

# MANUAL TECNICO



## Visión artificial para la detección de cansancio

Instituto Politécnico Nacional

Unidad Profesional Interdisciplinaria de Ingeniería  
Campus Zacatecas

---

2022

---

## Contenido

|  |    |
|--|----|
| Propósito .....  | 1  |
| Criterios de Entrada.....                                | 1  |
| Requerimientos de la aplicación móvil VADCC .....        | 1  |
| Requerimientos mínimos.....                              | 1  |
| Requerimientos recomendados.....                         | 1  |
| Requerimientos de la Raspberry .....                     | 1  |
| Requerimientos mínimos.....                              | 1  |
| Requerimientos recomendados.....                         | 1  |
| Puertos de la Raspberry utilizados .....                 | 2  |
| Herramientas desarrollo .....                            | 2  |
| Android Studio Dolphin 2021.3.1.....                     | 2  |
| Pycharm 2022.2.4 .....                                   | 2  |
| Thonny 4.0.1 .....                                       | 2  |
| Firebase Realtime Database.....                          | 2  |
| PyTorch.....   | 3  |
| Restricciones del producto .....                         | 3  |
| Permisos necesarios de la aplicación móvil.....          | 3  |
| Políticas de uso .....                                   | 3  |
| Restricciones de recursos.....                           | 3  |
| Restricciones de tiempo .....                            | 4  |
| Restricciones de alcance .....                           | 4  |
| Restricciones de calidad.....                            | 4  |
| Obtener acceso y/o descarga al producto .....            | 4  |
| Proceso de instalación y configuración del producto..... | 4  |
| Instalación de appVADCC .....                            | 4  |
| Instalación de la Raspberry .....                        | 8  |
| Instalación de Raspberry Pi SO .....                     | 8  |
| Instalación de Raspberry en el vehículo .....            | 23 |
| Instalar tapete en el vehículo.....                      | 23 |
| Despliegue del producto.....                             | 25 |

| CONTROL DE VERSIONES |                       |         |                        |        |          |
|----------------------|-----------------------|---------|------------------------|--------|----------|
| Autor(es)            | Fecha de modificación | Versión | Descripción del cambio | Revisó | Estado   |
| JJAG                 | 01/12/2022            | 1.0     | Creación del Documento | AISD   | Aprobado |

## *Propósito*

Este documento tiene como propósito describir las actividades para poner en marcha el producto; viendo su ambiente de operación, las restricciones y la instalación.

## *Criterios de Entrada*

### Requerimientos de la aplicación móvil VADCC

#### Requerimientos mínimos

- Android 8
- RAM 1G
- Cuenta con Bluetooth
- Cuenta con GPS

#### Requerimientos recomendados

- Android 11
- RAM 4G
- Cuenta con Bluetooth
- Cuenta con GPS
- Resolución de pantalla de 1600x720 o superior

### Requerimientos de la Raspberry

#### Requerimientos mínimos

- RAM 4G
- Memoria interna 32

#### Requerimientos recomendados

- RAM 8G
- Memoria interna 64G

## Puertos de la Raspberry utilizados

- 2 y 4: de 5V
- 22 (GIPO 25): puerto GIPO para alimentar el led amarillo
- 16 (GIPO 23): puerto GIPO para alimentar led verde
- 37 (GIPO 26): puerto GIPO para alimentar el sistema de buzzers
- 32 (GIPO 12, PWM0): puerto GIPO PWM para alimentar el sistema de módulos de vibración
- 6 y 34 (Ground): puertos de tierra

## *Herramientas desarrollo*

### Android Studio Dolphin 2021.3.1

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android, el cual fue utilizado para el desarrollo de la aplicación móvil con JAVA.



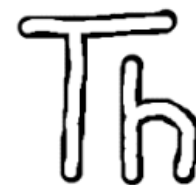
### Pycharm 2022.2.4

Es un entorno de desarrollo integrado utilizado en programación informática, concretamente para el lenguaje de programación Python. Está desarrollado por la empresa checa JetBrains. La cual fue utilizado para generar los códigos que serán utilizados en la raspberry



### Thonny 4.0.1

Es un entorno de desarrollo integrado para Python diseñado para principiantes. Fue creado por Aivar Annamaa, un programador estonio. El cual fue utilizado para comprobar que los códigos en generados en pycharm funcionen correctamente en la raspberry.



### Firestore Realtime Database

Es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles, ubicada en la nube, integrada con Google Cloud Platform, que usa un



conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad, haciendo posible el crecimiento del número de usuarios y dando resultado también a la obtención de una mayor monetización. Guarda y sincroniza datos entre tus usuarios en directo con Firebase Realtime Database. Optimiza para el uso sin conexión.

## PyTorch

Es una biblioteca de aprendizaje automático de código abierto basada en la biblioteca de Torch, utilizado para aplicaciones que implementan cosas como visión artificial y procesamiento de lenguajes naturales, principalmente desarrollado por el Laboratorio de Investigación de Inteligencia Artificial de Facebook (FAIR). Es un software libre y de código abierto liberado bajo la Licencia Modificada de BSD. A pesar de que la interfaz de Python está más pulida y es el foco principal del desarrollo.



## *Restricciones del producto*

### Permisos necesarios de la aplicación móvil

Para el funcionamiento pleno de la aplicación es necesario conceder los siguientes permisos a la aplicación:

- SMS
- Teléfono
- Ubicación

### Políticas de uso

El usuario de la aplicación debe tener una previa interacción con el dispositivo ya instalado en el vehículo, además de seguir todas las instrucciones plasmadas en el manual e usuario para que la eficiencia del producto sea lo más alta posible.

Además, se recomienda que los contactos de confianza ingresados en la aplicación tengan un dispositivo con la posibilidad de acceder a google maps para poder revisar la ubicación del conductor cuando se encuentre en peligro de sufrir un accidente. No obstante, se le notificara con un mensaje simple advirtiéndole que la persona que envió el mensaje está en riesgo de sufrir un accidente.

### Restricciones de recursos

Los recursos para firebase da permiso de un máximo de 100 conexiones en tiempo real por

lo que solo serán permitido ser instalado en 100 dispositivos móviles por ende 100 vehículos. Al menos que se cambie de plan sin costo a prepago de Firebase, más información: <https://firebase.google.com/pricing?authuser=0&hl=es>

## **Restricciones de tiempo**

Toda función para que sea eficaz

## **Restricciones de alcance**

El proyecto fue realizado con dimensiones para ser compatible para vehículos el segmento C. Por lo que para ser adaptado debe de tomar las medidas del vehículo y cambiar el cableado.

## **Restricciones de calidad**

### ***Obtener acceso y/o descarga al producto***

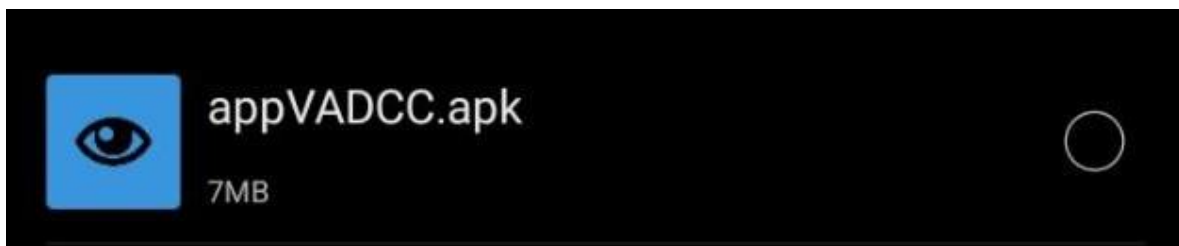
El apk de la appVADCC además de una Raspberry Pi OS, con los programas necesarios y códigos ya instalados se encuentran disponibles en el siguiente link de Dropbox <https://www.dropbox.com/sh/be9mci5o0b7ywk0/AABh88x1Msh-vcMNcjgqv43ia?dl=0+>

En caso de que no se encuentre disponible la apk se puede consultar el código en el github <https://github.com/vacctt>

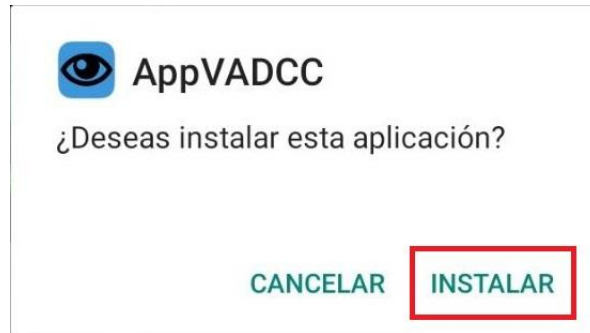
### ***Proceso de instalación y configuración del producto.***

#### **Instalación de appVADCC**

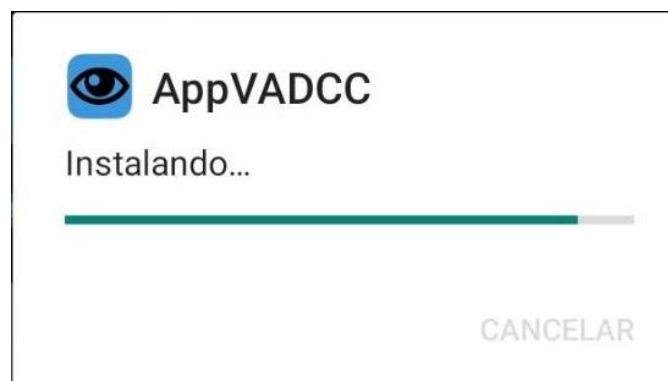
1. Seleccionar el archivo apk donde se descargó.



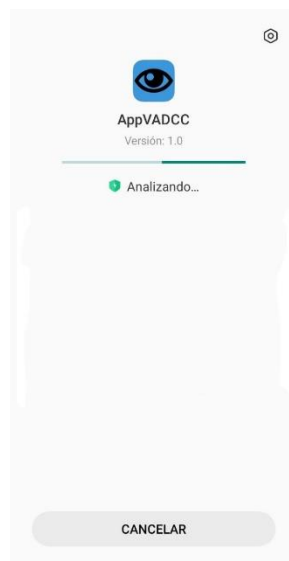
2. Una vez seleccionado en la ventana que se genera dar en instalar



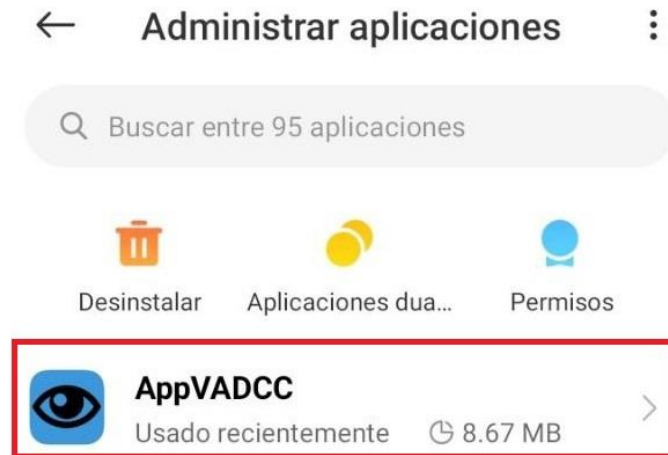
3. Esperar que se termine de instalar la aplicación.



4. Una vez instalada de ser necesario esperar a la verificación



5. Una vez ya instalada la aplicación se recomienda dar los permisos necesarios, buscando la aplicación dentro de administración de aplicaciones



6. Dentro de las características de la aplicación entrar al apartado de “Permisos de aplicación”

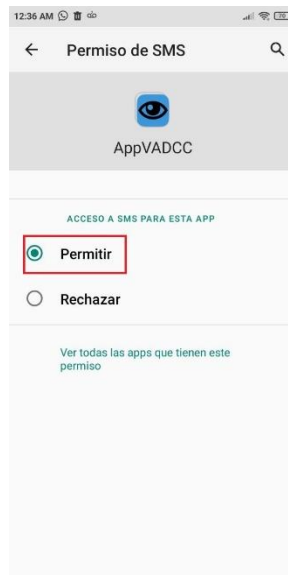


7. Seleccionar uno por uno los permisos solicitados por la aplicación,





8. Para cada permiso cambiar de “No permitir” a “Permitir”



# Instalación de la Raspberry

## Instalación de Raspberry Pi SO

### Instalar software Raspberry Pi Imager

Se descarga el software Raspberry Pi Imager desde la página oficial de raspberry <https://www.raspberrypi.com/software/>

#### Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. [Watch our 45-second video](#) to learn how to install an operating system using Raspberry Pi Imager.


Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

[Download for Windows](#)

[Download for macOS](#)

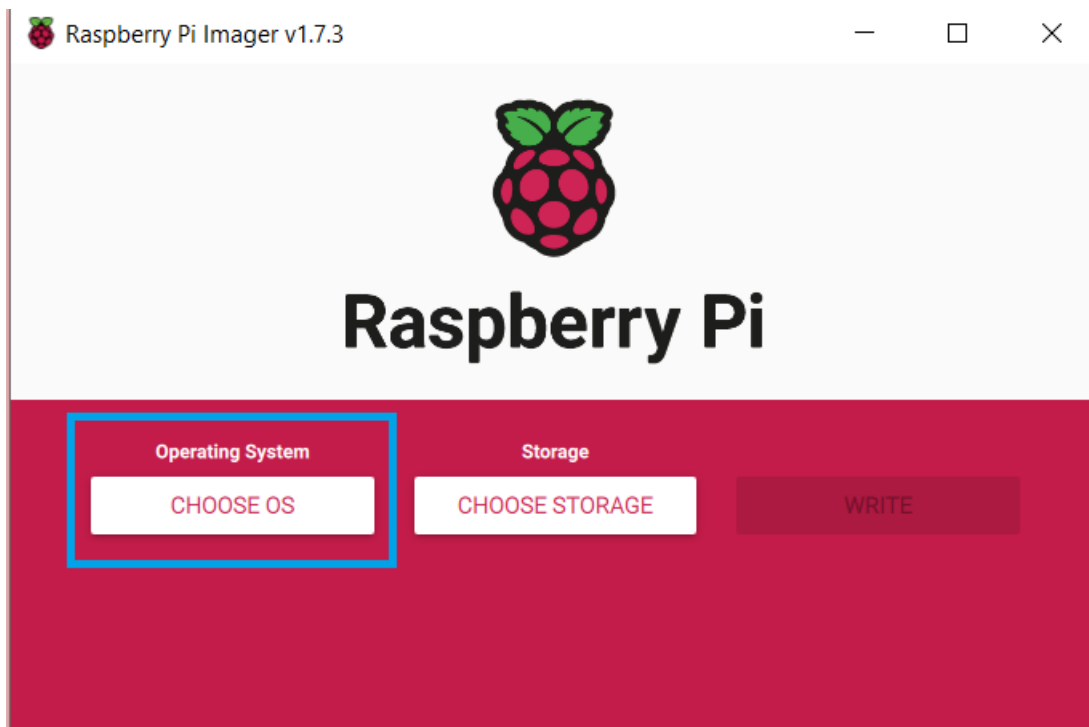
[Download for Ubuntu for x86](#)

To install on **Raspberry Pi OS**, type `sudo apt install rpi-imager` in a Terminal window.

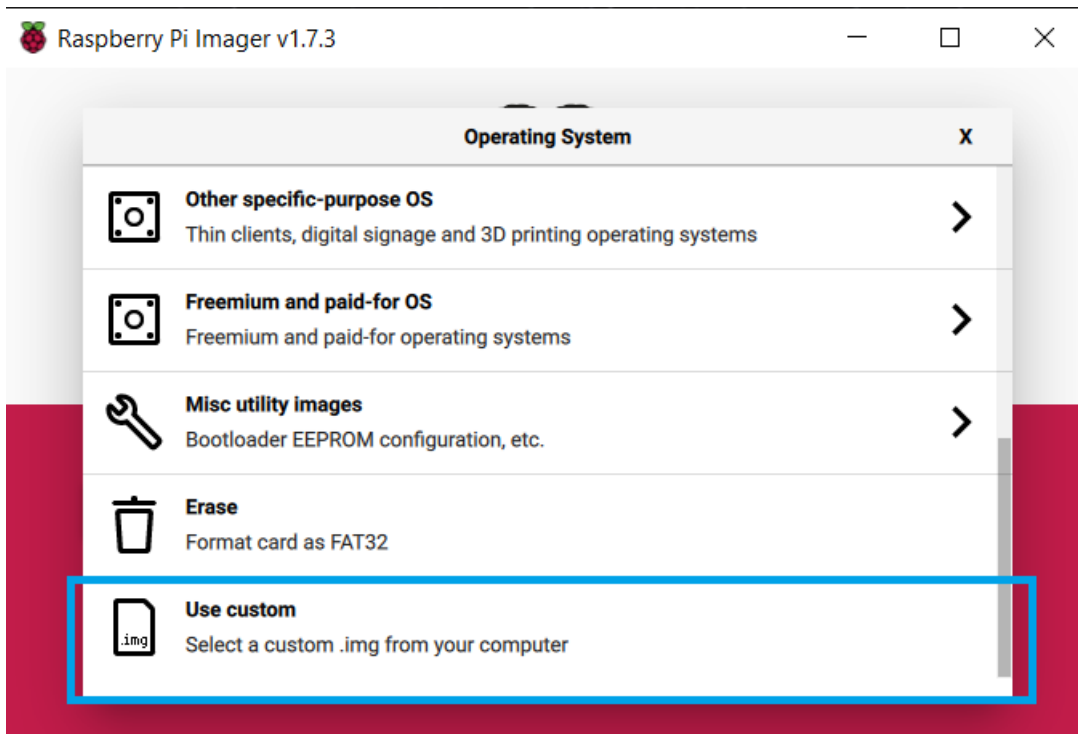


### Instalar el archivo VADCC.iso en la SD

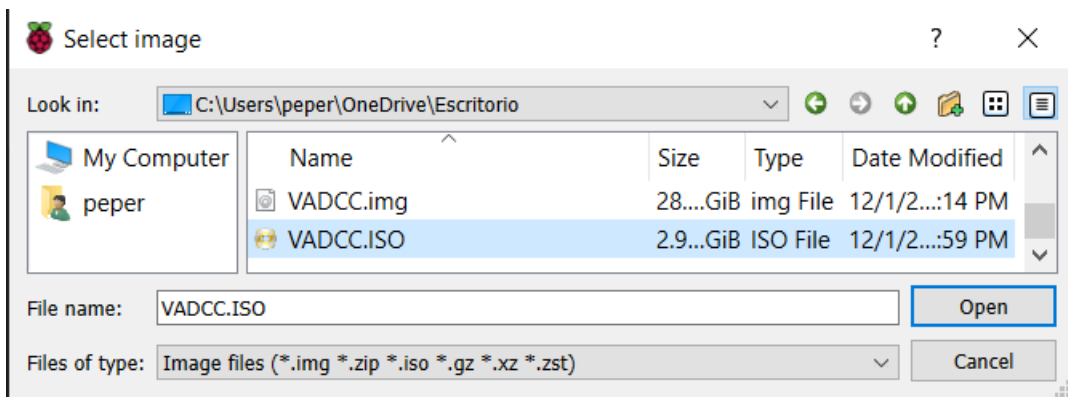
Se abre el software de Raspberry Pi Imager, y seleccionamos la sección de selección Sistema operativo a instalar.



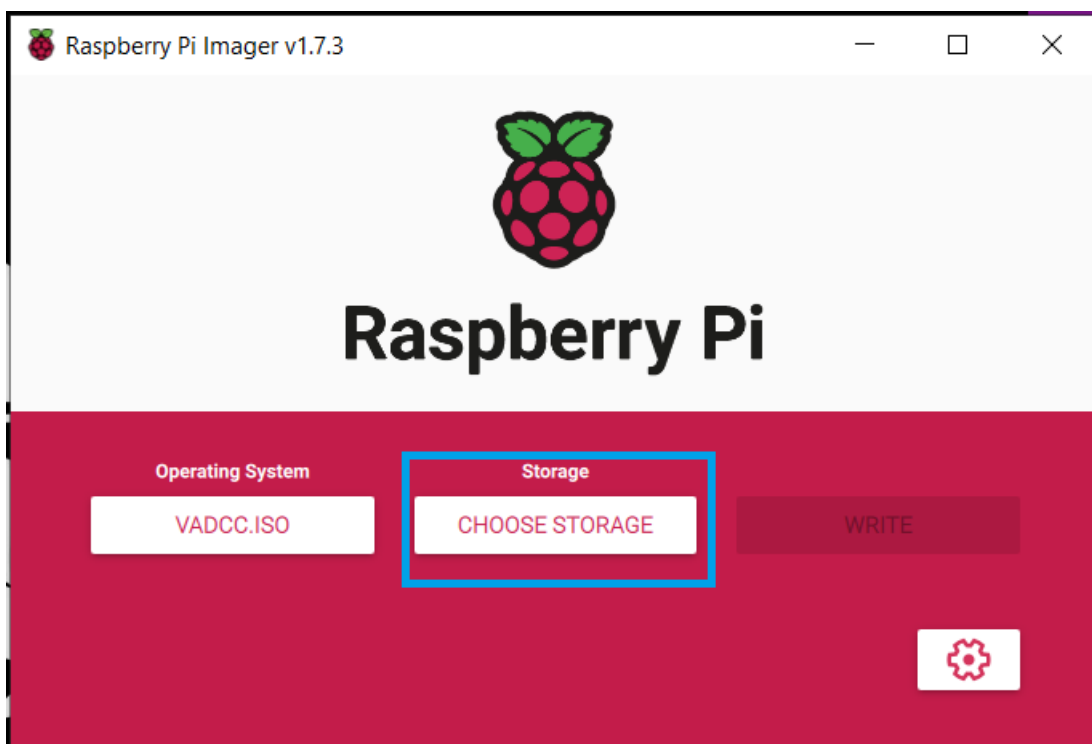
En la siguiente seccion se selecciona la opcion de “Use custom”



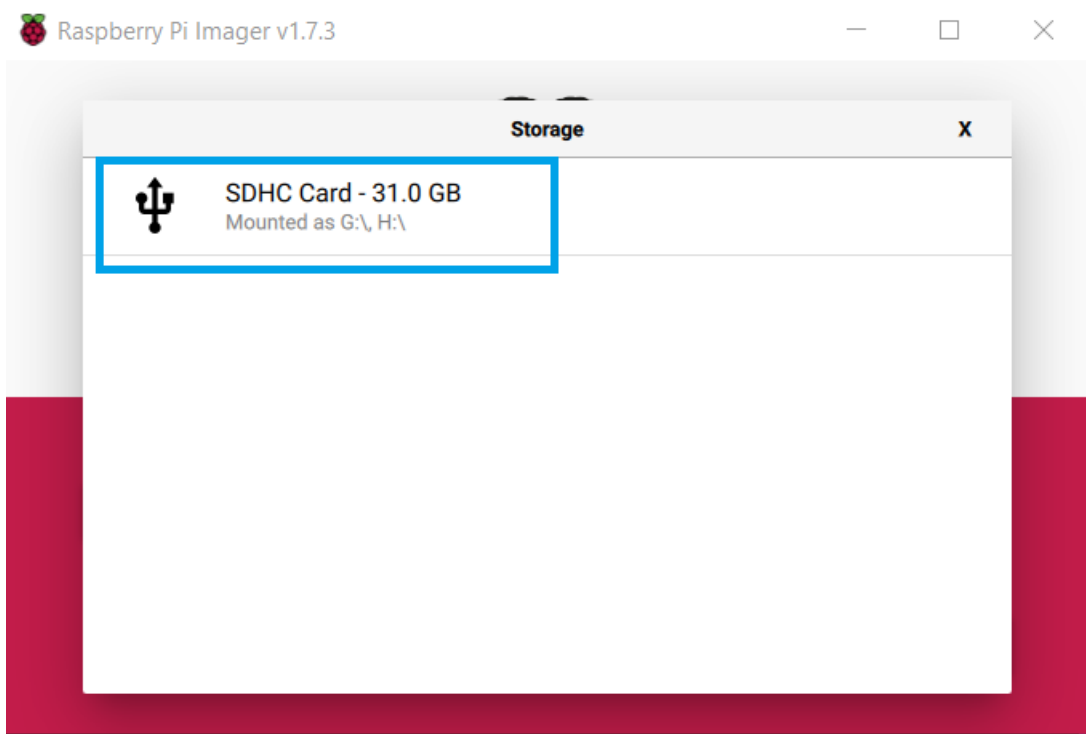
Para seleccionar el archivo ISO previamente descargado, y dar click en “Open”



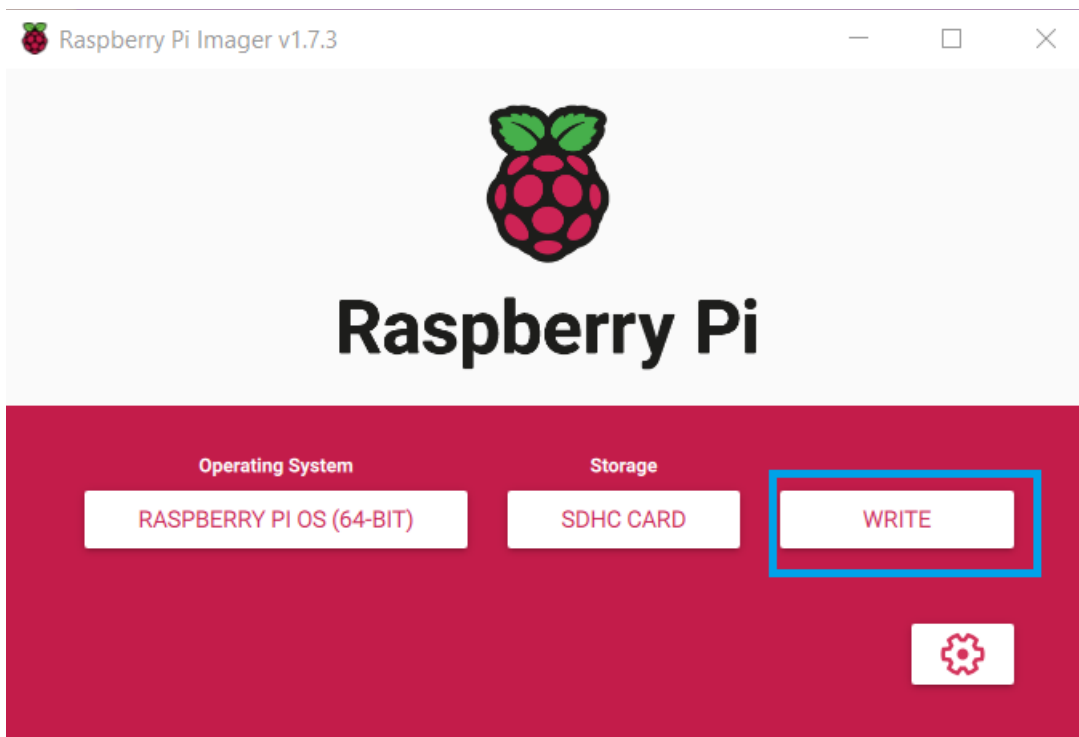
Posteriormente seleccionar la sd en el cual sera plasmado



Para despues seleccionar la sd que previamente debio de averse insertado en la computadora.



Por ultimo dar click en “Write” para instalar el sistema operativo.

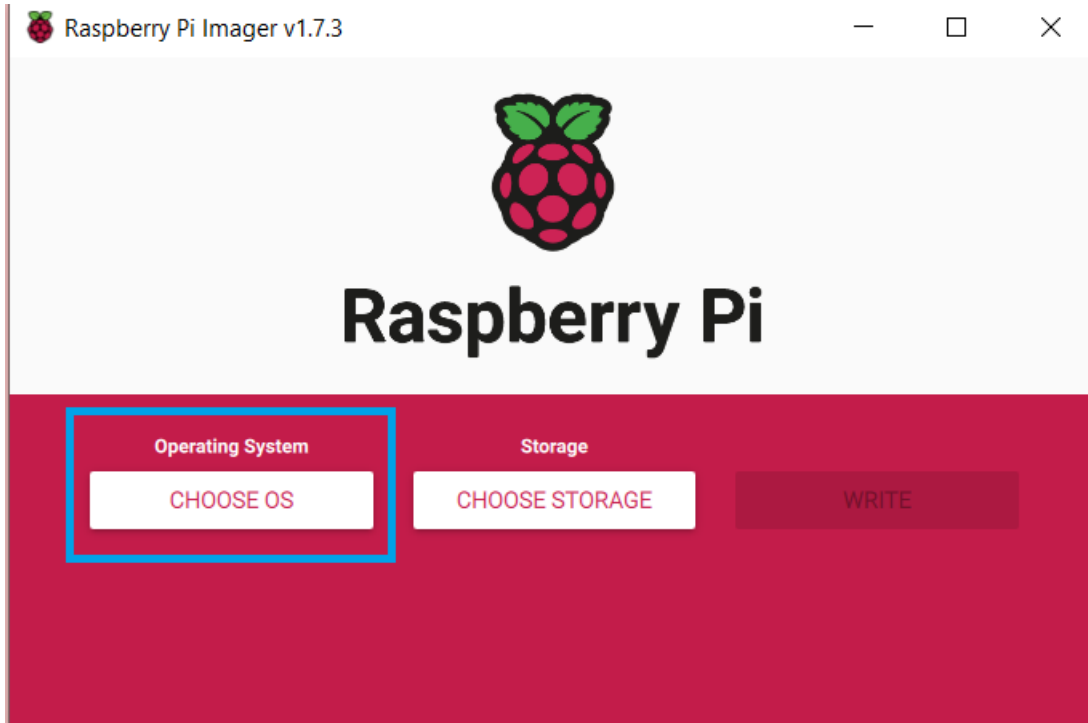


Una vez finalizao el proceso se pasara a ingresar la SD en la raspberry.

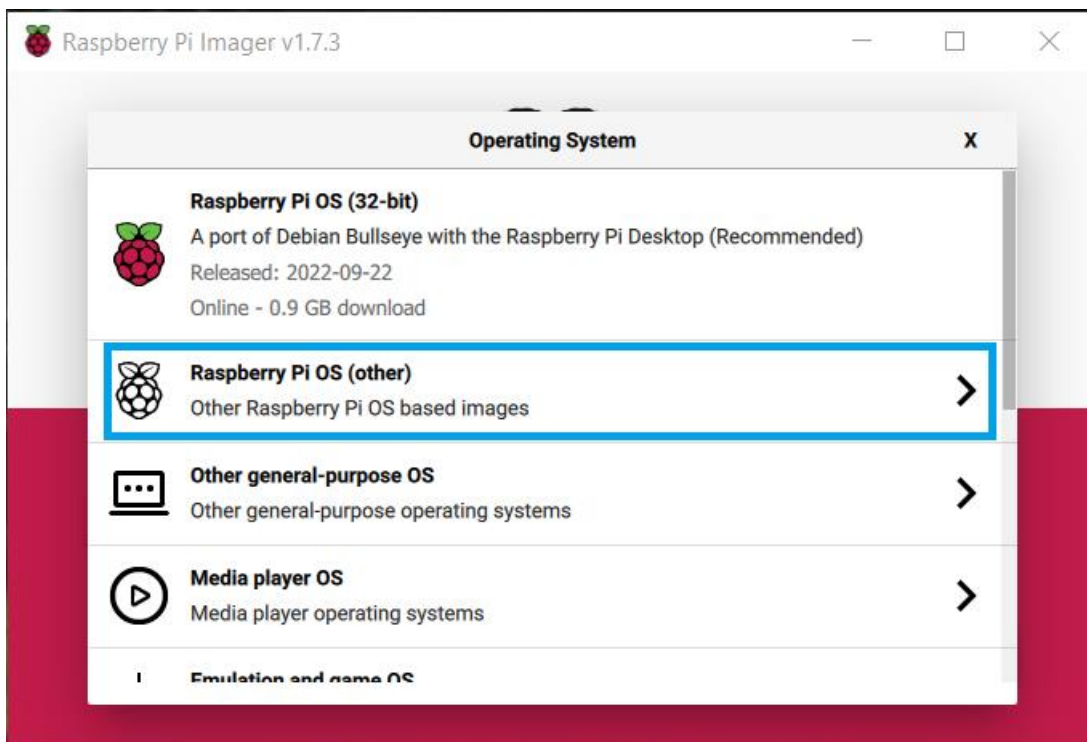
### Instalación manual de los componentes de la raspberry

En caso de que el archivo VADCC.iso este corrupto o deje de funcionar se deben seguir los siguientes pasos.

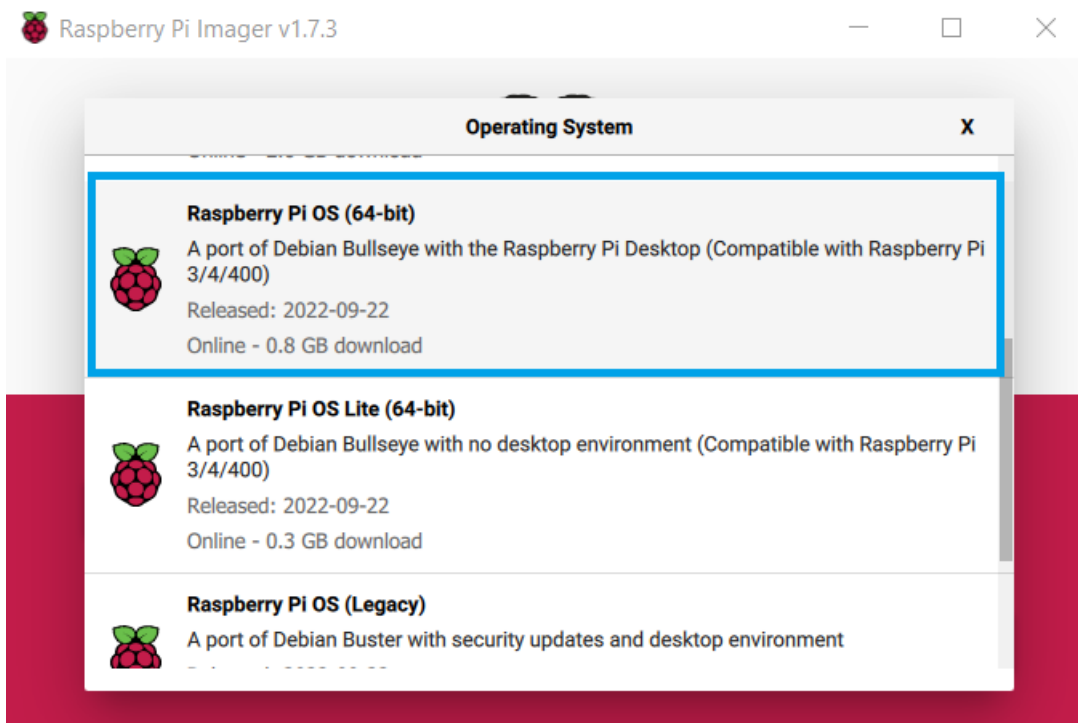
Se abre el software de Raspberry Pi Imager, y seleccionamos la sección de selección Sistema operativo a instalar.



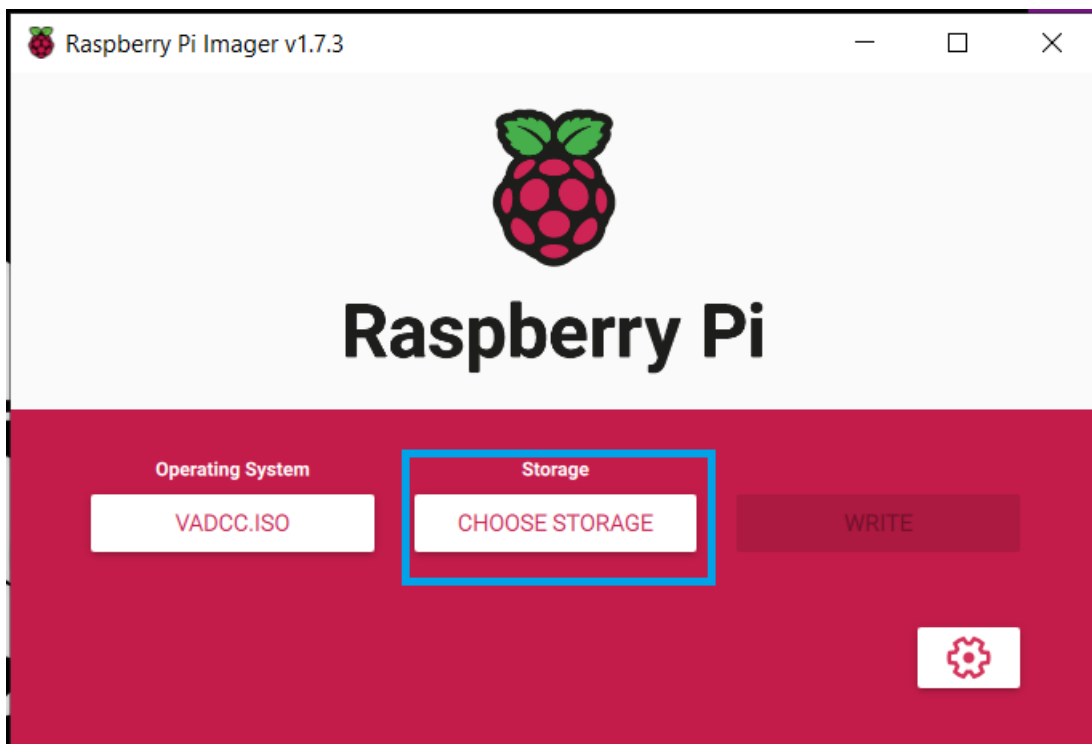
En la siguiente seccion se selecciona la opcion Raspberry



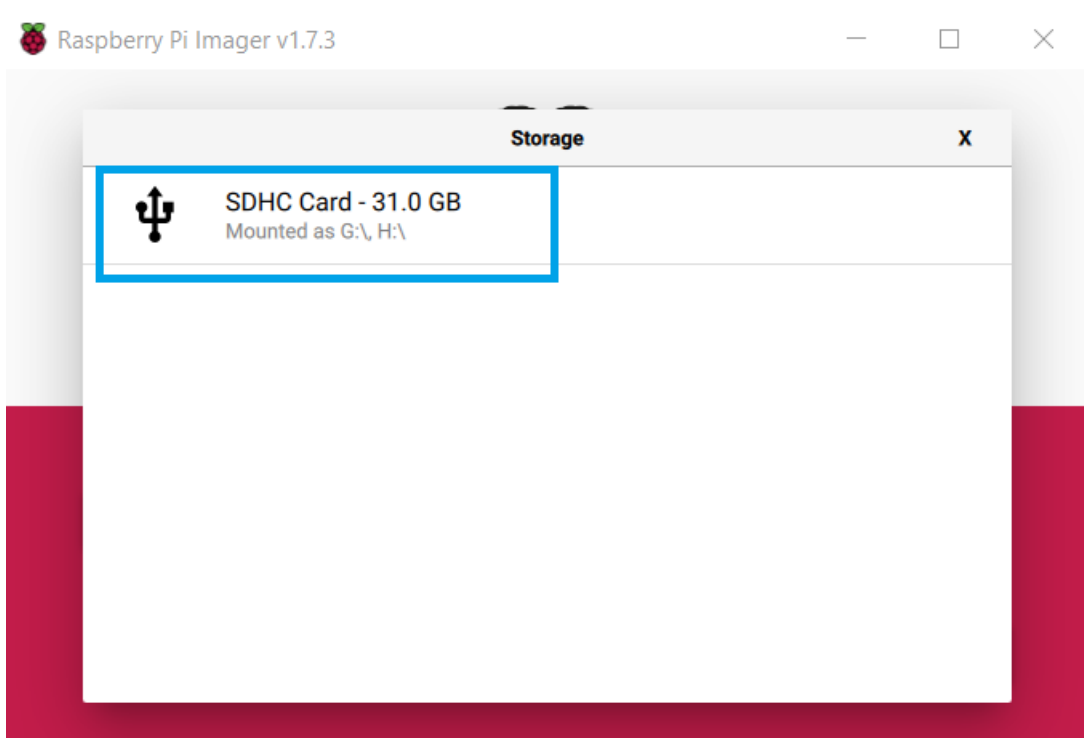
Posteriormente se selecciona



Posteriormente seleccionar la sd en el cual sera plasmado

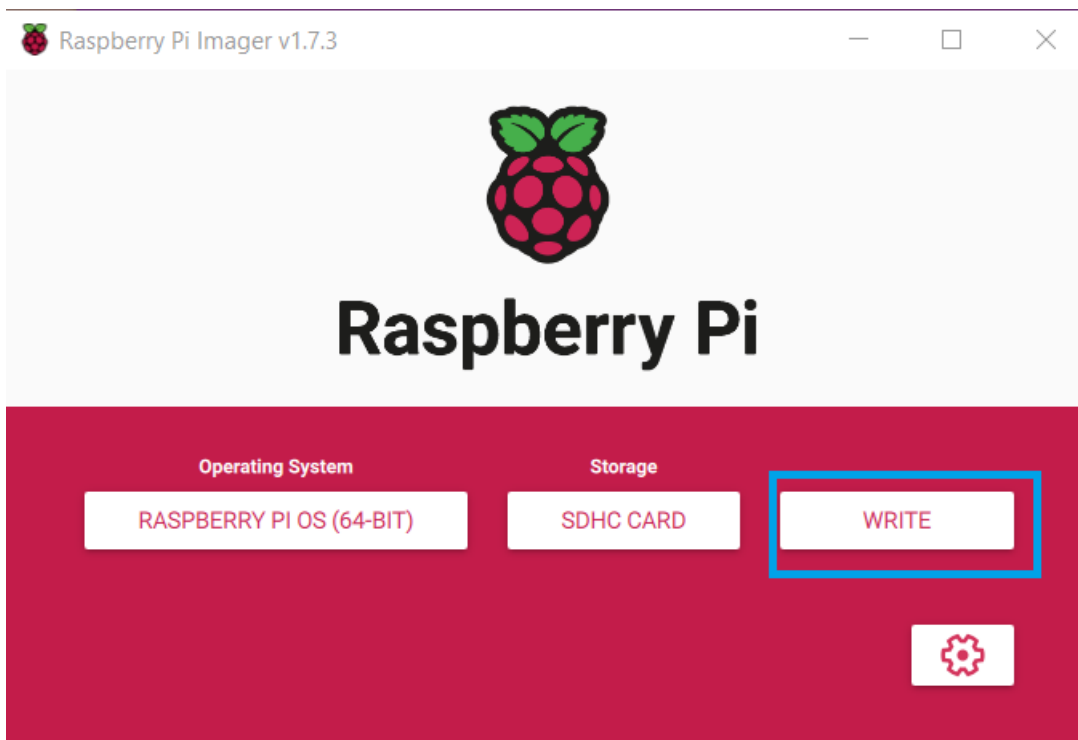


Para despues seleccionar la sd que previamente debio de averse insertado en la computadora.





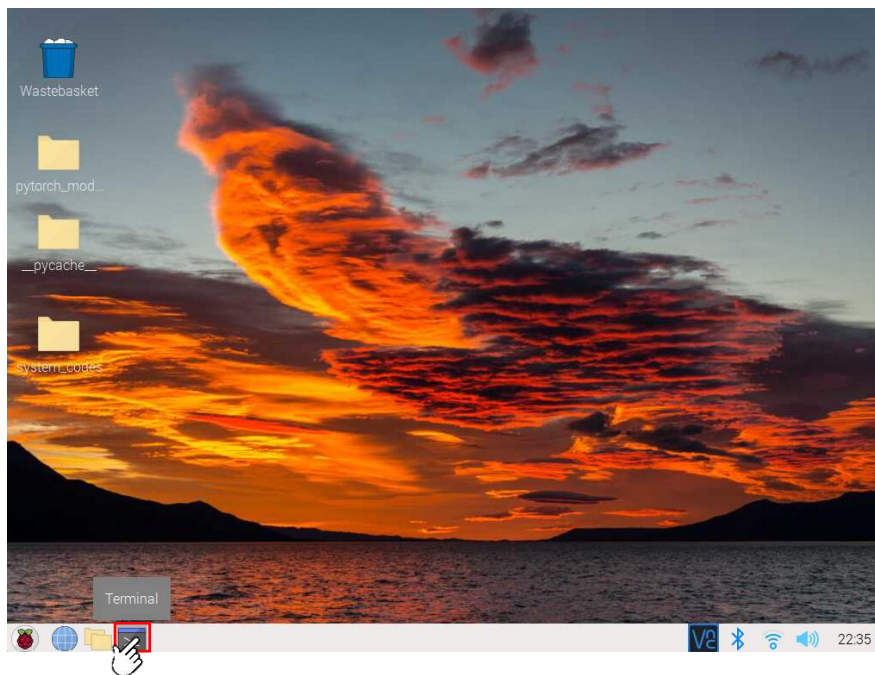
Por ultimo dar click en “Write” para instalar el sistema operativo.



Una vez finalizao el proceso se pasara a ingresar la SD en la raspberry.

Para su instalación es necesario contar con conexión a internet. Todas las librerías se instalan desde la terminal.

Abrimos la terminal que se encuentra en la barra de tareas:



Se ingresa el siguiente comando para actualizar el sistema antes de empezar con las instalaciones: **sudo apt-get update**

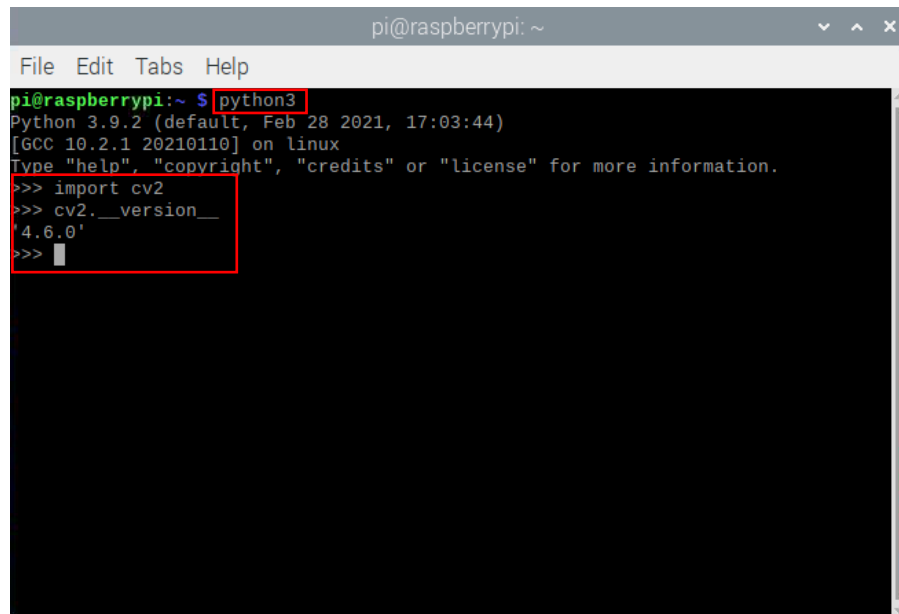
```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ sudo apt-get update  
Hit:1 http://security.debian.org/debian-security bullseye-security InRelease  
Hit:2 http://deb.debian.org/debian bullseye InRelease  
Hit:3 http://deb.debian.org/debian bullseye-updates InRelease  
Hit:4 http://archive.raspberrypi.org/debian bullseye InRelease  
Reading package lists... Done  
pi@raspberrypi:~ $
```

Con el sistema actualizado se puede proseguir con la instalación de las librerías.

## OpenCV

Ingresamos el siguiente comando: `sudo apt install python3-opencv`

Esperamos a que finalice el proceso. Una vez terminado verificamos que podamos usar la librería desde Python:



```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.6.0'
>>>
```

## Pytorch y Torchvision

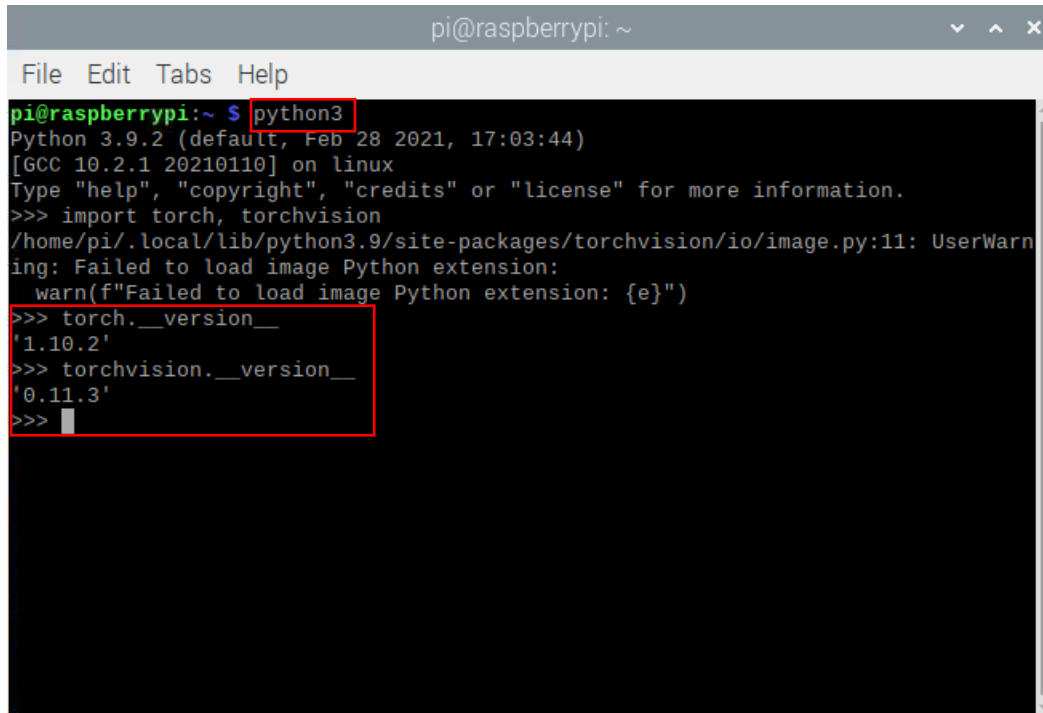
Para que pytorch y torchvision funcionen correctamente se requiere instalar versiones estables como se muestra a continuación:

| Operating system                                | PyTorch<br>1.12.0                    | PyTorch<br>1.11.0                    | PyTorch<br>1.10.0                    | PyTorch<br>1.9.0                     | PyTorch<br>1.8.0                     | PyTorch<br>1.7.0 |
|---|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|------------------|
| Raspberry Pi 64-bit<br>Bullseye<br>(Python 3.9) | Wheel<br>Vision<br>LibTorch<br>Guide | Wheel<br>Vision<br>LibTorch<br>Guide | Wheel<br>Vision<br>LibTorch<br>Guide | Wheel<br>Vision<br>LibTorch<br>Guide | Wheel<br>Vision<br>LibTorch<br>Guide |                  |

Fuente: Github Qengineering/PyTorch-Raspberry-Pi-64-OS

El sistema operativo en el que está construido el sistema es 64-bit Bullseye con Python 3.9 y se eligió la versión 1.10.X de PyTorch.

Ingresamos el siguiente comando: `sudo pip3 install torch==1.10.2 torchvision==0.11.3`. Y verificamos que se hayan instalado correctamente:



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ python3  
Python 3.9.2 (default, Feb 28 2021, 17:03:44)  
[GCC 10.2.1 20210110] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import torch, torchvision  
/home/pi/.local/lib/python3.9/site-packages/torchvision/io/image.py:11: UserWarning: Failed to load image Python extension:  
  warn(f"Failed to load image Python extension: {e}")  
>>> torch.__version__  
'1.10.2'  
>>> torchvision.__version__  
'0.11.3'  
>>>
```

Puede aparecer un error relacionado con la carga de una imagen de python pero es irrelevante.

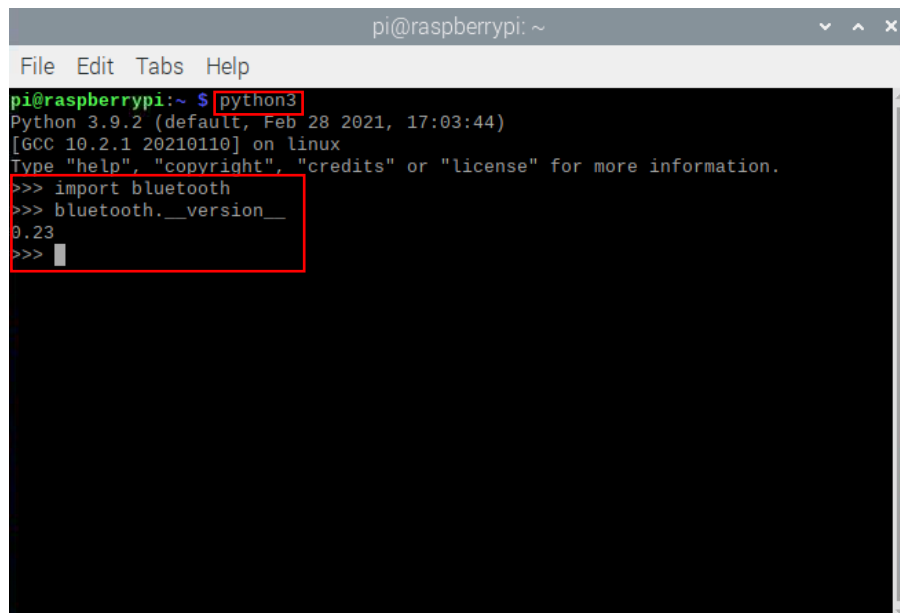
### Bluetooth para Python

Es necesario usar las librerías pybluez, libbluetooth, bluetooth y bluez. Las instalamos con los siguientes comandos:

```
sudo apt-get install bluetooth bluez libbluetooth-dev
```

```
sudo python3 -m pip install pybluez
```

Terminado el proceso de instalación verificamos que se haya instalado correctamente.

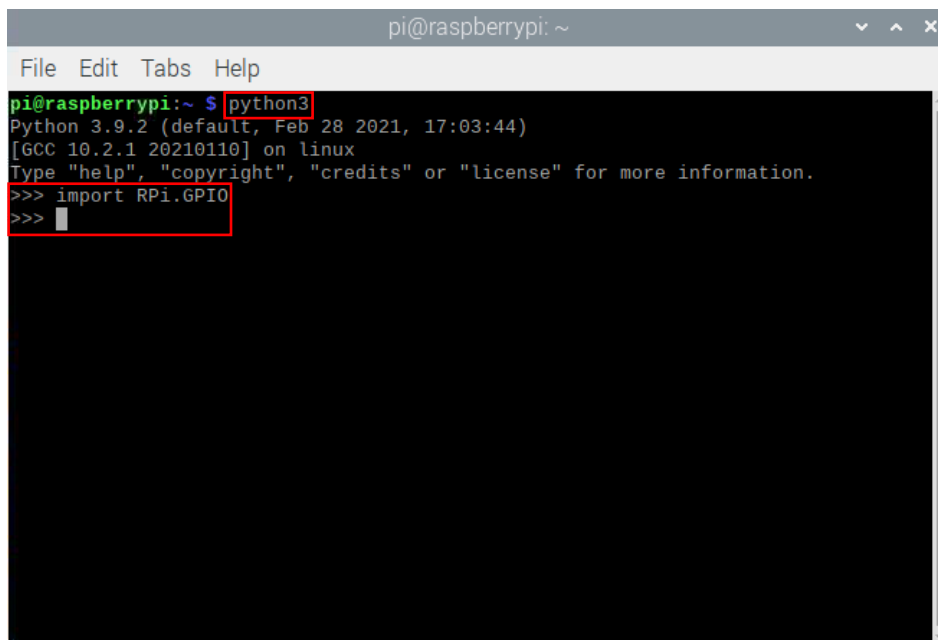


```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ python3  
Python 3.9.2 (default, Feb 28 2021, 17:03:44)  
[GCC 10.2.1 20210110] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import bluetooth  
>>> bluetooth.__version__  
0.23  
>>> 
```

## RPi.GPIO

La instalamos con el siguiente comando: `sudo apt-get install rpi.gpio`.

Terminada la instalación verificamos que se haya instalado correctamente:

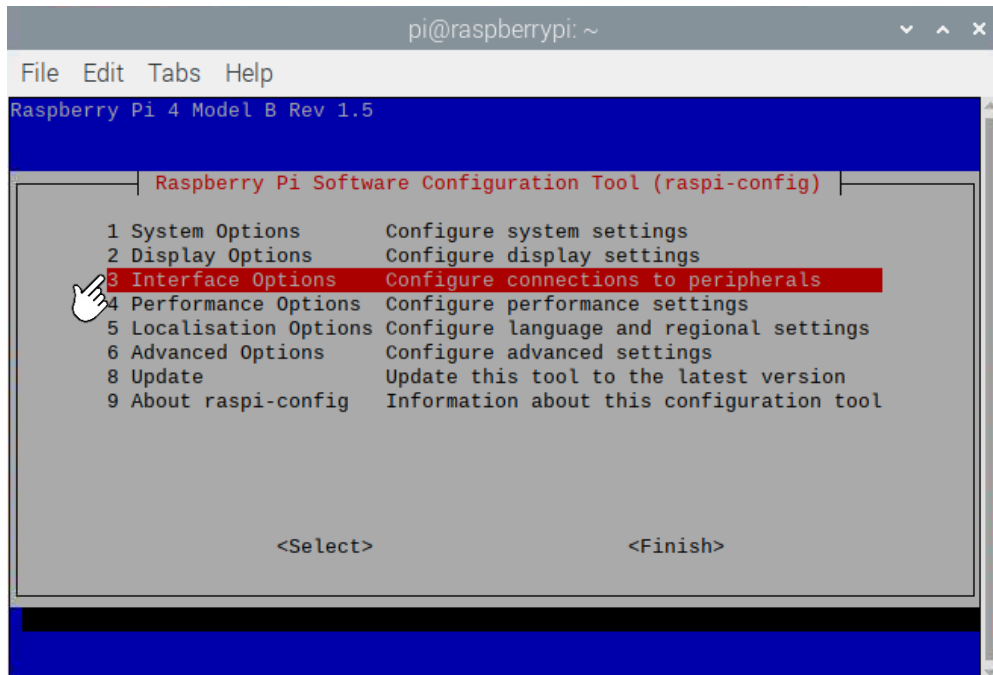


```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ python3  
Python 3.9.2 (default, Feb 28 2021, 17:03:44)  
[GCC 10.2.1 20210110] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import RPi.GPIO  
>>> 
```

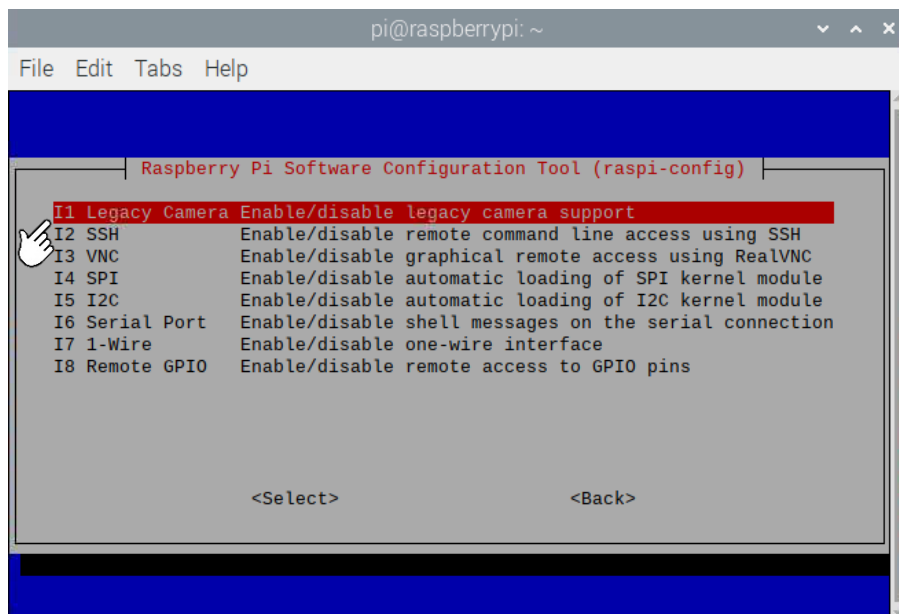
## Cámara

Por defecto el sistema operativo viene con una interfaz activada para soporte de la cámara legacy, sin embargo, para hacer uso de la cámara con la librería OpenCV (librería instalada en la sección anterior) se necesita desactivar dicho soporte.

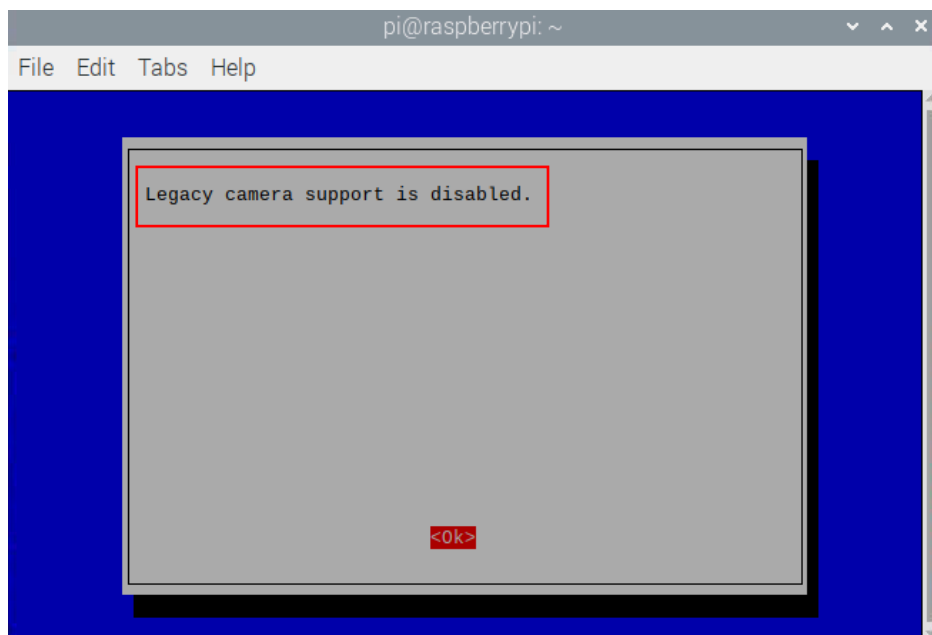
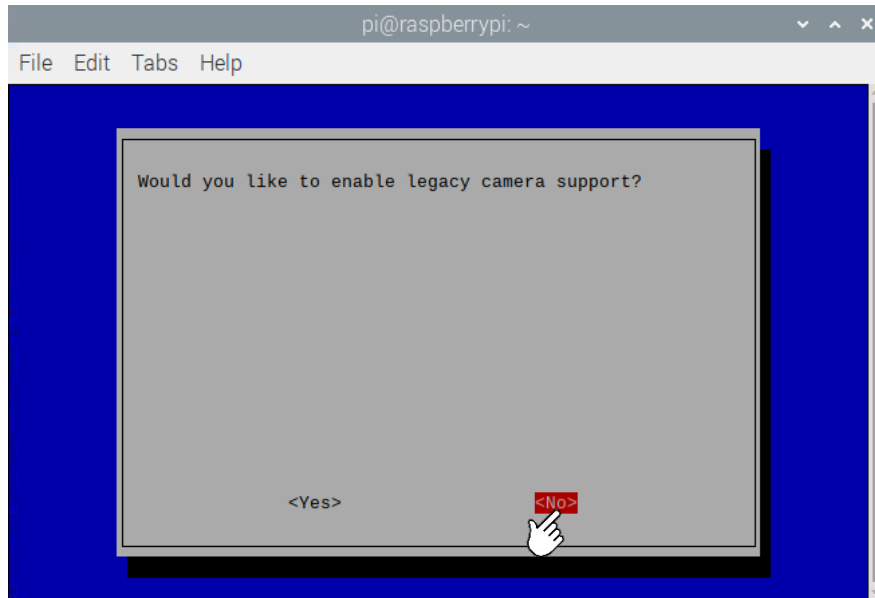
Ingresamos a la terminal e introducimos el siguiente comando: `sudo raspi-config`



Nos aparecerá una sección de configuraciones del Raspberry Pi. Mediante las flechas del teclado seleccionamos la opción “Interface Options”.



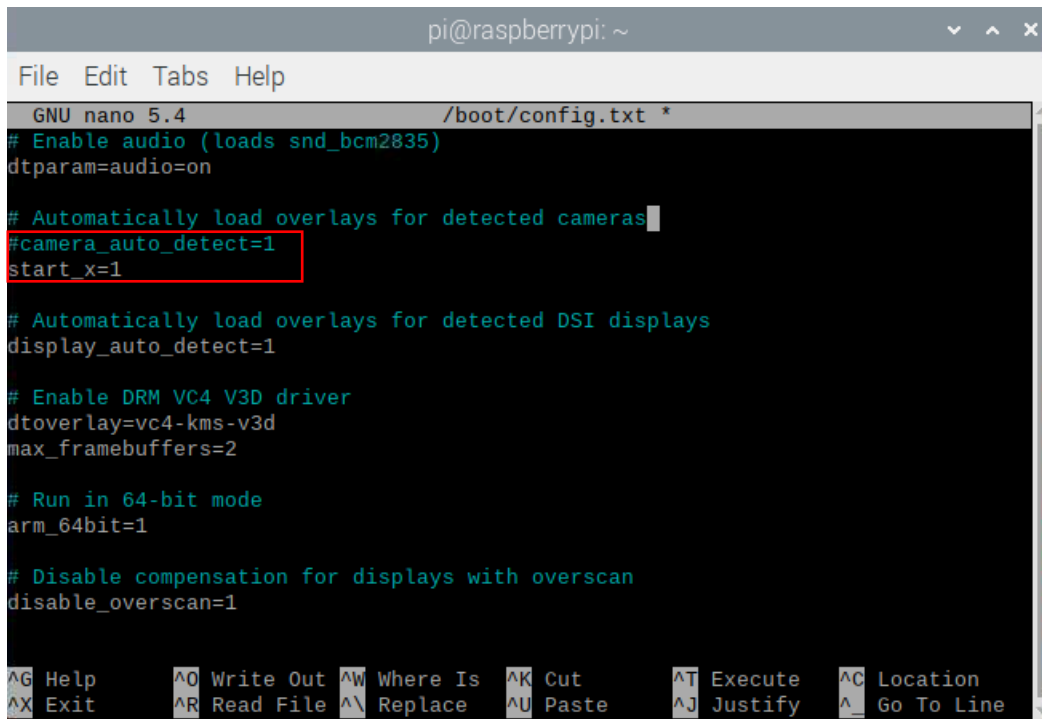
Seleccionamos la primera opción. En caso de pedirnos habilitarlo escogemos la opción “No”, en caso contrario escogemos la opción “Si”.



Al aparecernos el mensaje de confirmación que se ha deshabilitado el soporte (si estaba habilitado) nos salimos de la configuración.

## Habilitar detección usando OpenCV

Nos dirigimos a la terminal e ingresamos el comando **sudo nano /boot/config.txt**



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 5.4 /boot/config.txt *
# Enable audio (loads snd_bcm2835)
dtparam=audio=on

# Automatically load overlays for detected cameras
#camera_auto_detect=1
start_x=1

# Automatically load overlays for detected DSI displays
display_auto_detect=1

# Enable DRM VC4 V3D driver
dtoverlay=vc4-kms-v3d
max_framebuffers=2

# Run in 64-bit mode
arm_64bit=1

# Disable compensation for displays with overscan
disable_overscan=1

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Modificamos el archivo comentando **camera\_auto\_detect** y agregamos la línea **start\_x=1**.

Reiniciamos la Raspberry Pi para guardar los cambios realizados.

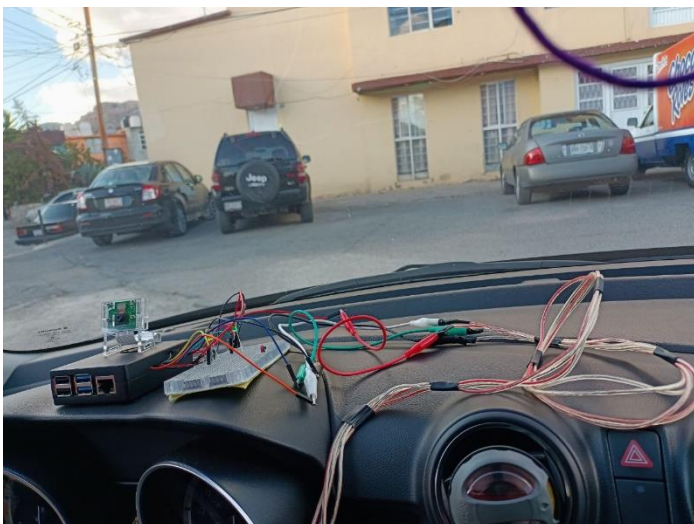


## Instalación de Raspberry en el vehículo

Se localiza un espacio disponible en el vehículo que no obstruya la visión del conductor



Se acopla las conexiones de la raspberry a las del tapete con los caimanes



## Instalar tapete en el vehículo

Se coloca el tapete en el asiento del piloto ajustado con el listón que posee en l parte de atras



Posteriormente se colocan los cables l lado del asiento, sin que le estorben al conductor



Se llevan los cables hasta la parte superior del vehículo



## ***Despliegue del producto***

Para la comparación e que todo funcione correctamente visite el video de YouTube

<https://youtu.be/NCMIfN3ey8Q>