

Materiales necesarios:

Jumpers (hembra macho, hembra hembra, macho macho (material de oficina)

Raspberry pi 4 (material de oficina)

Conecotor HDMI (material de oficina)

Monitor (material de oficina y propio)

Teclado de raspberry (material de oficina)

Mouse inalámbrico (material propio)

Pantalla LED (material de oficina)

Protoboard (material de oficina)

Sensor MQ2/sensor de gas (material propio)

Sensor MQ3/sensor de alcohol (material propio)

Sensor DHT11/sensor de temperatura y humedad (material de oficina)

Cubeta de helado (material de casa)

Tubo de plomería (material de casa)

Evidencia del hardware terminado

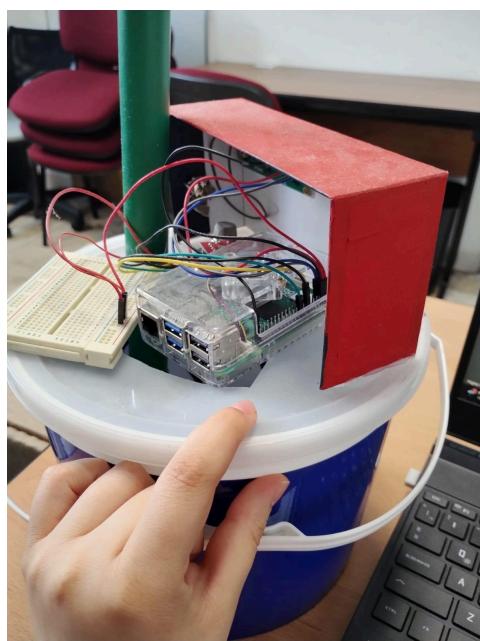
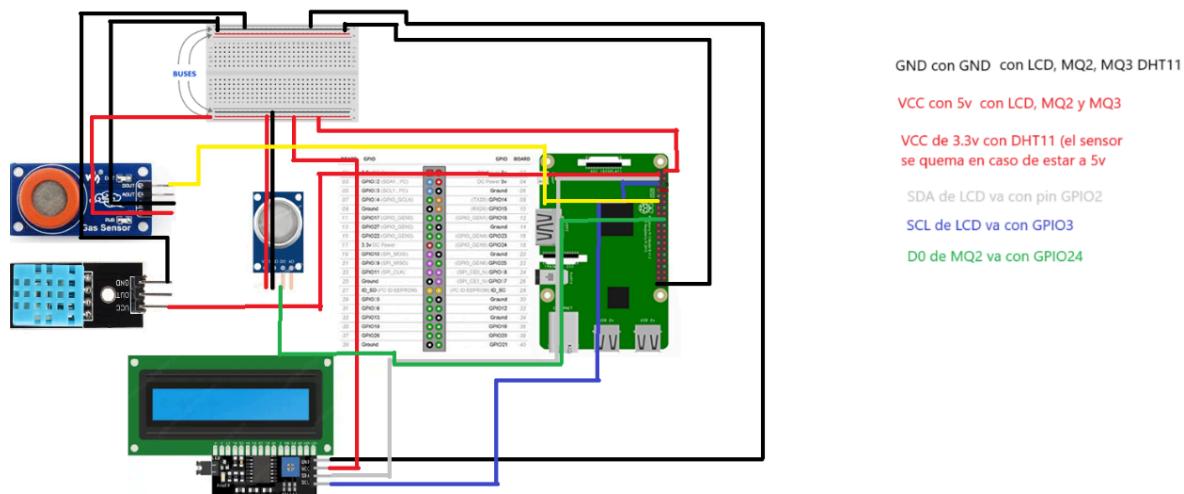


Diagrama de conexión con componentes:



Software utilizado:

- Uso del IDE Thonny instalado en la rasp
- Lenguaje de programación: python
- Funcionamiento para conexión al servidor (en este caso se usó el servidor de U.K)
- Conexión vía Json

Código:

```

import os
import time
import json
import board
import requests
import adafruit_dht
import spidev
import RPi.GPIO as GPIO
from RPLCD.i2c import CharLCD
from datetime import datetime

# --- Configuración WiFi y LCD ---
SSID = "namewifi"
PASSWORD = "passwordwifi"
LCD_ADDRESS = 0x27
URL_SERVIDOR = "linkserver"

# --- Inicialización ---
lcd = CharLCD('PCF8574', LCD_ADDRESS)
dhtDevice = adafruit_dht.DHT11(board.D4)

spi = spidev.SpiDev()

```

```

spi.open(0, 0)
spi.max_speed_hz = 1350000

MQ4_PIN = 24
GPIO.setmode(GPIO.BCM)
GPIO.setup(MQ4_PIN, GPIO.IN)

UMBRAL_ALCOHOL = 400 # Umbral para determinar nivel de alcohol

# --- Funciones ---

def mostrar_en_lcd(linea1="", linea2=""):
    lcd.clear()
    lcd.write_string(linea1[:16])
    lcd.cursor_pos = (1, 0)
    lcd.write_string(linea2[:16])

def conectar_wifi(ssid, password):
    mostrar_en_lcd("Conectando a", ssid)
    print(f"Conectando a WiFi {ssid}...")
    resultado = os.system(f"nmcli dev wifi connect '{ssid}' password '{password}'")
    if resultado == 0:
        mostrar_en_lcd("WiFi Conectado", ssid)
        print("WiFi conectado correctamente.")
        return True
    else:
        mostrar_en_lcd("Sin conexión", "WiFi falló")
        print("No se pudo conectar a WiFi.")
        return False

def leer_mq3(channel=0):
    adc = spi.xfer2([1, (8 + channel) << 4, 0])
    data = ((adc[1] & 3) << 8) + adc[2]
    return data

def leer_dht():
    try:
        temp = dhtDevice.temperature
        hum = dhtDevice.humidity
        print(f"DHT11 - T:{temp}C H:{hum}%")
        return temp, hum
    except RuntimeError as e:
        print(f"Error DHT11: {e}")
        return None, None

def leer_mq4():
    estado = GPIO.input(MQ4_PIN)
    gas_texto = "Gas Detectado" if estado == 0 else "Aire Limpio"

```

```

print(f"MQ4 - Estado: {gas_texto}")
return estado == 0, gas_texto

def enviar_datos(url, datos):
    try:
        headers = {'Content-Type': 'application/json'}
        r = requests.post(url, data=json.dumps(datos), headers=headers)
        print("Código HTTP:", r.status_code)
        if r.status_code == 200:
            mostrar_en_lcd("Enviado OK", "Servidor resp.")
            return True
        else:
            mostrar_en_lcd("Error Envío", f"HTTP {r.status_code}")
            return False
    except Exception as e:
        mostrar_en_lcd("Error Envío", "Exception")
        print(f"Excepción: {e}")
        return False

# --- Programa principal ---
def main():
    if not conectar_wifi(SSID, PASSWORD):
        return

    try:
        while True:
            temp, hum = leer_dht()
            valor_alcohol = leer_mq3()
            gas_detectado, estado_gas = leer_mq4()

            nivel_alcohol = "ALTO" if valor_alcohol > UMBRAL_ALCOHOL else "BAJO"
            print(f"MQ3 - Nivel de alcohol: {nivel_alcohol}")
            print(f"Gas metano: {estado_gas}")

            # Mostrar los datos en la LCD uno por uno con pausas
            if temp is not None and hum is not None:
                mostrar_en_lcd(f"Temp: {temp} C", f"Humedad: {hum} %")
                time.sleep(5)

                mostrar_en_lcd("Alcohol:", f"Nivel: {nivel_alcohol}")
                time.sleep(5)

                mostrar_en_lcd("Gas Metano:", estado_gas)
                time.sleep(5)

            # Obtener la hora actual en formato ISO 8601
            hora_actual = datetime.now().isoformat()

```

```
# Enviar datos al servidor
datos = {
    "temperatura": temp,
    "humedad": hum,
    "nivel_alcohol": nivel_alcohol,
    "gas_metano": estado_gas,
    "dispositivo": "RasPi01",
    "hora": hora_actual
}
enviar_datos(URL_SERVIDOR, datos)
else:
    mostrar_en_lcd("Error lectura", "DHT11")

except KeyboardInterrupt:
    print("Saliendo...")
    lcd.clear()
    lcd.write_string("Apagando...")
    spi.close()
    GPIO.cleanup()

if __name__ == "__main__":
    main()
```