

# Cámara

- [8 Cámara](#)
- [8.1 ¿Qué vamos a hacer?](#)
- [8.2 Configuración](#)
- [8.3 Motion](#)

[illegible]

# 8.1 ¿Qué vamos a hacer?

Manejar la cámara web es fácil si queremos que salga por la salida HDMI de la Raspberry, simplemente ejecutando el siguiente código Python sale, en este caso durante 10 segundos **Pero no sale por VNC ni por SSH:**

```
from picamera import PiCamera
from time import sleep
camera = PiCamera()
camera.start_preview()
sleep(10)
camera.stop_preview()
```

Pero nosotros necesitamos que **retransmita = streaming** las imágenes, pues el robot se mueve, no tiene instalado un monitor.

Encontrarás en Internet varias formas de hacerlo:

1. Utilizando un programa en Python
2. Utilizando MJPG STREAMER bajo un programa servidor WEBIOPI
3. Utilizando **Motion** (recomendamos)

La primera opción [video](#) o [este tutorial](#) dependes de tener todas las librerías instaladas, por ejemplo *limmal* ...

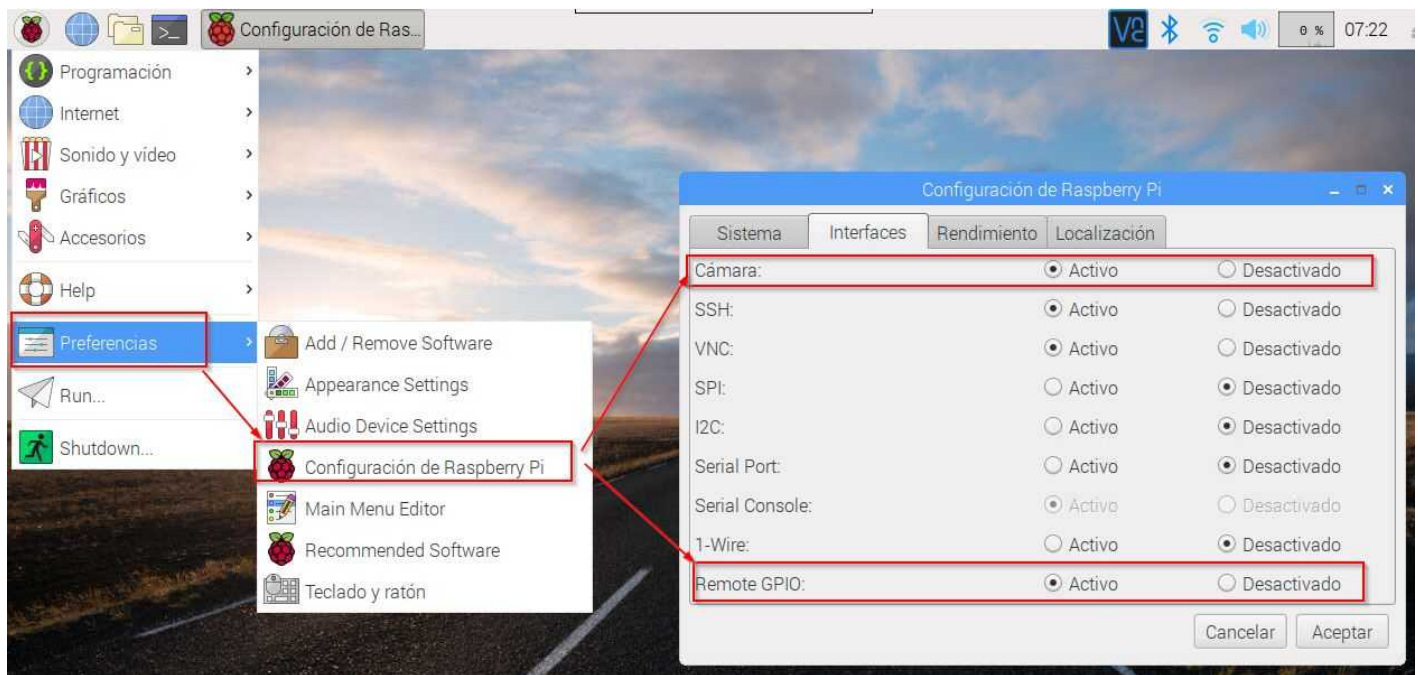
La segunda opción WEBIOPI (<http://webiopi.trouch.com/>) siguen con la versión 0.7.1 sin actualizar luego no lo recomendamos

Vamos a usar **Motion**, un programa diseñado para manejar la cámara en estos contextos y sí que está actualizado (actualmente por la 4.3.1) y muy extendido en el uso de cámaras web, lo que nos da unas garantías de no tener problemas, su página web oficial <https://motion-project.github.io/index.html> .

## 8.2 Configuración

Lo primero que tienes que hacer es activar la cámara y Remote GPIO

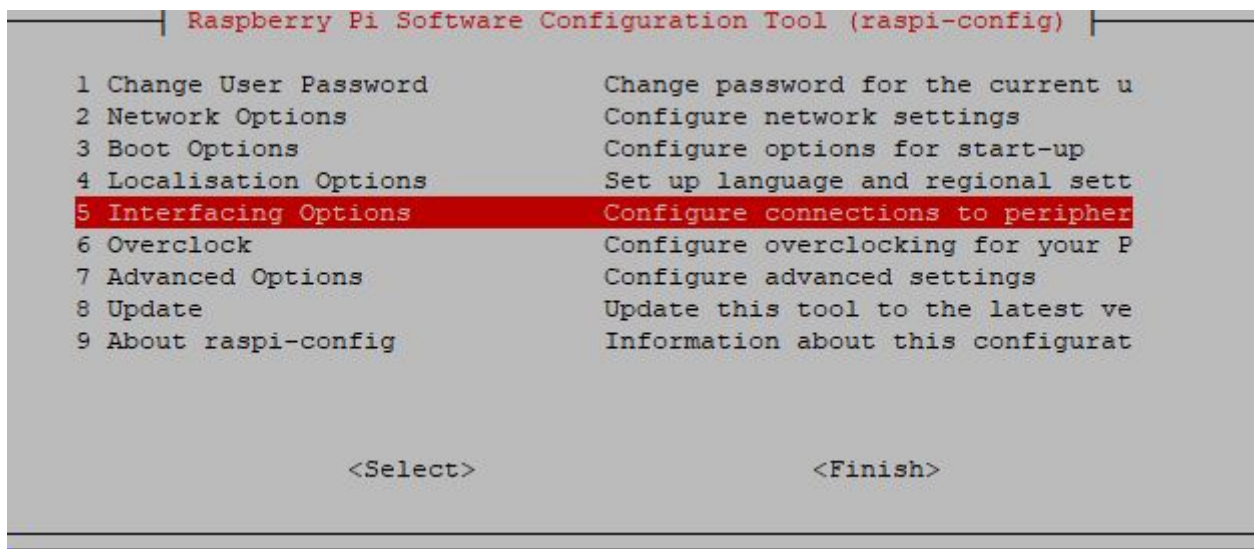
“ ¿Recuerdas? [En el capítulo 8.1 de Raspberry básico](#) ya activamos **VNC Server** y **SSH**, ahí también están la opciones de **activar la cámara y Remote GPIO**.



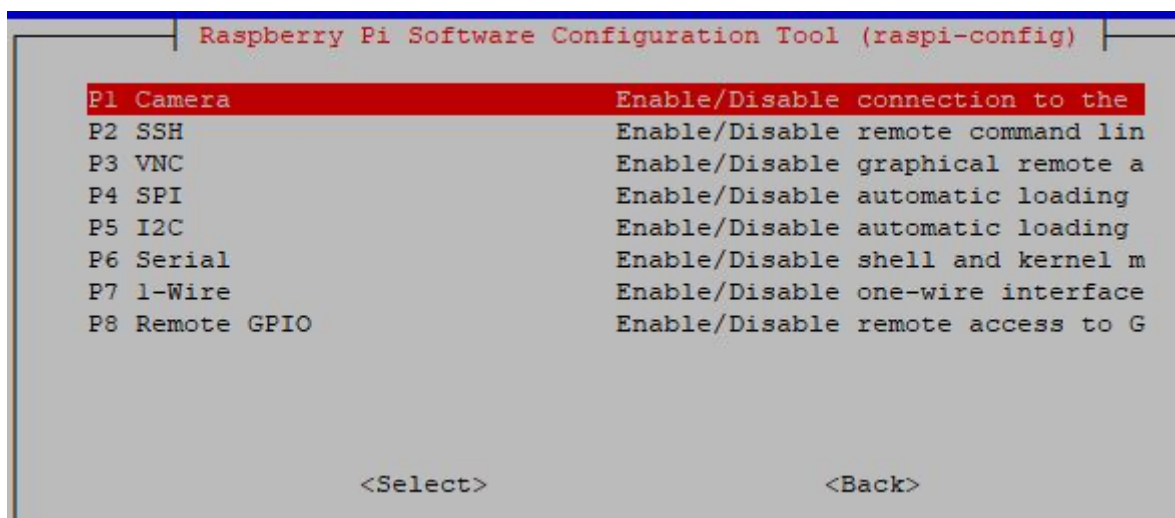
Si estás utilizando la Raspberry pero no de forma gráfica con VNC sino textual con SSH el comando a utilizar es

“ **sudo raspi-config**

entra en la opción 5



Y activas la cámara y Remote SSH en las opciones P1 y P8 :



## 8.3 Motion

Esta librería open-source muy utilizado en sistemas de alarma con la Raspberry (puedes hacer un sistema de videovigilancia de mi casa a distancia y monitorizarlo por muy bajo coste...), se pueden encontrar proyectos interesantes como :

- [Que grabe en un vídeo cuando detecta un movimiento](#)
- [Que nos envíe un email con una foto cuando detecta un movimiento](#)



Pero nosotros NO nos interesa que detecte movimiento, sino que simplemente haga streaming.

Para esto, simplemente modificaremos el fichero de configuración de la librería **motion.conf**. Es muy típico modificar ficheros de configuración tipo texto en los softwares abiertos, lo que muestra su versatilidad y potencialidad. Puedes ver las diferentes posibilidades de configuración de Motion [aquí](#)

## Cómo hacerlo

Abrimos una ventana de comandos, en SSH, [ya sabes cómo](#) y ejecutamos estas órdenes:

Instalamos MOTION :

## **sudo apt-get install motion**

Editamos el fichero de configuración motion.conf con el editor nano

## **sudo nano /etc/motion/motion.conf**

Buscamos estas líneas y las modificamos :

- stream\_localhost on lo cambiamos por off [si es on sólo localhost puede abrirlo, si es off pueden todos:](#)
- **stream\_localhost off**
- si vemos #stream\_port 8081 y como queremos abrirlo por ese puerto, le quitamos el hastag, o sea lo dejamos así,:
- **stream\_port 8081**
- Si vemos daemon off lo cambiamos por
- **daemon on**

Si estuvieran estas líneas webcam\_localhost on y webcam\_port 8080 las borramos, o mejor las dejamos como comentarios poniendo delante un hastag # así #webcam\_localhost on #webcam\_port 8080.

En esta [página](#) podemos ver otra configuración de motion.conf para el mismo propósito de streaming.

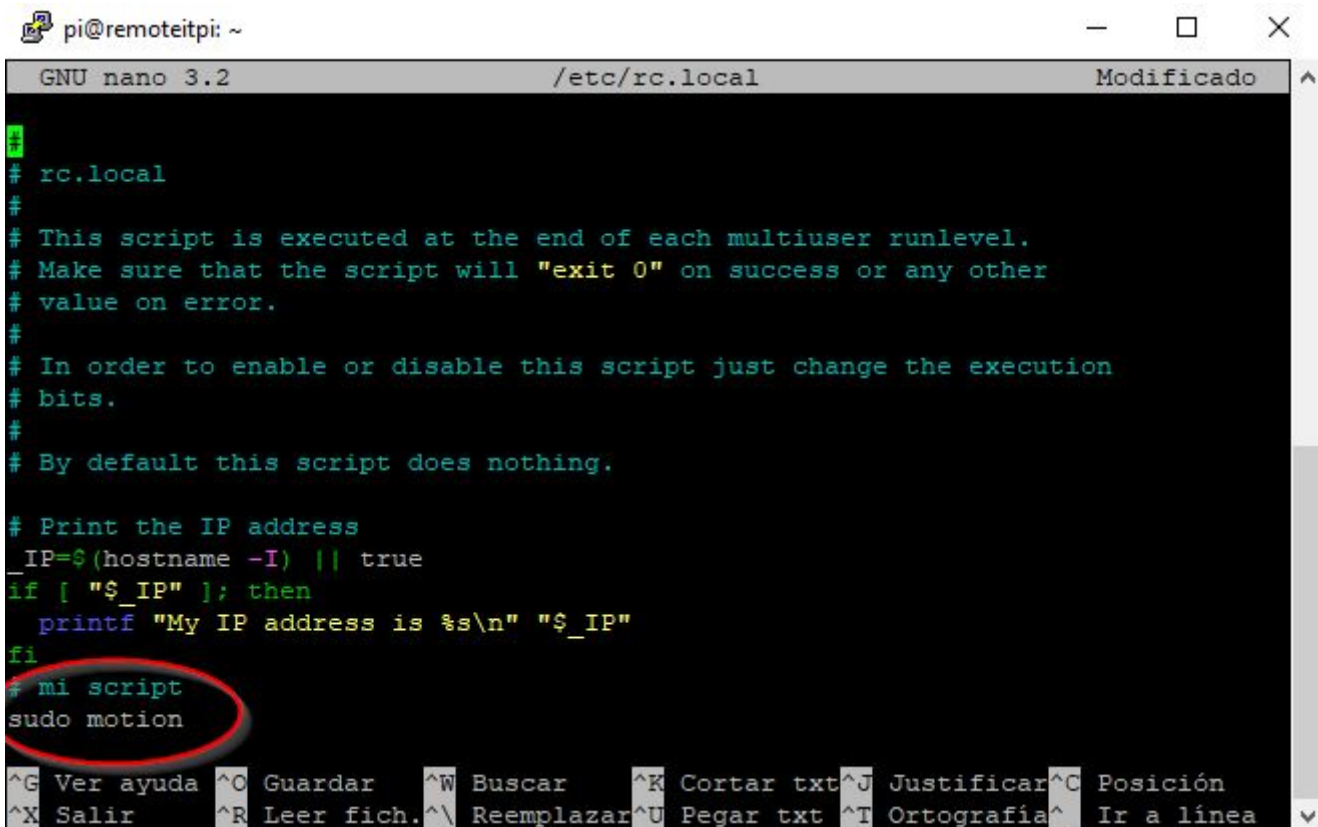
Grabamos el fichero: pulsando Ctrl+X se sale pero preguntará si queremos grabar el fichero con el mismo nombre, le decimos que sí

Finalmente ejecutamos motion con esta orden

## **sudo motion**

Si queremos que se ejecute de forma automática cuando arranque la raspberrypi editamos el fichero /etc/rc.local y al final le ponemos esa instrucción. Es decir

**sudo nano /etc/rc.local** y añadimos sudo motion al final (he puesto un comentario My script optativo)



```
pi@remoteitpi: ~
GNU nano 3.2 /etc/rc.local Modificado
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
#
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
# mi script
sudo motion

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

## ¿Cómo se ve desde la red local?

Pues abrimos un navegador y ponemos la dirección a través del puerto que le hemos dicho en stream\_port o sea 8081

http://---LA-DIRECCION-DE-LA-RASPBERRY---:8081

es decir si la dirección es 192.168.1.25 entonces tecleamos http://192.168.1.25:8081

Si queremos un protocolo seguro https mirar esta [página](#)

## ¿Y desde Internet?

### Opción instalar un nuevo servicio

No se puede hacer gráficamente en la página Remote.it no sabemos por qué. Hay que hacerlo con comandos con SSH.

**sudo remoteit add -h**



Nos sale una lista de servicios que podemos añadir, tecleamos el ID del servicio que queremos añadir en este caso vemos en la figura que el 7 es HTTP.

Nos pide el puerto, ponemos **8081** el mismo que en Motion


Un nombre para el servicio, le hemos puesto webcam, y hecho lo que tienes que hacer en la Raspberry.

```
pi@remoteitpi:~$ sudo remoteit add -h
Service Type is not specified or invalid. This is a list of valid service types

ID      | Name      | Default Port | Protocol | Description
-----|-----|-----|-----|-----
1       | TCP       | 0            | TCP      | Generic TCP
4       | VNC       | 5900         | TCP      | VNC remote desktop
5       | RDP       | 3389         | TCP      | Microsoft remote desktop
7       | HTTP      | 80           | TCP      | Web protocol
8       | HTTPS     | 443          | TCP      | Secure web protocol
28      | SSH       | 22           | TCP      | Secure shell terminal
34      | SMB/CIFS  | 445          | TCP      | Internet file sharing
37      | NxWitness | 7001         | TCP      | Nx Witness VMS
38      | Nextcloud | 443          | TCP      | Nextcloud hub
39      | OpenVPN   | 1194         | TCP      | OpenVPN server
41      | Minecraft | 25565        | TCP      | Minecraft server
42      | Admin Panel | 29999        | TCP      | remote.it admin panel
32769   | UDP       | 0            | UDP      | Generic UDP
32770   | WireGuard | 51820        | UDP      | WireGuard VPN server

Enter Service Type ('ID' or 'Name', not case sensitive, default is 'ssh'): 7
Enter Port (default is '80'): 8081
Enter Service Name (default is 'MyService'): webcam
[ ] ADD in progress ...
✓ Login with remote.it account
✓ Creating service named 'webcam' at '127.0.0.1:8081'
✓ NEXT-ACTION: run 'remoteit add' to add more services
pi@remoteitpi:~$
```

Entramos ahora en un ordenador a remote.it en nuestros "Devices" y pinchamos en el servicio que hemos creado:



javierquintana

Devices

Contacts

Scripting

Registrations

Products

Logs

≡

Devices

+ ADD DEVICE

GROUP BY ▾

ACTIONS ▾

?

☐

Status ▴▾

Device Name

☐

CONNECT ▾

webcam

MIHTTP

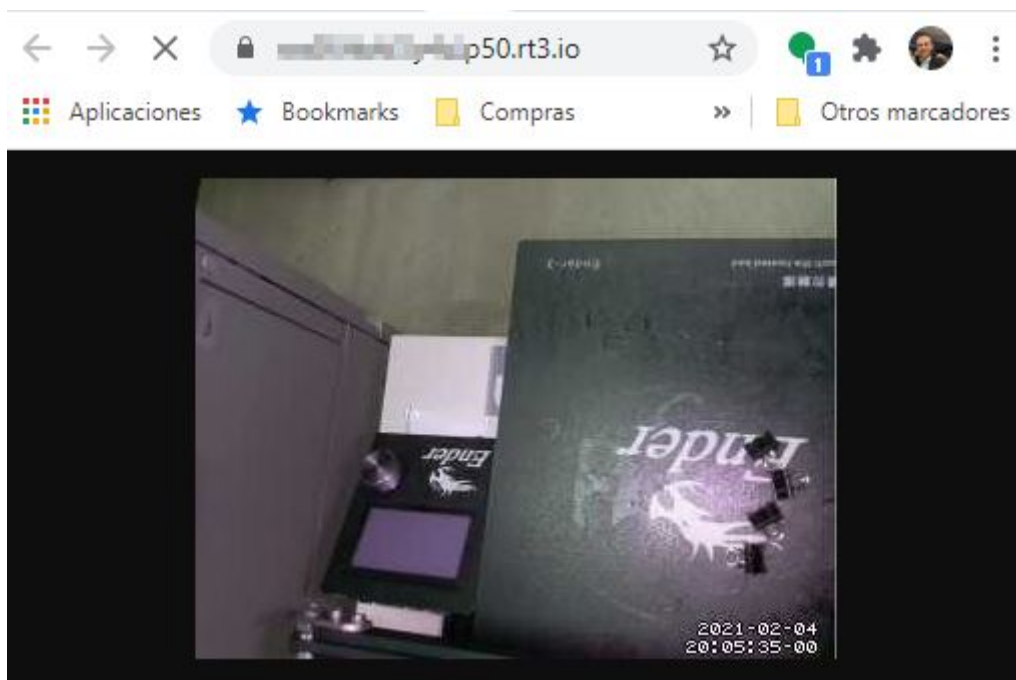
ssh

MyService

vnc

pi-concamara

Y automáticamente nos abre el navegador con la webcam funcionando



(en este caso he utilizado el robot para vigilar la impresora 3D)

## Vale, pero .. ¿y cómo se quita un servicio de Remoteit?

Entramos en al página web en el device en cuestión



Y copiamos el ID del servicio que queremos borrar:

# Device Services

Connect to your services.

The following Services are available on Device **pi-concamara**.

✓

pi-concamara

Bulk Service

Filter

Q

Status	Service	Application	Port	Service ID
✓	webcam	HTTP	8081	
✓	MIHTTP	HTTP	80	
✓	ssh	SSH	22	
✓	MyService	HTTP	81	80:00:00:00:01:0A:18:DF
✓	vnc	VNC	5900	

**Note on connections:** From the remote.it portal, connections will either be IP restricted or reverse proxy (HTTP/Secure Web).

**Reverse proxy:** For security purposes, this will be a randomly generated text string value for each connection, making it highly unlikely that anyone will discover the link. Once the connection closes, the link will no longer be valid, even to you.

**IP restriction:** Limits the connection to the originating connection IP address.

**Having trouble connecting?**

Mobile providers may use multiple source IP addresses. You may need to disable IP restrictions to connect to your devices from these platforms. NOTE: IP restriction will remain off for the duration of the connection.

☐ **Make connections public (IP Restriction will be disabled)**

Y ejecutamos la orden **sudo remoteit remove --id** y la ID que queremos borrar es decir en mi caso :

```
sudo remoteit remove --id 80:00:00:00:01:0A:18:DF
```

```
pi@remoteitpi:~$ sudo remoteit remove --id 80:00:00:00:01:0A:18:DF
□ REMOVE in progress ...
✓ Login with remote.it account
✓ Removing service '80:00:00:00:01:0A:18:DF'
✓ NEXT-ACTION: run 'sudo remoteit status' for status
pi@remoteitpi:~$
```

## Opción cutre con VNC

Si lo anterior por alguna razón fallara o remote.it quita el servicio HTTP, puedes ver la cámara por VNC.

Tienes que acceder a la Raspberry desde Internet con VNC mira estos [apuntes](#)

Una vez accedido por VNC remotamente podemos abrir el navegador de la misma Raspberry y la IP de él mismo es 127.0.0.1 luego abrir

`http://127.0.0.1:8081`

ojo en el navegador de la Raspberry no en tu ordenador.