP32/ESP8266): + [0mESP32
Thanks! You've chosen ESP32 as your platform.
```

Figura 11 - Seleção do Microcontrolador

4. De seguida é pedido para seleccionar a placa que está a utilizar, de notar que é apresentada uma lista das mesmas (Figura 12).

```

===== STEP 2 =====

ESP

=====
Now I'd like to know what microcontroller you're using so that I can compile firmwares for it.
Are you using an ESP32 or ESP8266 platform? (Choose ESP8266 for Sonoff devices)

Please enter either ESP32 or ESP8266.
->[01;37m(ESP32/ESP8266): +[0mESP32
Thanks! You've chosen ESP32 as your platform.

Next, I need to know what board you're using.
Please go to http://docs.platformio.org/en/latest/platforms/espressif32.html#boards and choose a board.
(Type esp01_1m for Sonoff devices)

For example "nodemcu-32s".
Options: alksesp32, bpi-bit, d-duino-32, esp-wrover-kit, esp32-devkitlipo, esp32-evb, esp32-gateway, esp32-poe, esp32-poe-iso, esp32-pro, esp320, esp32cam, esp32dev, es
p32doit-devkit-v1, esp32thing, esp32vn-iot-uno, espea32, espectro32, espino32, featheresp32, firebeetle32, fm-devkit, frogboard, heltec_wifi_kit_32, heltec_wifi_lora_32
, heltec_wifi_lora_32_V2, heltec_wireless_stick, hornbill32dev, hornbill32minima, intorobot, iotaa magnolia, iotbusio, iotbusproteus, lolin32, lolin_d32, lolin_d32_pro
, lopy, lopy4, m5stack-core-esp32, m5stack-fire, m5stack-greys, m5stick-c, magicbit, mhetesp32devkit, mhetesp32minikit, microduino-core-esp32, nano32, nina_w10, node32s,
nodemcu-32s, odroid_esp32, onehorse32dev, oroca_edubot, pico32, pocket_32, pycom_gpy, quantum, sparkfun_lora_gateway_1-channel, tinypico, ttgo-lora32-v1, ttgo-t-beam,
ttgo-t-watch, ttgo-t1, turta_10t_node, vintlabs-devkit-v1, wemos_d1_mini32, wemosbat, wesp32, widora-air, xinabox_cw02
->[01;37m(board): +[0m

```

Figura 12 - Seleção da Placa

5. O próximo passo será introduzir as credenciais da sua rede WiFi, primeiramente é-lhe pedido para introduzir o *ssid* e de seguida a *password* da rede (Figura 13).

```

===== STEP 3 =====

WiFi

=====
In this step, I'm going to create the configuration for WiFi.

First, what's the SSID (the name) of the WiFi network livingroom I should connect to?
For example "Abraham Linksys".
->[01;37m(ssid): +[0mVodafone-DF79CC
Thank you very much! You've just chosen "Vodafone-DF79CC" as your SSID.

Now please state the password of the WiFi network so that I can connect to it (Leave empty for no password)
For example "PASSWORD42"
->[01;37m(PSK): +[0mVodafone-DF79CC
Perfect! WiFi is now set up (you can create static IPs and so on later).

```

Figura 13 - Configuração da Rede WiFi

- O último passo será para introduzir uma *password* para realizar *uploads* dos seus *nodes* via WiFi, caso não pretenda definir uma *password* basta clicar na tecla “ENTER” (Figura 14).

```
===== STEP 4 =====
      O T A
=====
Almost there! ESPHome can automatically upload custom firmwares over WiFi (over the air) and integrates into Home Assistant with a native API.
This can be insecure if you do not trust the WiFi network. Do you want to set a password for connecting to this ESP?

Press ENTER for no password
~[01;37m(password): ~[0m1234

DONE! I've now written a new configuration file to myfirst.yaml

Next steps:
> Check your Home Assistant "integrations" screen. If all goes well, you should see your ESP being discovered automatically.
> Then follow the rest of the getting started guide:
> https://esphome.io/guides/getting_started_command_line.html

C:\Users\Marco\DEV\ESPHome_workspace>
```

Figura 14 – *Password OTA*

3.2.Editar o Projeto

- Agora que criou o seu projeto, abra o ficheiro “*myfirst.yaml*” com um editor à sua escolha (Figura 15).

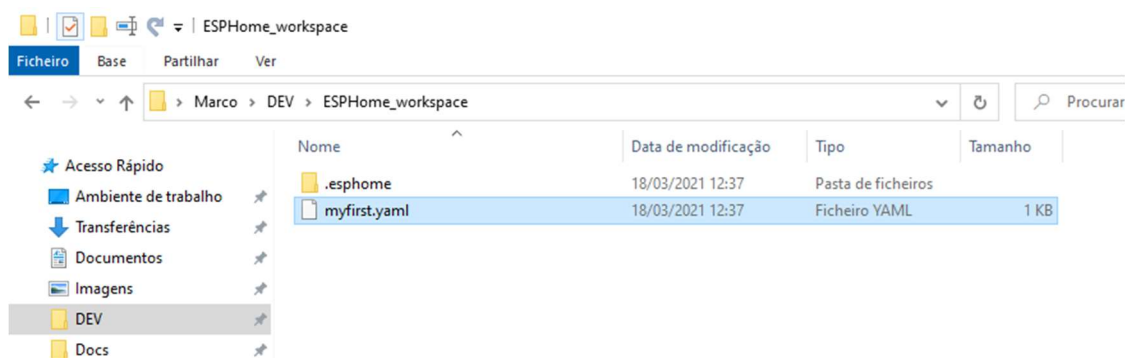
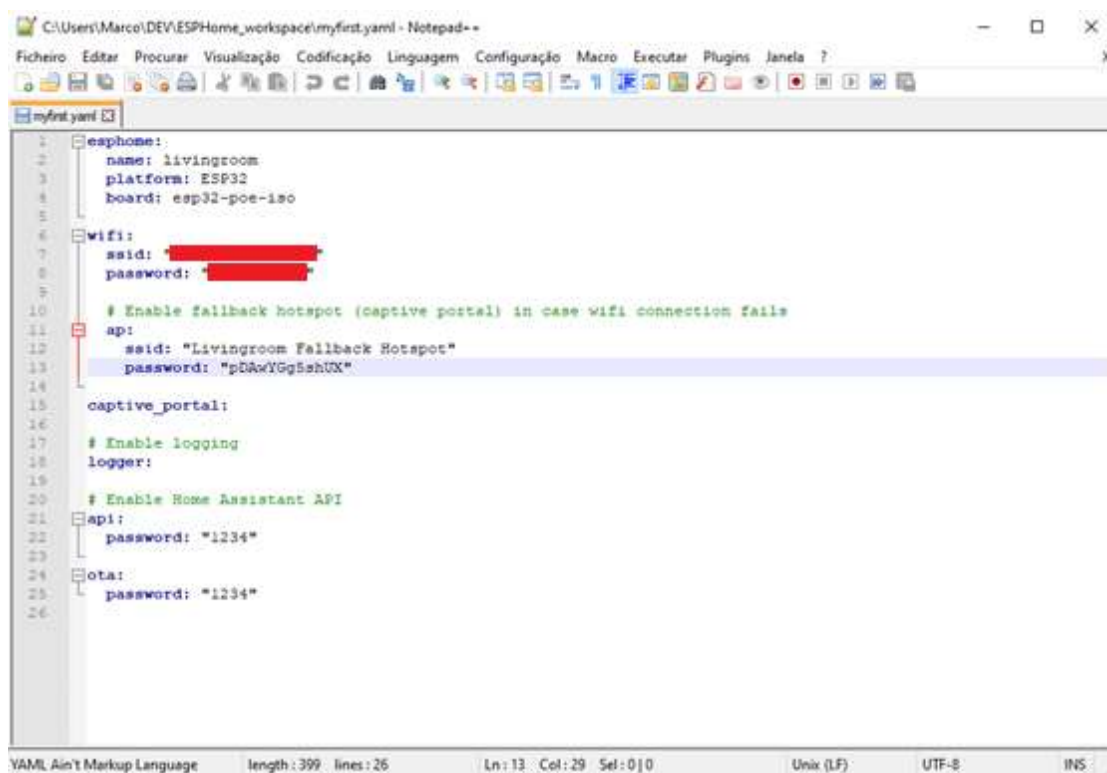


Figura 15 - Abrir o Ficheiro “*myfirst.yaml*”

2. Pode verificar que todas as configurações estão contidas neste ficheiro e podem ser alteradas (Figura 16).



```

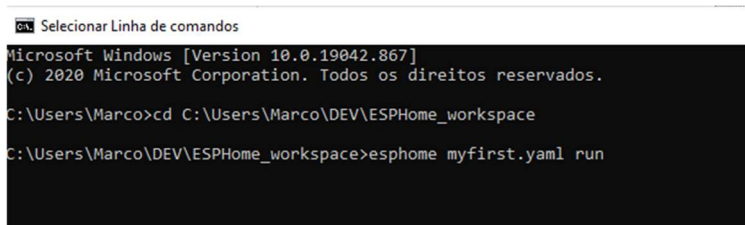
1 esphome:
2   name: livingroom
3   platform: ESP32
4   board: esp32-poe-iso
5
6   wifi:
7     ssid: [REDACTED]
8     password: [REDACTED]
9
10    # Enable fallback hotspot (captive portal) in case wifi connection fails
11    ap:
12      ssid: "Livingroom Fallback Hotspot"
13      password: "pD8xYGg5shUX"
14
15    captive_portal:
16
17    # Enable logging
18    logger:
19
20    # Enable Home Assistant API
21    api:
22      password: "1234"
23
24    ota:
25      password: "1234"
26
  
```

Figura 16 - Conteúdo do Ficheiro

3.3.Upload do Projeto

1. Neste momento necessita de conectar a sua placa à porta USB do seu computador.
2. Voltando à linha de comandos, pode agora, executar o comando “*esphome myfirst.yaml run*” de modo a compilar o ficheiro “*myfirst.yaml*” e a criar e fazer *upload* do *firmware* para a sua placa (Figura 17).

Nota: Se ao fazer este passo, ocorrer um erro a informar que a placa não está conectada, muito provavelmente necessitará de instalar os *drivers* da mesma, pode encontrar alguns dos *drivers* aqui: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/establish-serial-connection.html>. Caso esteja a utilizar a placa *ESP32-PoE-ISO*, pode encontrar os *drivers* para a mesma no seguinte link: <https://www.olimex.com/Products/IoT/ESP32/ESP32-POE-ISO/open-source-hardware> ou seguir o tutorial “6_Tutorial_Instalacao_de_Drivers_ESP32_PoE_ISO”.



```

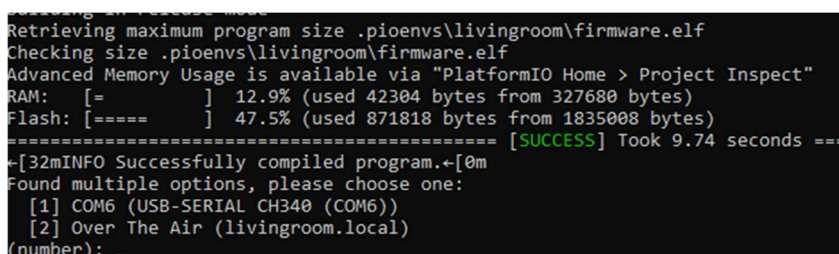
Selecionar Linha de comandos
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Marco>cd C:\Users\Marco\DEV\ESPHome_workspace

C:\Users\Marco\DEV\ESPHome_workspace>esphome myfirst.yaml run
  
```

Figura 17 - Comando para Compilar

3. Após executar o comando irá aparecer a seguinte informação, onde terá de escolher se pretende realizar o *upload* através da porta USB ou *Over The Air* (OTA), ou seja, via WiFi, como é a primeira vez que irá fazer o *upload* tem de ser através da porta USB, por isso escolha a opção 1 (Figura 18).



```

Retrieving maximum program size .pioenvs\livingroom\firmware.elf
Checking size .pioenvs\livingroom\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 12.9% (used 42304 bytes from 327680 bytes)
Flash: [=====] 47.5% (used 871818 bytes from 1835008 bytes)
===== [SUCCESS] Took 9.74 seconds =====
+32mINFO Successfully compiled program.←[0m
Found multiple options, please choose one:
[1] COM6 (USB-SERIAL CH340 (COM6))
[2] Over The Air (livingroom.local)
(number):
  
```

Figura 18 - Opção de Upload

4. Quando o *upload* estiver concluído com sucesso irá surgir uma mensagem do género da apresentada na Figura 19. Caso surja o erro apresentado na Figura 20 deverá fazer o *upload* novamente, mas tentando uma das seguintes hipóteses:
 1. Manter o botão de *boot* pressionado até que o *upload* esteja concluído.
 2. Manter o botão de *boot* pressionado, clicar e soltar o botão de *reset* e quando o *upload* estiver concluído soltar o botão de *boot*.
 3. Conectar o pino GND ao pino GPIO0 enquanto o *upload* é executado, seguindo o esquema apresentado na Figura 21.

```

Linha de comandos - esphome myfirstyaml run
[15:15:01][C][wifi:319]:   DNS2: 0.0.0.0
[15:15:01][D][wifi:466]: Disabling AP...
[15:15:01][C][ota:029]: Over-The-Air Updates:
[15:15:01][C][ota:030]:   Address: livingroom.local:3232
[15:15:01][C][ota:032]:   Using Password.
[15:15:01][C][api:022]: Setting up Home Assistant API server...
[15:15:01][I][app:059]: setup() finished successfully!
[15:15:01][I][app:105]: ESPHome version 1.16.2 compiled on Mar 18 2021, 12:50:05
[15:15:01][C][wifi:443]: WiFi:
[15:15:01][C][wifi:303]:   SSID: 'Vodafone-DF79CC'
[15:15:01][C][wifi:304]:   IP Address: 192.168.1.128
[15:15:01][C][wifi:306]:   BSSID: 48:FD:8E:DF:79:D4
[15:15:01][C][wifi:307]:   Hostname: 'livingroom'
[15:15:01][C][wifi:311]:   Signal strength: -59 dB
[15:15:01][C][wifi:315]:   Channel: 5
[15:15:01][C][wifi:316]:   Subnet: 255.255.255.0
[15:15:01][C][wifi:317]:   Gateway: 192.168.1.1
[15:15:01][C][wifi:318]:   DNS1: 192.168.1.1
[15:15:01][C][wifi:319]:   DNS2: 0.0.0.0
[15:15:01][C][logger:185]: Logger:
[15:15:01][C][logger:186]:   Level: DEBUG
[15:15:01][C][logger:187]:   Log Baud Rate: 115200
[15:15:01][C][logger:188]:   Hardware UART: UART0
[15:15:01][C][captive_portal:169]: Captive Portal:
[15:15:01][C][ota:029]: Over-The-Air Updates:
[15:15:01][C][ota:030]:   Address: livingroom.local:3232
[15:15:01][C][ota:032]:   Using Password.
[15:15:01][C][api:095]: API Server:
[15:15:01][C][api:096]:   Address: livingroom.local:6053

```

Figura 19 - *Upload* Concluído com Sucesso

```

AVAILABLE: esp-prog, espota, esptool, iot-bus-jtag, jlink, minipro, olimex-arm-usb-ocd, olimex-arm-usb-ocd-h, olimex-arm-usb-tiny-h,
, olimex-jtag-tiny, tumpa
CURRENT: upload_protocol = esptool
Looking for upload port...
Use manually specified: COM6
Uploading .pioenvs\livingroom\firmware.bin
esptool.py v2.6
Serial port COM6
Connecting.....
A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header
*** [upload] Error 2
===== [FAILED] Took 34.30 seconds =====

```

Figura 20 - Erro no *Upload*

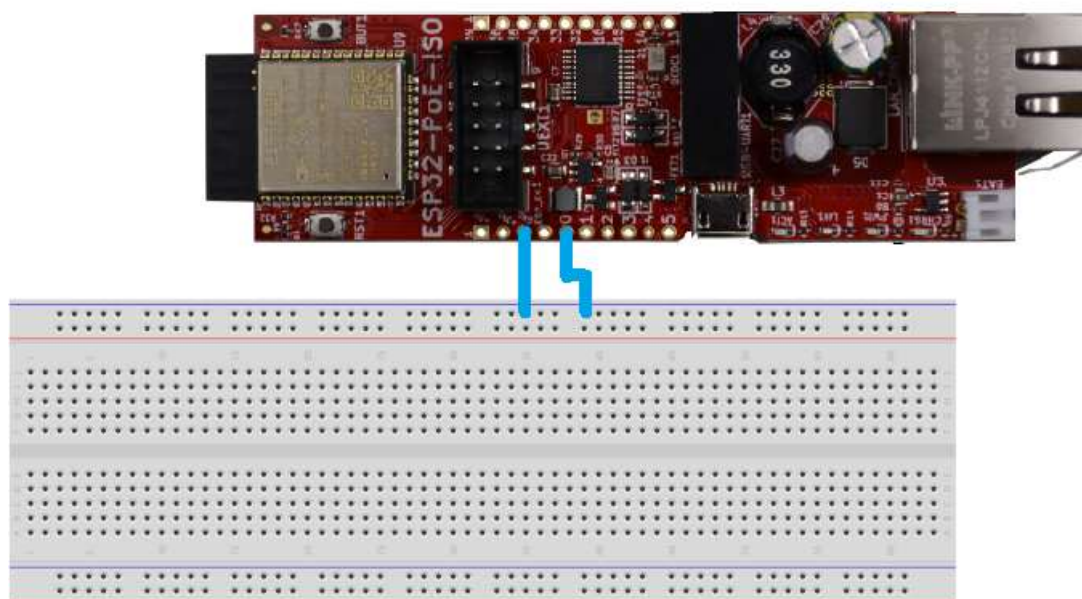


Figura 21 - Esquema de Ligação