

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

LABORATORIO DE ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES 1

Manual técnico

NOMBRE COMPLETO	CARNÉ
Henry David Quel Santos	202004071
Pablo Alejandro Marroquin Cutz	202200214
Eric David Rojas de León	202200331
David Isaac García Mejía	202202077
Jorge Alejandro De León Batres	202111277
Roberto Miguel García Santizo	202201724
Jose Javier Bonilla Salazar	202200035
Gerardo Leonel Ortiz Tobar	202200196

Introducción

Este manual técnico describe el desarrollo y funcionamiento del sistema de monitoreo y riego automatizado implementado en la Fase 3 del proyecto. Se utilizó lenguaje ensamblador ARM64 para el análisis de datos y Python para la exportación de los resultados a un archivo CSV. Este documento incluye la configuración de hardware y software, las principales funcionalidades y el flujo del código.

Requisitos

Hardware:

- Raspberry Pi (modelos 3B+ o superiores con arquitectura ARM64).
- Sensores:
 - DHT11 para temperatura y humedad relativa.
 - BMP280 para presión barométrica.
 - Anemómetro para medir velocidad del viento.

Software:

- Lenguaje de programación: Python y ensamblador ARM64.
- Framework Flask para el backend.
- Base de datos: MongoDB.

Estructura del Proyecto

Descripción General:

El sistema recopila datos meteorológicos como temperatura, humedad y velocidad del viento mediante sensores conectados a una Raspberry Pi. Estos datos se almacenan en MongoDB y se analizan utilizando un programa en ensamblador y Python que calcula estadísticas básicas y las compara.

Dependencias:

Hardware:

1. Raspberry Pi con pines GPIO habilitados.
2. Sensor DHT11.
3. Sensor BMP280.
4. Anemómetro conectado al pin GPIO24.

Software:

1. Sistema operativo Raspbian o similar.
2. Bibliotecas de Python:

adafruit-circuitpython-dht

adafruit-circuitpython-bmp280

RPi.GPIO

Flask

pymongo

python-weather

Conexiones Físicas:

1. Conectar el DHT11 al pin GPIO4.
2. Conectar el BMP280 a los pines I2C (SDA al pin 3 y SCL al pin 5).
3. Conectar el anemómetro al pin GPIO24.

Funcionamiento del Sistema

Inicialización:

- Se configuran los sensores y los pines GPIO de la Raspberry Pi.
- El servidor Flask se ejecuta en el puerto 5000, esperando solicitudes POST para almacenar los datos en MongoDB.

Captura de Datos:

1. Temperatura y Humedad:

- Se mide con el sensor DHT11.

2. Presión Barométrica:

- Se mide con el sensor BMP280.

3. Velocidad del Viento:

- Se mide utilizando el anemómetro, contando sus revoluciones por segundo (RPS).

Envío de Datos al Servidor:

Los datos se empaquetan en formato JSON y se envían al servidor Flask mediante una solicitud POST.

Ejemplo de JSON enviado:

```
{  
  "temperatura": 25,  
  "humedadRelativa": 65,  
  "presionBarometrica": 1013,  
  "velocidadViento": 3.5  
}
```

Funciones del Código en Ensamblador:

1. **Menú Principal:** Presenta al usuario las opciones para mostrar los integrantes, realizar el análisis estadístico, generar archivos TXT y salir del programa.
2. **Mostrar Integrantes:** Muestra el nombre completo y carnet de los integrantes del grupo.
3. **Cálculo de Promedios:** La función `calculate_averages` suma los valores registrados y los divide por 5 para obtener los promedios.
4. **Conversión de Números a Texto:** La función `double_to_string` convierte los valores en punto flotante a cadenas de texto para su exportación.
5. **Escritura de Archivos:** La función `write_to_file` crea un archivo de salida y escribe los promedios calculados.
6. **Generación de TXT:** Se generan tres archivos TXT:
 - `promedios.txt` con los promedios de los parámetros.
 - `maxmin.txt` con los valores máximos y mínimos.
 - `medianas.txt` con las medianas de los parámetros.

Cálculos Clave

1. **Promedio:** Se calcula la media de los parámetros medidos (temperatura, humedad, presión, velocidad del viento).
2. **Moda:** Se determina el valor que más se repite en cada parámetro.
3. **Valor Mínimo y Máximo:** Se identifican los valores extremos de cada parámetro.
4. **Rango de Temperatura:** Se calcula la diferencia entre las temperaturas interna y externa más alta y más baja.

Exportación de Datos

El sistema permite exportar los datos obtenidos en dos formatos:

1. **CSV:** Generado mediante la ruta /generar_csv.
2. **TXT:** Exportado por el programa en ensamblador tras realizar el análisis estadístico.

Anexos

Estadísticas

Archivo CSV

Subir CSV

Contenido de archivo CSV

Archivo TXT

Subir TXT

Contenido de archivo TXT

Tabla de Medidas

Medidas	Python	Assembly
Moda	12	10
Promedio	20	19
Máximo	12	12
Mínimo	1	1
Rango	12	12