Reto capa de aplicación: Interfaz

**Equipo de Trabajo 01 – Pareja 2**

*Jhon Jairo Moreno Trujillo*

*Jonathan Andrés Verdugo Romero*

1. **Modificar el código dado en el tutorial para el procesamiento de un nuevo evento.**
2. **¿Qué buenas prácticas de diseño de interfaz observa en el tablero REMA?**

comportamiento

1. **¿Hay alguna mala práctica en el diseño de interfaz? ¿Cuál? ¿Cómo la corregiría?**
2. **¿Qué ventajas/desventajas sobre las consultas trae consigo el uso del patrón Blob en el almacenamiento?**
3. **¿Qué ventajas/desventajas tiene el uso de un patrón de actualización tipo “pull” como Grafana?**
4. **¿Qué impacto tiene la frecuencia de muestreo de los dispositivos IoT sobre la interfaz gráfica?**
5. **En el tutorial ¿cómo se integra Grafana con el resto de las capas del sistema IoT? ¿A través de APIs o fuentes de datos?**

Condición: Se observa el comportamiento y la variación de una variable como humedad o temperatura durante la última hora con el fin que se encuentre dentro de los límites permitidos.

Acción: Encender LED cerca al sensor para notificar del evento.

Se hace referencia a la variación estándar de las mediciones, para lo cual utilizamos las funciones estadísticas *built-in* de python.

Una variación por encima de los rangos en un instante de tiempo indica que los datos se extienden sobre un rango de valores amplio, en nuestro caso, es de interés conocer estos eventos ya que puede significar errores de calibración o medición en los sensores.

Por otro lado, si hay cambios repentinos en la temperatura o humedad también se observaría una variación elevada, es por esto que se configura una alerta para que se ejecute únicamente cuando la variación estándar sea mayor a 5.

Del lado de la aplicación, se utilizó la interfaz para registrar las nuevas variables:

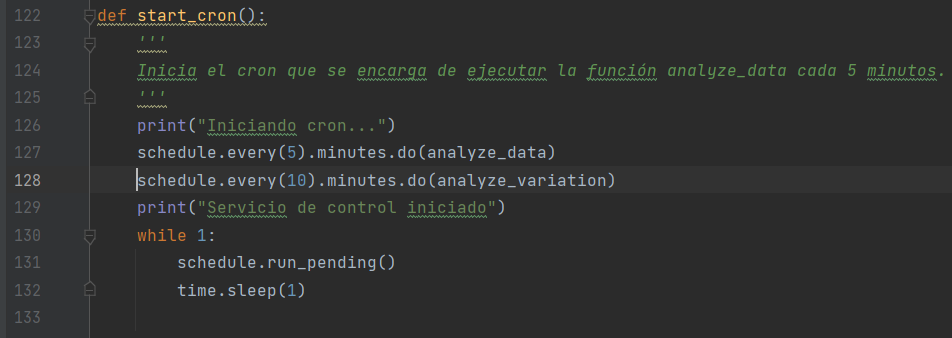
A picture containing chart

Description automatically generated

Graphical user interface, application

Description automatically generated

Modificación del código: Se creó un cron cada 10 minutos para realizar el análisis:



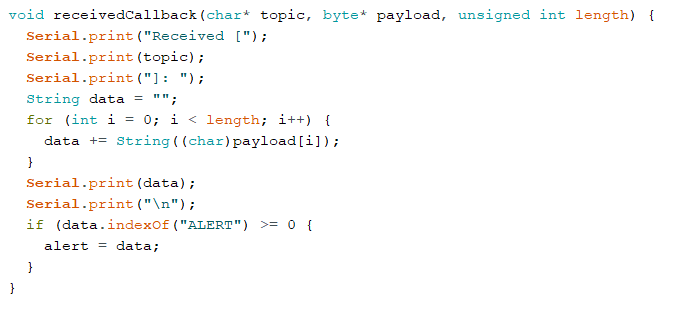
Posteriormente se define la lógica que decidirá si se activa o no una alerta, consultando los datos de las mediciones de última hora e iterando sobre el resultado. Para esto se calcula la cariación así como las otras variables para darle contexto al evento que se está procesando.

Finalmente, si la variación calculada supera el valor definido en el paso anterior, se envía un mensaje al tópico de MQTT con el mensaje de ALERTA DE VARUACION junto con la información relacionada:

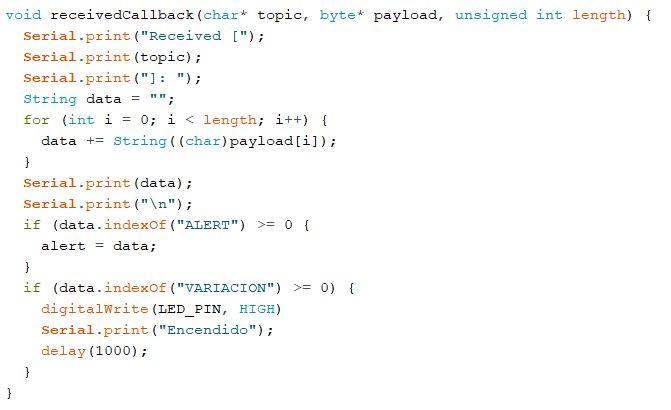
Text

Description automatically generated

A continuación se muestra la recepción del mensaje a través de los logs que se incluyeron dentro del método *receivedCallback*



Por último, dentro del método *receivedCallback* se incluyó una validación en caso de que el mensaje sea una alerta de validación, encendiéndose el led que se tiene previamente configurado.



El código modificado se encuentra en el siguiente repositorio:   
  
<https://github.com/jmorenotuniandes/IOTMonitoringServer>