

PUZZLE 1

- Objectiu

L'objectiu d'aquest puzzle és ser capaç de connectar el nostre Raspberry Pi 5 al nostre ordinador i configurar-ho de manera que puguem visualitzar desde el nostre ordinador el seu escriptori. A continuació, després de configurar tot el necessari al nostre Raspberry, haurem de descarregar i utilitzar les llibreries necessàries per poder programar i treballar amb el display que ens hagi deixat prestat el nostre professor, en el meu cas, el lector RFID amb cable USB.

- Configuració Raspberry:

El primer pas que vaig fer va ser instal·lar-me el programa Raspberry Pi Image.



Aquest programa serveix per poder instal·lar el sistema que utilitzarà el Raspberry. Podem configurar també l'usuari, contrasenya de la nostra sessió al raspberry i també el nom al qual s'identifica el nostre Raspberry als altres dispositius. També és molt important activar l'opció SSH al Raspberry.

On guardem aquest sistema és en una tarjeta micro SD on utilitzarem un adaptador USB per gravar-hi dins.

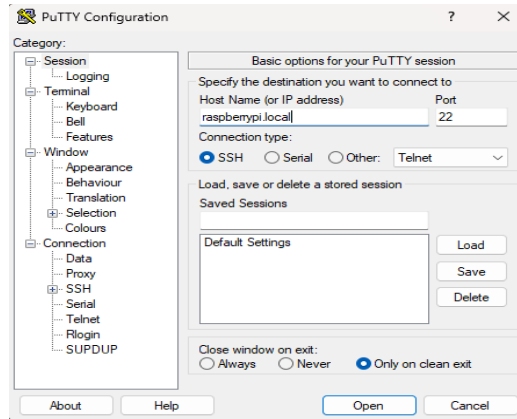
Després d'haver acabat aquest pas, a continuació connectarem el Raspberry i l'ordinador mitjançant Internet.

PROBLEMES:

De primeres, vaig intentar fer aquesta connexió mitjançant WI-FI. Un dels problemes que va aparèixer es que per la configuració del Raspberry hauria de treballar sempre amb la mateixa red WI-FI. Si volgués treballar amb altres reds tenia que tornar a grabar en la tarjeta el sistema configurant desde allà la nova red. També la IP que s'assignava no era fixa, cosa que feia més complicada la connexió.

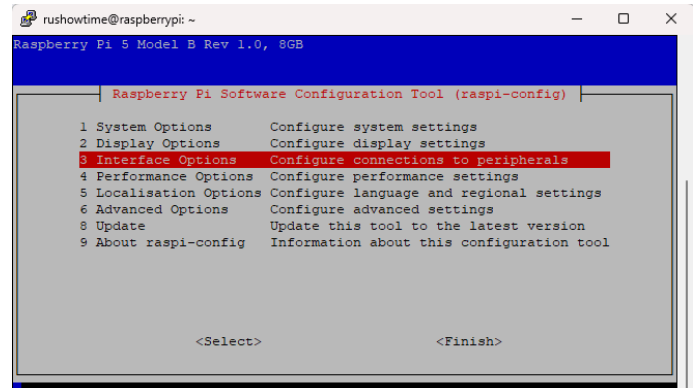
SOLUCIÓ:

Al final em vaig decantar per l'opció de connexió de cable Ethernet, ja que és una connexió directa a l'ordinador i no m'haig de preocupar amb quina red WI-FI estic connectat. Ademés la direcció IP és fixa. També per habilitar aquesta opció hem d'anar a l'apartat de "Ver connexiones" de Windows i en l'apartat de propietats WI-FI activem el permís d'ús compartit de la xarxa.



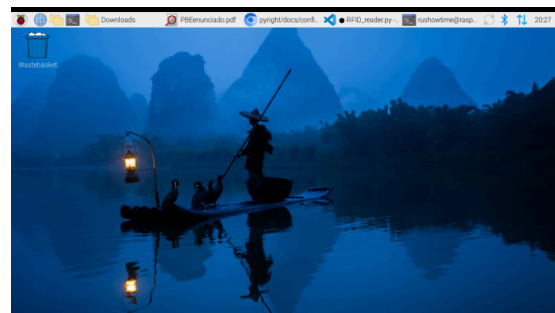
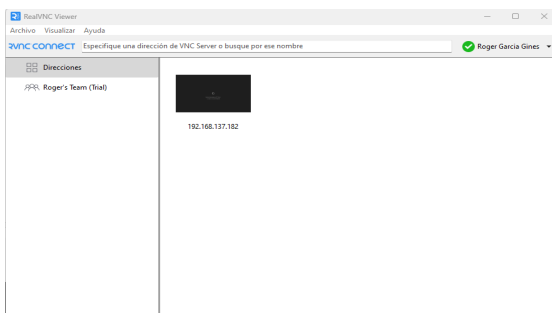
Després de fer connexió via internet hem de trobar quina direcció IP s'ha assignat. Per veure aquesta direcció IP he utilitzat l'aplicació PUTTY que ens permet buscar el nom del dispositiu en el nostre entorn mitjançant l'internet.

A continuació s'obrirà una pestanya on serà el terminal del nostre Raspberry. Iniciem sessió en el nostre Raspberry i utilitzem el comando "**sudo raspi-config**" que ens permetrà entrar a "**Software configuration Tool**" del Raspberry. Desde allà en l'apartat "**Interface Options**" activem la opció "**VNC**".



Després d'aquest pas tornem a la terminal i introduïm el següent comando: "**ifconfig**". Aquest comando ens permetrà saber la direcció IP del Raspberry.

Per últim, per visualitzar el nostre escriptori del nostre Raspberry Pi ens descarquem l'aplicació "**Real VNC Viewer**". Serà l'aplicació que ens permetrà veure visualment l'escriptori. Introduïm l'IP que hem trobat i haurem d'iniciar sessió del Raspberry.



Fent això haurem completat la configuració de connexió del Raspberry.

- Llibreria i codi:

A continuació s'ha de configurar el nostre display. Com que en el meu cas és un display amb cable usb no m'haig de preocupar connexions amb cablejat. Les llibreries que he utilitzat són llibreries fetes ja per Python que he trobat recomanades en "GIT-HUB". Les llibreries són les següents:

```
import serial
import time
import serial.tools.list_ports
```

Amb aquestes llibreries podem controlar els ports USB, tant per escriure com per llegir els diferents displays.

El codi que he utilitzat amb aquestes llibreries és el següent:

FUNCIONS PER LLEGIR:

```
def read_uid(self):
    self.ser.fd=0
    if self.ser.in_waiting > 0:
        #Fica en la variable UID lo que ha enviat el lector
        uid = self.ser.readline().decode('utf-8').strip()
        return uid_hexadecimalstr(uid) # type: ignore
    return None
```

```
def uid_hexadecimalstr(uid):
    #Converteix l'uid llegit en bytes
    uid_bytes = uid.encode('utf-8')
    #Converteix l'uid en bytes a hexadecimal
    uid_hex = uid_bytes.hex()
    return uid_hex
```

Aquestes funcions que he creat serveix per llegir el que envia el port serial. Quan haguem llegit alguna dada, convertim el que hem llegit en hexadecimal.

FUNCIO PER TROBAR EL PORT SERIAL:

```
#Troba el port que esta connectat al dispositiu
def find_port(self):
    ports = serial.tools.list_ports.comports()
    for port in ports:
        #Mostra els ports trobats disponibles
        print(port.device, port.description)
        #Si la descripcio del port coincideix amb el del lector
        if "PCPlay Barcode PCP-BCG4209" in port.description.lower():
            print(port)
            return port.device
```

Amb aquesta funció hauríem de poder trobar de tots els ports serials disponibles quin és el que està connectat amb el nostre display.

MAIN:

```
if __name__ == "__main__":
    rf = Rfid()
    try:
        print("Llegint UID")
        while True:
            #Llegeix el que envia el lector
            uid = rf.read_uid()
            if uid:
                #Escriu el que ha arribat al lector
                print("UID llegit:", uid)
                time.sleep(1) # Espera un segon abans d'acabar la lectura
    except KeyboardInterrupt:
        print("Terminant el programa.")
    finally:
        rf.close()
```

Aquest tros de codi és el nostre main on executa les funcions esmentades anteriorment.

PROBLEMES:

Un dels primers problemes que he trobat ha sigut trobar llibreries, codis o informació sobre el meu display, ja que, tota la informació que trobava era sobre el component R522 que és connecta mitjançant "GPIO" amb cablejat.

L'altre problema que he trobat és que el Raspberry no reconeix o no troba el dispositiu connectat, qual cosa que fa que no es pugui comprovar el codi amb el display perquè no trobo el port serial que és connectat el display o simplement el Raspberry no el reconeix.