# Raspberry Pi

Configuración – Introducción a IoT utilizando Adafruit.io

### Raspberry Pi 3 Model A+

Our third-generation single-board computer, now in the A+ format

BUY NOW >

### Raspberry Pi 3 Model B+

The latest revision of our third-generation single-board computer

BUY NOW >

or buy for business

### Raspberry Pi 3 Model B

Our third-generation single-board computer

BUY NOW >

or buy for business

### Raspberry Pi 2 Model B

The Raspberry Pi 2 Model B is the second-generation Raspberry Pi

BUY NOW >

or buy for business

### Raspberry Pi 1 Model B+

The Model B+ is the final revision of the original Raspberry Pi

BUY NOW >

or buy for business

### Raspberry Pi 1 Model A+

The Model A+ is the low-cost variant of the Raspberry Pi

BUY NOW >

or buy for business

### Raspberry Pi Zero W

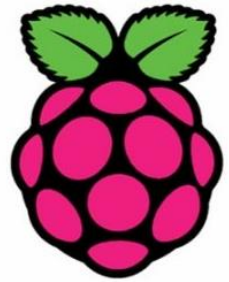
Single-board computer with wireless and Bluetooth connectivity

BUY NOW >

### Raspberry Pi Zero

Our lowest-cost single-board computer

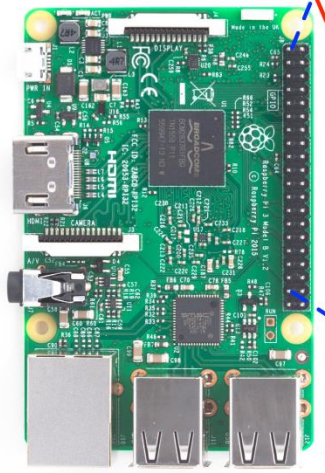
BUY NOW >



| - | Raspberry Pi 1 Modelo A | Raspberry Pi 1 Modelo B | Raspberry Pi 1 Modelo B+ | Raspberry Pi 2 Modelo B | Raspberry Pi 3 Modelo B | Raspberry Pi 3 Modelo B+ |
|---|---|---|---|---|---|---|
| SoC:[4] | Broadcom BCM2835 (CPU + GPU + DSP + SDRAM + puerto USB)[2] | | | Broadcom BCM2836 (CPU + GPU + DSP + SDRAM + Puerto USB) | Broadcom BCM2837 (CPU + GPU + DSP + SDRAM + Puerto USB) | Broadcom BCM2837 (CPU + GPU + DSP + SDRAM + Puerto USB) |
| CPU: | ARM 1176JZF-S a 700 MHz (familia ARM11)[2] | | | 900 MHz quad-core ARM Cortex A7 | 1.2GHz 64-bit quad-core ARMv8 | 1.4GHz 64-bit quad-core ARMv8 |
| Juego de instrucciones: | RISC de 32 bits | | | | RISC de 64 bits | RISC de 64 bits |
| GPU: | Broadcom VideoCore IV,,[55] OpenGL ES 2.0, MPEG-2 y VC-1 (con licencia),[56] 1080p30 H.264/MPEG-4 AVC[2] | | | | | |
| Memoria (SDRAM): | 256 MiB (compartidos con la GPU) | 512 MiB (compartidos con la GPU)[57] desde el 15 de octubre de 2012 | | 1 GB (compartidos con la GPU) | | |
| Puertos USB 2.0:[58] | 1 | 2 (vía hub USB integrado)[59] | 4 | | | |
| Entradas de vídeo:[60] | Conector MIPI CSI que permite instalar un módulo de cámara desarrollado por la RPF | | | | | |
| Salidas de vídeo:[4] | Conector RCA (PAL y NTSC), HDMI (rev1.3 y 1.4),[61] Interfaz DSI para panel LCD[62] [63] | | | | | |
| Salidas de audio:[4] | Conector de 3.5 mm, HDMI | | | | | |
| Almacenamiento integrado: | SD / MMC / ranura para SDIO | | MicroSD | | | |
| Conectividad de red:[4] | Ninguna | 10/100 Ethernet (RJ-45) vía hub USB[59] | | | 10/100 Ethernet (RJ-45) vía hub USB,[64] Wifi 802.11n, Bluetooth 4.1 | 10/100/1000 Ethernet (RJ-45) vía hub USB Max 300Mbits/s ,[64] Wifi 802.11n/ac, Bluetooth 4.2 BLE |
| Periféricos de bajo nivel: | 8 x GPIO, SPI, I²C, UART[55] | | | 17 x GPIO y un bus HAT ID | | |

**Pi Zero**

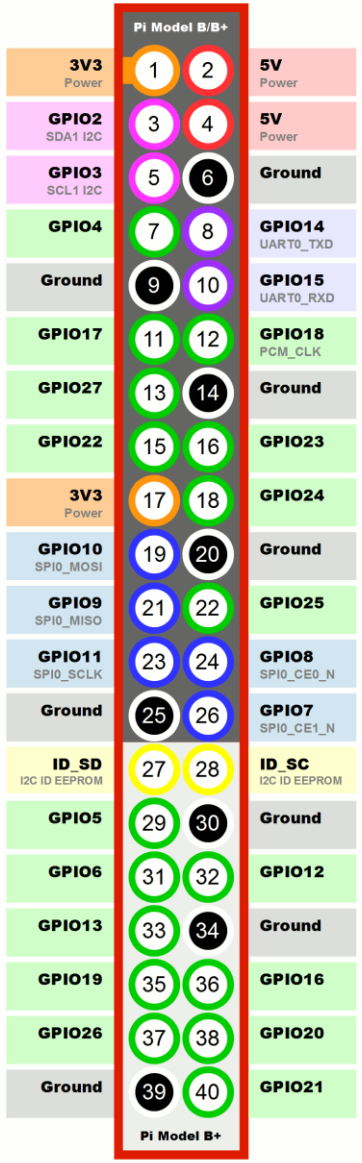
**Pi 3**



| | Pin No. | | |
|---|---|---|---|
| 3.3V | 1 | 2 | 5V |
| GPIO2 | 3 | 4 | 5V |
| GPIO3 | 5 | 6 | GND |
| GPIO4 | 7 | 8 | GPIO14 |
| GND | 9 | 10 | GPIO15 |
| GPIO17 | 11 | 12 | GPIO18 |
| GPIO27 | 13 | 14 | GND |
| GPIO22 | 15 | 16 | GPIO23 |
| 3.3V | 17 | 18 | GPIO24 |
| GPIO10 | 19 | 20 | GND |
| GPIO9 | 21 | 22 | GPIO25 |
| GPIO11 | 23 | 24 | GPIO8 |
| GND | 25 | 26 | GPIO7 |
| DNC | 27 | 28 | DNC |
| GPIO5 | 29 | 30 | GND |
| GPIO6 | 31 | 32 | GPIO12 |
| GPIO13 | 33 | 34 | GND |
| GPIO19 | 35 | 36 | GPIO16 |
| GPIO26 | 37 | 38 | GPIO20 |
| GND | 39 | 40 | GPIO21 |

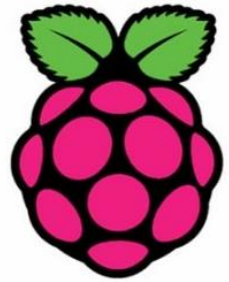
| Key | |
|---|---|
| Power + | UART |
| GND | SPI |
| I²C | GPIO |





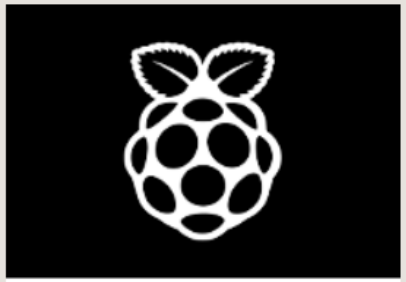
https://www.raspberrypi.org/downloads/raspbian/

Blog Downloads Community Help Forums Education



# Downloads

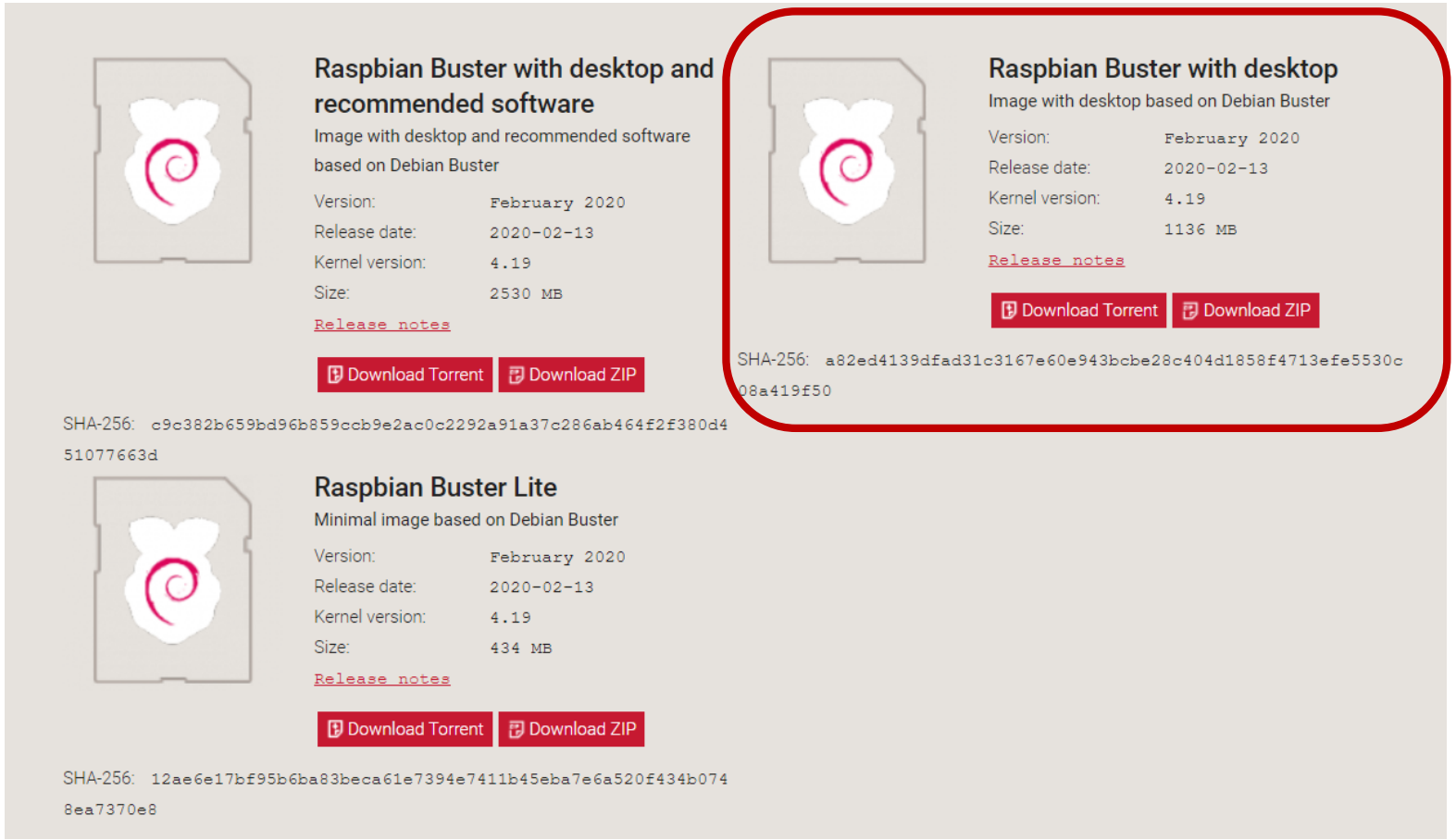
**Raspbian** is our official operating system for **all** models of the Raspberry Pi.
Download it here, or use **NOOBS**, our easy installer for Raspbian and more.



NOOBS

Raspbian

Seleccionar Raspbian



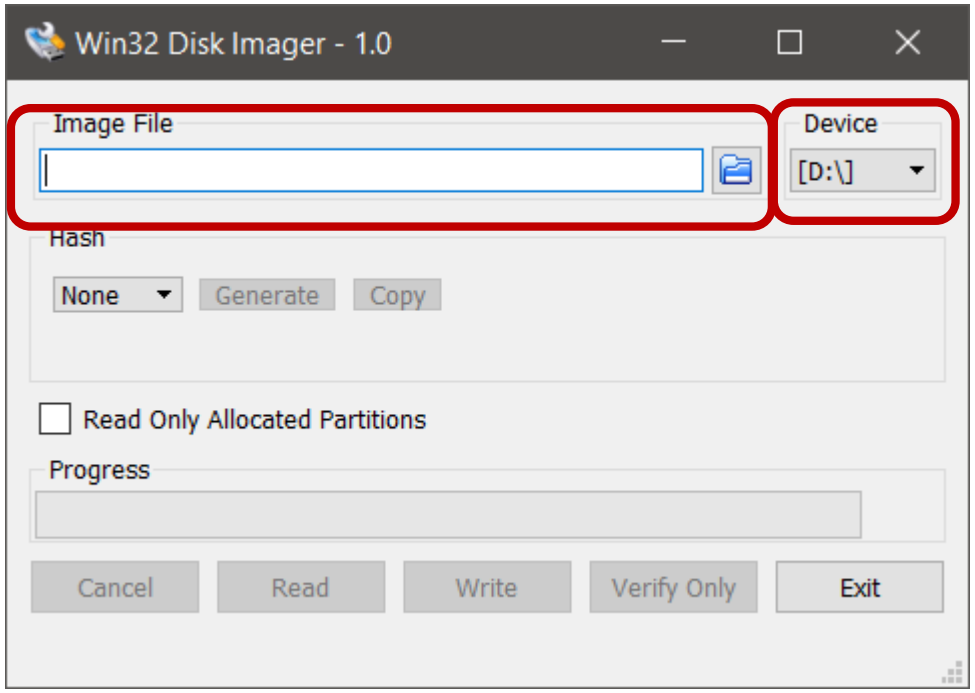

Descargar la última versión de Raspbian al momento, de preferencia la que incluye el escritorio

https://www.sdcard.org/downloads/formatter_4/

https://sourceforge.net/projects/win32diskimager/

Una vez descargamos la imagen de Raspbian

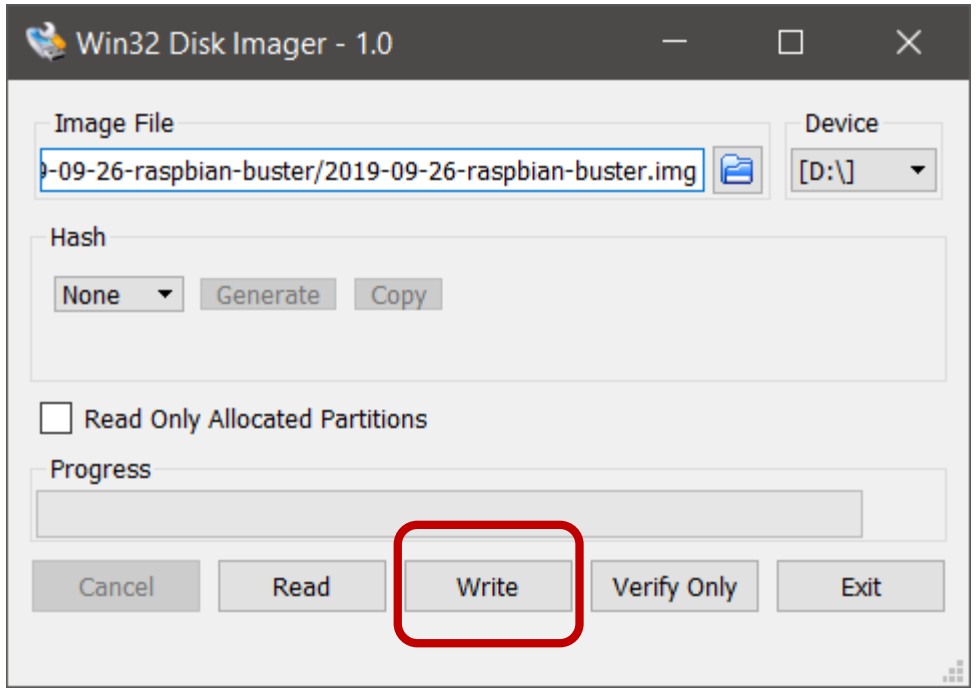
La seleccionamos dicha imagen




Verificamos que la unidad en donde vamos a montar el sistema operativo sea el adecuado, ya que se realizará un formateo de la unidad

https://sourceforge.net/projects/win32diskimager/

Una vez tengamos seleccionada la imagen

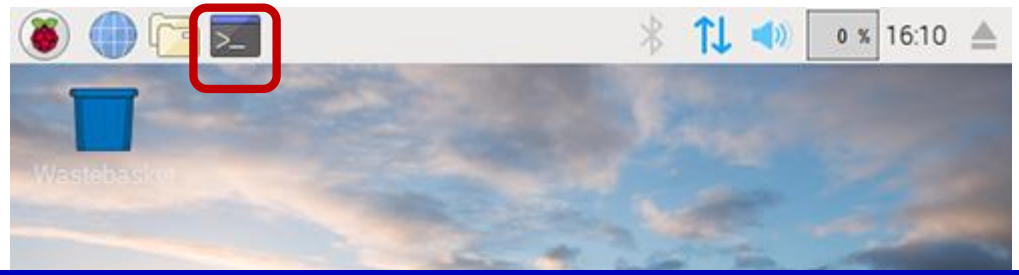


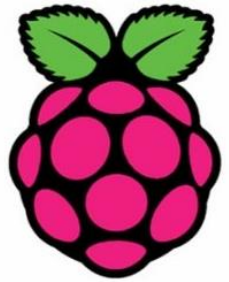

Hacemos clic en Write para montar el sistema operativo en la microSD

https://sourceforge.net/projects/win32diskimager/

Para configurar la Raspberry nos metemos a la terminal

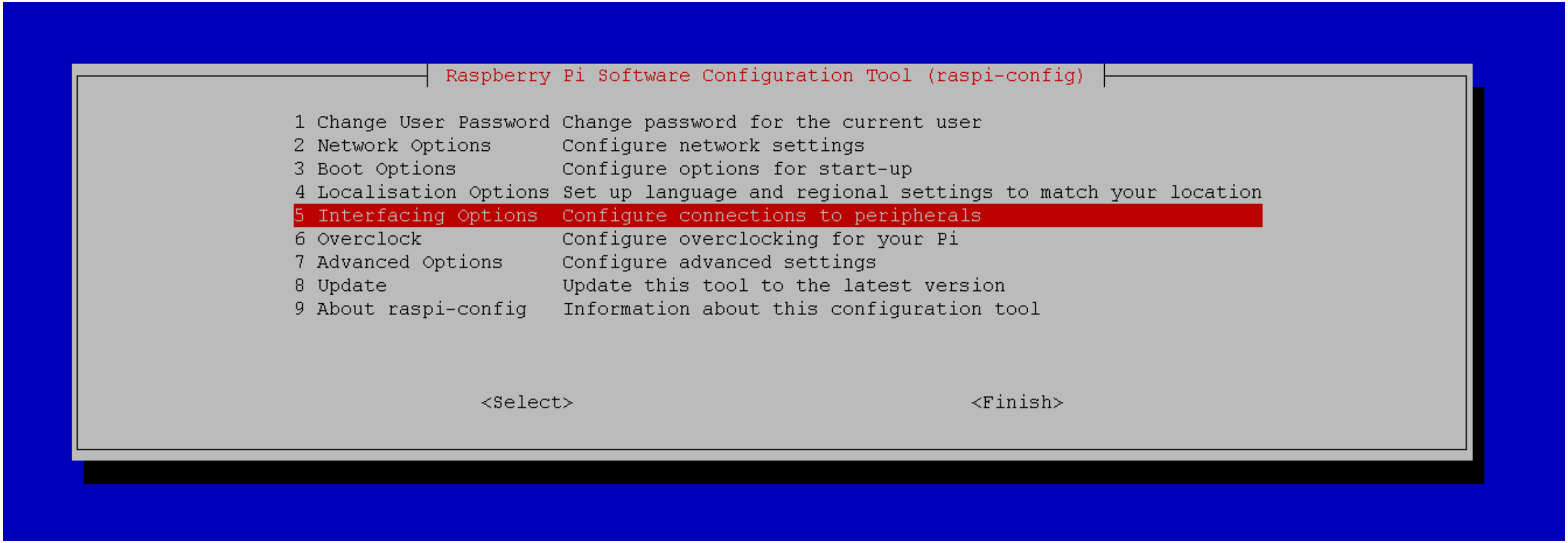
```
pi@raspberrypi:~ $ sudo raspi-config
```



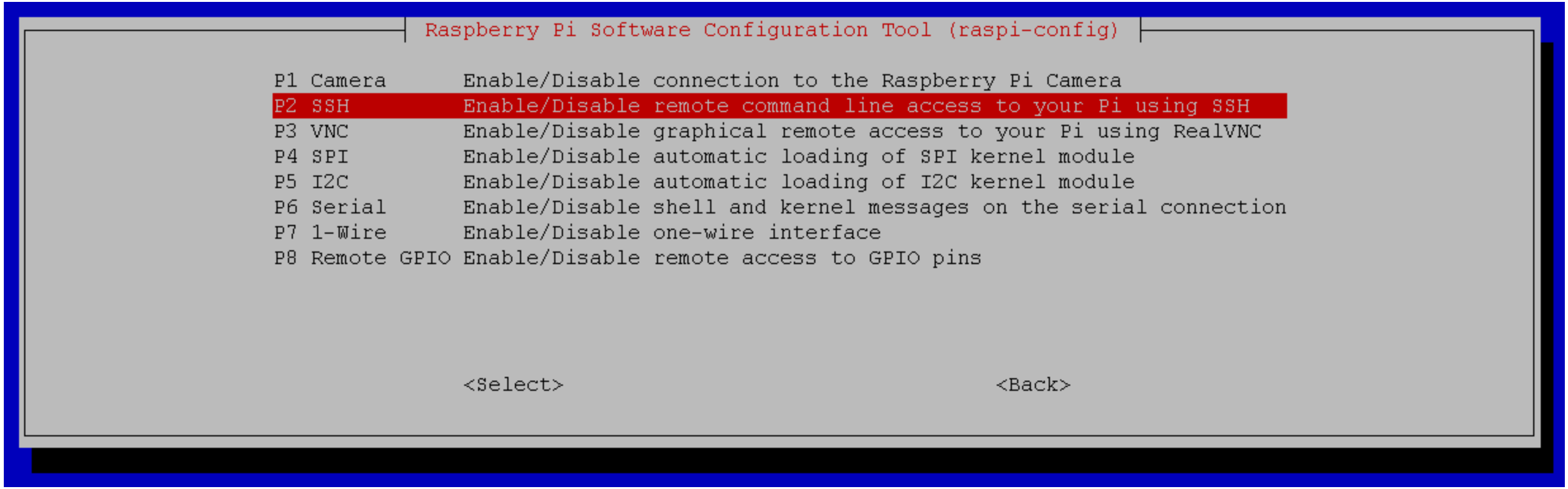



```
                Raspberry Pi Software Configuration Tool (raspi-config)

    1 Change User Password Change password for the current user
    2 Network Options      Configure network settings
    3 Boot Options         Configure options for start-up
    4 Localisation Options Set up language and regional settings to match your location
    5 Interfacing Options  Configure connections to peripherals
    6 Overclock            Configure overclocking for your Pi
    7 Advanced Options     Configure advanced settings
    8 Update               Update this tool to the latest version
    9 About raspi-config   Information about this configuration tool


                      <Select>                              <Finish>
```

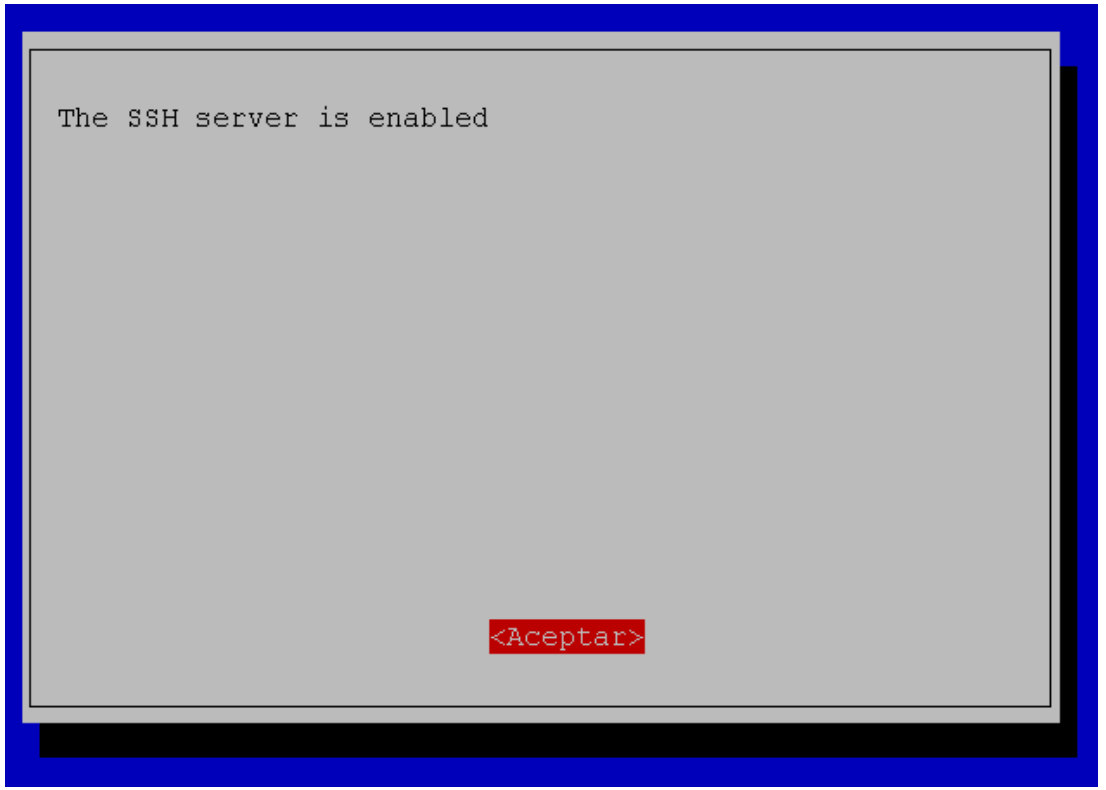
Habilitamos comunicación SSH, I2C, SPI y Serial



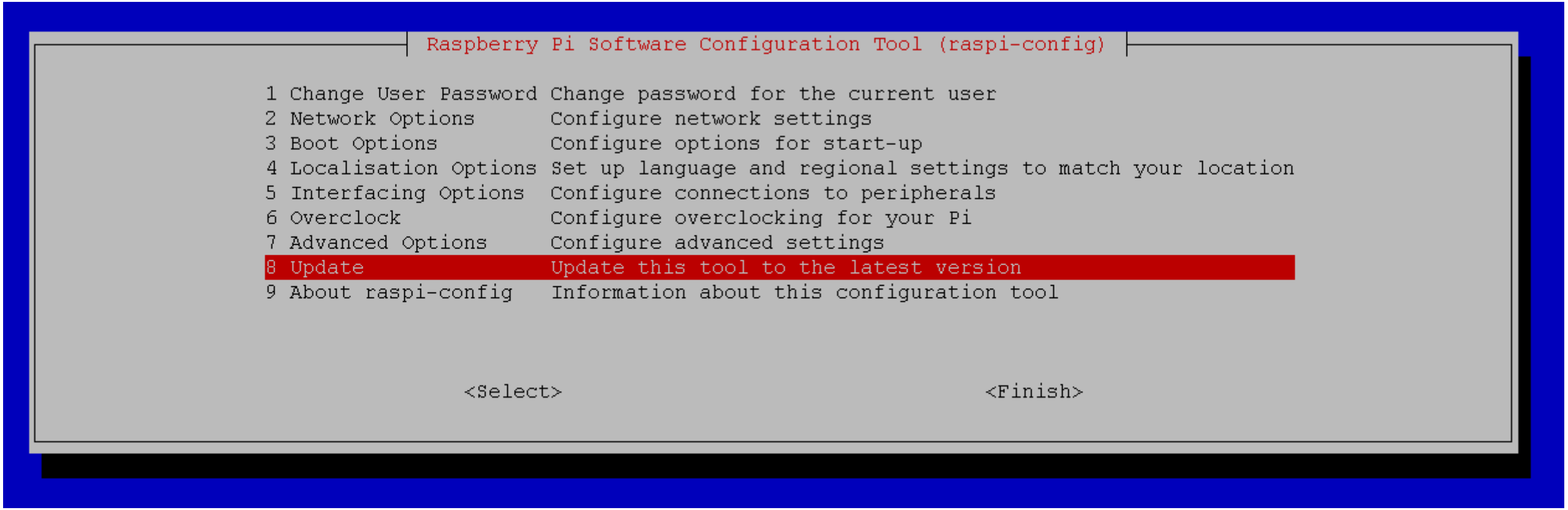
Habilitamos comunicación SSH, I2C, SPI y Serial



# Habilitamos comunicación SSH, VNC, I2C, SPI y Serial

Buscamos actualizaciones del sistema operativo



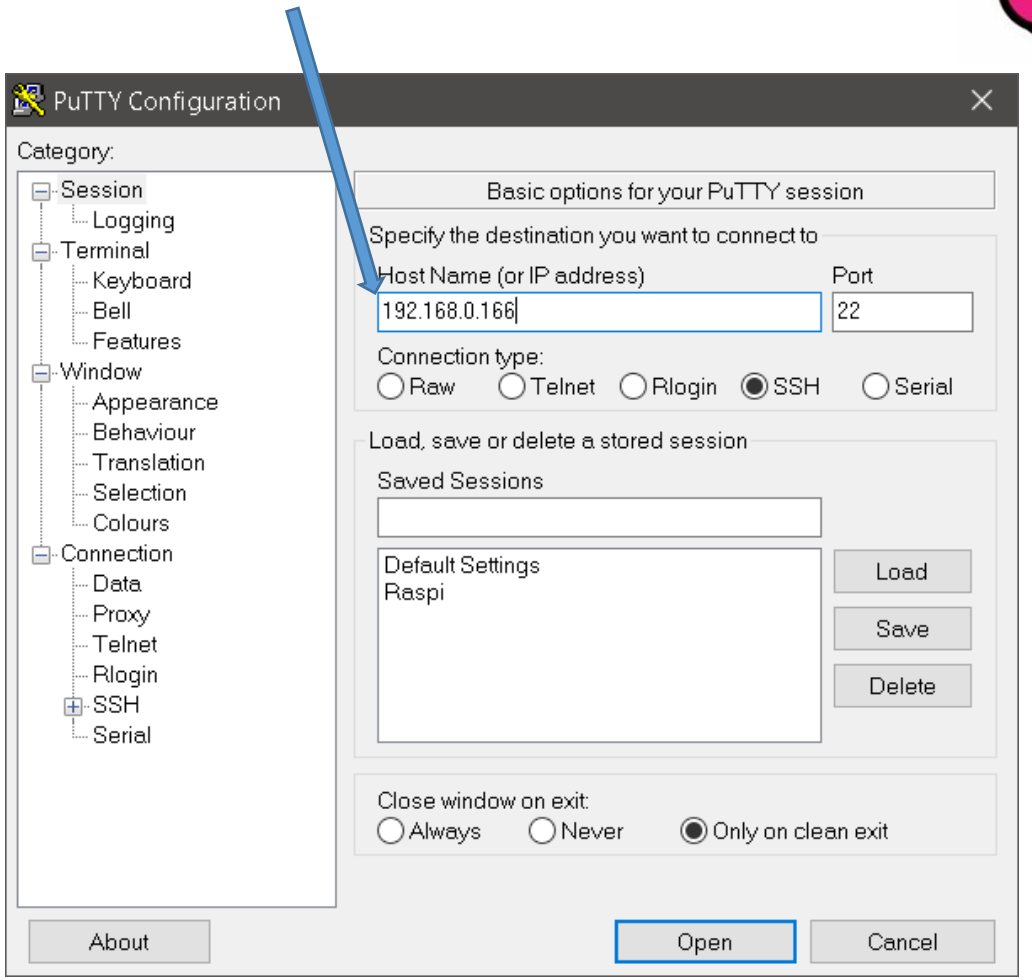
**NOTA: esto solo funciona si están conectados bajo la misma red**

Para conocer las propiedades de las redes

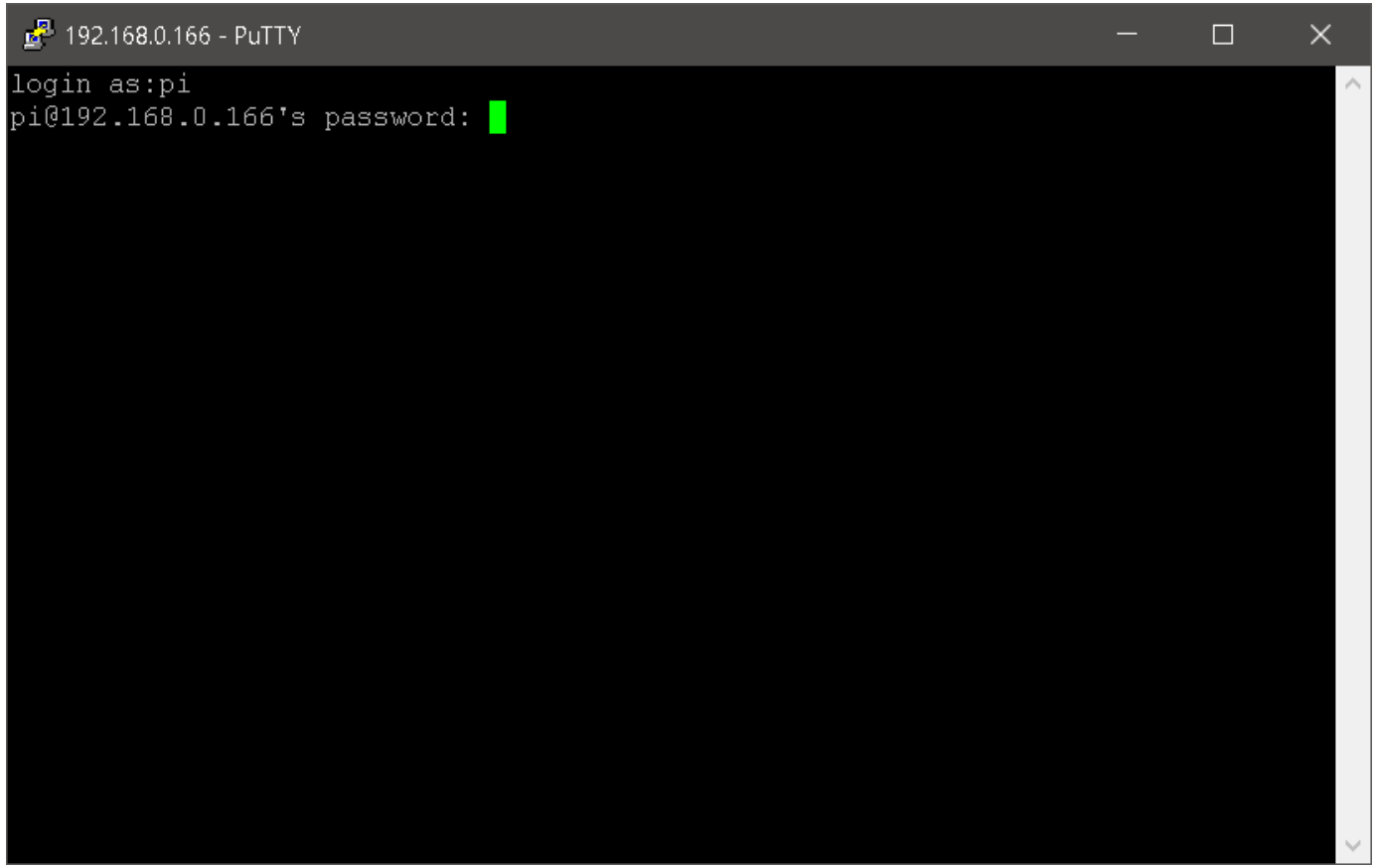
```
pi@raspberrypi:~ $ ifconfig
```

```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b8:27:eb:c3:44:09  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 20  bytes 1136 (1.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20  bytes 1136 (1.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.166  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::8655:104a:149c:a19a  prefixlen 64  scopeid 0x20<link>
        ether d8:eb:97:2f:cd:47  txqueuelen 1000  (Ethernet)
        RX packets 783  bytes 168864 (164.9 KiB)
        RX errors 0  dropped 164  overruns 0  frame 0
        TX packets 242  bytes 40278 (39.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

pi@raspberrypi:~ $
```

Colocamos el número de IP que tengamos registrado según la red (wlan0 ó eth0)



Introducimos las credenciales




Por default
User: pi
Password: raspberry

Lo podemos cambiar desde *sudo raspi-config*

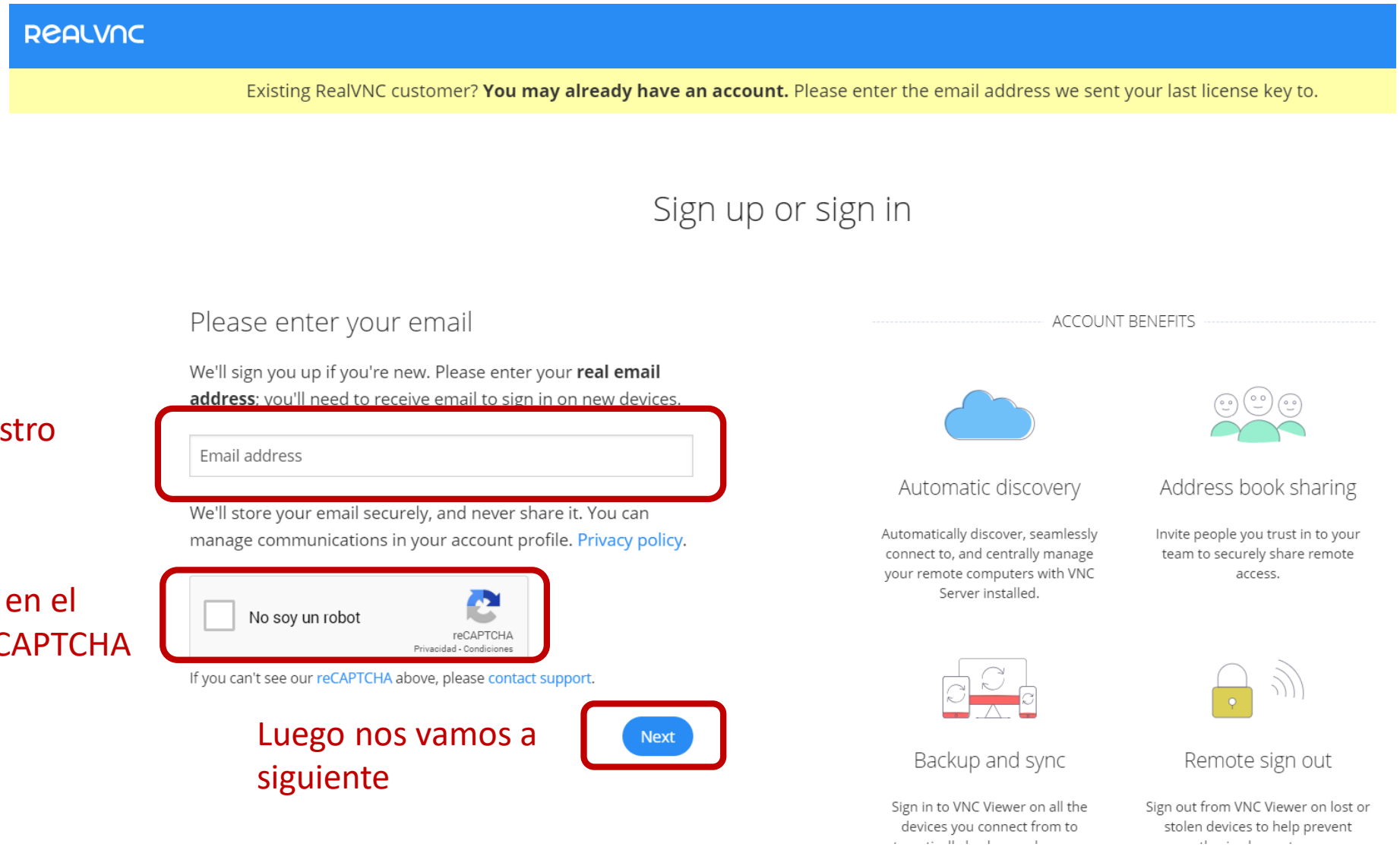
**NOTA: si no estamos bajo la misma red deberemos utilizar VNC**

Podemos descargar el programa de VNC Viewer desde el siguiente enlace:
https://www.realvnc.com/es/connect/download/viewer/


Al correrlo nos aparecerá la siguiente ventana


Hacemos clic en Iniciar sesión


Crearemos una cuenta



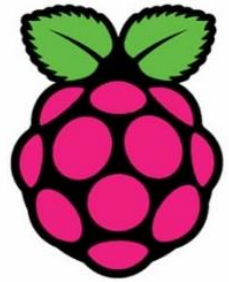

Ingresamos nuestro correo

Hacemos clic en el cuadro de reCAPTCHA

Luego nos vamos a siguiente

Llenamos el formulario

Seleccionamos que vamos a utilizar VNC como uso personal



## Create an account

We just need a few details to get started. Again, please make sure this is your **real email address**.

pdmazariegos@uvg.edu.gt

Change email

Choose a password

Minimum 8 characters. Please make your password difficult to guess, and do not re-use a password from another online service.

First name

Last name

I plan to use VNC Connect for:

- business use
- personal use

Phone number

Optional. Please include your country code, e.g. +1-800-555-0199 for the US.

Country or region

- I am over the age of 16
- I've read and accept the T&Cs
- Keep me up-to-date with RealVNC® news

Les enviarán un correo de confirmación, solo darle clic a verificar correo

Hi Pablo,

Welcome to RealVNC! To get full access to your account, please verify your email address by clicking the button below.

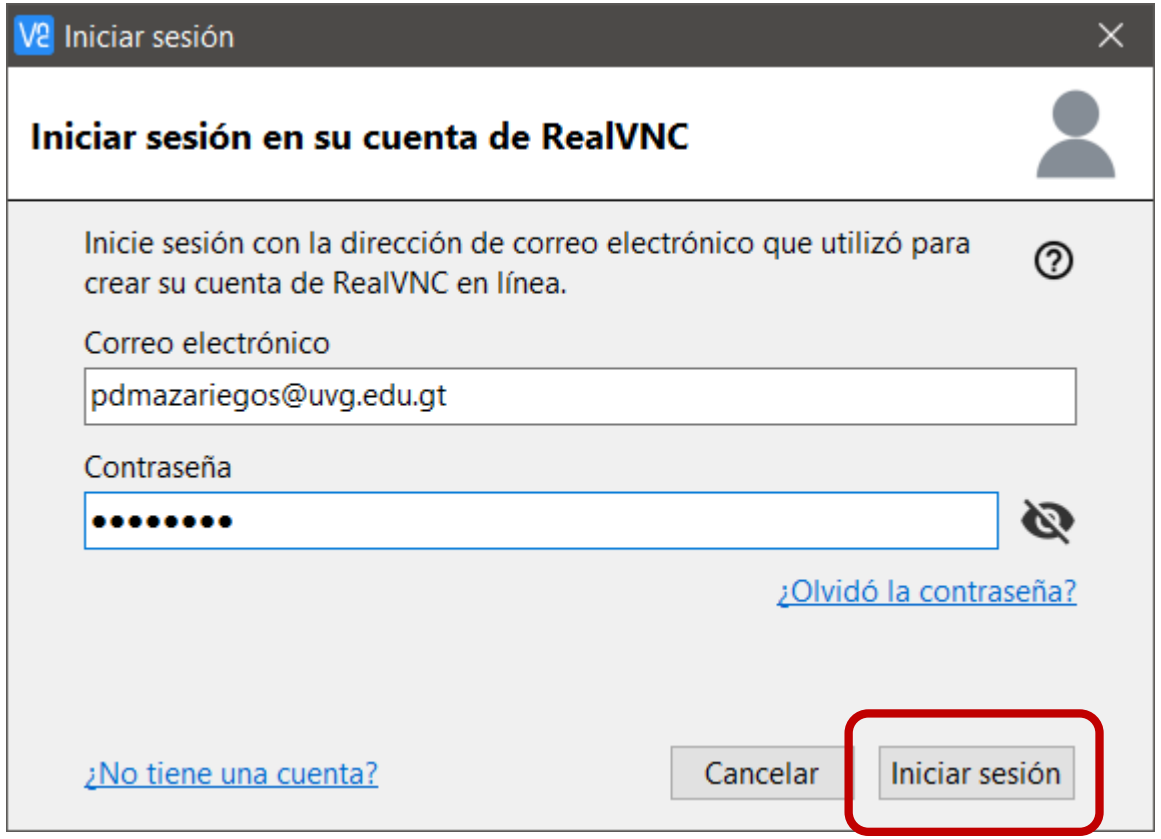
VERIFY EMAIL

**Note:** This must be your **real email address** to authorize each new device you use with VNC Connect, so please make sure it is correct. If not, please verify anyway and then change it on your account's **Profile** page.

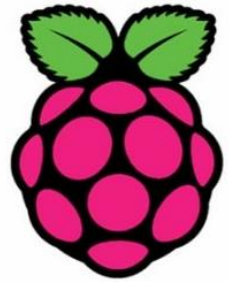
Many thanks,

The RealVNC team

Ya que tenemos nuestra cuenta activa llenamos los campos para iniciar sesión



Por motivos de seguridad nos enviarán otro correo diciendo que nuestras credenciales han sido utilizadas en el siguiente dispositivo. Le damos clic al botón de seguir con la sesión



REALVNC

Hi Pablo,

Your RealVNC account credentials were just used to sign in:

Time: **2020-02-25 01:58:47 UTC**

Device: **Microsoft Windows 10, version 1909 (VNC Viewer)**

Location: **Guatemala City, Guatemala (190.56.75.151)**

CONTINUE SIGNING IN

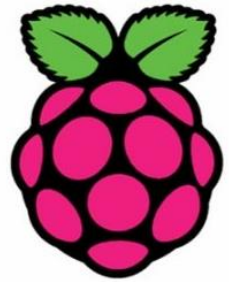
**Not you?**

Don't worry, we won't let this person sign in. But we strongly recommend you change your account password as soon as possible. To do this, sign in yourself at https://manage.realvnc.com and navigate to the **Security** page, where you should also enable 2-step verification (2FA).

Many thanks,

The RealVNC team

Nos llevarán a una página donde autorizaremos al dispositivo a ingresar a nuestra cuenta



# Confirm sign-in

Again, please check you recognize this device, since authorizing will sign it in:

**Time**

2020-02-25 01:58:47 UTC

**Device**

Microsoft Windows 10, version 1909 (VNC Viewer)

**Location**

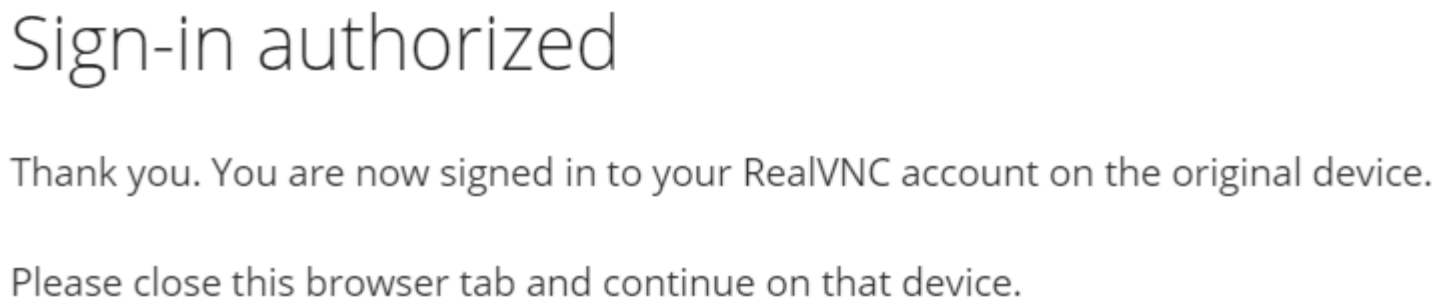
Guatemala City, Guatemala (190.56.75.151)

Authorize sign-in

**Don't recognize this device?**

We strongly recommend you change your RealVNC account password as soon as possible. To do this, sign in yourself and navigate to the **Security** page.

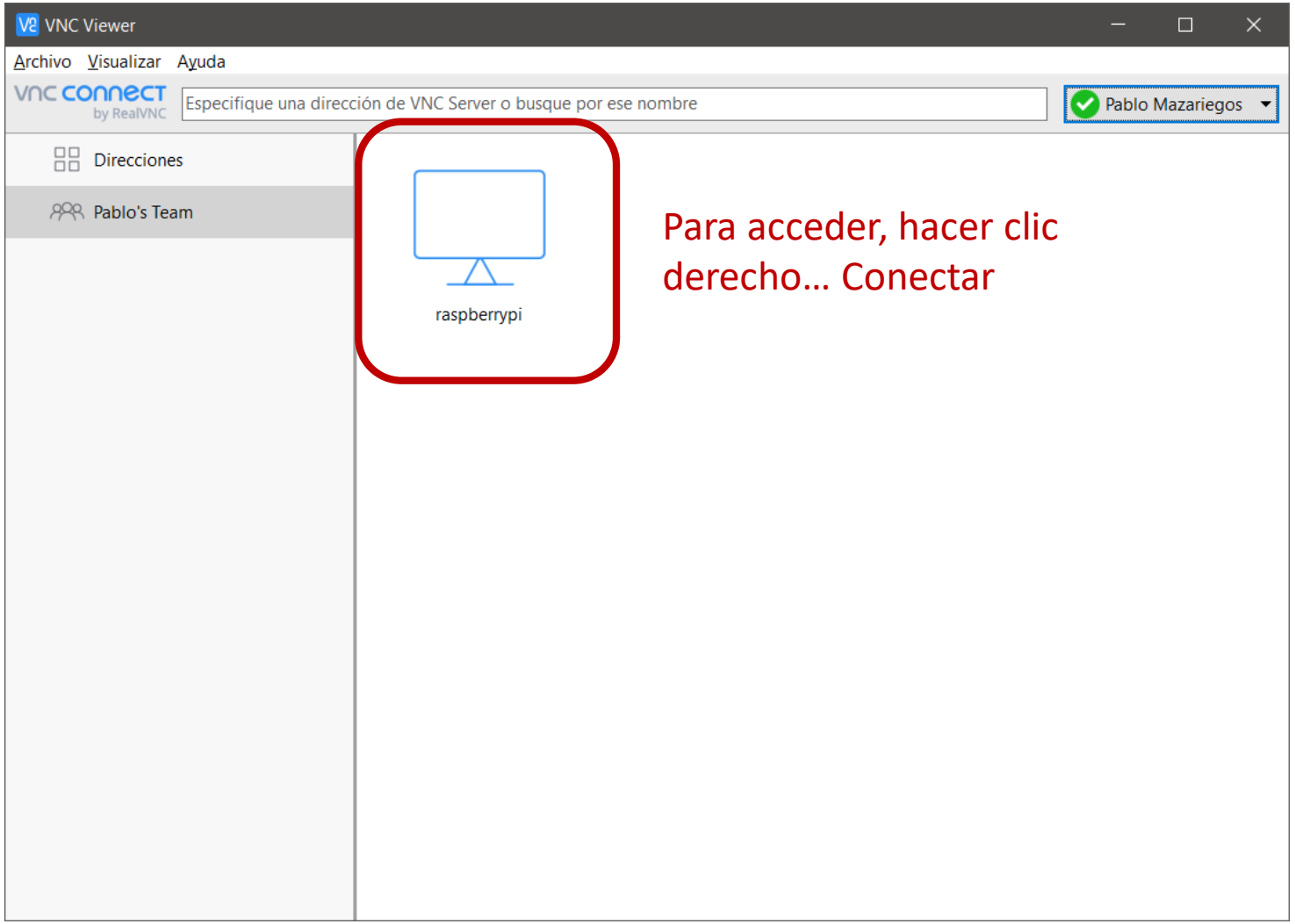
Al finalizar nos aparecerá el siguiente mensaje



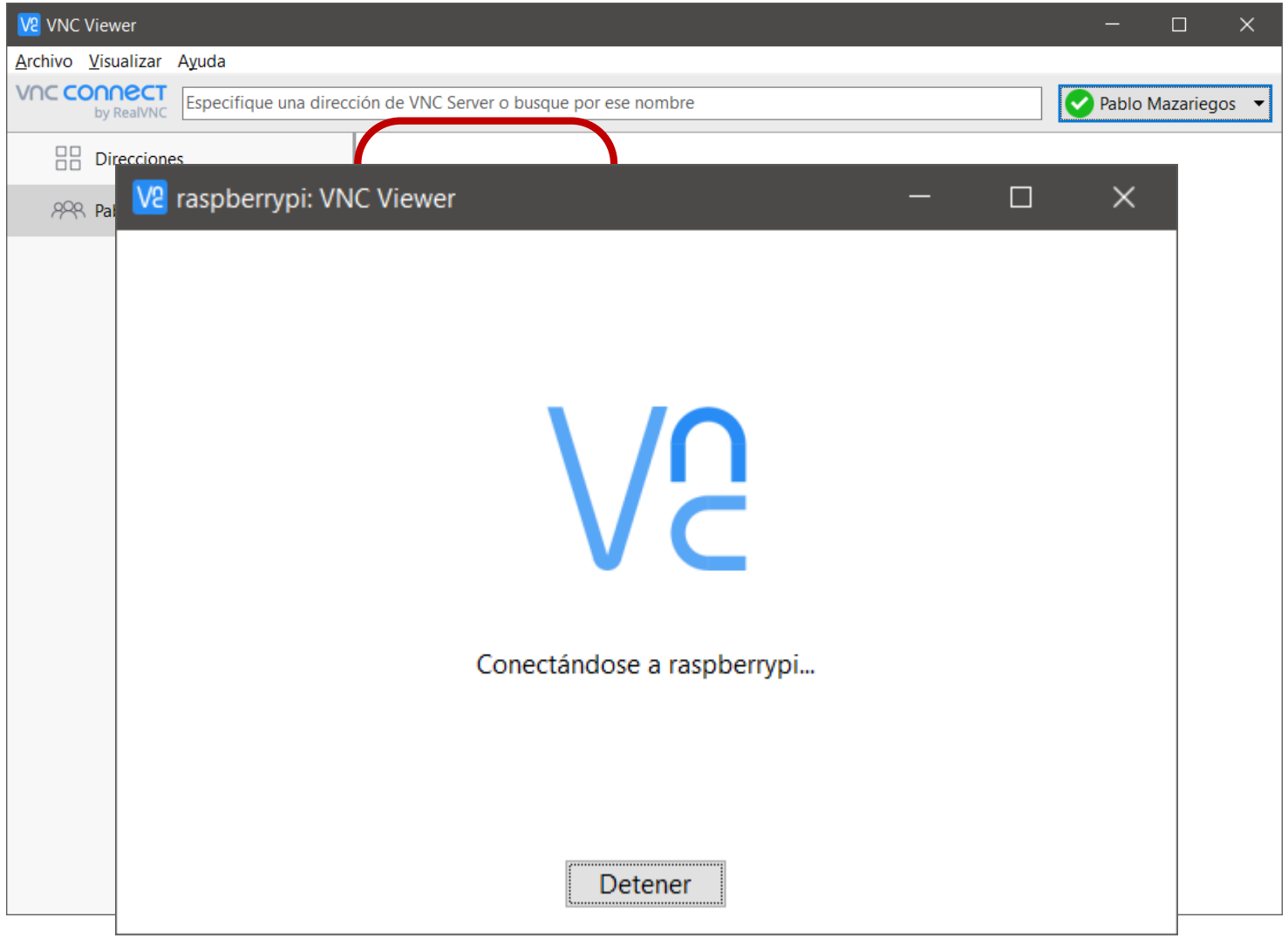

Ahora solo queda activar nuestra cuenta también en el VNC de la raspberry, realizaremos el mismo procedimiento.

Cuando se les pregunte sobre la suscripción se deberá seleccionar Suscripción a domicilio (uso estrictamente no comercial)

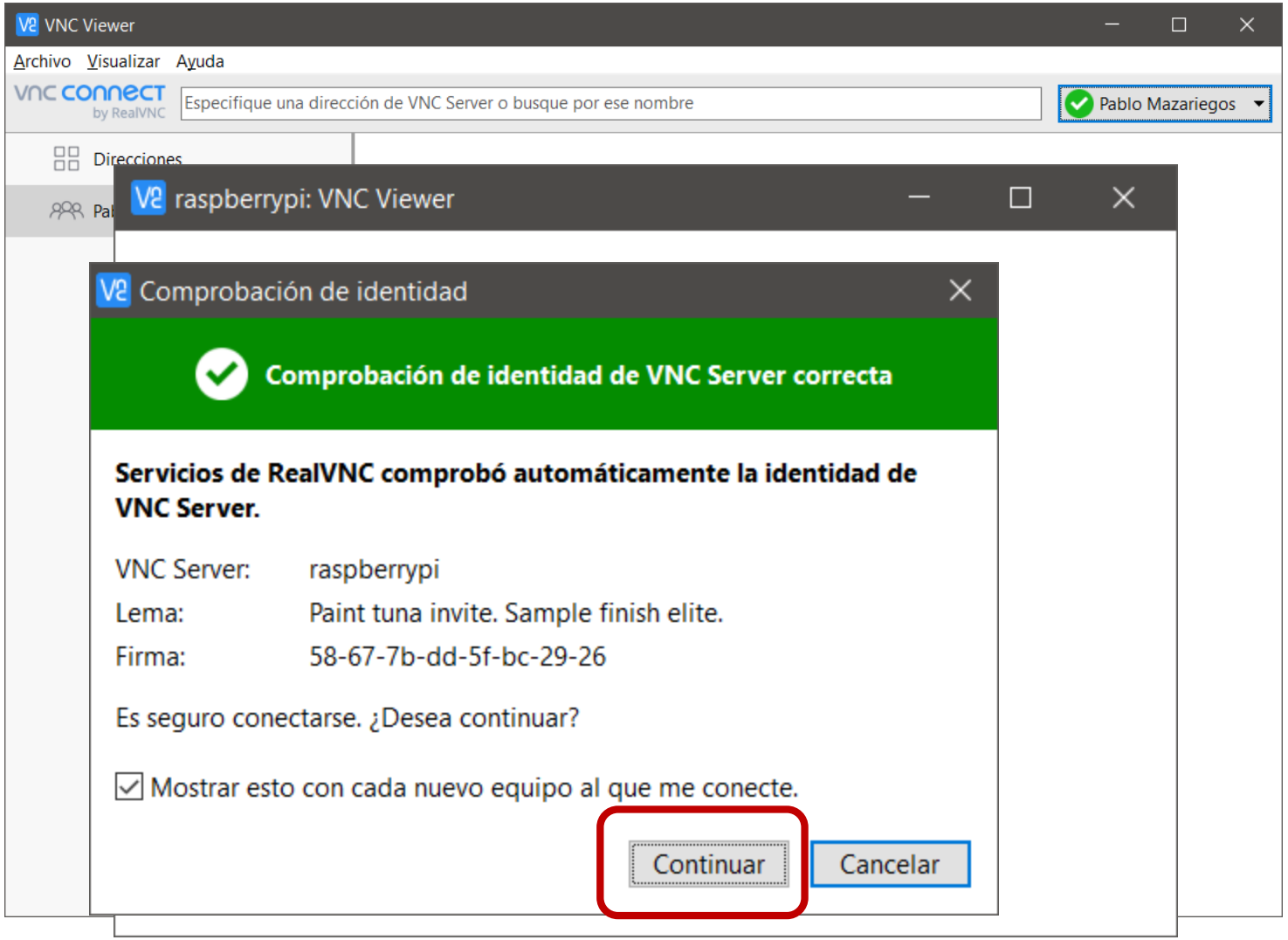
Una vez hayamos ingresado con nuestra cuenta en la raspberry, desde nuestro PC podremos acceder a la raspberry



Nos aparecerá la siguiente ventana

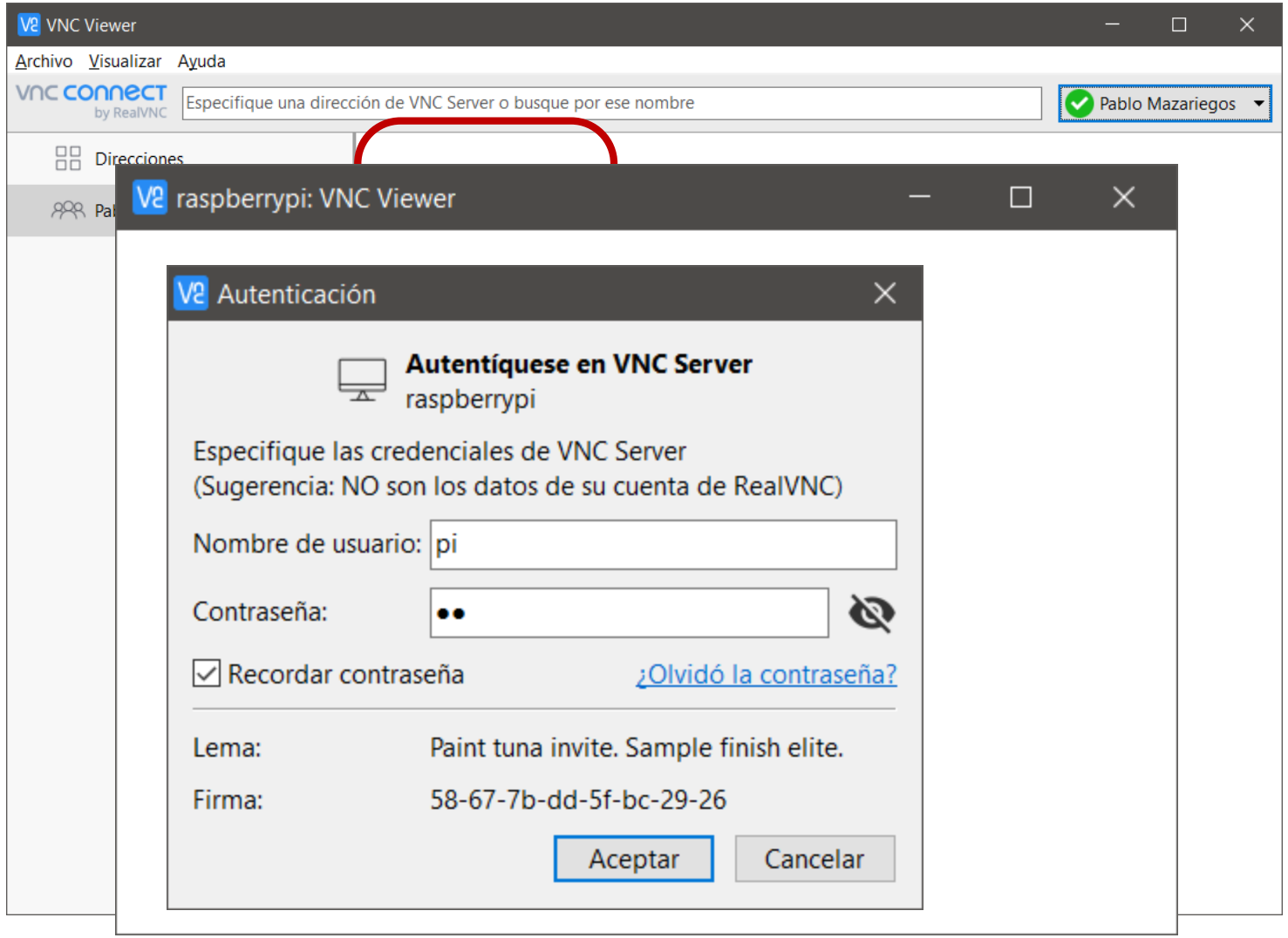


Seleccionamos continuar

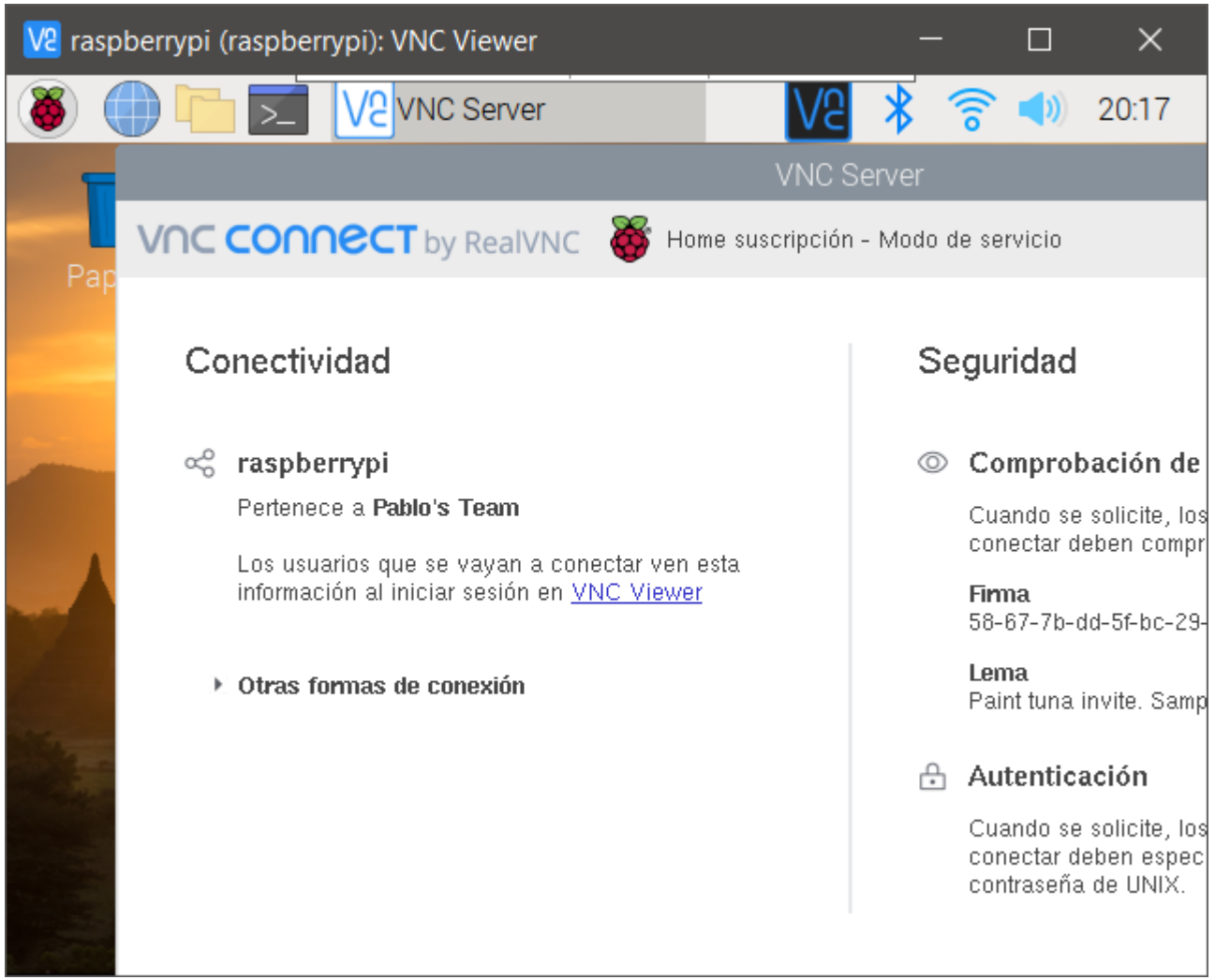


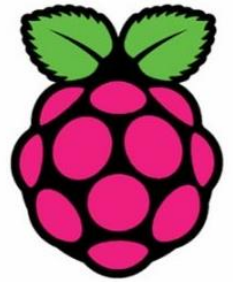
Nos aparecerá la siguiente ventana para autenticarnos

Utilizamos las credenciales que dejamos en la raspberry



Ya que ingresamos, podremos utilizar nuestra raspberry desde nuestro PC



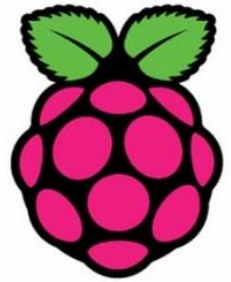


https://io.adafruit.com/

https://learn.adafruit.com/welcome-to-adafruit-io?view=all

https://github.com/adafruit/Adafruit_IO_Python

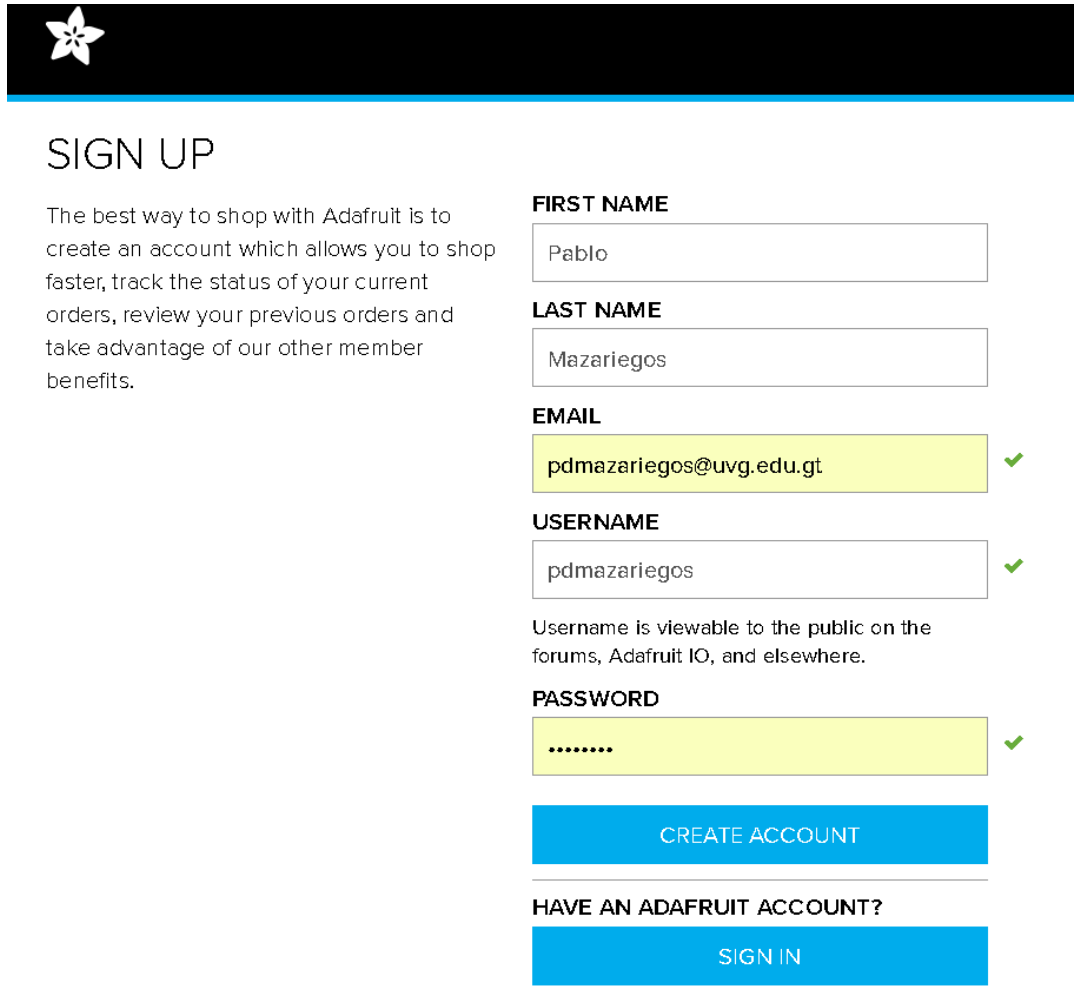
https://learn.adafruit.com/series/adafruit-io-basics



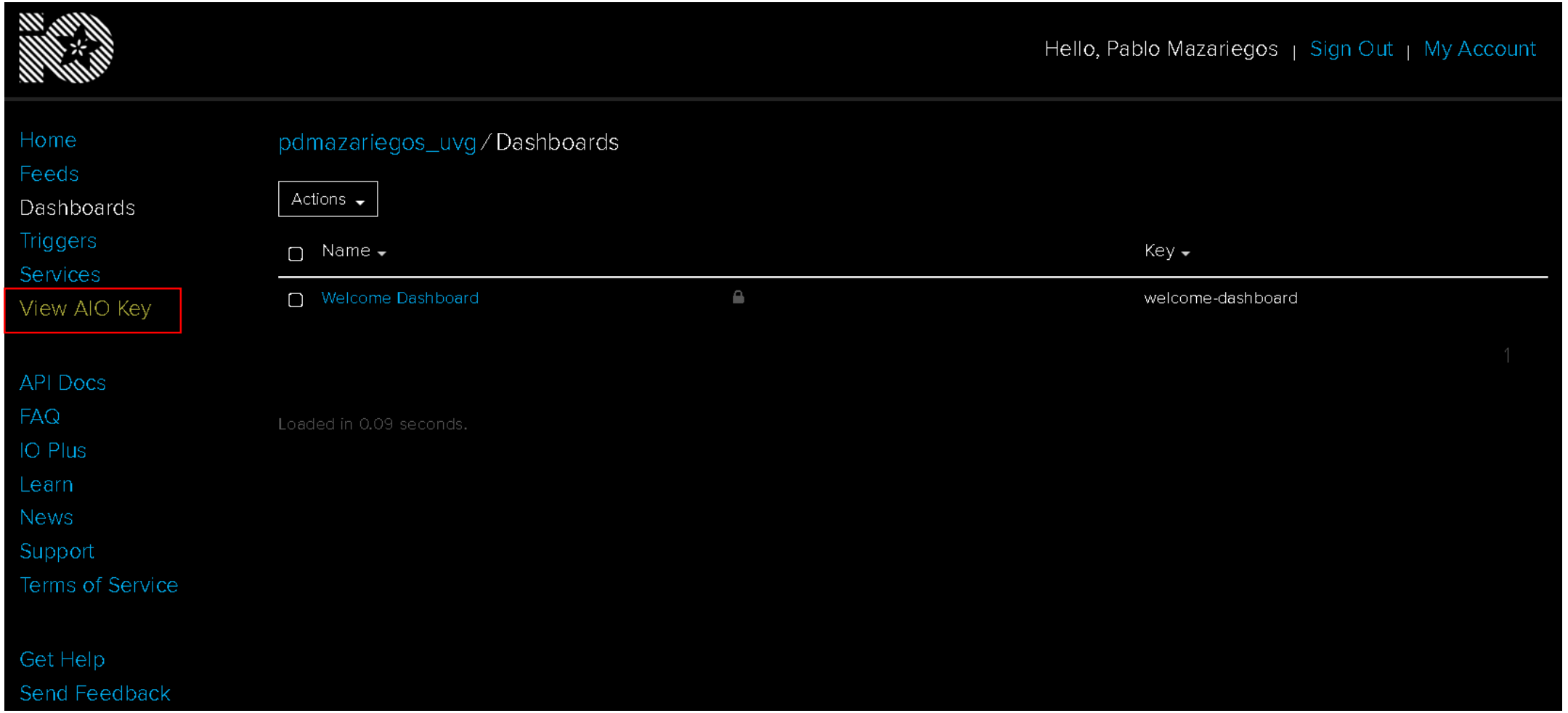
Instalamos la librería de adafruit-io

```
pi@raspberrypi:~ $ sudo pip install adafruit-io
Collecting adafruit-io
  Downloading https://files.pythonhosted.org/packages/a6/1a/864394af8ed308cf3b40a01ca739dc0eb8eefddd63119ac50a32f6225257/adafruit-io-2.1.tar.gz (43kB)
    100% |████████████████████████████████| 51kB 87kB/s
Requirement already satisfied: paho-mqtt in /usr/local/lib/python2.7/dist-packages (from adafruit-io)
Requirement already satisfied: requests in /usr/lib/python2.7/dist-packages (from adafruit-io)
Building wheels for collected packages: adafruit-io
  Running setup.py bdist_wheel for adafruit-io ... done
  Stored in directory: /root/.cache/pip/wheels/3a/f7/e3/4d4e4361324779a756fd6447bcf076ec2944b785444ad6691d
Successfully built adafruit-io
Installing collected packages: adafruit-io
Successfully installed adafruit-io-2.1
```

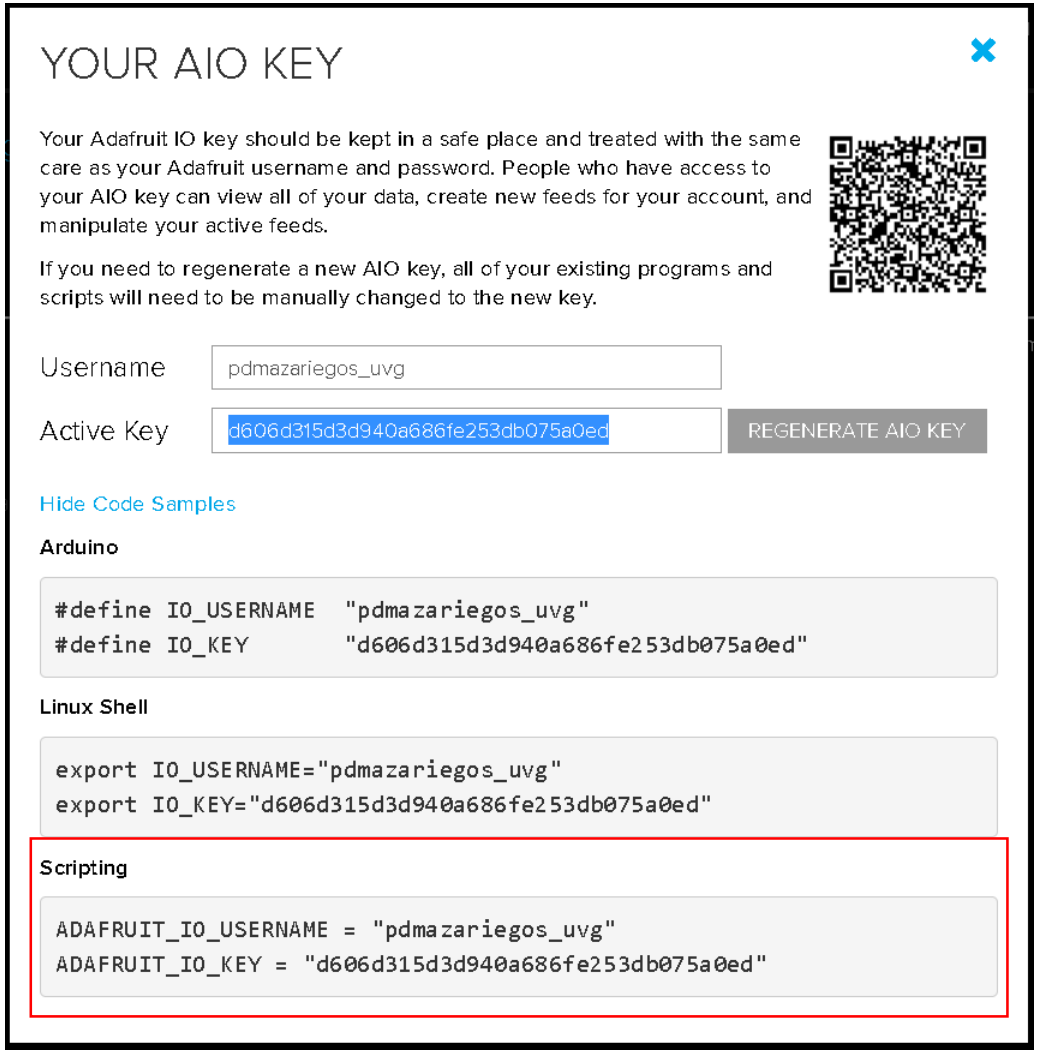
# Creamos una cuenta en Adafruit.io



Obtenemos nuestro AIO Key para poder acceder a nuestra cuenta



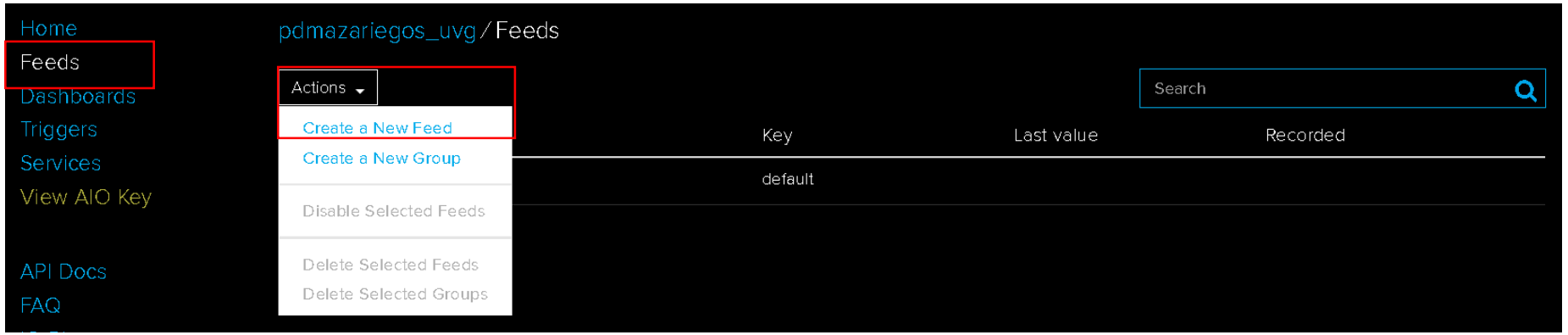
# Obtenemos nuestro AIO Key para poder acceder a nuestra cuenta



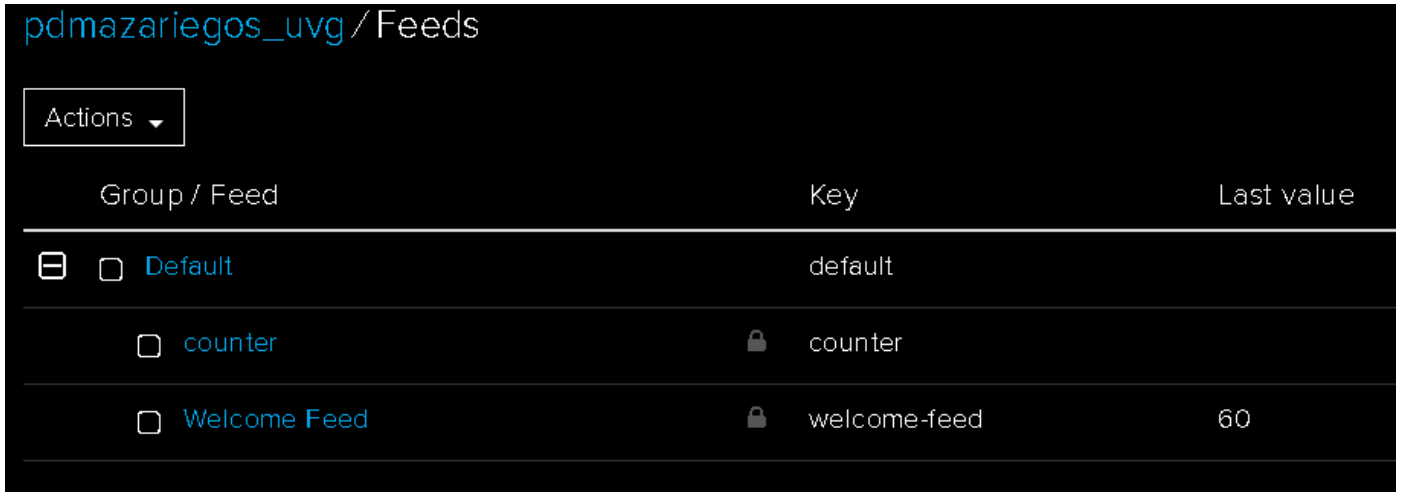

Reemplazamos con los valores que nos aparecen en el código de Python

```
# Set to your Adafruit IO key.
# Remember, your key is a secret,
# so make sure not to publish it when you publish this code!
ADAFRUIT_IO_KEY = 'YOUR_AIO_KEY'

# Set to your Adafruit IO username.
# (go to https://accounts.adafruit.com to find your username)
ADAFRUIT_IO_USERNAME = 'YOUR_AIO_USERNAME'
```

# Creamos un nuevo Feed (un canal)

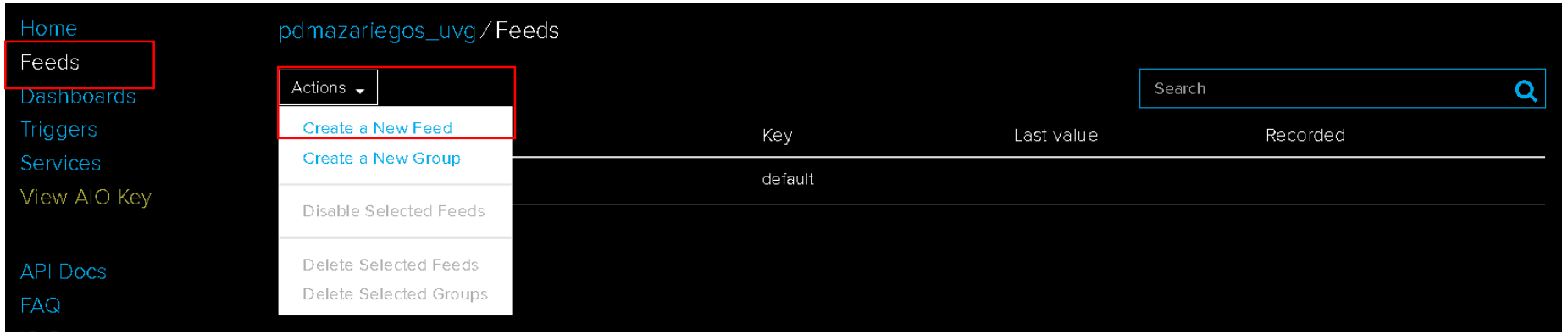




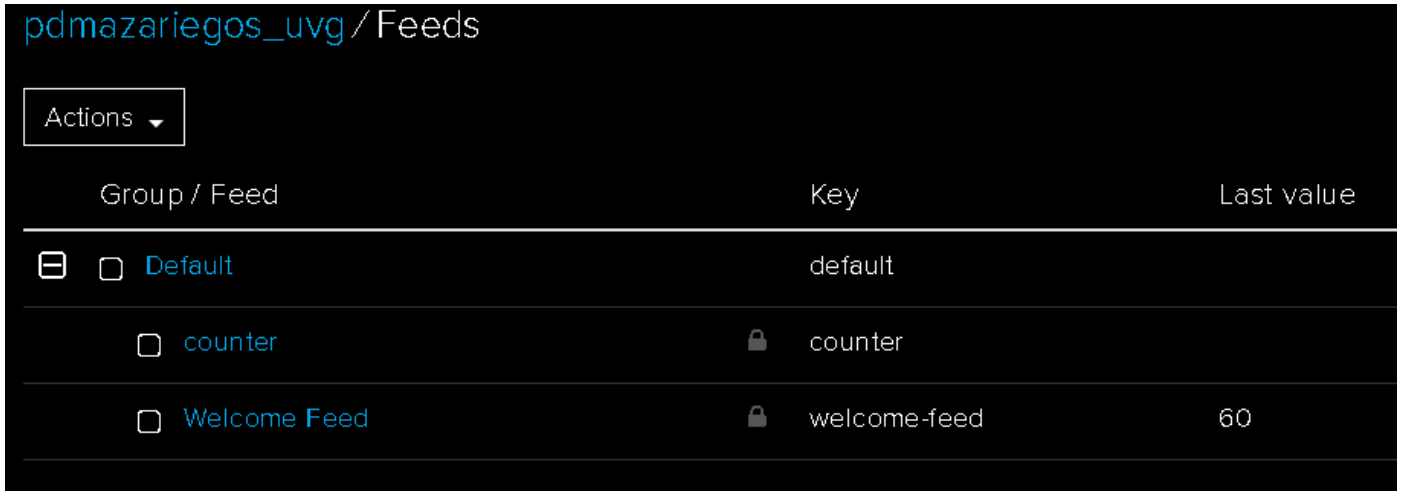



```
# Set to the ID of the feed to subscribe to for updates.
FEED_ID = 'counter'
```

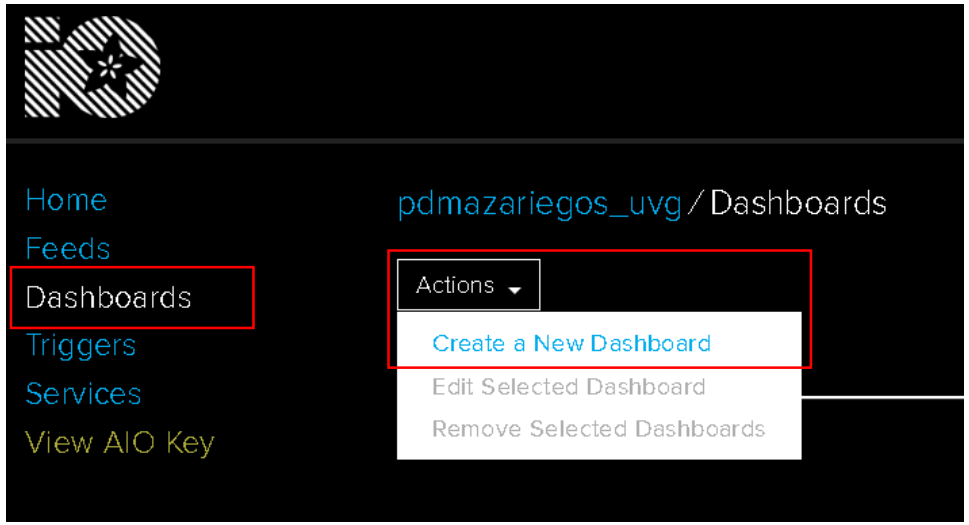
# Creamos un nuevo Feed (un canal)

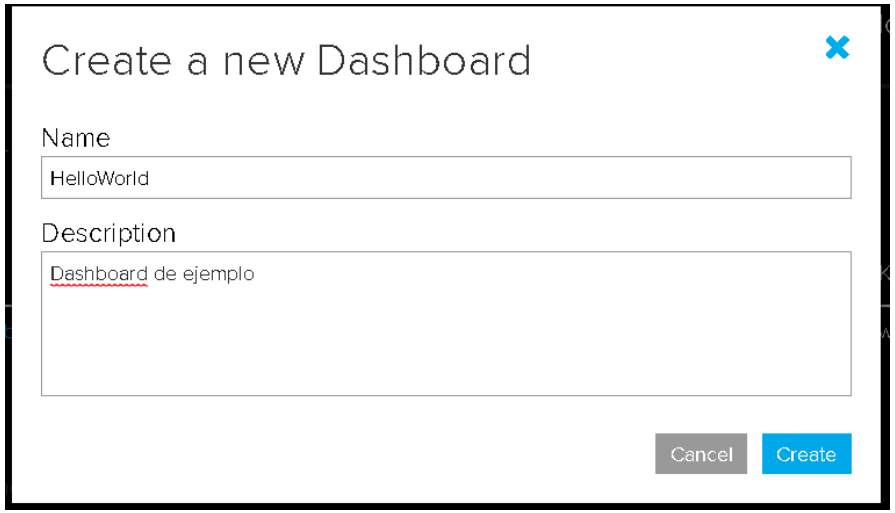


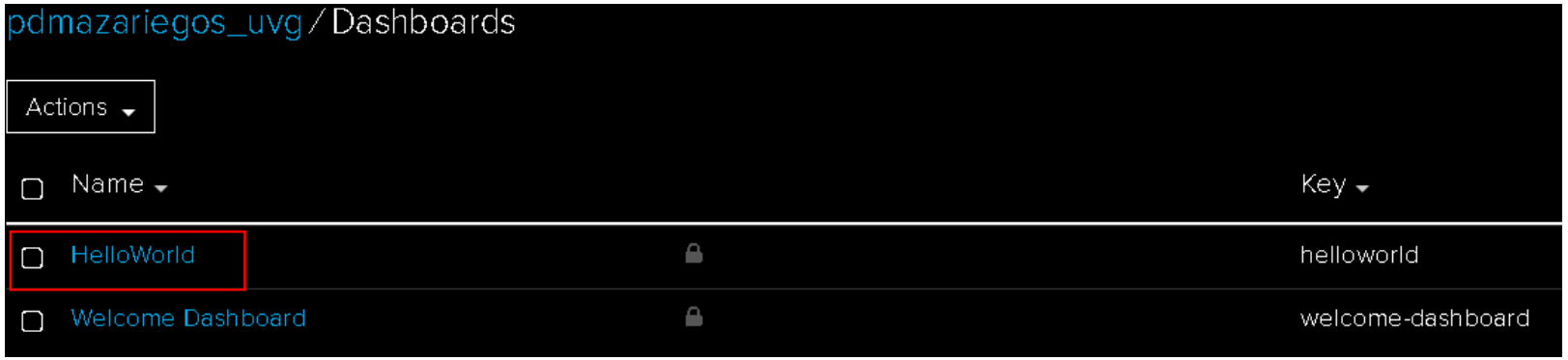




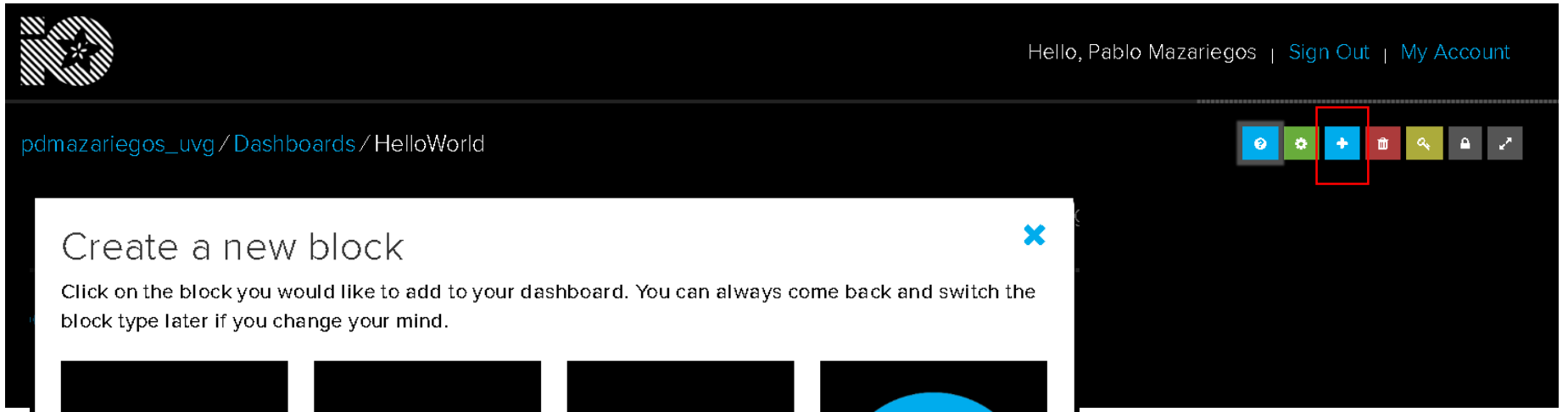
# Creamos un nuevo dashboard

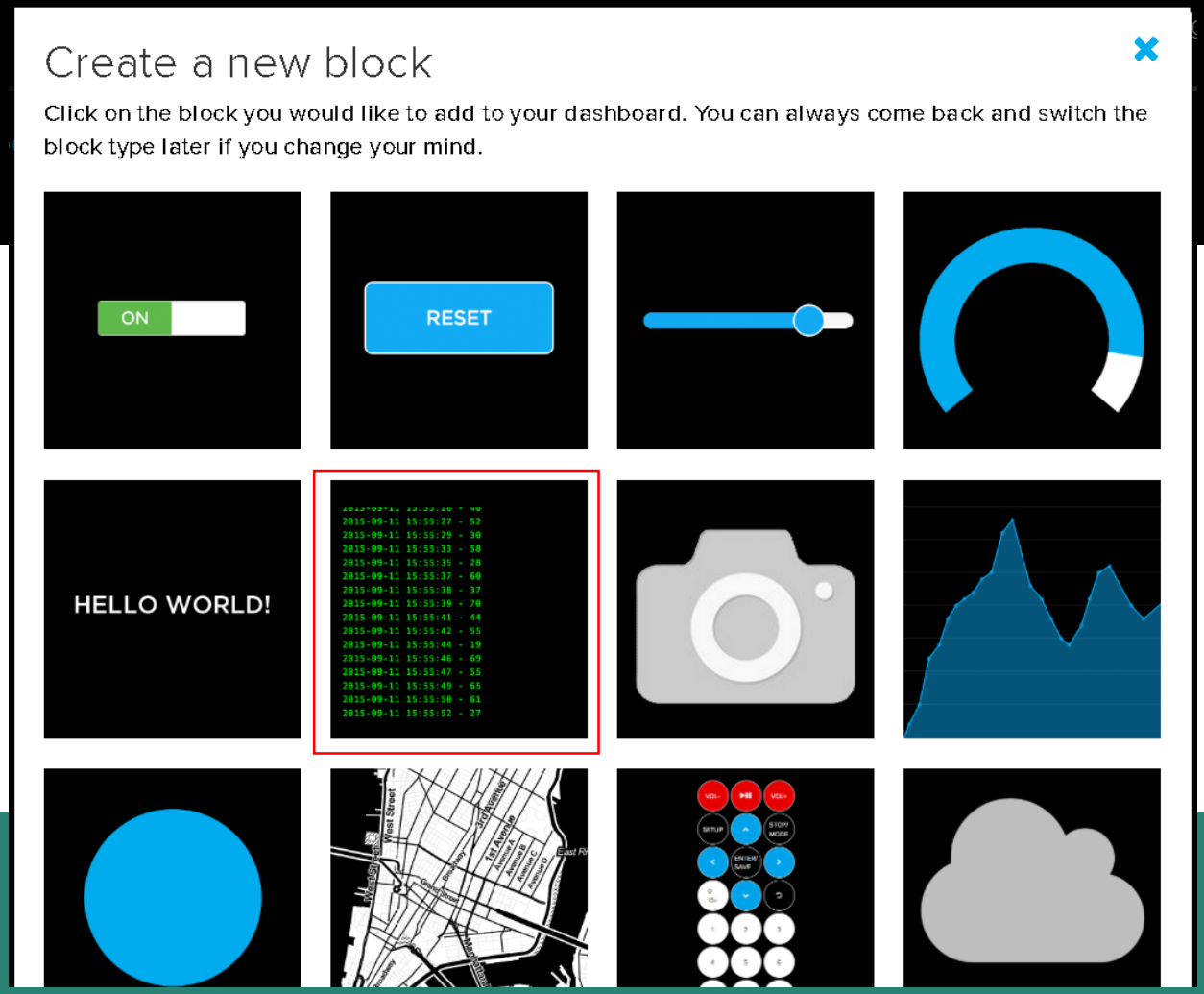


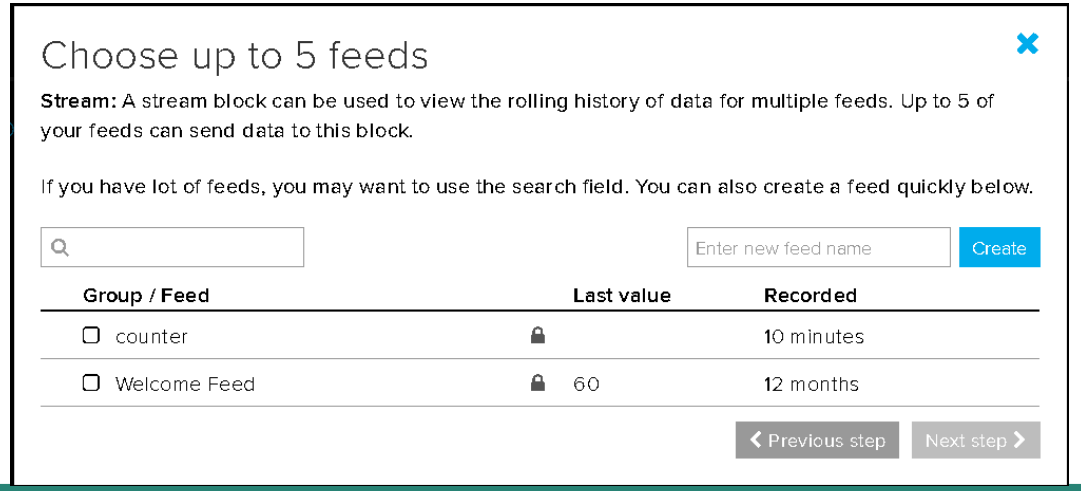



# Creamos un nuevo dashboard

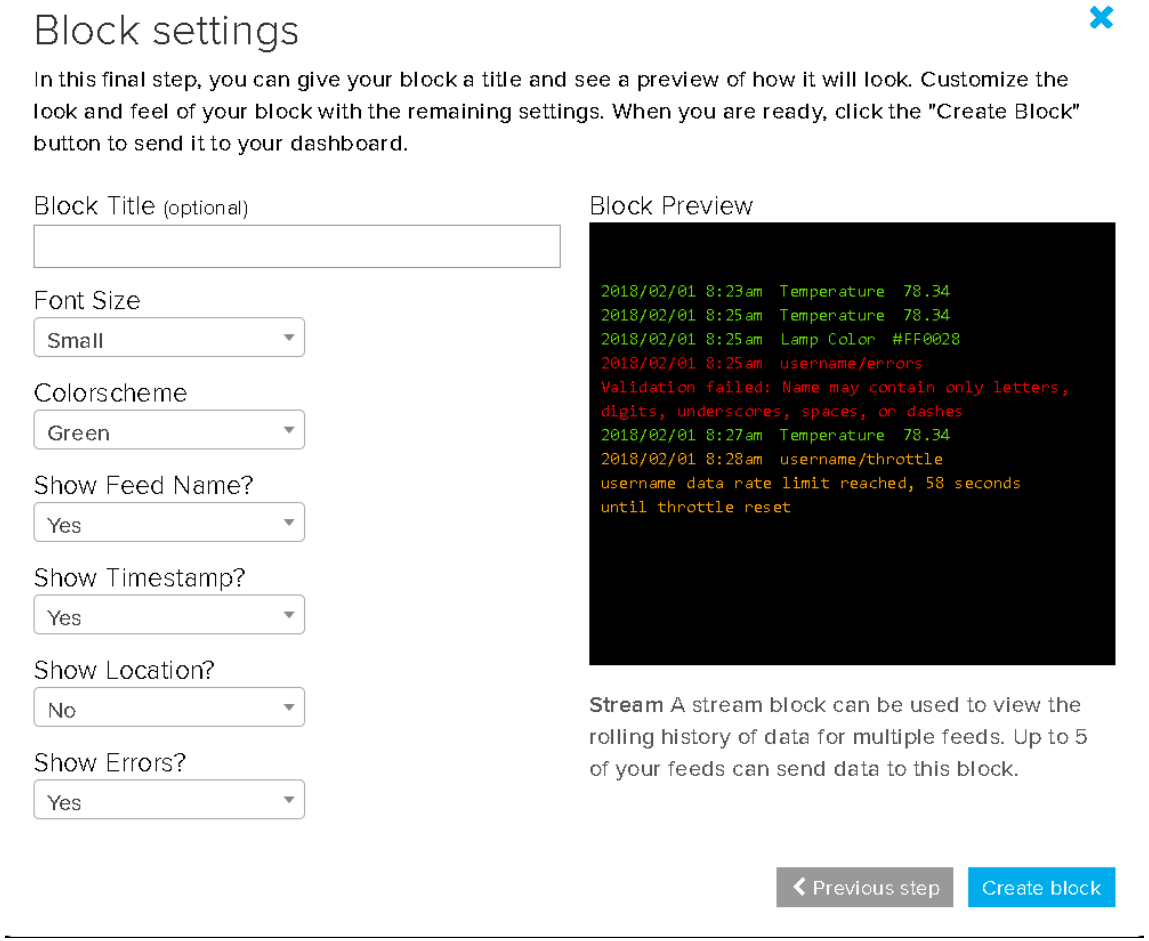


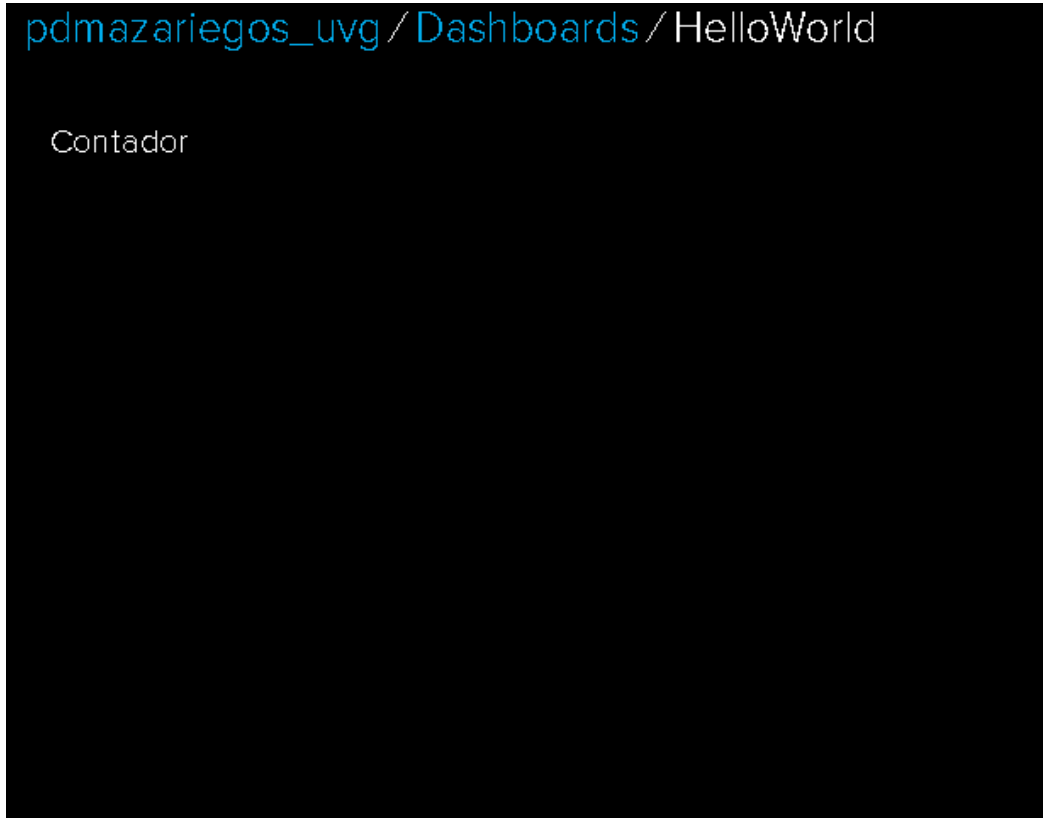




## Seleccionamos el Feed que creamos anteriormente

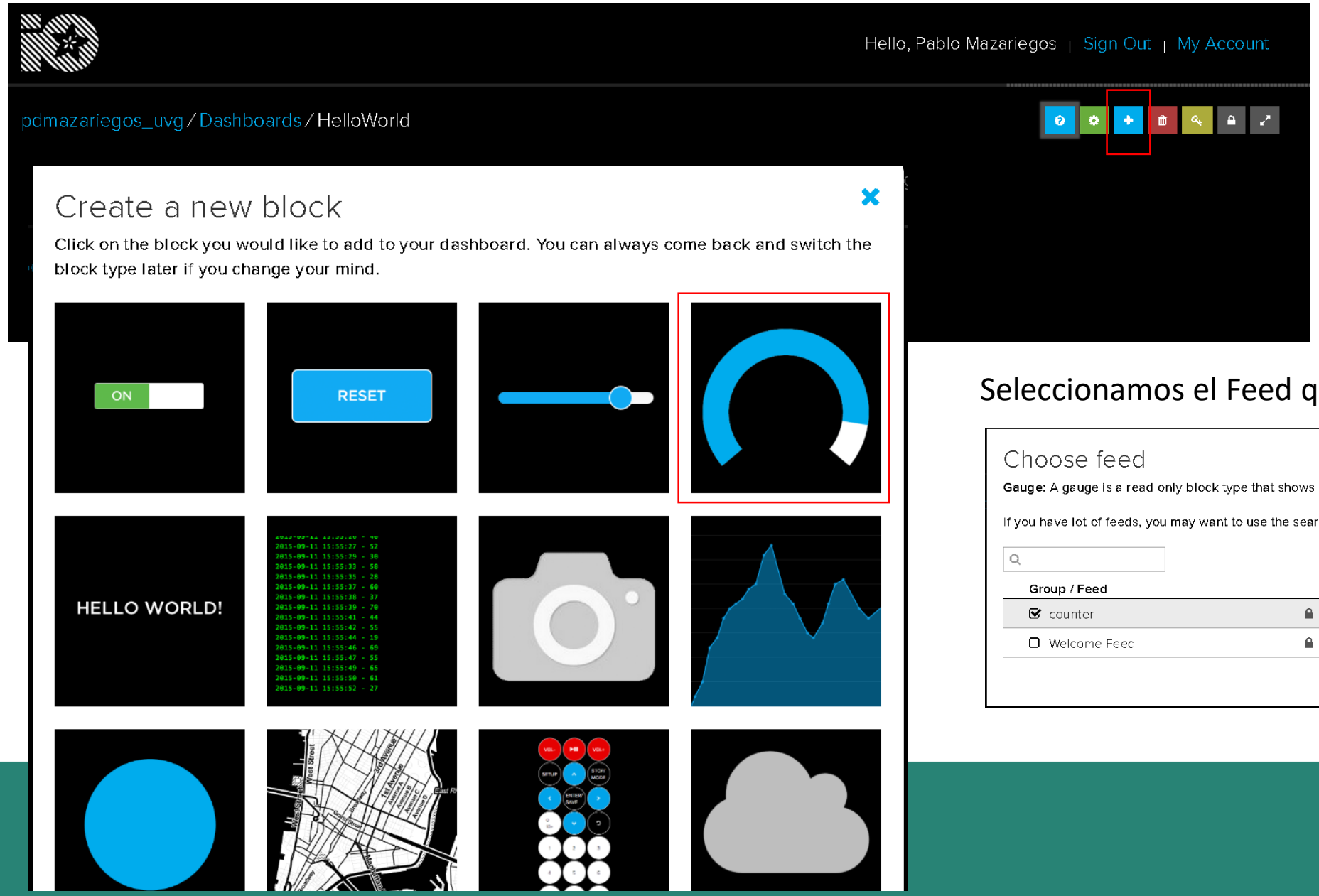


# Llenamos la configuración del bloque



# Añadimos otro bloque




## Seleccionamos el Feed que creamos anteriormente

# Llenamos la configuración del bloque

## Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)
Contador

Gauge Min Value
0

Gauge Max Value
100

Gauge Width
25px

Gauge Label
Contador

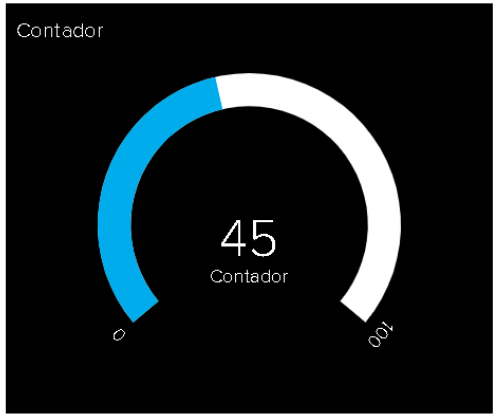
Low Warning Value

Optional. If no low warning value is given, the gauge will only change color when the value is out of bounds.

High Warning Value

Optional. If no high warning value is given, the gauge will only change color when the value is out of bounds.

Block Preview

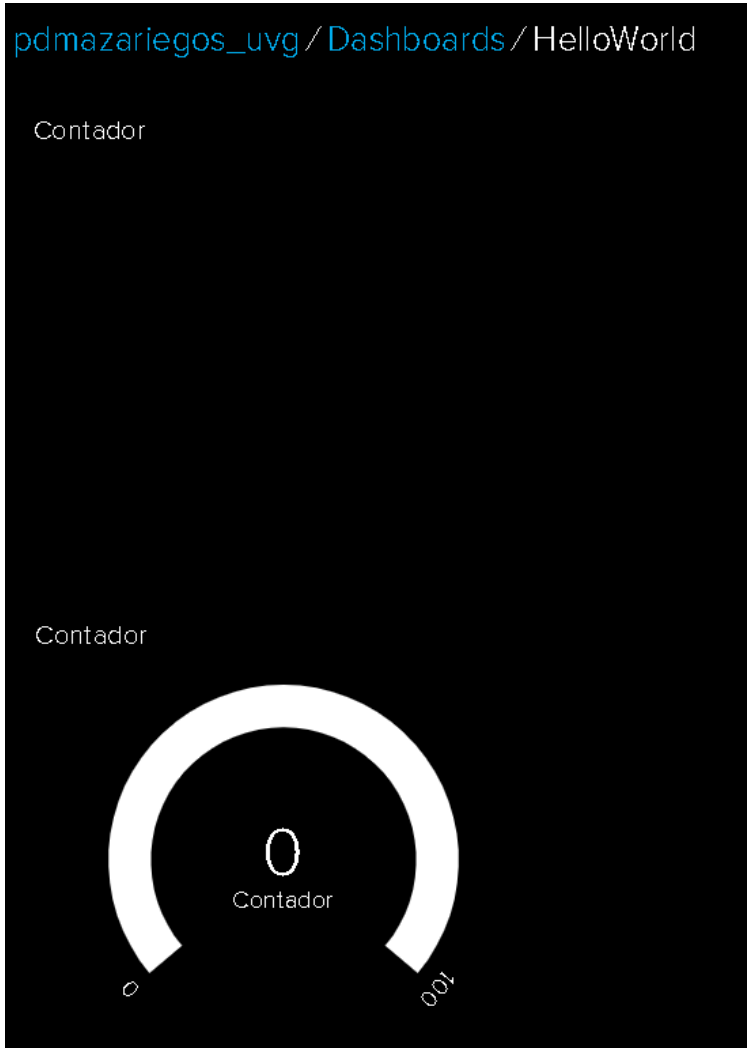



**Gauge** A gauge is a read only block type that shows a fixed range of values.

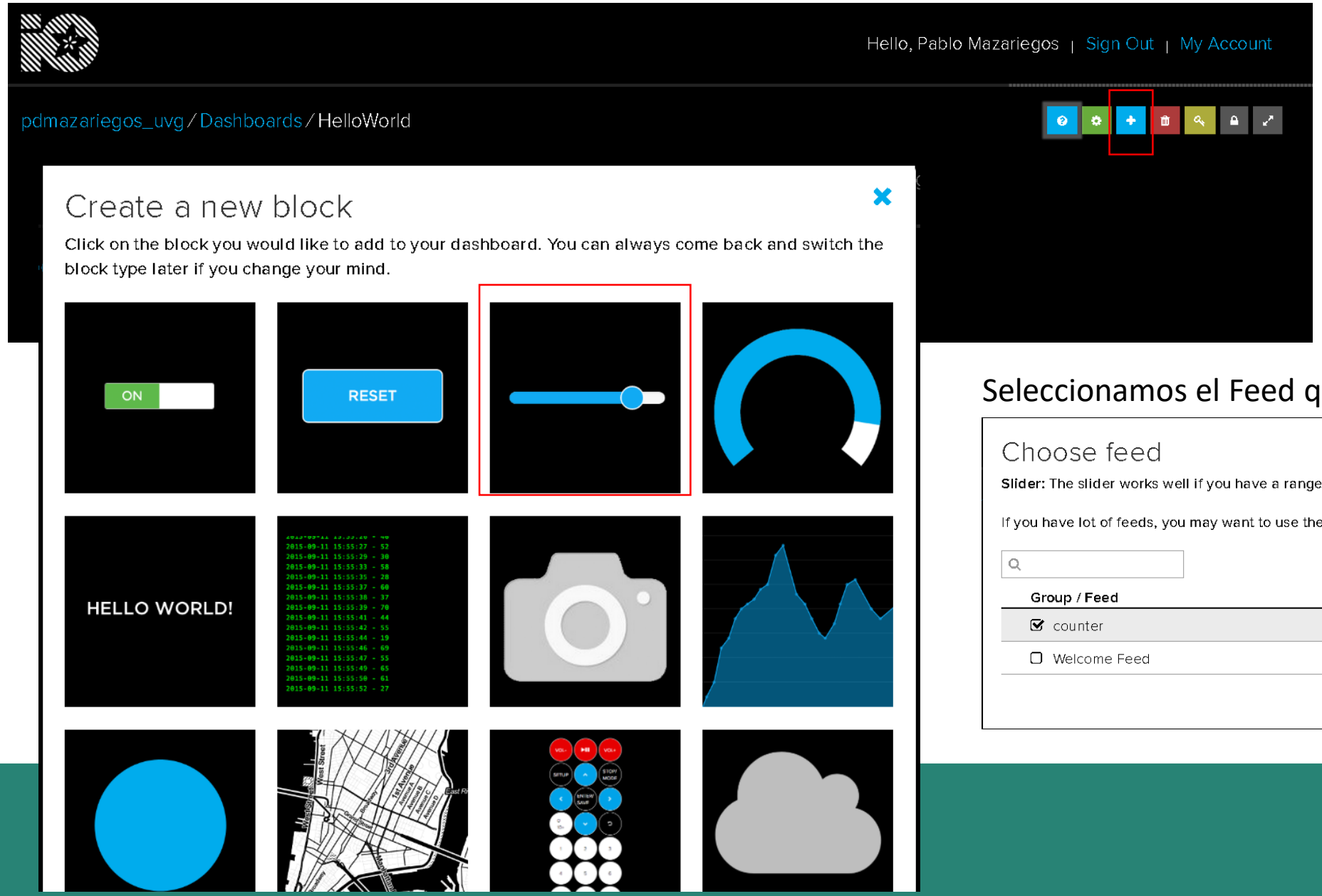
Test Value
45

Published Value

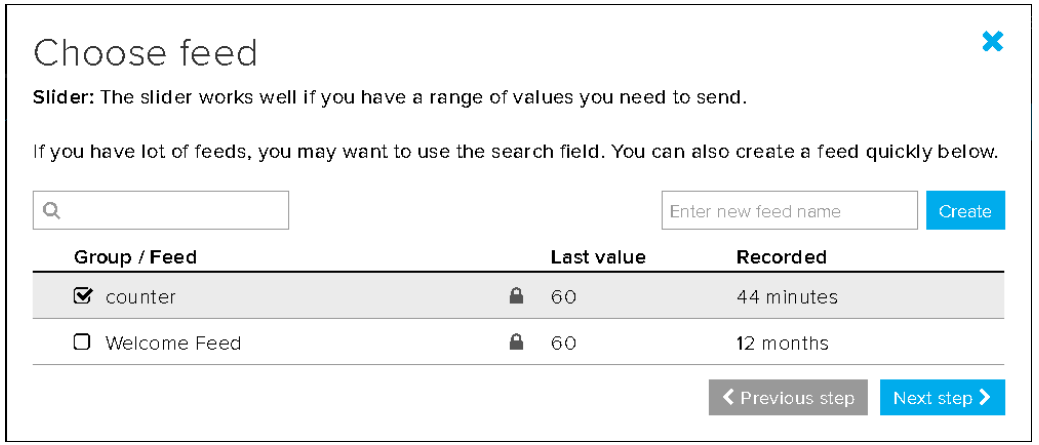
`0 bytes`



# Añadimos otro bloque




## Seleccionamos el Feed que creamos anteriormente



# Llenamos la configuración del bloque

## Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)

Dato a enviar

Slider Min Value

0

Slider Max Value

100

Slider Step Size

10

Slider Label

Value

Block Preview

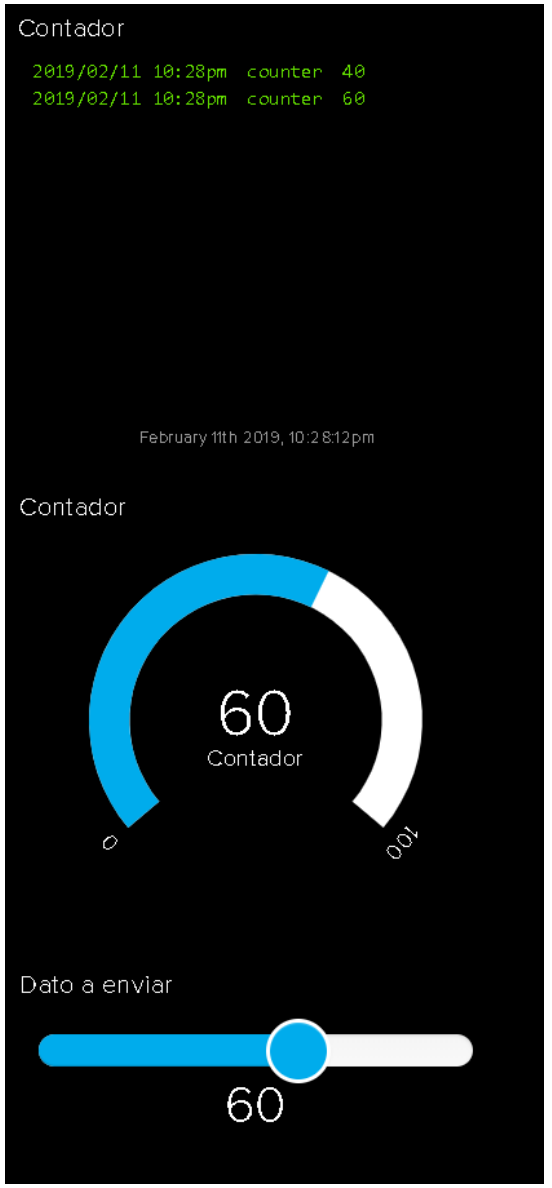

**Slider** The slider works well if you have a range of values you need to send.

Test Value

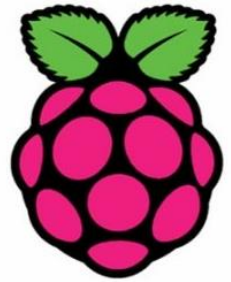
45

Published Value

0 bytes



Abrimos ejemplo ***suscribe.py*** y llenamos los parámetros



```
"""
'subscribe.py'
==========================
Subscribes to an Adafruit IO Feed
Author(s): Brent Rubell, Todd Treece for Adafruit Industries
"""
# Import standard python modules.
import sys

# This example uses the MQTTClient instead of the REST client
from Adafruit_IO import MQTTClient

# Set to your Adafruit IO key.
# Remember, your key is a secret,
# so make sure not to publish it when you publish this code!
ADAFRUIT_IO_KEY = 'd606d315d3d940a686fe253db075a0ed'

# Set to your Adafruit IO username.
# (go to https://accounts.adafruit.com to find your username)
ADAFRUIT_IO_USERNAME = 'pdmazariegos_uvg'

# Set to the ID of the feed to subscribe to for updates.
FEED_ID = 'counter'

# Define callback functions which will be called when certain events happen.
def connected(client):
    """Connected function will be called when the client is connected to
    Adafruit IO.This is a good place to subscribe to feed changes.  The client
    parameter passed to this function is the Adafruit IO MQTT client so you
    can make calls against it easily.
    """
    # Subscribe to changes on a feed named Counter.
    print('Subscribing to Feed {0}'.format(FEED_ID))
    client.subscribe(FEED_ID)
    print('Waiting for feed data...')

def disconnected(client):
    """Disconnected function will be called when the client disconnects."""
    sys.exit(1)

def message(client, feed_id, payload):
    """Message function will be called when a subscribed feed has a new value.
    The feed_id parameter identifies the feed, and the payload parameter has
    the new value.
    """
    print('Feed {0} received new value: {1}'.format(feed_id, payload))
```

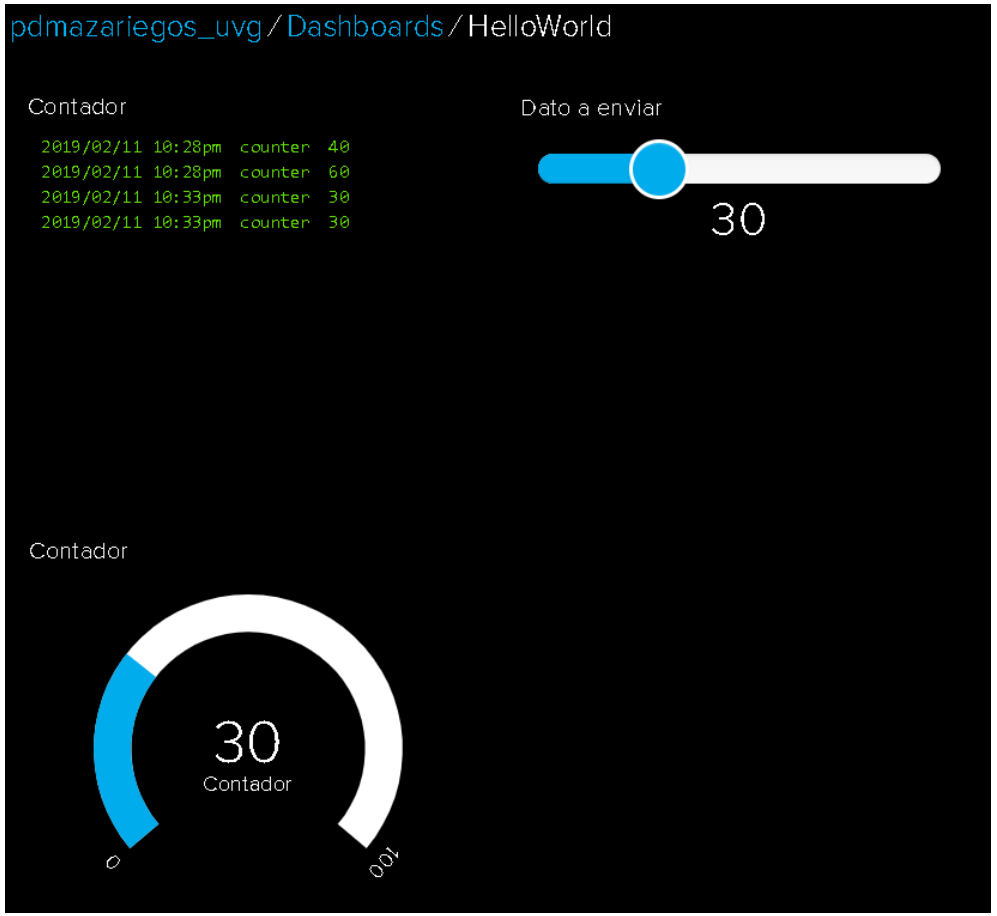
```
# Create an MQTT client instance.
client = MQTTClient(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)

# Setup the callback functions defined above.
client.on_connect = connected
client.on_disconnect = disconnected
client.on_message = message

# Connect to the Adafruit IO server.
client.connect()

# The first option is to run a thread in the background so you can continue
# doing things in your program.
client.loop_blocking()
```

Abrimos el editor de texto ***nano*** y pegamos nuestro código y lo guardamos como ***subscribe.py***

```
pi@raspberrypi:~ $ sudo nano subscribe.py
```

Para correr nuestro programa lo realizamos de la siguiente forma:

```
pi@raspberrypi:~ $ sudo python subscribe.py
```




```
pi@raspberrypi:~ $ python subscribe.py
Connected to Adafruit IO!
Subscribing to Feed counter
Waiting for feed data...
Feed counter received new value: 40
Feed counter received new value: 60
Feed counter received new value: 30
Feed counter received new value: 30
Feed counter received new value: 30
```

# Enlaces importantes

- Tutorial GPIO de Raspberry Pi de Adafruit

https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup?view=all

- Convertidor Lógico 3.3V a 5V

https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide

- Tutorial 1 SPI Raspberry Pi API

http://tightdev.net/SpiDev_Doc.pdf

- Tutorial 2 de SPI de la Raspberry Pi

https://www.takaitra.com/spi-device-raspberry-pi/

- Tutorial 3 de SPI de  la Raspberry Pi

https://raspberrypi-aa.github.io/session3/spi.html

- Tutorial SPI e I2C de Raspberry en Sparkfun en Lenguaje C

https://learn.sparkfun.com/tutorials/raspberry-pi-spi-and-i2c-tutorial