

**UNIVERSIDAD DE SAN CARLOS**

**FACULTAD DE INGENIERIA**

**ESCUELA DE SISTEMAS**



## **Manual Técnico**

### ***Proyecto 2, Grupo # 1***

Arquitectura de Computadoras y Ensambladores I

Sección S

Ing. Luis Espino

## **Introducción**

Este proyecto tiene como objetivo desarrollar una estación meteorológica utilizando Raspberry Pi y diversos sensores especializados en la medición de luz, humedad y temperatura. En respuesta al desafío global del cambio climático, caracterizado por el aumento de la temperatura, cambios en los patrones de precipitación y eventos climáticos extremos más frecuentes e intensos, esta estación será fundamental para monitorear de manera continua y precisa las condiciones meteorológicas locales.

La implementación de esta estación no solo busca diagnosticar el clima de forma efectiva, sino también proporcionar datos cruciales para la investigación climática. Además, servirá como herramienta invaluable para la gestión de recursos naturales, permitiendo a los investigadores, administradores y comunidades locales entender mejor los efectos del cambio climático y tomar medidas mitigadoras apropiadas.

Al integrar tecnología accesible como Raspberry Pi y sensores especializados, este proyecto no solo promueve la innovación en la monitorización ambiental, sino que también contribuye significativamente a la protección y resiliencia de las comunidades frente a eventos climáticos adversos.

## **Marco teórico**

### **Fotorresistencias**

Las fotorresistencias, también conocidas como LDR (del inglés Light Dependent Resistor), son componentes que varían su resistencia eléctrica en función de la intensidad de luz incidente sobre ellas. Este cambio en resistencia se debe a la capacidad del material semiconductor que la compone para conducir electricidad de manera más efectiva bajo la influencia de la luz. Se utilizan en aplicaciones como fotodetectores y en circuitos que requieren control automático basado en niveles de luz ambiental.

### **Sensores de Humedad**

Los sensores de humedad detectan y cuantifican la cantidad de vapor de agua presente en el aire u otros medios. Utilizan principios capacitivos o resistivos para medir la humedad relativa, proporcionando una señal eléctrica que varía proporcionalmente con la humedad ambiental. Estos sensores son esenciales en aplicaciones de control ambiental, agricultura de precisión, y climatización, entre otros.

### **Sensores de Temperatura**

Los sensores de temperatura son dispositivos que convierten cambios de temperatura en una señal eléctrica, generalmente una resistencia, voltaje o corriente que varía con la temperatura. Utilizan termistores, termopares o semiconductores para medir temperaturas en rangos específicos, y son fundamentales en aplicaciones industriales, electrónicas y de control climático.

### **Sensores de Humo**

Los sensores de humo detectan partículas de humo en el aire y activan una alarma o dispositivo de seguridad cuando se supera un umbral crítico. Pueden basarse en tecnologías ópticas, iónicas o fotoeléctricas, detectando la presencia de humo antes de que se convierta en un

incendio completo. Estos sensores son cruciales para la seguridad en viviendas, edificios comerciales y vehículos.

### **Sensor de Velocidad de Viento**

Los sensores de velocidad de viento miden la velocidad del aire en unidades como metros por segundo o kilómetros por hora. Utilizan tecnologías como anemómetros ultrasónicos, de copa o de hélice para registrar la velocidad y a veces la dirección del viento. Estos sensores son vitales en aplicaciones meteorológicas, estudios de aerodinámica, y control de turbinas eólicas.

### **Manejo de Arreglos en ARMx64**

El manejo de arreglos en arquitecturas ARMx64 implica la gestión eficiente de memoria y procesamiento paralelo mediante instrucciones SIMD (Single Instruction, Multiple Data). Estas instrucciones permiten operaciones simultáneas en múltiples elementos de datos dentro de arreglos, optimizando el rendimiento en aplicaciones de procesamiento intensivo como la manipulación de señales, imágenes y datos científicos.

### **Desviación Estándar**

La desviación estándar es una medida de dispersión que indica cuánto varían los datos respecto a la media. Se calcula como la raíz cuadrada de la varianza y se utiliza para evaluar la uniformidad o la variabilidad de un conjunto de datos.

### **Moda**

La moda es el valor que aparece con mayor frecuencia en un conjunto de datos. Es útil para identificar el valor más típico o común en una distribución de datos.

### **Mediana**

La mediana es el valor que ocupa la posición central en un conjunto de datos ordenado. Es menos sensible a valores extremos que la media y proporciona una medida robusta de la ubicación central de los datos.

### **Media**

La media, o promedio aritmético, es la suma de todos los valores en un conjunto de datos dividida por el número de elementos. Es la medida de tendencia central más comúnmente utilizada y puede verse afectada por valores atípicos en los datos.

Estos conceptos y herramientas son fundamentales en el análisis y la interpretación de datos obtenidos de sensores y otras fuentes de información en diversas aplicaciones científicas, tecnológicas e industriales.

**Material gastado**

<b>Descripción</b>	<b>Precio</b>
Sensor de gas MQ2	Q.35.00
Sensor de temperatura y humedad DHT11	Q.25.00
Sensor de velocidad	Q.30.00
Sensor de luminosidad	Q.15.00
Barómetro	Q.26.00
Total	Q.131.00

## Código ARM

Figura 1: Contador

```

1  .global _start
2  .bss
3  input:
4      .space 100
5  .text
6  _start:
7      mov x0, 0
8      ldr x1, =input
9      mov x2, 100
10     mov x8, 63
11     svc 0
12     ldr x0, =input
13     mov x3, 1
14     bucle_contar:
15         ldrb w1, [x0], 1
16         cmp w1, 0
17         beq fin
18         cmp w1, 44
19         beq incrementar
20         b bucle_contar
21     incrementar:
22         add x3, x3, 1
23         b bucle_contar
24     fin:
25         mov x0, x3
26         mov x8, 93
27         svc 0
28

```

---

Fuente: Elaboración propia

**Figura 2: Cálculo de desviación estándar en ASM**

```

_start:
    mov x0, #-100
    ldr x1, =input_file
    mov x2, #0
    mov x8, #56
    svc #0
    cbz x0, error
    mov x9, x0

    mov x0, x9
    ldr x1, =buffer
    mov x2, #1024
    mov x8, #63
    svc #0
    cbz x0, error
    mov x19, x0

    mov x20, #0
    ldr x21, =numbers
    mov x22, #0
    ldr x1, =buffer

parse_loop:
    mov x0, x1
    bl atoi

    cmp x0, #-1
    beq parse_end

    str w0, [x21, x22, lsl #2]
    add x22, x22, #1
    add x20, x20, x0

```

**Fuente: Elaboración propia**



**Figura 3: Cálculo de máximo en la lista en ASM**

```

_start:
    mov x0, 0
    ldr x1, =input
    mov x2, 1000
    mov x8, 63
    svc 0
    mov x3, 0
    mov x4, 0
    ldr x0, =input

bucle_lectura:
    ldrb w1, [x0], 1
    cmp w1, 44
    beq procesar_numero
    cmp w1, 10
    beq fin
    sub w1, w1, 48
    mov x2, x3
    mov x3, 10
    mul x2, x2, x3
    add x3, x2, x1
    b bucle_lectura

procesar_numero:
    cmp x3, x4
    bgt actualizar_mayor
    mov x3, 0
    b bucle_lectura

actualizar_mayor:
    mov x4, x3
    mov x3, 0
    b bucle_lectura

```

**Fuente: Elaboración propia**

**Figura 4: Cálculo de mínimo en la lista en ASM**

```

7      _start:
8          mov x0, 0
9          ldr x1, =input
10         mov x2, 100
11         mov x8, 63
12         svc 0
13
14         mov x3, 0
15         mov x4, 0
16         mov x5, 0x7FFFFFFFFFFFFFFF
17         ldr x0, =input
18
19     bucle_lectura:
20         ldrb w1, [x0], 1
21         cmp w1, 44
22         beq procesar_numero
23         cmp w1, 10
24         beq fin
25         sub w1, w1, 48
26         mov w2, w4
27         mov w4, 10
28         mul w2, w2, w4
29         add w4, w2, w1
30         b bucle_lectura
31
32     procesar_numero:
33         cmp x4, x5
34         blt actualizar_menor
35         mov x4, 0
36         b bucle_lectura

```

**Fuente: Elaboración propia**

**Figura 5: Cálculo de la mediana en ASM**

```

7      _start:
8          mov x0, 0
9          ldr x1, =input
10         mov x2, 100
11         mov x8, 63
12         svc 0
13
14         mov x4, 0
15         mov x5, 0
16         ldr x0, =input
17         ldr x1, =numeros
18
19     leer_numeros:
20         ldrb w2, [x0], 1
21         cmp w2, 44
22         beq guardar_numero
23         cmp w2, 10
24         beq ordenar
25         sub w2, w2, 48
26         mov w3, w4
27         mov w4, 10
28         mul w3, w3, w4
29         add w4, w3, w2
30         b leer_numeros
31
32     guardar_numero:
33         str w4, [x1], 4
34         mov w4, 0
35         add x5, x5, 1
36         b leer_numeros
37
38     ordenar:
39         str w4, [x1], 4
40         add x5, x5, 1
41         mov x6, 0
42
43     bucle externo:

```

**Fuente: Elaboración propia**

**Figura 6: Cálculo de la moda en ASM**

```

_start:
mov x0, 0
ldr x1, =input
mov x2, 100
mov x8, 63
svc 0
ldr x0, =frecuencias
mov x1, 0
mov x2, 100
inicializar_frecuencias:
str xzr, [x0], 8
add x1, x1, 1
cmp x1, x2
bne inicializar_frecuencias
mov x4, 0
ldr x0, =input
contar_frecuencias:
ldrb w1, [x0], 1
cmp w1, 44
beq sumar_frecuencia
cmp w1, 10
beq encontrar_moda
sub w1, w1, 48
mov w2, w4
mov w4, 10
mul w2, w2, w4
add w4, w2, w1
b contar_frecuencias
sumar_frecuencia:
ldr x1, =frecuencias
lsl x2, x4, 3
add x1, x1, x2
ldr x2, [x1]
add x2, x2, 1

```

**Fuente: Elaboración propia**

**Figura 7: Cálculo de promedio**


---

```

14  _start:
15      mov x0, #-100
16      ldr x1, =input_file
17      mov x2, #0
18      mov x8, #56
19      svc #0
20      cbz x0, error
21      mov x9, x0
22
23      mov x0, x9
24      ldr x1, =buffer
25      mov x2, #1024
26      mov x8, #63
27      svc #0
28      cbz x0, error
29      mov x19, x0
30
31      mov x20, #0
32      ldr x21, =numbers
33      mov x22, #0
34      ldr x1, =buffer
35
36  parse_loop:
37      mov x0, x1
38      bl atoi
39
40      cmp x0, #-1
41      beq parse_end
42
43      str w0, [x21, x22, lsl #2]
44      add x22, x22, #1
45      add x20, x20, x0
46
47  parse_next:
48      ldrb w2, [x1], #1
49      cmp w2, #','
50      b eq parse_loop

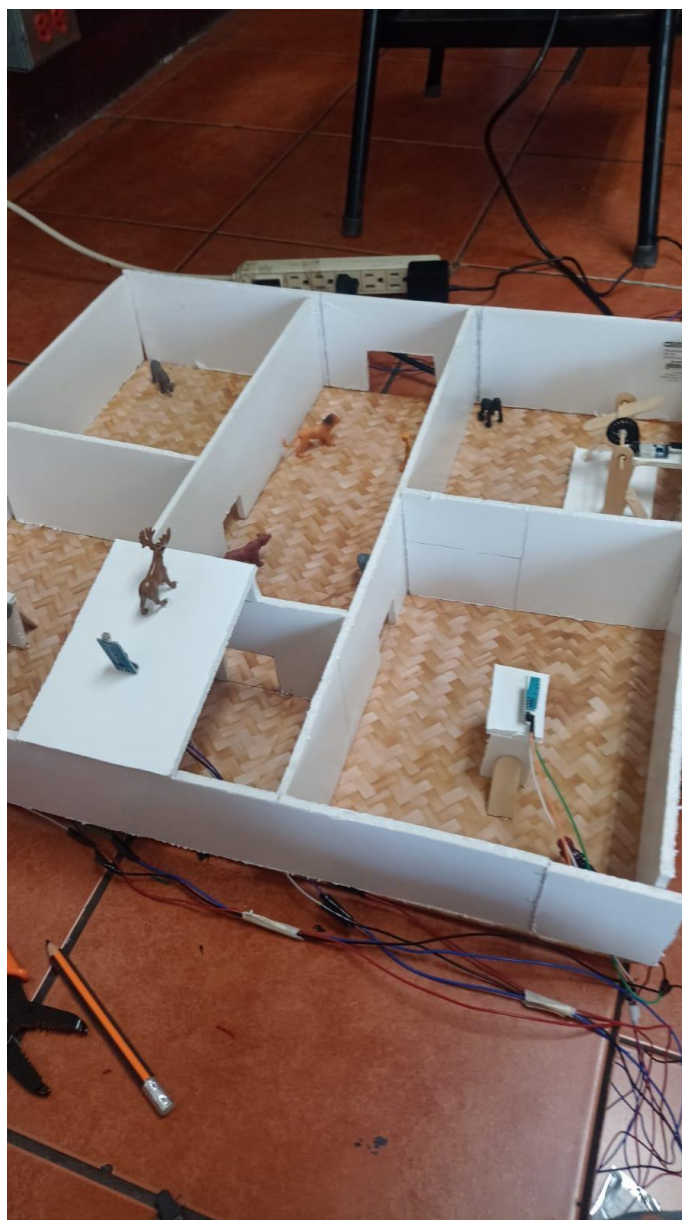
```

---

**Fuente: Elaboración propia**

## Anexos

**Figura 8: Maqueta terminada y sensores**



**Fuente: Elaboración propia**

## **Bibliografía**

[Estadística básica: Media, mediana y moda \(gcfglobal.org\)](https://www.gcfglobal.org/es/curso/estadistica-basica-media-mediana-y-moda)

[Raspberry Pi Documentation](https://www.raspberrypi.org/documentation/)

[Documentation – Arm Developer](https://developer.arm.com/documentation/)