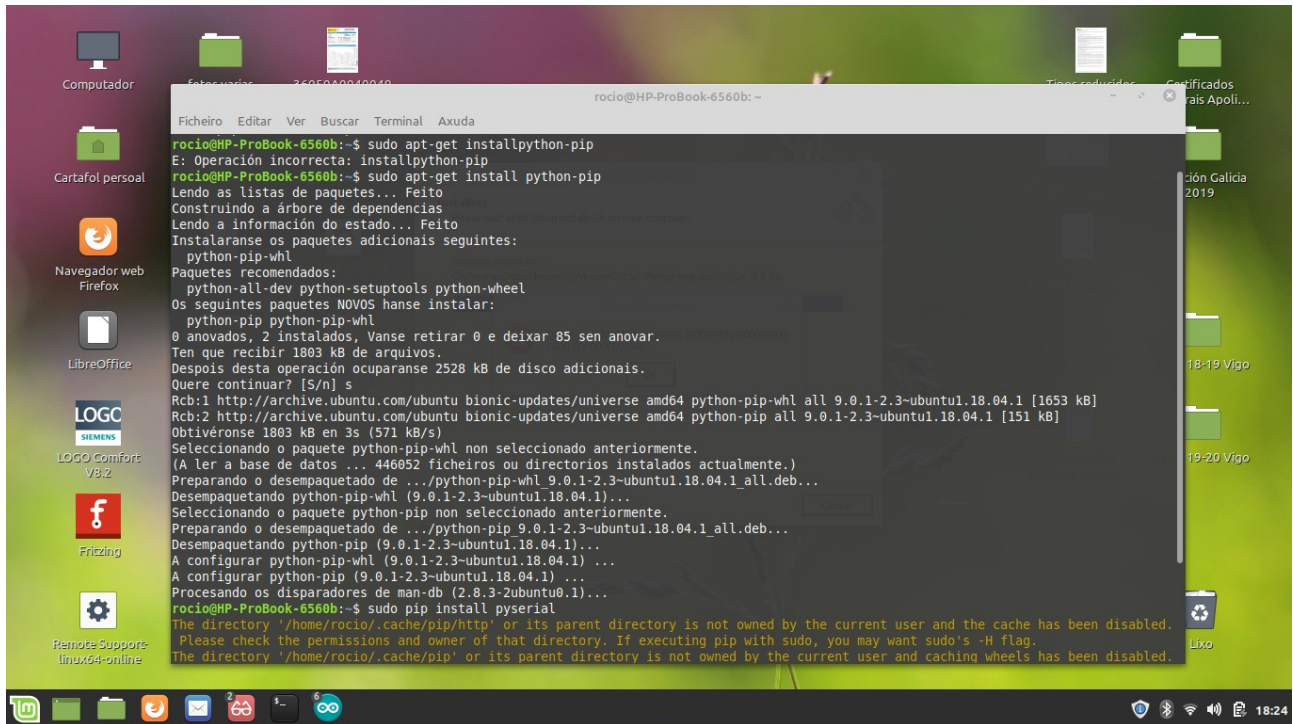


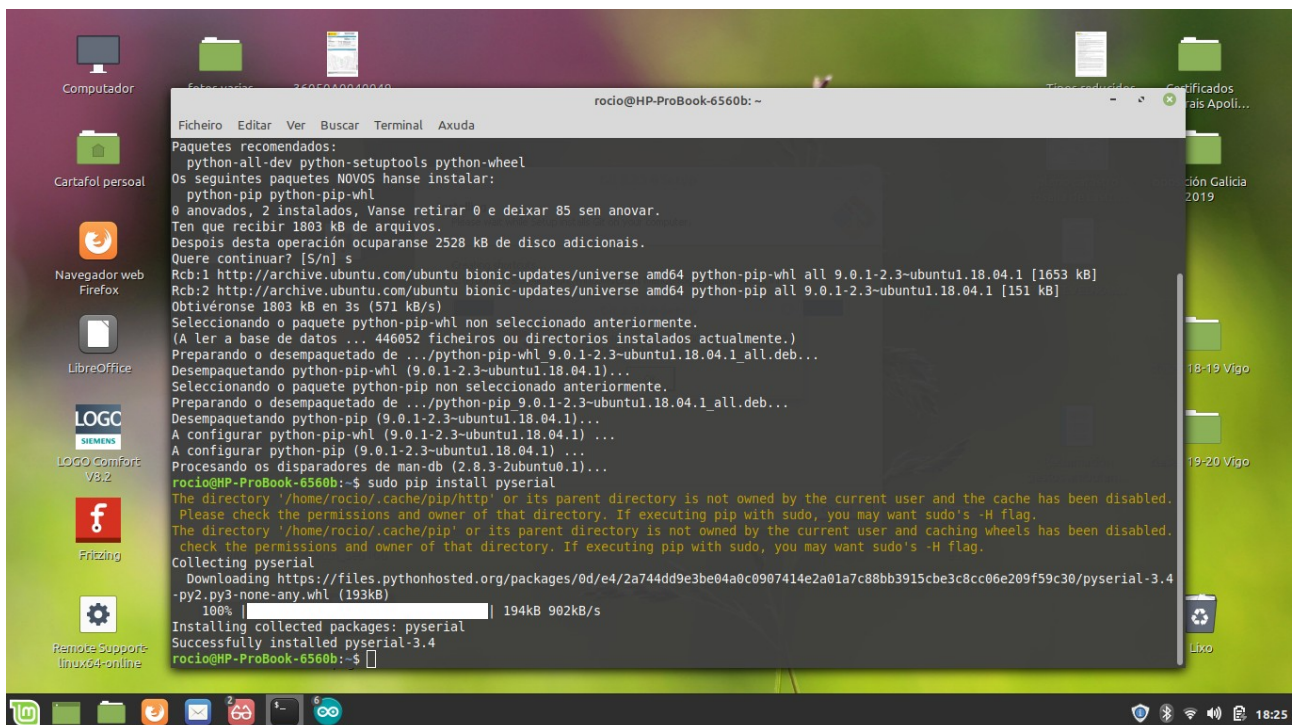
Como engadimos as placas ESP no noso Arduino IDE

1. Vamos precisar engadir Python no noso equipo, e para elo abrimos o terminal e escribimos estes 2 comandos, sempre que traballemos con Linux

sudo apt-get install python-pip

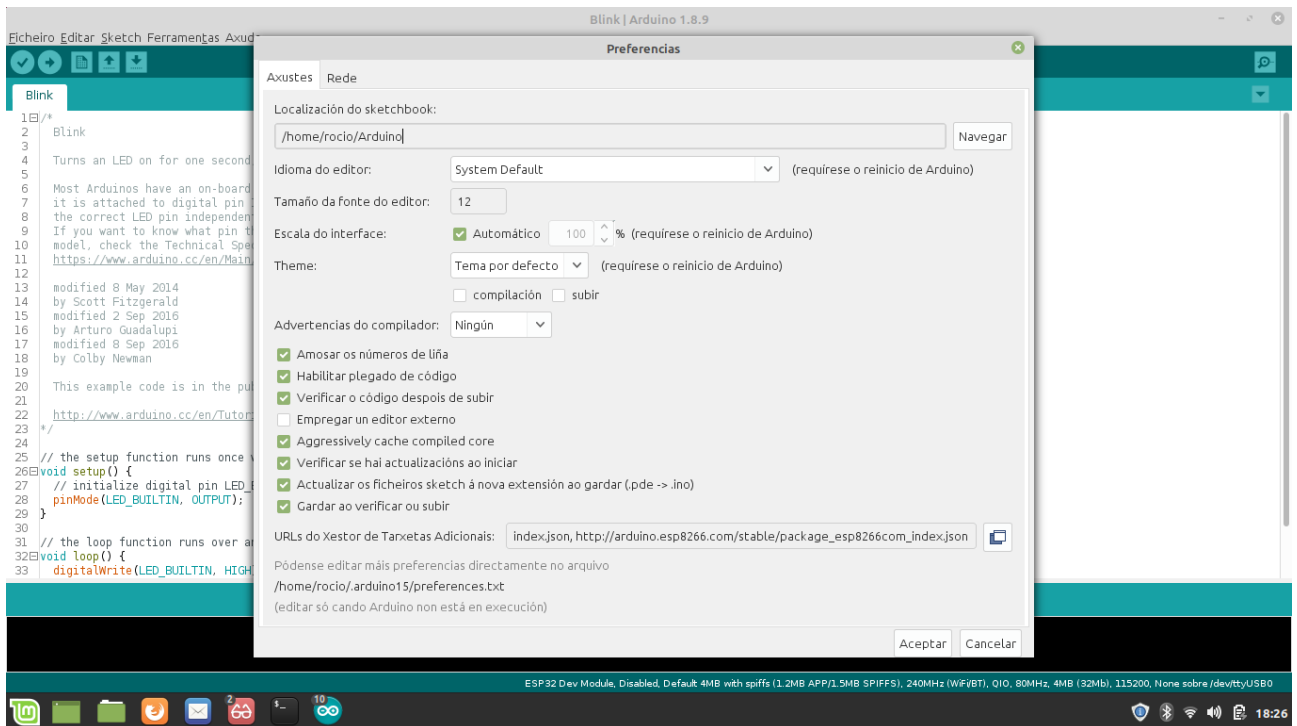


sudo pip install pyserial



2. A continuación temos que recoller a “base de datos” das placas ESP no noso Arduino, que xa temos instalado.

Para elo abrimos Ficheiro – Preferencias e engadimos a URL do xestor de tarxetas adicionais



3. A dirección é:

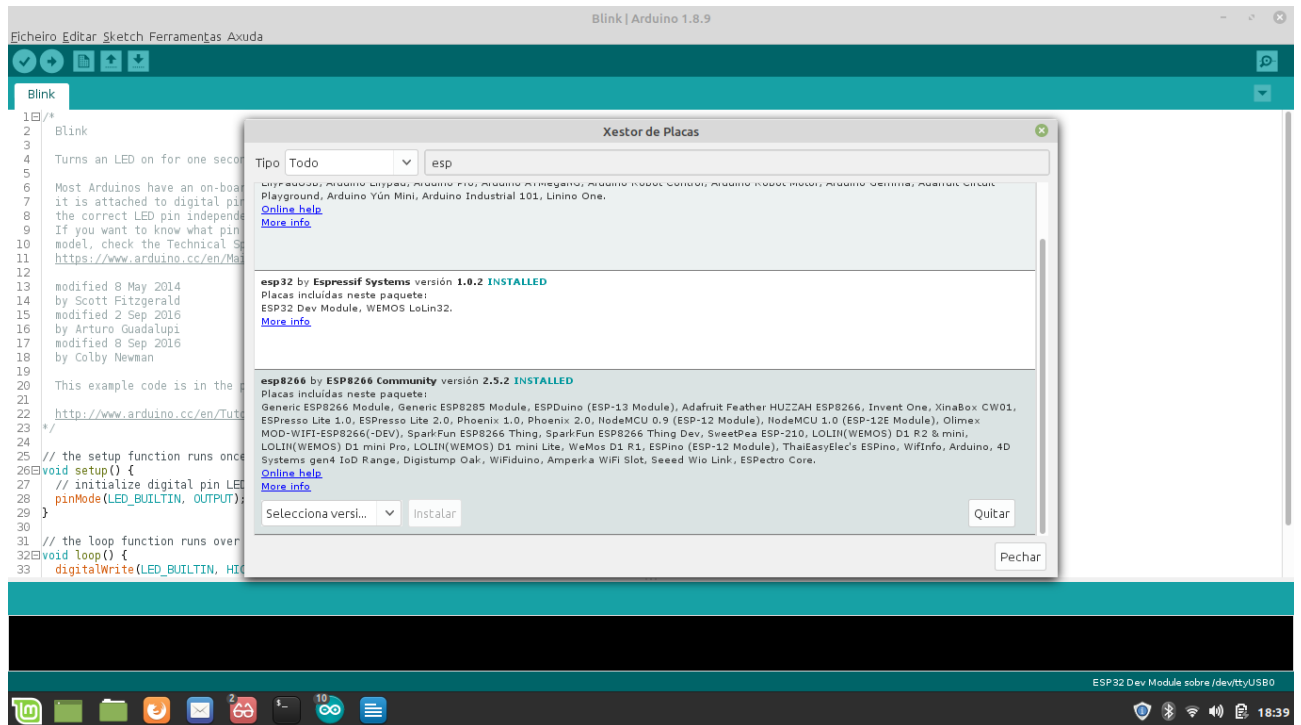
https://dl.espressif.com/dl/package_esp32_index.json

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Se queremos só empregar os modelos ESP32 collemos a primeira liña, se queremos empregar só os modelos ESP82 collemos a 2ª liña, e para incluír todas poñemos as 2 liñas

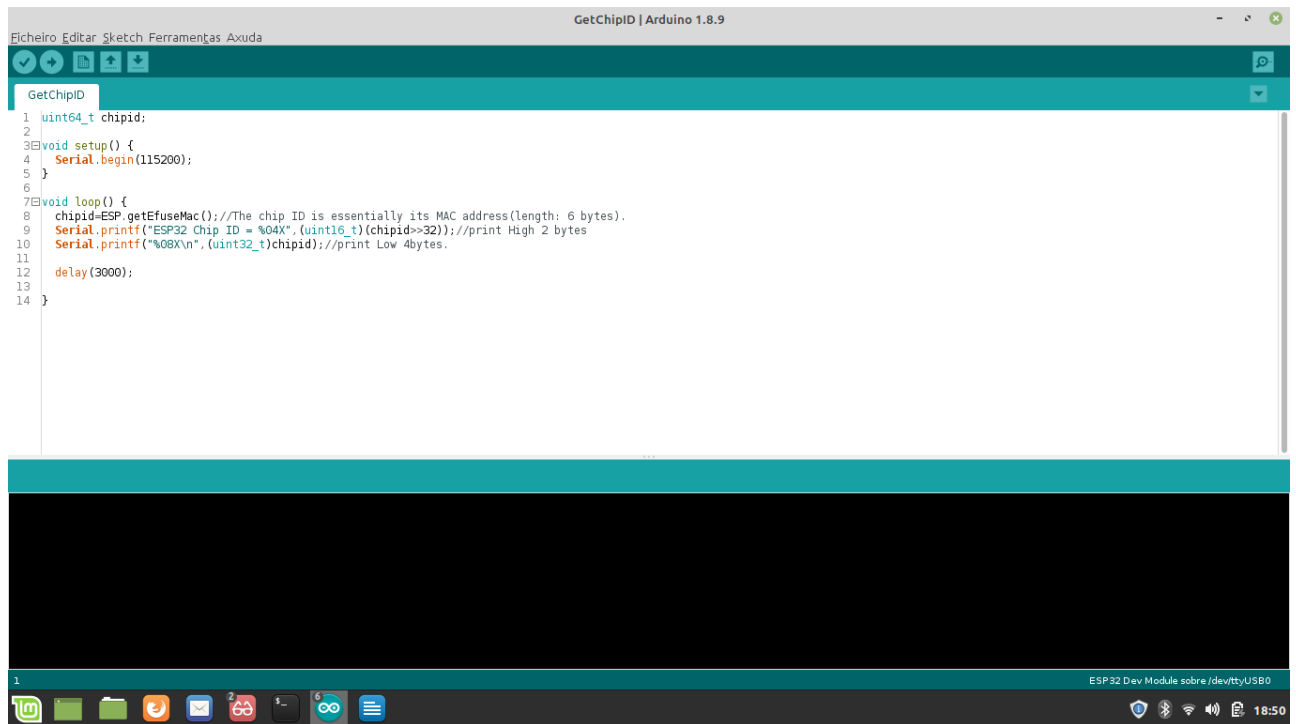
4. Aínda nos queda instalar estas tarxetas no xestor de placas. Entón imos a Ferramentas – Placa – Xestor de placas
E escribimos no buscador ESP

Aparecerannos as seguintes:

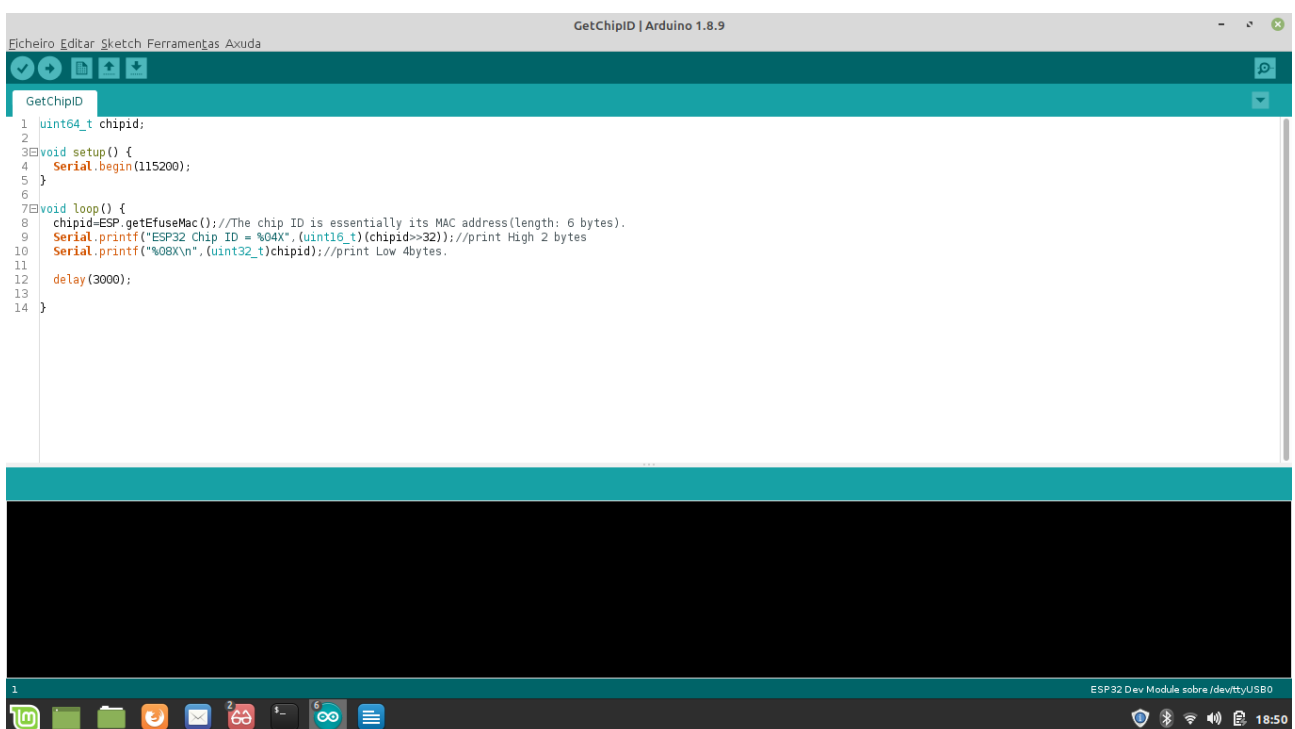


5. Temos que comprobar que esté instalada a ESP32 de Espressif ou a ESP82, ou ambas, segundo precisemos. Se non están instalámolas
6. É preciso reiniciar o Arduino para que recoñeza ben as novas placas
7. Agora vamos realizar a primeira proba co Espressif 32, aproveitando os exemplos da librería, para elo vamos a Ficheiro – Exemplos – ESP32 – ChipID – GetChipID

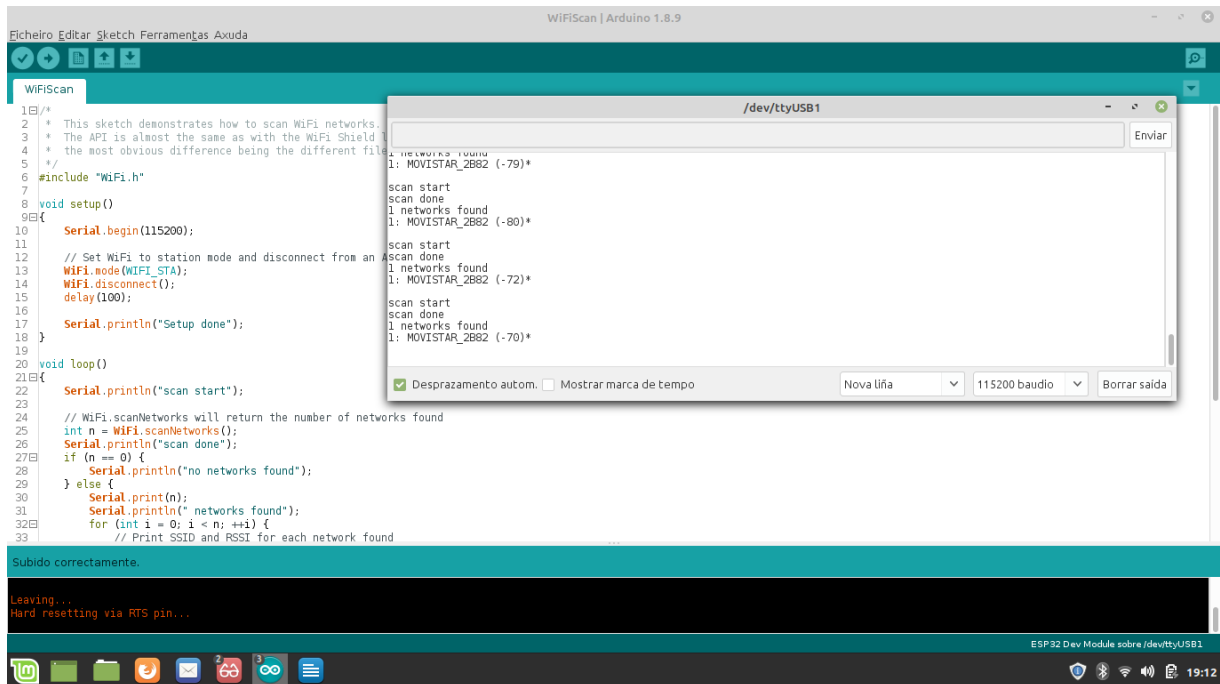
Abrimos o arquivo e o primeiro que podemos observar é que a velocidade de transmisión do serie son 115200 en lugar dos 9600 das placas arduino



8. Nas ferramentas temos que escoller placa ESP32 Dev Module, upload speed 115200 e o porto que nos recoñeza o noso equipo no meu caso USB0
9. Para cargar o programa acordémonos de premer o BOOT
10. E para comezar a executar o programa temos que premer EN
11. Abrimos o monitor serie xa mentres cargamos para non perder información, ou incluso antes de cargar o programa, e así mostraranos o chip do noso ESP32



Tamén temos outros exemplos como a wifi que podemos empregar do noso ESP32. Se collemos Ficheiro- Exemplos – Wifi – WifiScan poderemos ver nuns minutos cales son as redes wifi que ten o alcance o ESP32



The screenshot shows the Arduino IDE interface with the 'WifiScan' sketch loaded. The sketch is a C++ program that demonstrates how to scan for WiFi networks using the WiFi Shield library. It includes comments explaining the API and the difference between the WiFi Shield and the ESP32. The code sets the WiFi module to station mode, disconnects from any existing network, and then scans for available networks. The results are printed to the serial monitor.

```
1 // This sketch demonstrates how to scan WiFi networks.
2 // * The API is almost the same as with the WiFi Shield
3 // * the most obvious difference being the different file
4 // */
5 #include "WiFi.h"
6
7 void setup()
8 {
9   Serial.begin(115200);
10
11   // Set WiFi to station mode and disconnect from an AP
12   // if it has previously been configured.
13   WiFi.mode(WIFI_STA);
14   WiFi.disconnect();
15   delay(100);
16   Serial.println("Setup done");
17 }
18
19 void loop()
20 {
21   Serial.println("scan start");
22
23   // WiFi.scanNetworks will return the number of networks found
24   int n = WiFi.scanNetworks();
25   Serial.println("scan done");
26   if (n == 0) {
27     Serial.println("no networks found");
28   } else {
29     Serial.print(n);
30     Serial.println(" networks found");
31     for (int i = 0; i < n; ++i) {
32       // Print SSID and RSSI for each network found
33     }
```

The serial monitor window shows the output of the sketch. It displays the following text:

```
scan start
scan done
1 networks found
1: MOVISTAR_2B82 (-79)*
scan start
scan done
1 networks found
1: MOVISTAR_2B82 (-80)*
scan start
scan done
1 networks found
1: MOVISTAR_2B82 (-72)*
scan start
scan done
1 networks found
1: MOVISTAR_2B82 (-70)*
```

The status bar at the bottom of the IDE indicates that the sketch was uploaded successfully: "Subido correctamente." It also shows the board name "ESP32 Dev Module" and the port "/dev/ttyUSB1".

E tamén podemos subir un programa para traballar co sensor dht11 por exemplo, pero hai que ter en conta que pins temos dispoñibles nesta placa