



**UNIVERSIDAD
DE GRANADA**

Informática Industrial

Como crear nuestro propio sistema de vigilancia si
tenemos una Raspberry Pi.

Jonathan Molinero Castillo

¿Qué es una Raspberry Pi?

La Raspberry Pi es un mini ordenador completo que dispone de una placa de reducidas dimensiones. Lleva todas las conexiones habituales de un ordenador estándar: puertos USB, conexión de red, salidas de audio y video, etc. Con los periféricos necesarios gracias a su salida de video HDMI, lo podemos conectar a un televisor o un monitor y utilizarlo como un ordenador de sobremesa o como un centro multimedia.

¿Qué necesitaremos para nuestro sistema?

- Raspberry Pi 3
- Tarjeta Micro SD 16GB
- Cable alimentación Raspberry Pi
- Cable HDMI
- Ratón y teclado
- Cámara de Raspberry Pi

Una vez puesta en marcha la Raspberry con su sistema operativo instalado

Lo primero que debemos hacer es actualizar el sistema y los paquetes que tenemos instalados en la Raspberry Pi.

Para ello abrimos una terminal y ejecutamos lo siguiente, para actualizar todos los repositorios:

```
Sudo apt-get update
```

```
Sudo apt-get upgrade
```

En cuanto acabe procederemos a instalar los paquetes básicos de Python

Instalación de Python:

Escribiremos los siguientes comandos en la terminal:

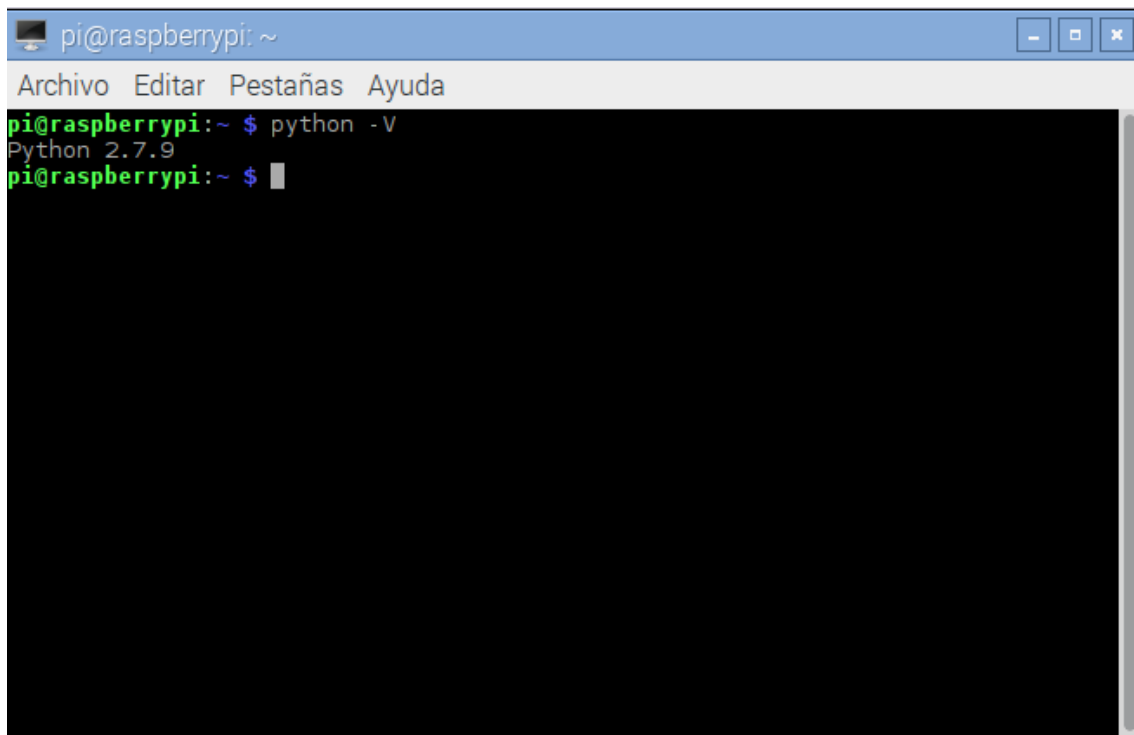
```
Sudo apt-get install build-essential python-dev python-distlib python-setuptools python-pip  
python-wheel libzmq-dev libgdal-dev
```

Esto nos va a instalar los paquetes:

- build-essential
- python-dev
- python-distlib
- python-setuptools
- python-pip
- python-wheel
- libzmq-dev
- libgdal-dev

Para comprobar que versión de Python se ha instalado lo podemos ver con el comando siguiente

`python -V.`

A screenshot of a terminal window titled 'pi@raspberrypi: ~'. The window has a menu bar with 'Archivo', 'Editar', 'Pestañas', and 'Ayuda'. The terminal shows the command 'python -V' being executed, resulting in the output 'Python 2.7.9'. The prompt 'pi@raspberrypi:~ \$' is visible at the end of the line.

Para poder trabajar con OpenCV y Python, necesitamos ciertos paquetes que nos van a facilitar las cosas bastante.

```
sudo apt-get install python-numpy python-matplotlib python-mpltoolkits.basemap python-scipy  
python-sklearn python-statsmodels
```

Este comando instala los siguientes paquetes:

- python-numpy

- python-matplotlib
- python-mpltoolkits.basemap
- python-scipy
- python-sklearn
- python-statsmodels

Instalar OpenCV

Lo haremos con el siguiente comando

```
sudo apt-get install libopencv-dev python-opencv
```

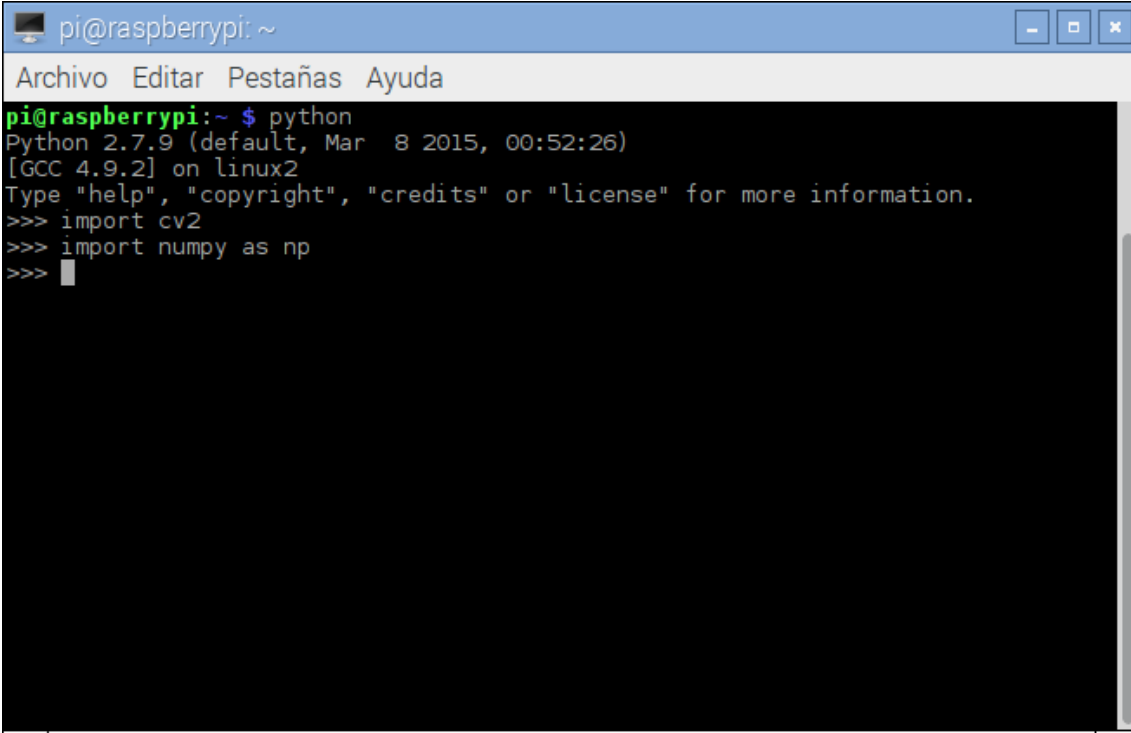
Para comprobar que todo funciona correctamente, escribimos lo siguiente en la terminal

```
python
```

Intentaremos importar estas dos librerías para ver que todo esta bien:

```
Import cv2
```

```
Import numpy as np
```



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~ $ python  
Python 2.7.9 (default, Mar  8 2015, 00:52:26)  
[GCC 4.9.2] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import cv2  
>>> import numpy as np  
>>> 
```

Instalación camara

Para instalar la cámara en la Raspberry Pi, solo tienes que insertar un extremo de la cinta en la placa.



Instalación de PICAMARA

Para acceder a la cámara vamos a necesitar un paquete para Python. El paquete se llama Picamara. Y Para instalarlo escribimos lo siguiente en la terminal:

```
Sudo apt-get install Python-picamera python3-picamera
```

Una vez instalado actualizamos todos los paquetes de nuevo

```
Sudo apt-get update
```

```
Sudo apt-get upgrade
```

Algoritmo para la detección de movimiento

```
from picamera import PiCamera
import time
import cv2

# Inicializamos la cámara con resolución 640x480
camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 32
rawCapture = PiRGBArray(camera, size=(640, 480))

# Tiempo de espera para que la cámara arranque
time.sleep(0.5)

# Inicializamos el primer frame a vacío.
# Nos servirá para obtener el fondo
fondo = None

# Capturamos frame a frame de la cámara
for frame in camera.capture_continuous(rawCapture, format="bgr",
use_video_port=True):
    # Obtenemos el array en formato NumPy
    image = frame.array

    # Convertimos a escala de grises
    gris = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Aplicamos suavizado para eliminar ruido
    gris = cv2.GaussianBlur(gris, (21, 21), 0)

    # Si todavía no hemos obtenido el fondo, lo obtenemos
    # Será el primer frame que obtengamos
```

if fondo is None:

```
fondo = gris
```

```
# Calculo de la diferencia entre el fondo y el frame actual
```

```
resta = cv2.absdiff(fondo, gris)
```

```
# Aplicamos un umbral
```

```
umbral = cv2.threshold(resta, 25, 255, cv2.THRESH_BINARY)[1]
```

```
# Dilatamos el umbral para tapar agujeros
```

```
umbral = cv2.dilate(umbral, None, iterations=2)
```

```
# Copiamos el umbral para detectar los contornos
```

```
contornosimg = umbral.copy()
```

```
# Buscamos contorno en la imagen
```

```
contornos, hierarchy =  
cv2.findContours(contornosimg, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
```

```
# Recorremos todos los contornos encontrados
```

```
for c in contornos:
```

```
    # Eliminamos los contornos más pequeños
```

```
    if cv2.contourArea(c) < 500:
```

```
        continue
```

```
    # Obtenemos el bounds del contorno, el rectángulo mayor  
    que engloba al contorno
```

```
    (x, y, w, h) = cv2.boundingRect(c)
```

```
    # Dibujamos el rectángulo del bounds
```

```
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
```

```
# Mostramos las diferentes capturas
```

```
cv2.imshow("Imagen Movimiento", image)
```

```
cv2.imshow("Umbral", umbral)
```

```
cv2.imshow("Resta", resta)
```

```
cv2.imshow("Contornos", contornosimg)
```

```
key = cv2.waitKey(1) & 0xFF
```

```
# Reseteamos el archivo raw para la siguiente captura
```

```
rawCapture.truncate(0)
```

```
# Con la letra s salimos de la aplicación
```

```
if key == ord("s"):
```

```
    break
```

La información e imágenes la he consultado de varios sitios:

<https://programarfacil.com/podcast/87-video-con-raspberry-pi/>

<https://programarfacil.com/podcast/87-video-con-raspberry-pi/>