

Requisitos:

Qué necesitamos?

- **RAK2287 Pi HAT WisLink LPWAN Concentrator**
- **RAK2287 Pi HAT**
- **Raspberry Pi 3B+ o Raspberry Pi 4**
- Tarjeta SD de 16GB + Lector de tarjetas
- Adaptador USB o USB C (Pi 4) de 5V de por lo menos 2 A
- Un ordenador Windows/Mac OS/Linux
- Raspberry Pi Imager
- El ultimo firmware de RAK2287 Pi HAT aunque en este manual no lo necesitemos

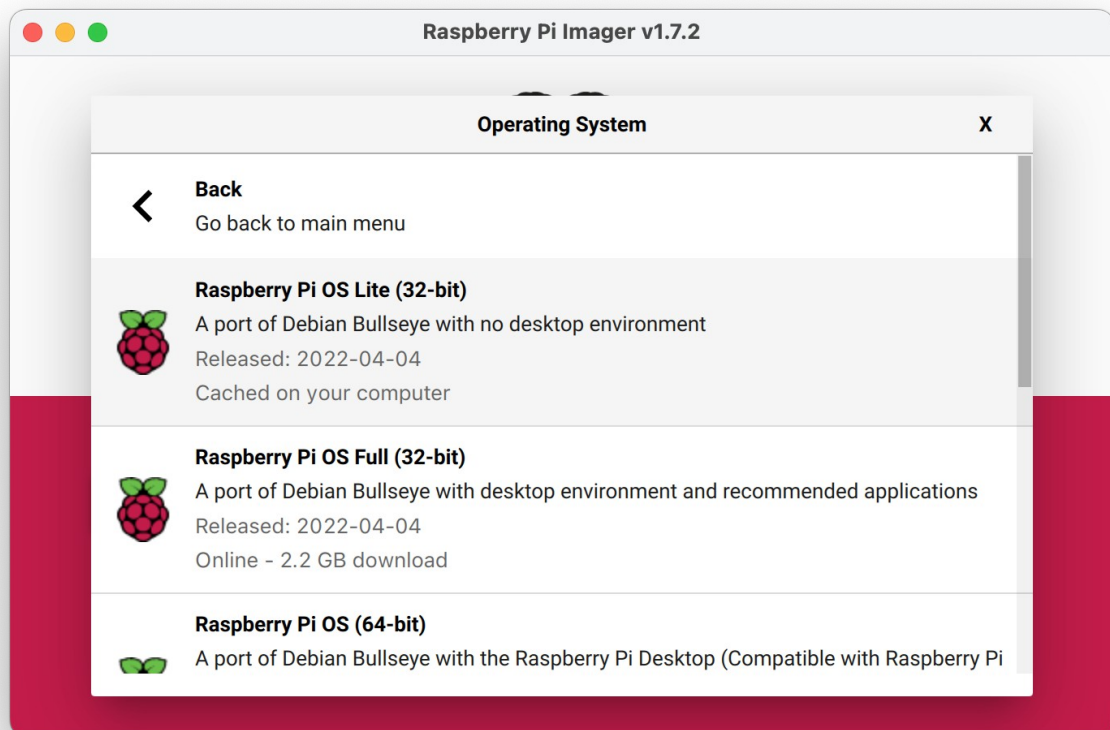
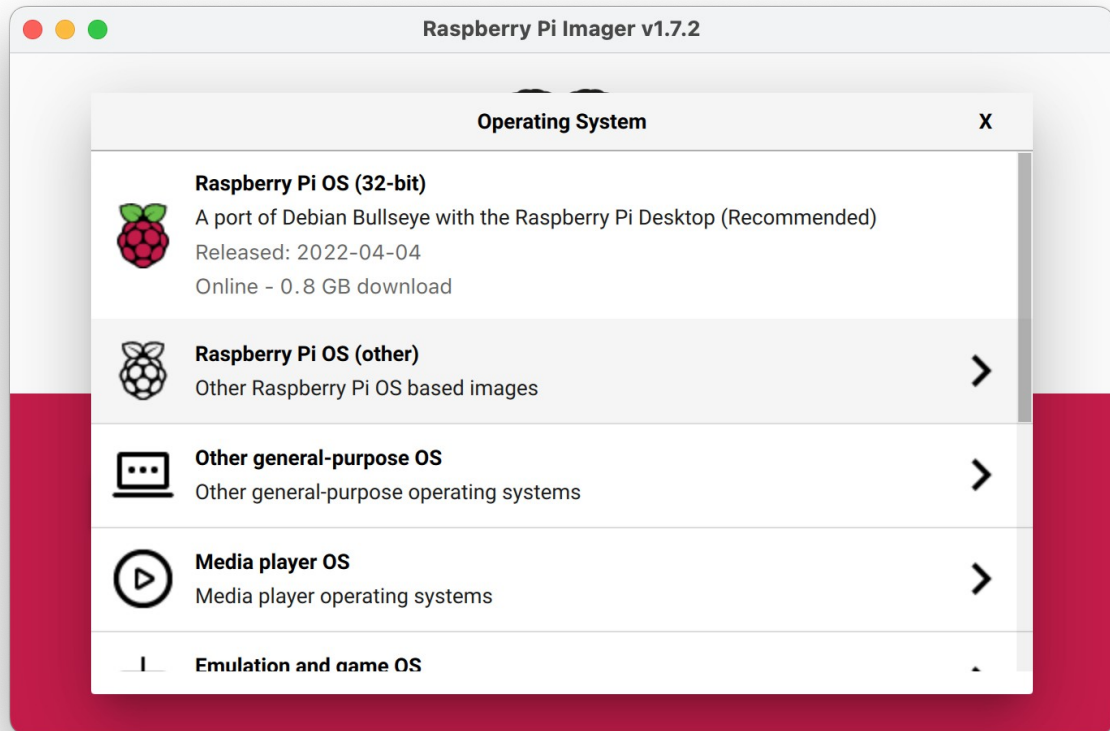
Instalacion en la raspberry:

Descargar [**Raspberry Pi Imager**](#)

Abrir **Raspberry Pi Imager** que se puede descargar de la pagina oficial de Raspberry.

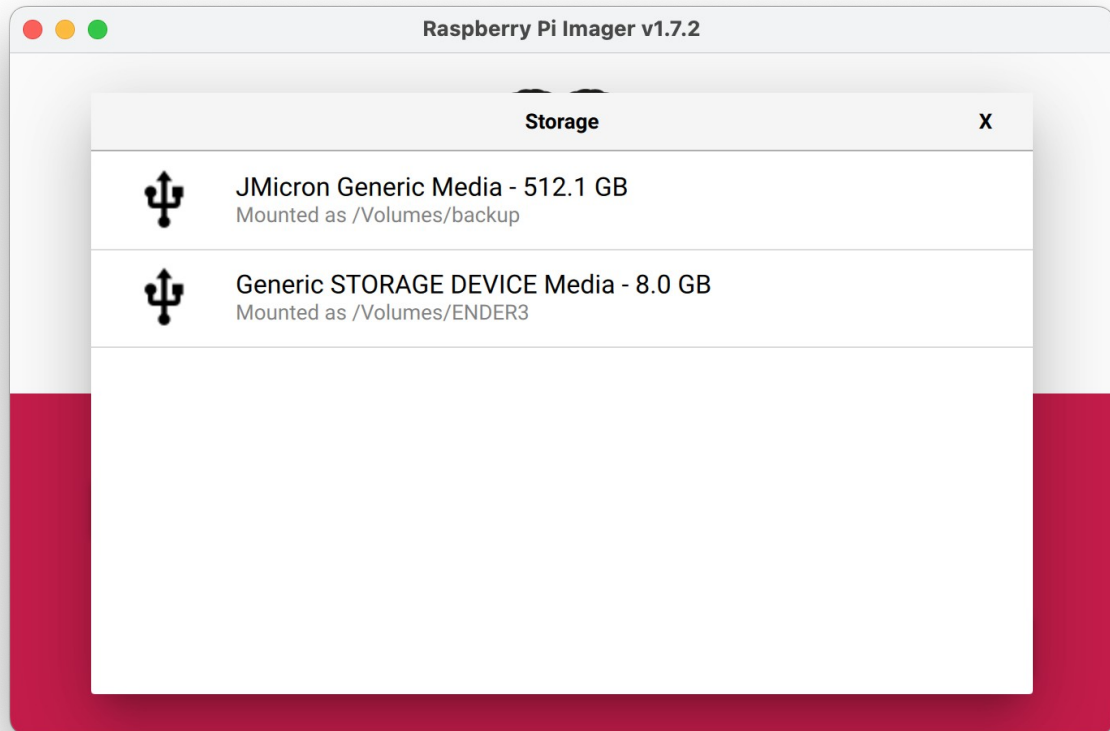


Elegimos OS Raspberry Pi OS (32-bit)

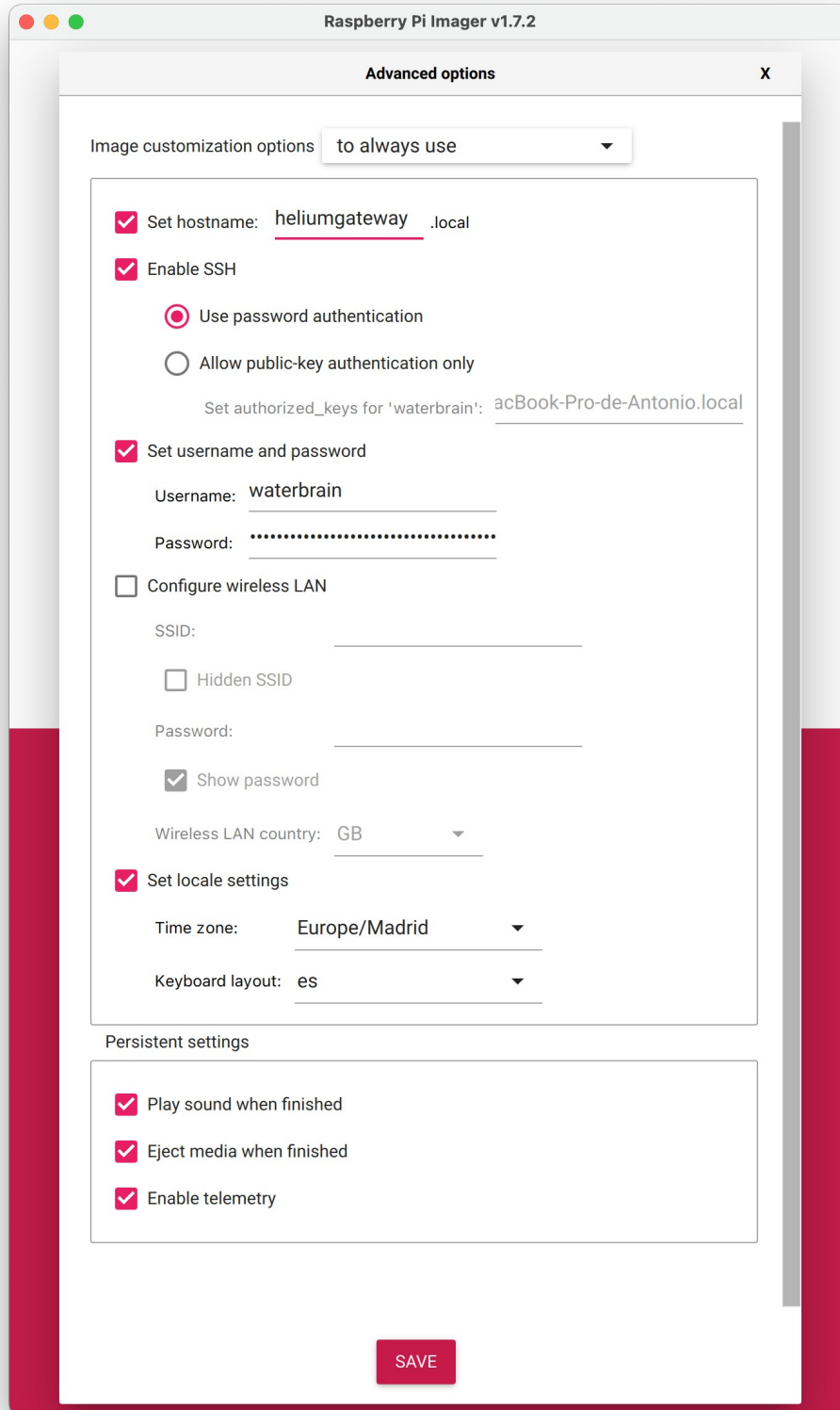


A continuación elegimos la tarjeta:

CHOOSE STORAGE



Aquí aparece la ruedecita de configuracion la pulsamos y  por razones legales y de seguridad para evitar ataques con el usuario pi yo lo dejo asi:



Raspberry Pi Imager v1.7.2

Advanced options X

Image customization options to always use ▼

☒ Set hostname: heliumgateway.local

☒ Enable SSH

☒ Use password authentication

☐ Allow public-key authentication only

Set authorized_keys for 'waterbrain': acBook-Pro-de-Antonio.local

☒ Set username and password

Username: waterbrain

Password:

☐ Configure wireless LAN

SSID: _____

☐ Hidden SSID

Password: _____

☒ Show password

Wireless LAN country: GB ▼

☒ Set locale settings

Time zone: Europe/Madrid ▼

Keyboard layout: es ▼

Persistent settings

☒ Play sound when finished

☒ Eject media when finished

☒ Enable telemetry

SAVE

WRITE

y ya le podemos dar a write. Esperamos a que acabe. Introducimos la imagen en la Raspberry y arracamos.

Buscamos en que IP está, yo empleo [Fing](#) que en versión libre esta para cualquier ordenador incluido iPhone o Android.

Dispositivos	Red	Seguridad	Internet	Gente	
UPnP-IPTV	Raspberry Pi • Raspberry Pi	Raspberry	Raspbian	192.168.1.18	lun 18 0:14
homeassistant	Raspberry Pi • Raspberry Pi	Raspberry	Windows	192.168.1.19	
server20	Raspberry Pi • Raspberry Pi	Raspberry	Raspbian	192.168.1.20	mar 19 18:39
MAINMOTIONEYE	Raspberry Pi • Raspberry Pi	Raspberry	Raspbian	192.168.1.21	6 ene 19:25
linux-3	Amazon			192.168.1.33	mar 12 12:20
wlan0	Tuya		Smart Device	192.168.1.34	mar 19 18:40
Generic	Huawei			192.168.1.36	mar 19 11:05
ESP_57F19A	Sonoff • Basic	Smart Device	Android	192.168.1.37	mar 19 11:05
wlan0	Tuya		Smart Device	192.168.1.38	mar 19 21:06
UPnP-IPTV	Raspberry Pi • Raspberry Pi	Raspberry	Raspbian	192.168.1.41	9:27
none	Amazon • Echo Dot	Voice Control	Android	192.168.1.43	mar 12 12:44
MacBook-Pro-de-Antonio-179	Apple • MacBook Pro 15" (Mid 2015)	Laptop	And	192.168.1.44	29 nov 2021 ...

veo que a las 9:27 ha aparecido un dispositivo nuevo con la ip 192.168.1.41 por lo que se en que direccion esta la Rasberry.

Abrimos una pantalla de terminal:

- ssh [waterbrain@192.168.1.41](https://192.168.1.41)

Yo personalmente prefiero ssh, aunque se pude utilizar cualquier herramienta como [PuTTY](https://www.putty.org/). Por ser la primera vez nos pide que añadamos la clave de la direccion de la Raspberry a su fichero de hosts conocidos, le decimos que si y voila:

```

-- waterbrain@rak-gateway: ~ -- ssh waterbrain@192.168.1.41 -- 178x60
$ ssh waterbrain@192.168.1.41
The authenticity of host '192.168.1.41 (192.168.1.41)' can't be established.
ECDSA key fingerprint is SHA256:q25N/ynA2189TfR8e7uR9p0u8eT0r1214AA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.41' (ECDSA) to the list of known hosts.
waterbrain@192.168.1.41's password:
Linux rak-gateway 5.15.32-v7+ #1538 SMP Thu Mar 31 19:38:48 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

=====
IOT Made Easy
=====
Last login: Wed Apr 20 07:57:58 2022 from 192.168.1.44

Mi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

waterbrain@rak-gateway:~$

```

Si queremos que no nos pida la contraseña: salimos de la Raspberry y ejecutamos:

- ssh-copy-id [waterbrain@192.168.1.41](https://192.168.1.41)

en nuestra terminal. Previo a esto debemos tener una llave o clave única generada. Yo no lo hago pero por si acaso no lo hemos hecho utilizaremos:

- **ssh-keygen -b 4096 -t rsa**

Esto generará una llave pública. Para los que tengan más interés en este tema, solo tienen que googlear un poco.

```

~ -- waterbrain@rak-gateway: ~ -- ssh waterbrain@192.168.1.41
~ -- ssh waterbrain@192.168.1.41
~ -- ssh-copy-id waterbrain@192.168.1.41
/opt/local/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/opt/local/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
waterbrain@192.168.1.41's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'waterbrain@192.168.1.41'"
and check to make sure that only the key(s) you wanted were added.
~ -- ssh waterbrain@192.168.1.41
Linux rak-gateway 5.15.32-v7+ #1538 SMP Thu Mar 31 19:38:48 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

=====
                    IoT Made Easy
=====
Last login: Wed Apr 20 08:35:35 2022 from 192.168.1.44

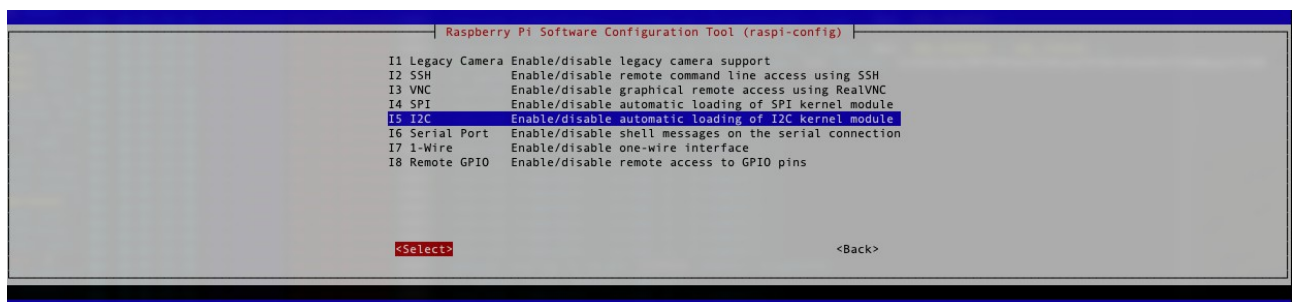
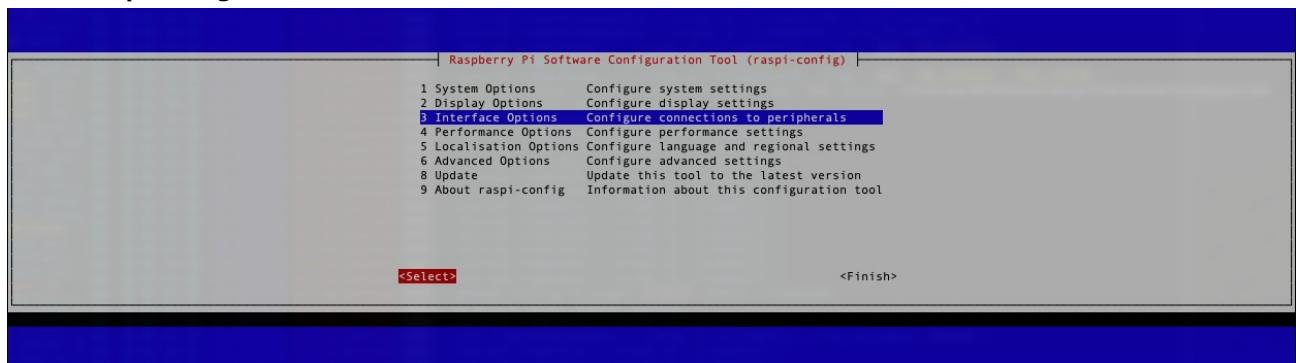
Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

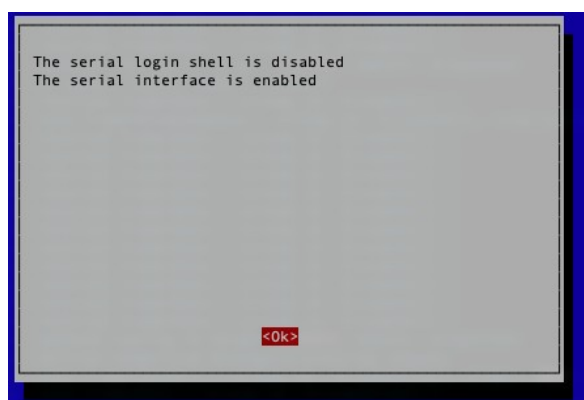
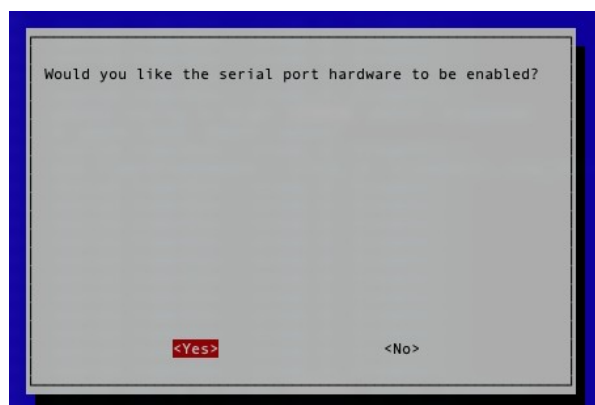
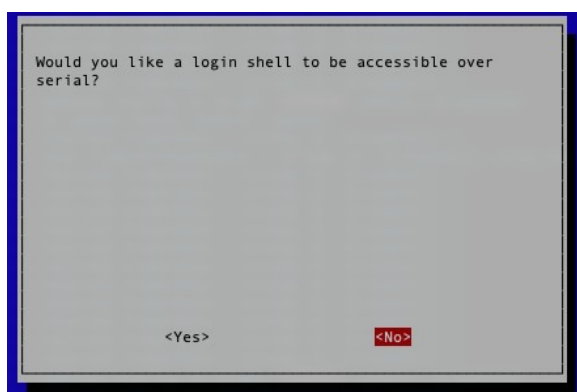
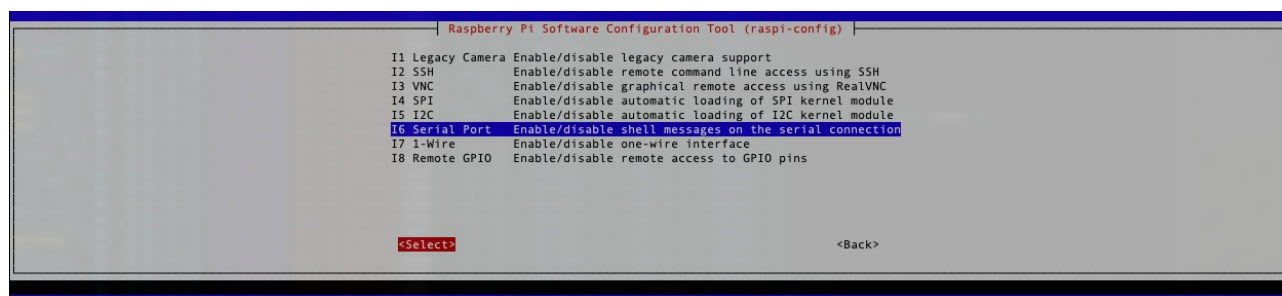
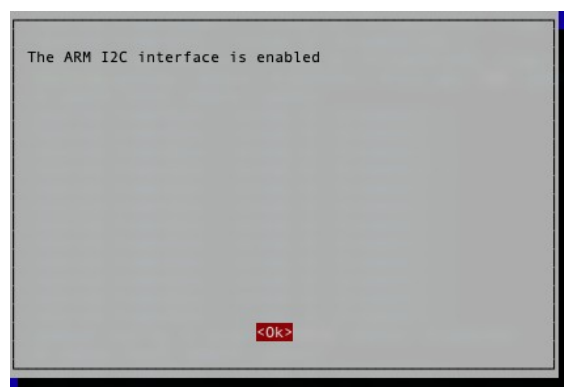
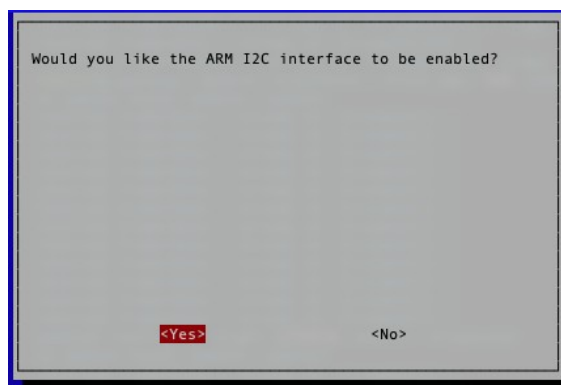
waterbrain@rak-gateway:~$

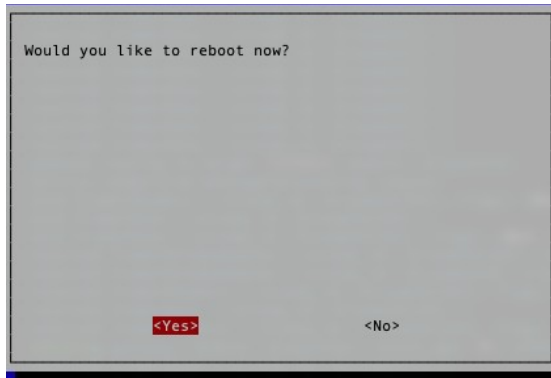
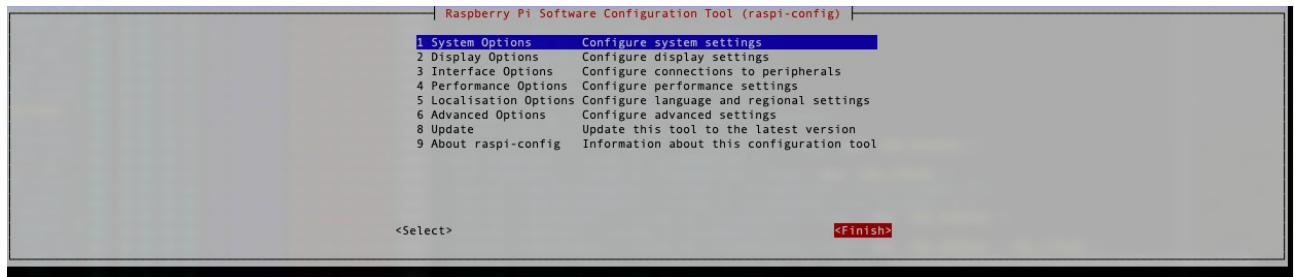
```

2.-:

- **sudo raspi-config**







3.- Editar `.bash_aliases`, yo suelo utilizar **nano**, y poner los alias siguientes:

- **nano ~/ .bash_aliases**

```
alias confrak='sudo nano /opt/ttn-gateway/packet_forwarder/lora_pkt_fwd/global_conf.json'
alias confrakloger='sudo nano /opt/ttn-gateway/lora_gateway/util_pkt_logger/global_conf.json'
alias raklog='sudo tail -f /var/log/syslog | grep ttn-gateway | ccze -A'
alias rakstop='sudo systemctl stop ttn-gateway'
alias rakstart='sudo systemctl start ttn-gateway'
alias heliumlog='sudo tail -f /var/log/syslog | ccze -A | grep helium_gateway'
alias heliumstop='sudo systemctl stop helium_gateway'
alias heliumstart='sudo systemctl start helium_gateway'
alias heliumrestart='sudo systemctl restart helium_gateway'
alias rakrestart='sudo systemctl restart ttn-gateway'
alias heliuminfo='helium_gateway --stdin info | ccze -A'
alias confhelium='sudo nano /etc/helium_gateway/settings.toml'
alias heliumversions='helium_gateway update list'
alias heliumverupdate='helium_gateway update download'
alias nrlog='pm2 logs node-red | ccze -A'
alias nrstop='pm2 stop node-red'
alias nrres='pm2 reload node-red;pm2 logs node-red | ccze -A; pm2 logs node-red | ccze -A'
alias nrstart='pm2 start node-red;pm2 logs node-red | ccze -A'
alias nrrestart='pm2 restart node-red;pm2 logs node-red | ccze -A'
alias nrconf='sudo nano /lib/systemd/system/nodered.service'
alias nract='bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)'
alias nrdir='cd .node-red/'
_dir() { [ ! -x "$(command -v ccze)" ] && sudo apt-get install ccze; \dir "$@" | ccze -A; }; alias dir='_dir'
_ifconfig() { [ ! -x "$(command -v ccze)" ] && sudo apt-get install ccze; \ifconfig "$@" | ccze -A; }; alias
ifconfig='_ifconfig'
upgr() { [ ! -x "$(command -v ccze)" ] && sudo apt-get install ccze; echo "*** Updating"; sudo apt-get update | ccze
-A; echo "*** Upgrading"; sudo apt-get upgrade | ccze -A; }
alias ap2log='sudo journalctl -f -n 50 -u apache2 -o cat | ccze -A'
alias ap2stop='sudo service apache2 stop'
alias ap2start='sudo service apache2 start'
alias ap2restart='sudo service apache2 restart'
alias ap2reload='sudo systemctl reload apache2'
alias ngstop='sudo /etc/init.d/nginx stop'
alias ngstart='sudo /etc/init.d/nginx start'
alias ngstatus='sudo /etc/init.d/nginx status'
alias model='cat /proc/device-tree/model | ccze -A;echo'
alias cputype='cat /proc/device-tree/cpus/cpu@0/compatible | ccze -A;echo'
```

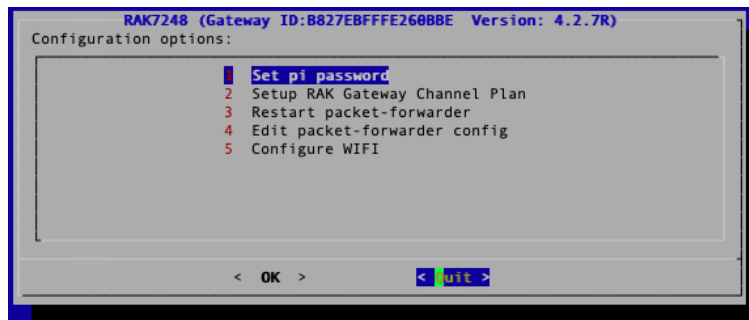
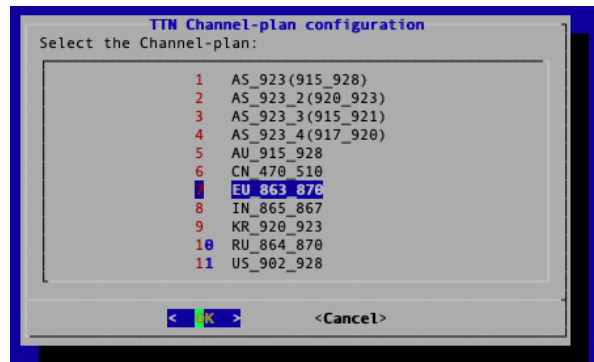
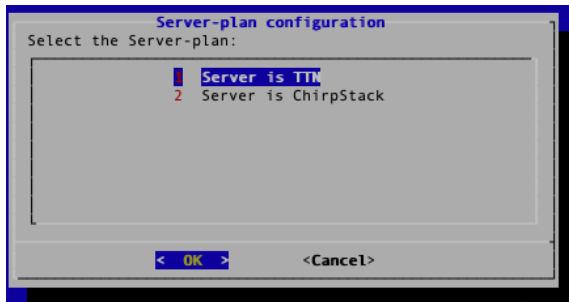
Ya sabreis que para salir de **nano** utilizamos **ctr-X** y contestar afirmativamente cuando nos pregunte si queremos guardar el archivo.

4.-

- sudo apt-get install git ccze mc macchanger
- git clone https://github.com/RAKWireless/rak_common_for_gateway.git
- cd ~/rak_common_for_gateway
- sudo ./install.sh
- sudo reboot

5. configurar el gateway de rak:

- **sudo gateway-config**



- **confrak**

```

GNU nano 5.4 antoniomaza - pi@rak-gateway: ~ -- ssh pi@192.168.1.018 -- 201x59
/opt/ttn-gateway/packet_forwarder/lora_pkt_fwd/global_conf.json

    ("rf_power": 15, "pa_gain": 1, "pwr_idx": 7),
    ("rf_power": 16, "pa_gain": 1, "pwr_idx": 8),
    ("rf_power": 17, "pa_gain": 1, "pwr_idx": 9),
    ("rf_power": 18, "pa_gain": 1, "pwr_idx": 10),
    ("rf_power": 19, "pa_gain": 1, "pwr_idx": 11),
    ("rf_power": 20, "pa_gain": 1, "pwr_idx": 12),
    ("rf_power": 21, "pa_gain": 1, "pwr_idx": 13),
    ("rf_power": 22, "pa_gain": 1, "pwr_idx": 14),
    ("rf_power": 23, "pa_gain": 1, "pwr_idx": 16),
    ("rf_power": 24, "pa_gain": 1, "pwr_idx": 17),
    ("rf_power": 25, "pa_gain": 1, "pwr_idx": 18),
    ("rf_power": 26, "pa_gain": 1, "pwr_idx": 19),
    ("rf_power": 27, "pa_gain": 1, "pwr_idx": 22)
  ],
  "radio_1": {
    "enable": true,
    "type": "SX1250",
    "freq": 868500000,
    "rssi_offset": -215.4,
    "rssi_comp": {"coeff_a": 0, "coeff_b": 0, "coeff_c": 20.41, "coeff_d": 2162.56, "coeff_e": 0},
    "tx_enable": false
  },
  "chan_multisf_all": {"spreading_factor_enable": [ 5, 6, 7, 8, 9, 10, 11, 12 ]},
  "chan_multisf_0": {"enable": true, "radio": 1, "if": -400000},
  "chan_multisf_1": {"enable": true, "radio": 1, "if": -200000},
  "chan_multisf_2": {"enable": true, "radio": 1, "if": 0},
  "chan_multisf_3": {"enable": true, "radio": 0, "if": -400000},
  "chan_multisf_4": {"enable": true, "radio": 0, "if": -200000},
  "chan_multisf_5": {"enable": true, "radio": 0, "if": 0},
  "chan_multisf_6": {"enable": true, "radio": 0, "if": 200000},
  "chan_multisf_7": {"enable": true, "radio": 0, "if": 400000},
  "chan_lora_scd": {"enable": true, "radio": 1, "if": -200000, "bandwidth": 250000, "spread_factor": 7,
    "implicit_header": false, "implicit_payload_length": 17, "implicit_crc_on": false, "implicit_coderate": 1},
  "chan_FSK": {"enable": true, "radio": 1, "if": 300000, "bandwidth": 125000, "datarate": 50000}
},
"gateway_conf": {
  "gateway_id": "AA555A0000000000",
  /* change with default server address/ports
  configuration para loara
  "server_address": "eu1.cloud.thethings.network",
  "serv_port_up": 1700,
  "serv_port_down": 1700, */
  /* configuration para helium: */
  "server_address": "127.0.0.1",
  "serv_port_up": 1680,
  "serv_port_down": 1680,
  /* adjust the following parameters for your network */
  "keepalive_interval": 10,
  "stat_interval": 30,
  "push_timeout_s": 100,
  /* forward only valid packets */
  "forward_crc_valid": true,

```

Yo lo dejo así para poder jugar con lora o helium.

- **rakrestart**

6.- instalar la parte de helium:

- **cd /tmp**
- **wget https://github.com/helium/gateway-rs/releases/download/v1.0.0-alpha.22/helium-gateway-v1.0.0-alpha.22-raspi234.deb**
- **sudo dpkg -i helium-gateway-v1.0.0-alpha.22-raspi234.deb**
- **cd**
- **sudo nano /etc/helium_gateway/settings.toml**
- **sudo systemctl restart helium_gateway**

7.- ponerle una ip fija (opcional, pero muy conveniente):

Nos conectamos a la raspberry y ejecutamos:

- **sudo nano /etc/dhcpd.conf**

y añadimos al final lo siguiente:

```

interface eth0
static ip_address=192.168.1.18/24
static routers=192.168.1.1
noip6
static domain_name_servers=80.58.61.250 80.58.61.254
static domain_search=

```

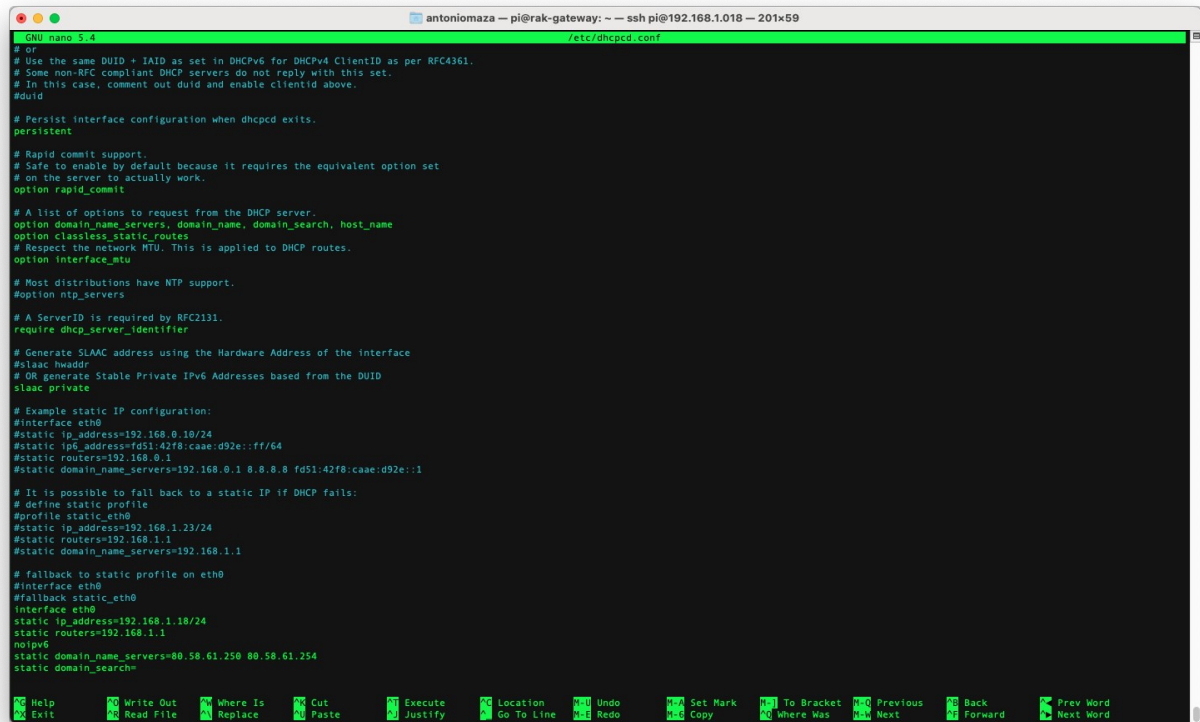
- **sudo nano -m /etc/network/interfaces**

```

auto lo
allow-hotplug wlan0
iface wlan0 inet static
    address 192.168.1.105
    netmask 255.255.255.0
    broadcast 192.168.1.255
    network 192.168.1.0
    gateway 192.168.1.1
    wpa-ssid "el ssid de tu router wifi"

```

```
wpa-psk "tu password"
wpa-ap-scan 1
wpa-scan-ssid 1
wpa-proto RSN
wpa-pairwise CCMP
wpa-key-mgmt WPA-PSK
dns-nameservers 80.58.61.250 80.58.61.254# interfaces(5) file used by ifup(8) and ifdown(8)
```



```
GNU nano 2.4 /etc/dhcpd.conf
# or
# Use the same DUID + IAID as set in DHCPv6 for DHCPv4 ClientID as per RFC4361.
# Some non-RFC compliant DHCP servers do not reply with this set.
# In this case, comment out duid and enable clientid above.
#duid
# Persist interface configuration when dhcpd exits.
persistent
# Rapid commit support.
# Safe to enable by default because it requires the equivalent option set
# on the server to actually work.
option rapid_commit
# A list of options to request from the DHCP server.
option domain_name_servers, domain_name, domain_search, host_name
option classless_static_routes
# Respect the network MTU. This is applied to DHCP routes.
option interface_mtu
# Most distributions have NTP support.
#option ntp_servers
# A ServerID is required by RFC2131.
require dhcp_server_identifier
# Generate SLAAC address using the Hardware Address of the interface
# OR generate Stable Private IPv6 Addresses based from the DUID
slaac private
# Example static IP configuration:
#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1
# fallback to static profile on eth0
#interface eth0
#fallback static_eth0
#interface eth0
#static ip_address=192.168.1.10/24
#static routers=192.168.1.1
#noipv6
#static domain_name_servers=80.58.61.250 80.58.61.254
#static domain_search=
```

Como ya he mencionado antes, esto se realiza con nano, y ya sabéis, una vez actualizado como salimos guardando el archivo.

le damos a re-arrancar la próxima vez ya estará en esa ip.

- **sudo reboot**

ya podemos utilizar con018 y una vez hecho el login a jugar.....

- **raklog**
- **heliumlog**
- **heliuminfo**
- **confrak**
- **confhelium**
- **heliuminfo**

o cualquiera de los alias que hemos añadido anteriormente

Corolario:

yo en mi caso ya tengo lo siguiente a mi entorno de shell

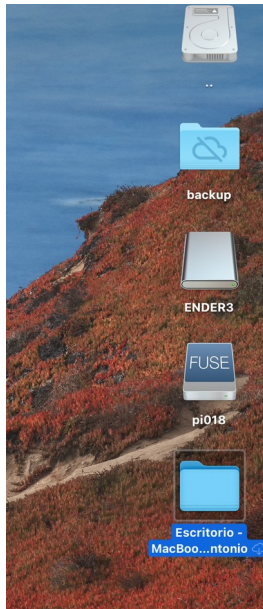
alias conecta018='ssh waterbrain@192.168.1.018'

y si tenemos alguna herramienta como **fuse** en el mac podemos añadir tambien **alias conecta018='sshfs waterbrain@192.168.1.18:/ pi018 -ovolname=pi018'**

Previamente debemos tener en nuestro home una carpeta vacía con el nombre pi018

ya podemos usar tanto un alias como otro

si le damos a conectar veremos en nuestro escritorio algo parecido a esto:



Sí clicamos en el disco pi018:

!MAGIA! Como podéis ver todo el contenido de la raspberry, algo parecido podemos hacer con **FileCilla** que esta disponible tanto para mac como para windows, pero a mi esto me parece más correcto y funcional.

