

Desafio – Processo Seletivo

Rafaela de Moraes Papale

- Primeira parte – Configuração de rede

Através da leitura do desafio, tem-se a informação de que a *raspberrry* está com a imagem do sistema operacional sem nenhum ajuste. Dessa forma, ela não está configurada com os dados da rede wi-fi e o serviço ssh está desativado. Portanto, é preciso realizar as configurações necessárias de rede para que o projeto possa ser realizado.

O primeiro passo é abrir uma janela de terminal e digitar o seguinte comando:

```
sudo raspi-config
```

Então, é preciso selecionar `advanced options` e, em seguida, navegar até `ssh` e selecionar `Ativar o servidor ssh`.

Após configurar o servidor ssh, é necessário digitar os seguintes comandos no terminal:

```
sudo /etc/network/interfaces
auto lo
iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant.conf

iface default inet dhcp
```

A configuração da rede sem fio é feita através do ficheiro `/etc/network/interfaces/wpa_supplicant.conf` e é necessário adicionar as seguintes informações:

```
country=BR
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="nome da rede"
    psk="senha da rede"
}
```

Depois de realizar as configurações, é necessário gravar e reiniciar a raspberry usando o comando:

```
sudo reboot
```

- Segunda parte – Programando a *raspberry*

O código realizado que atende as especificações propostas no desafio está descrito na abaixo.

```
#Inserção de bibliotecas
import socket
import threading
import Rpi.GPIO as gpio

host = '192.168.0.110' # Endereço IP do servidor
port = # Porta que o servidor está

# Envio
def send(interface):
    while True:
        try:
            #mensagem: leitura do botão
            message = gpio.input(18)
            interface.sendall(message.encode("utf-8"))
        except socket.error:
            break
    print("Envio de mensagens encerrado!")

# Recebimento
def receive(interface):
    while True:
        message = interface.recv(1024)
        if not(len(message)):
            break
        # Se a mensagem recebida for que o botão foi pressionado o led acende
        elif message == 1:
            gpio.output(13, gpio.HIGH)
        # Se a mensagem recebida for que o botão não está pressionado o led permanece apagado
        elif message == 0:
            gpio.output(13, gpio.LOW)
        else:
            gpio.output(13, gpio.LOW)
    print("Conexão com o servidor encerrada!")

def main():
    # Configurações iniciais
    gpio.setmode(gpio.BOARD)
    gpio.setup(13, gpio.OUT)
    gpio.setput(18, gpio.IN)

    # Criar um socket TCP
    interface = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    # Habilitar a reutilização do endereço/porta
    interface.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    # Conectando o socket no servidor
    server_address = (host, port)
    interface.connect(server_address)

    send_thread = threading.Thread(target = send, args = (interface,))
    receive_thread = threading.Thread(target = receive, args = (interface,))
```

- Terceira parte – Circuito elétrico

O hardware a ser desenvolvido consiste em utilizar uma porta da *raspberry* como saída, onde será ligado o led, e uma como entrada, onde será ligado o botão.

Os materiais a serem utilizados para a construção do circuito são:

- 1 LED
- 2 Resistores (1 limitador de corrente para o led e 1 como resistor *pull-down*)
- 1 *push-button*

Montando o circuito, teremos o esquema abaixo:

