

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Arquitectura de Computadores y Ensambladores 2
Catedrático: Ing. Gabriel Díaz
Auxiliares: Daniel Monterroso
Guillermo Orellana
Diego Martinez



Práctica 1

Estación Meteorológica IOT

Objetivos:

- Diseñar un dispositivo destinado a medir y registrar regularmente, diversas variables meteorológicas.
- Implementar una aplicación en Processing que permita visualizar mediciones y observaciones puntuales de los diferentes parámetros meteorológicos.
- Aprender a desarrollar una manera correcta de visualización de datos mediante la implementación del framework de lot

Descripción:

La temperatura global en los últimos tres años ha sido influenciada por el efecto de una prolongada “la niña”, efecto que provoca un enfriamiento temporal en la temperatura global, dicho efecto está terminando y la tierra está retornando a condiciones relativamente más cálidas que en 2022. En relación con las actividades diarias del ser humano puede que este cambio afecte la salud de las personas y prácticas de agricultura. Por lo que sabiendo el futuro que nos espera existe la necesidad de saber las condiciones meteorológicas con un dispositivo meteorológico inteligente para pronosticar y tomar las debidas precauciones.

Las estaciones meteorológicas son instalaciones permiten monitorear y realizar acciones preventivas en cuanto a las variaciones de los factores como la temperatura, la lluvia, la velocidad del viento, entre otras.

Como ya es de costumbre hoy día, los datos generados y almacenados por cualquier dispositivo deben poder ser monitoreados, visibles y de fácil comprensión para cualquier tipo de usuario, desde el más experto hasta el más inexperto, por ello se le solicita también integrar una interfaz que permita interpretar las lecturas de una forma gráfica y animada.

El diseño del dispositivo para su implementación e integración de los módulos, sensores y placa de microcontrolador queda a discreción del equipo desarrollador, ***pero debe poder soportar la intemperie al 100%, independientemente del clima ya que estará en el exterior.***

Funciones:

Al finalizar el proyecto se dispondrá de la siguiente información:

- **Temperatura externa**

El dispositivo permitirá conocer la temperatura actual del ambiente, lectura de las medidas de temperatura en diversas horas del día con la unidad de medida grados centígrados (°C)

- **Humedad relativa**

El dispositivo debe ser capaz de medir y registrar la cantidad de vapor de agua en el aire, mostrando la humedad relativa (°C), humedad absoluta (Gr/m³) y punto de rocío (Gr/Kg) del ambiente.

- **Velocidad del viento:**

El dispositivo debe ser capaz de medir y registrar la velocidad del viento al momento de colocarlo en el exterior mostrando la velocidad del viento expresada en Kilómetros por hora (km/h).

- **Dirección del viento:**

El dispositivo debe ser capaz de calcular las cuatro direcciones básicas: norte (N), sur (S), este (E) y oeste (O), así como sus variables cardinales.

- **Presión barométrica:**

El dispositivo tiene que ser capaz de almacenar la medida de la presión que la atmósfera ejerce sobre la tierra. Esta variable de presión cambia con las condiciones climáticas y es útil para pronosticar el tiempo.

Interfaz de comunicación con el humano (HMI):

La principal función de Estación Meteorológica IOT es proporcionar al usuario mediciones en tiempo real de las variables meteorológicas así como el comportamiento de dichas variables a lo largo de un periodo de tiempo.

Para la medición en tiempo real, debe desarrollar una aplicación en processing que muestre en una aplicación tipo [dashboard \(como este ejemplo\)](#) las distintas mediciones recolectadas por el dispositivo IoT, la representación de los datos medidos queda a discreción del desarrollador, pero es importante tomar en cuenta que todas la magnitudes deben ser visualizadas y evaluadas en la misma representación.

- Temperatura del ambiente(°C).
- Humedad relativa (°C)
- Humedad absoluta (g/m^3)
- Velocidad del viento (km/h).
- Dirección del viento (Puntos cardinales)
- Presión barométrica (mmHg)

Para el comportamiento de las variables meteorológicas a lo largo del tiempo debe desarrollar una aplicación en processing que muestre en una aplicación las gráficas relativas a cada uno de las 6 magnitudes representadas en el dashboard, para acceder a la [gráfica \(como en este ejemplo\)](#) debe dar click en el elemento de dashboard que la representa y mostrar

- Temperatura ambiente a lo largo del tiempo.
- Humedad relativa a lo largo del tiempo
- Humedad absoluta a lo largo del tiempo
- Velocidad del viento a lo largo del tiempo
- Presión barométrica a lo largo del tiempo
- Gráfica radial de las direcciones del viento.

Anemómetro:

Para las variables relacionadas con el viento se sugiere crear su propio [anemómetro como se muestra en este ejemplo](#)

Conectividad:

El dispositivo estará compuesto por un arduino que recolecta la información de los sensores del medio donde se coloque el dispositivo, y enviados mediante un cable que estará conectado a una computadora, esta información deberá almacenarse en una base de datos transaccional **local o en la nube** y deberán ser persistentes y a su vez procesados y desplegados en tiempo real mediante la aplicación desarrollada en processing.

Los datos que serán enviados por medio de la API son:

- Temperatura del ambiente(°C).
- Humedad relativa (°C)
- Humedad absoluta (Gr/m³)
- Velocidad del viento (K/h).
- Dirección del viento (Puntos cardinales)
- Presión barométrica (mmHg)

Experimentos a calificar:

- **Temperatura externa**

Debe realizar un experimento en interiores recreando un ambiente en el que aumente la temperatura ambiente de forma considerable para que el sensor detecte el cambio, no se permite tocar con los dedos o acercar llamas al sensor.

- **Humedad relativa**

Debe realizar un experimento en interiores recreando un ambiente en el que aumente la humedad ambiente de forma considerable para que el sensor detecte el cambio, no se permite utilizar agua directamente sobre el sensor.

- **Velocidad del viento:**

Debe realizar un experimento en interiores recreando un ambiente de viento en el que la velocidad aumente y disminuya.

- **Dirección del viento:**

Debe realizar un experimento en interiores recreando un ambiente de viento en el que la dirección del viento provenga de distintos ángulos.

- **Intemperie**

Realizar un video donde el dispositivo se encuentre en el exterior recolectando datos y el dashboard a un lado mostrando las métricas y reportes.

Repositorio de GitHub:

Todo el código utilizado y la documentación deberá ser subido a un repositorio de github y al momento de la entrega solo se mandara la documentación la cual deberá contener el link del repositorio, esto con el fin de evitar inconvenientes por el tamaño de los archivos al momento de la entrega, para la creación de dicho repositorio tomar en cuenta las siguientes indicaciones:

- **Nombre del repositorio:** **ACE2_1S23_G#GRUPO**, ejemplo **ACE2_1S23_G12**
- Agregar al usuario del auxiliar como colaborador a su repositorio de github:
 - **Grupos 0-7:** **Diamg502**
 - **Grupos 8-11, 16-19:** **dmonterroso1**
 - **Grupos 12-15, 20-24:** **Guillermo-O-C**
- Hacer por lo menos 1 commit por semana durante el desarrollo.
- Todo código o documento que no se encuentre en el repositorio no será tomado en cuenta para la calificación.

Contenido obligatorio del repositorio:

- Código de Arduino utilizado.
- Código de la aplicación en processing
- Script utilizados en la base de datos
- Todo el código utilizado para la implementación del servidor local o nube
- Fotos del prototipo final
- Documentación completa.

Estructura del repositorio:

Debido a que se usará el mismo repositorio durante todo el semestre se solicita que este contenga en su raíz únicamente 4 carpetas dentro de las cuales se almacenará todo lo referente a cada práctica y proyecto conforme se vayan desarrollando, los nombres de las carpetas serán:

- Practica 1
- Practica 2
- Proyecto 1
- Proyecto 2

Además en el README del repositorio deberán de colocar el número de grupo y los datos de sus integrantes.

Restricciones:

- La práctica se deberá realizar en grupos no mayor a 5 integrantes.
- El grupo debe contar con un coordinador de grupo
- La aplicación deberá estar realizada en procesamiento y el lenguaje utilizado para el envío de la información hacia la base de datos queda a discreción del grupo de trabajo.
- Todo deberá de ir implementado en un único dispositivo, no se permitirán varios.
- Se deberá implementar un servidor local o nube para almacenar y analizar los datos.
- Se deberán respetar los roles definidos para cada estudiante durante el desarrollo.
- **Durante la calificación se validará que se tengan como mínimo 3 días de datos recolectados por el dispositivo.**

Documentación:

En la documentacion debiera de llevar todo lo correspondiente al desarrollo tomando como base el Framework de IoT, dicho lo anterior se solicita:

- Introducción
- Bocetos del prototipo
- Imagenes de construcción del prototipo
- Pantallas de la aplicación web
- Capas del framework de IoT.
- Link del repositorio de github.

Consideraciones:

- Se calificará solamente lo que sea completamente funcional.
- La comunicación entre el dispositivo, la aplicación y el servidor deberá de estar implementada y funcional.
- Se deberán de mandar todos los entregables en la fecha establecida, de no ser así se tendrá una penalización del 50%.
- **Fecha de entrega: 17 de febrero de 2023 antes de las 23:59**
- El archivo de la documentación deberá de ser entregado en la plataforma de UEDi en el área destinada para ello, únicamente 1 integrante del grupo deberá de realizar la entrega.
- También se deberá enviar la documentación por correo electrónico a las siguientes direcciones según el número de grupo con el asunto **[ACE2]Practica1_G#GRUPO** como medida de precaución en caso de problemas con la plataforma UEDi:

-
- **Grupos 0-7:** 3042499630114@ingenieria.usac.edu.gt
- **Grupos 8-11, 16-19:** 3001403550101@ingenieria.usac.edu.g
- **Grupos 12-15, 20-24:** 2994293770101@ingenieria.usac.edu.gt