

PROYECTO DE CURSO – PRIMERA ENTREGA

INTRODUCCIÓN A LOS SISTEMAS DISTRIBUIDOS

INGENIERÍA DE SISTEMAS



ESTUDIANTE:

KAMILT ANDRÉS BEJARANO DÍAZ

DOCENTE:

MARIELA JOSEFINA CURIEL HUERFANO

FECHA:

22 DE MAYO DE 2024

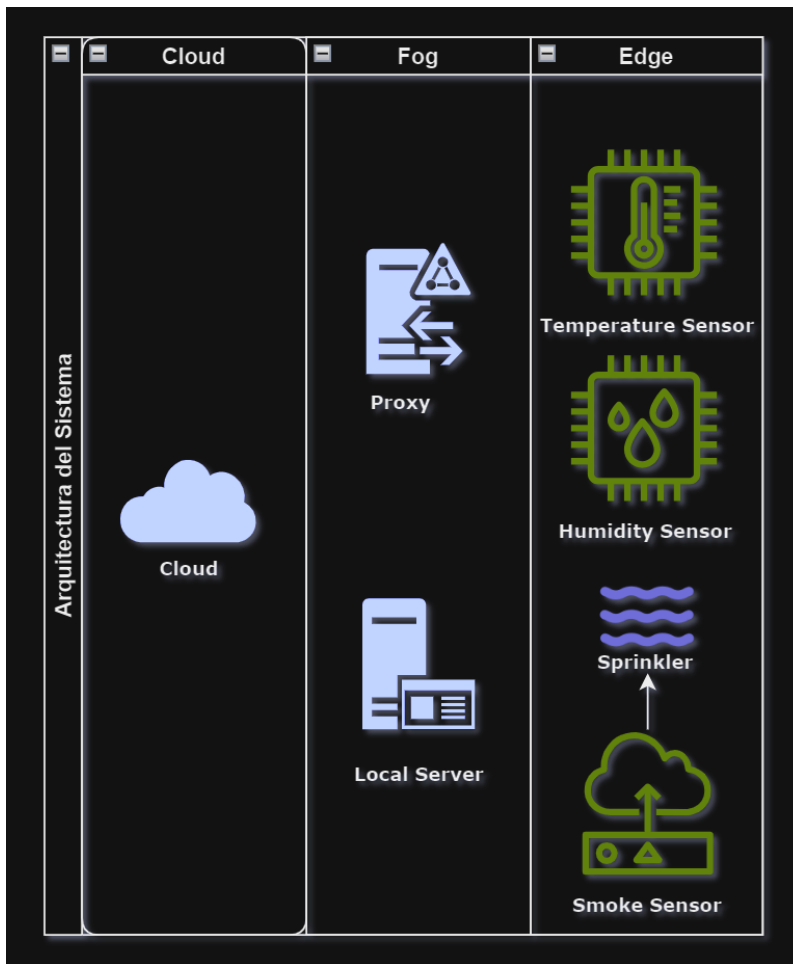
Introducción

En la presente época, los incendios forestales constituyen una amenaza significativa para el entorno natural y la seguridad pública, lo que enfatiza la necesidad urgente de implementar sistemas de monitoreo y detección anticipada eficaces. En respuesta a esta imperiosa necesidad, se ha propuesto el desarrollo de un sistema distribuido innovador que hace uso de tecnologías de Edge, Fog y Cloud computing para supervisar de forma continua las condiciones del suelo en áreas boscosas, con especial atención en los cerros orientales de Bogotá. Este sistema pretende identificar anomalías como cambios de temperatura, presencia de humo y variaciones en la humedad ambiental, para prevenir y controlar incendios forestales a tiempo.

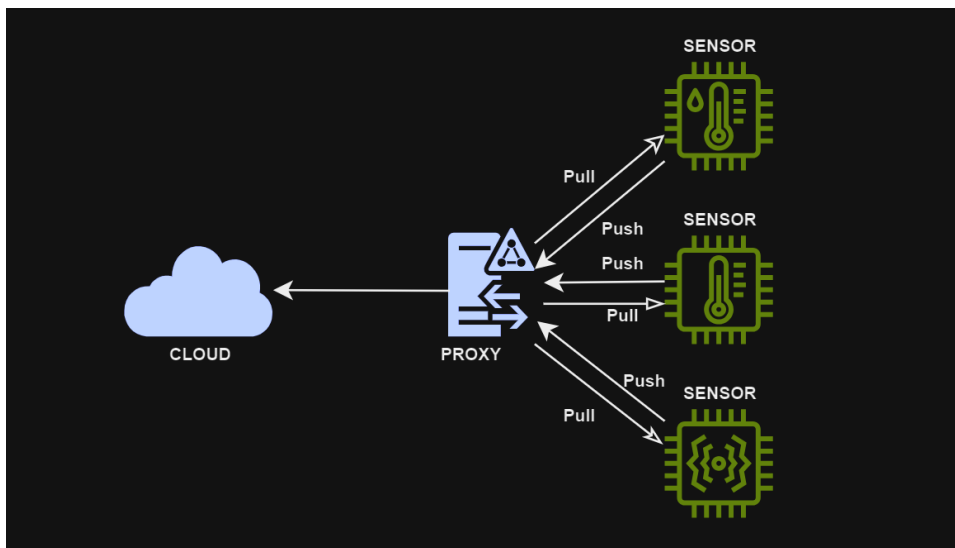
En la búsqueda de mejorar la precisión y eficiencia en la monitorización ambiental, se ha implementado un sistema de sensores de humedad y temperatura. Este sistema tiene como objetivo proporcionar datos precisos y continuos que puedan ser utilizados para diversas aplicaciones, desde la agricultura hasta el control de calidad en entornos industriales. Los sensores empleados son dispositivos avanzados capaces de operar en una amplia gama de condiciones ambientales y transmitir datos de forma inalámbrica a una plataforma de almacenamiento en la nube. Este documento presenta los resultados obtenidos, la tecnología utilizada, las pruebas realizadas y el análisis detallado del rendimiento del sistema, destacando tanto sus fortalezas como áreas de mejora.

1. Modelos del Sistema

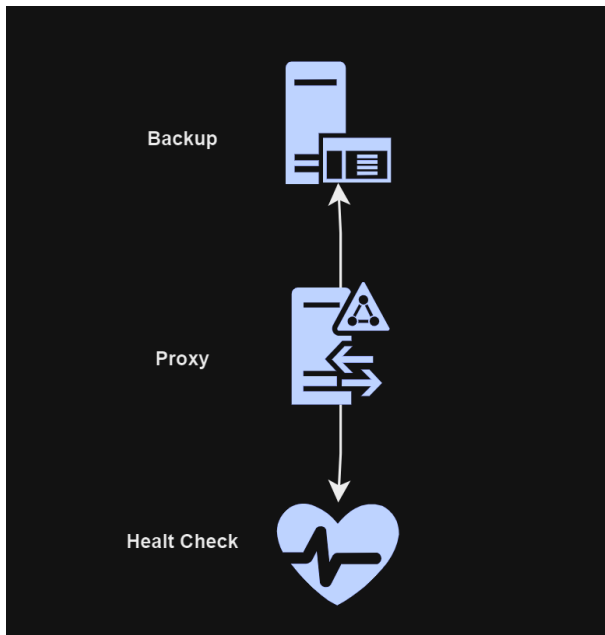
a) Modelo Arquitectónico: El sistema distribuido para monitoreo de incendios forestales se basa en una arquitectura de tres capas: Edge, Fog y Cloud Computing. La capa Edge comprende los sensores de humo, temperatura y humedad, la capa Fog se encarga del procesamiento intermedio y toma de decisiones, mientras que la capa Cloud almacena datos, realiza análisis estadístico y genera alertas. Se implementa un modelo de tolerancia a fallas para garantizar la disponibilidad del sistema.



b) Modelo de Interacción: La interacción entre los componentes del sistema se realiza a través de comunicación distribuida utilizando el modelo Pipeline de ZeroMQ para la transmisión de datos entre sensores y proxies, y el modelo request-reply para la comunicación entre los componentes de cada capa y sus sistemas de calidad.

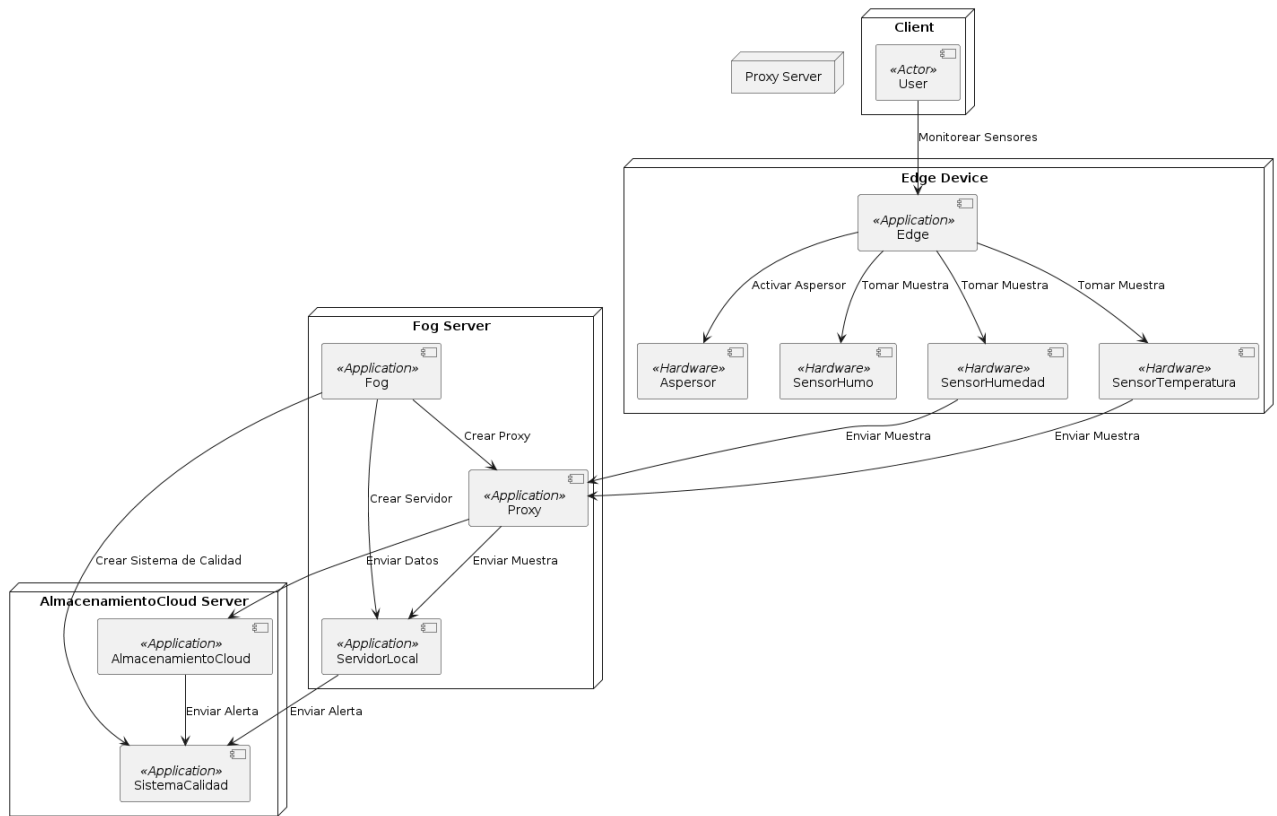


c) Modelo de Fallos y Seguridad: Se implementan mecanismos de tolerancia a fallos, incluyendo una réplica de respaldo para el componente proxy y un proceso de health check para detectar y reemplazar fallos automáticamente. Además, se asegura la comunicación segura entre componentes utilizando protocolos de cifrado.

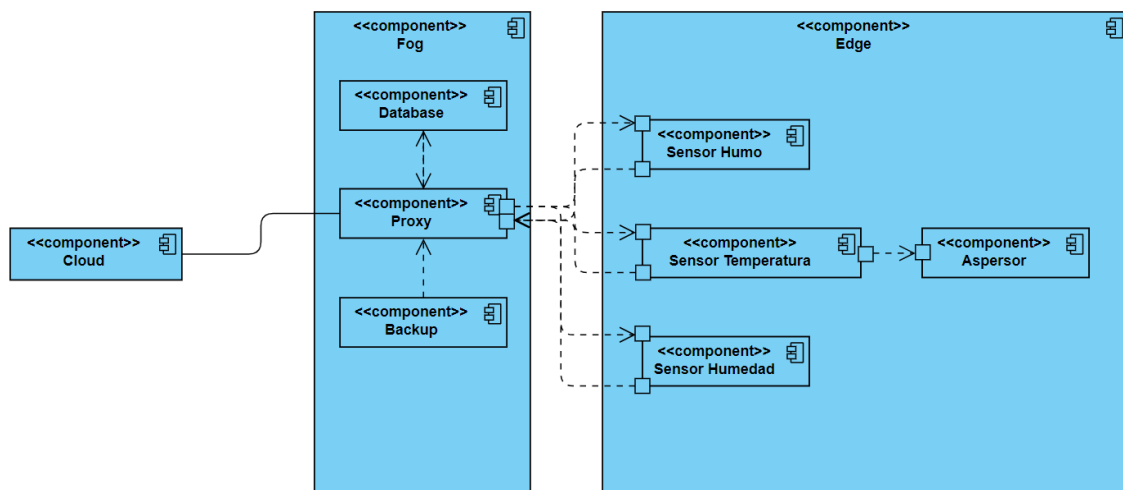


2. Diseño del Sistema

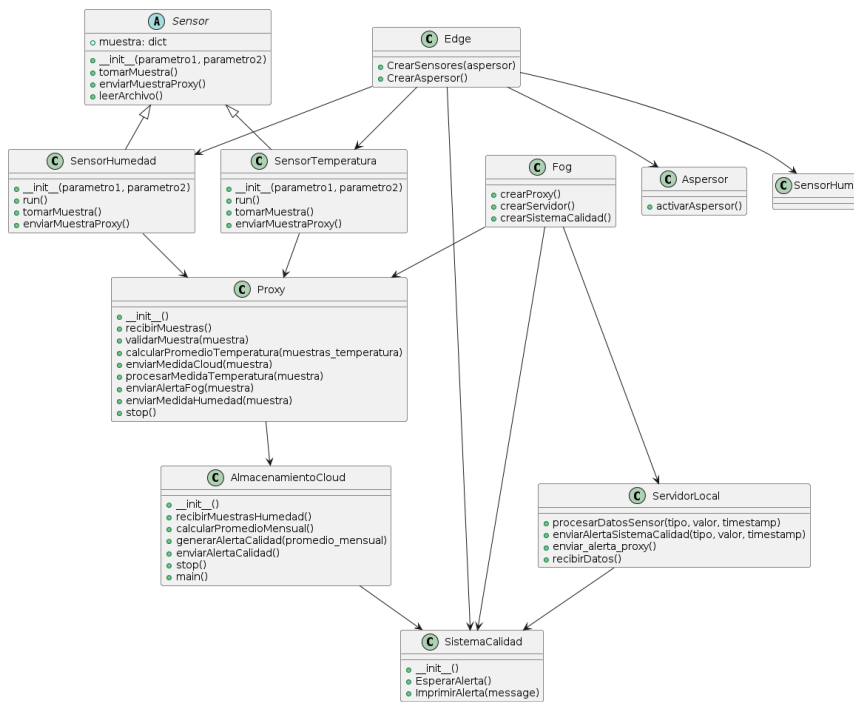
a) **Diagrama de Despliegue:** Este diagrama muestra los diferentes componentes del sistema y cómo están desplegados en varias máquinas o nodos.



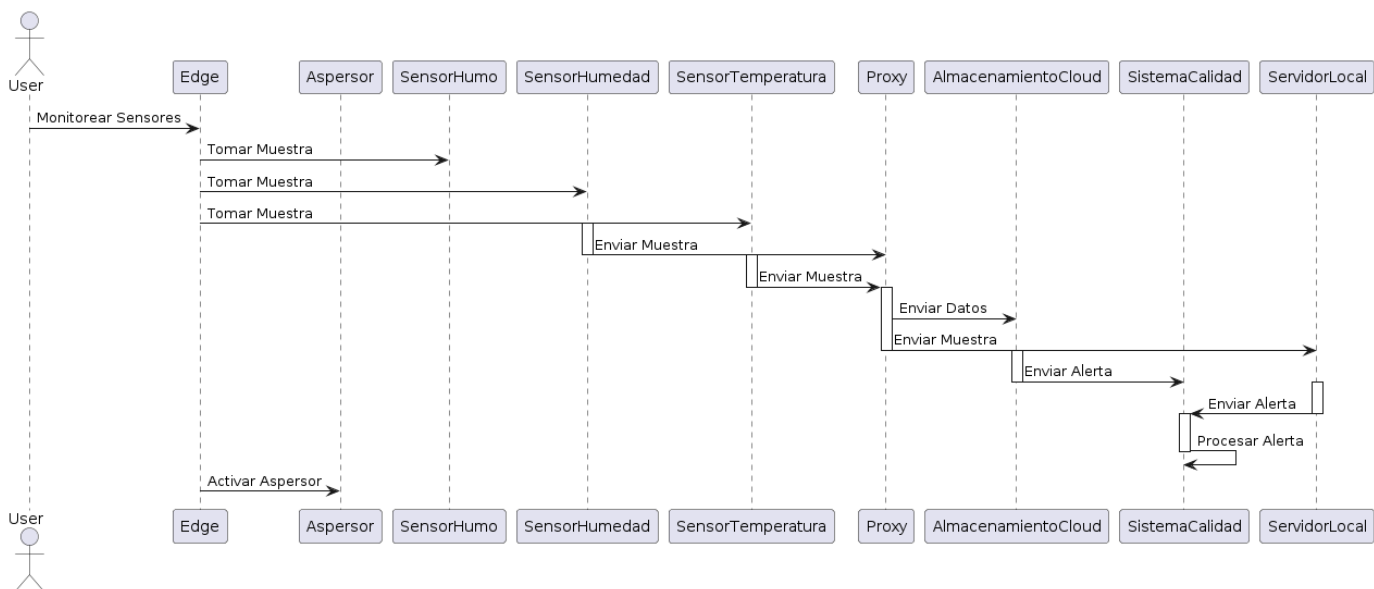
b) Diagrama de Componentes: El diagrama de componentes ilustra la estructura de componentes del sistema, incluyendo sensores, proxies, sistemas de calidad y bases de datos. Se muestra cómo interactúan entre sí para lograr los objetivos del sistema.



c) Diagrama de Clases: Este diagrama ilustra las relaciones entre las diferentes clases y cómo interactúan entre sí en el sistema descrito.



d) Diagrama de secuencia: Este diagrama muestra las interacciones entre los diferentes componentes del sistema durante el flujo de envío y procesamiento de muestras de sensores.



3. Protocolo de Pruebas

Se utilizará un protocolo de pruebas exhaustivo que cubra todos los aspectos funcionales y no funcionales del sistema, incluyendo pruebas de unidad, integración, aceptación y rendimiento. Se emplearán herramientas como Apache JMeter y Grafana k6 para generar carga y evaluar el rendimiento del sistema.

1. Configuración Incorrecta

Se simulará una configuración incorrecta en el archivo de configuración de sensores para asegurarse de que el sistema puede manejar y resolver este tipo de errores de manera adecuada.

2. Valor fuera de rango

Se enviará un valor fuera del rango establecido a uno de los sensores para verificar la capacidad del sistema para detectar y manejar alarmas por valores anómalos.

3. Sensor Desconocido

Se inicializará un monitor con un tipo de sensor no reconocido por el sistema para verificar la preparación del sistema para manejar casos inesperados.

4. Problemas de Red

Se simularán problemas de red, como pérdida de paquetes o intermitencia en la conexión de sensores y monitores, para comprobar la capacidad del sistema para responder a estas situaciones.

5. Valor Negativo

Se enviará un valor negativo a un sensor durante la ejecución para verificar la capacidad del monitor para manejar la alarma correspondiente de forma correcta.

6. Múltiples Monitores

Se ejecutarán múltiples monitores para un mismo tipo de sensor y se verificará que el sistema de calidad maneje las múltiples alarmas correctamente.

7. Cambios en la configuración en tiempo real

Durante la ejecución del sistema, se modificará el archivo de configuración para verificar que los cambios se reflejen correctamente y sean aceptados por el sistema.

8. Sobrecarga del sistema

Se aumentará el número de sensores y monitores para observar cómo el sistema maneja una mayor cantidad de datos entrantes sin degradación significativa del rendimiento.

9. Conflictos en puertos

Se iniciará el sistema con varios sensores o monitores en puertos conflictivos para verificar que el sistema pueda manejar estos conflictos durante la ejecución.

10. Pausa en la generación de datos

Se detendrá temporalmente la generación de datos de un sensor para verificar que el sistema detecte y maneje correctamente la pausa generada.

11. Congestión de la red

Se simulará una saturación de la red con un alto volumen de datos de sensores para verificar cómo el sistema maneja la congestión sin pérdida de datos críticos.

12. Cierre Abrupto de Sensores o Monitores

Durante la ejecución, se cerrará de forma abrupta un sensor o monitor para verificar que el sistema detecte la desconexión y maneje adecuadamente el error.

13. Cambios de IP de Sensores o Monitores

Durante la ejecución, se cambiará la dirección IP de un sensor o monitor en tiempo real para verificar la capacidad del sistema para mantener la comunicación entre ambos.

14. Errores de tiempo

Se modificará el reloj de forma manual durante la ejecución para observar cómo afecta la sincronización de las alarmas y cómo el sistema responde a estos cambios.

15. Conflictos con los Nombres de Sensores

Se hará que los sensores tengan nombres duplicados durante la ejecución para verificar que el sistema pueda identificar y distinguir entre ellos correctamente.

4. Obtención de Métricas de Desempeño

Para medir el rendimiento del sistema, se registrarán el total de mensajes enviados y el tamaño en bytes en todo el sistema durante su funcionamiento. Se calcularán los tiempos

de comunicación entre el proxy y el nivel Cloud, incluyendo tiempos promedio y desviación estándar asociada.

Para evaluar el rendimiento del sistema distribuido se debe considerar diferentes variables que permitan medir su eficiencia y calidad. Estas son las variables mas importantes:

- 1. Latencia de Procesamiento:** Tiempo desde que se genera una medición en los sensores hasta que se almacena en el sistema.
- 2. Latencia de Alerta:** Tiempo desde que se detecta una anomalía hasta que se envía una alerta al sistema de calidad.
- 3. Utilización de Recursos:** Uso de CPU, memoria y ancho de banda de la red durante la operación normal y en situaciones de carga.
- 4. Calidad de las Mediciones:** Porcentaje de mediciones correctas, mediciones fuera de rango y mediciones erróneas.
- 5. Carga del Sistema:** Número de sensores activos y su impacto en la capacidad de procesamiento del sistema.
- 6. Cantidad de Alertas Generadas:** Número de alertas generadas por los sensores y su distribución por tipo de anomalía.

Para llevar a cabo la evaluación de las métricas de desempeño mencionadas anteriormente, se utilizarán herramientas especializadas que permitan monitorear y analizar el funcionamiento del sistema distribuido. Entre las herramientas recomendadas se encuentran:

- 1. Scalene:** Herramienta de perfilado de Python que permite medir el rendimiento, analizar el uso de CPU, memoria y otros recursos en aplicaciones Python. Es útil para identificar cuellos de botella y problemas de rendimiento en el código Python del sistema.
- 2. VisualVM:** Herramienta de monitoreo y perfilado de Java que proporciona información detallada sobre el rendimiento de aplicaciones Java en tiempo real. Es útil para analizar el rendimiento de componentes Java, identificar problemas de memoria y CPU, y realizar perfiles de código Java.
- 3. Pyroscope:** Herramienta de perfilado de alto rendimiento diseñada para aplicaciones escritas en varios lenguajes de programación, incluido Python. Permite medir y analizar el tiempo de CPU utilizado por procesos Python y evaluar el rendimiento general del sistema distribuido.
- 4. Jensor:** Herramienta de perfilado de Java utilizada para medir el rendimiento y analizar aplicaciones Java. Proporciona métricas valiosas relacionadas con la latencia y el rendimiento en tiempo real, siendo útil para evaluar componentes Java en el sistema distribuido.

5. Comandos del sistema operativo: Se utilizarán comandos del sistema operativo como top y htop en sistemas Unix/Linux para monitorear en tiempo real los procesos en ejecución, el uso de CPU, la memoria y otros recursos del sistema.

5. Implementación

Se ha implementado la generación de mediciones desde los sensores y su recepción por parte de los procesos proxy. Este sistema inicial está funcionando en al menos dos computadoras, demostrando la comunicación efectiva entre los componentes del sistema distribuido.

```
Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.988229554572905, 'tiempo': '', 'hora': '2024-05-23 23:41:10.974211'}
Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.988229554572905, 'tiempo': '', 'hora': '2024-05-23 23:41:10.974211'}
Promedio mensual de humedad: {'humedad': 1.27850764721649}
Almacenamiento Cloud: {'promedio'}
Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 5.674372987061986, 'tiempo': '', 'hora': '2024-05-23 23:41:14.599302'}
Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 5.674372987061986, 'tiempo': '', 'hora': '2024-05-23 23:41:14.599302'}
Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.7217749481835355, 'tiempo': '', 'hora': '2024-05-23 23:41:15.041277'}
Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.7217749481835355, 'tiempo': '', 'hora': '2024-05-23 23:41:15.041277'}
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS	COMMENTS
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 1.7349854702829757, 'tiempo': '', 'hora': '2024-05-23 23:41:00.154493'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 1.7349854702829757, 'tiempo': '', 'hora': '2024-05-23 23:41:00.154493'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.9315511226798012, 'tiempo': '', 'hora': '2024-05-23 23:41:00.180492'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.9315511226798012, 'tiempo': '', 'hora': '2024-05-23 23:41:00.180492'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.9546209504343814, 'tiempo': '', 'hora': '2024-05-23 23:41:00.170492'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.9546209504343814, 'tiempo': '', 'hora': '2024-05-23 23:41:00.170492'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.9503990796898789, 'tiempo': '', 'hora': '2024-05-23 23:41:00.939093'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.9503990796898789, 'tiempo': '', 'hora': '2024-05-23 23:41:00.939093'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.803008649856861, 'tiempo': '', 'hora': '2024-05-23 23:41:00.950095'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.803008649856861, 'tiempo': '', 'hora': '2024-05-23 23:41:00.950095'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 2.9877335116075705, 'tiempo': '', 'hora': '2024-05-23 23:41:00.952093'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 2.9877335116075705, 'tiempo': '', 'hora': '2024-05-23 23:41:00.952093'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.7449012553132196, 'tiempo': '', 'hora': '2024-05-23 23:41:00.962093'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.7449012553132196, 'tiempo': '', 'hora': '2024-05-23 23:41:00.962093'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.8060807453456778, 'tiempo': '', 'hora': '2024-05-23 23:41:00.963680'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.8060807453456778, 'tiempo': '', 'hora': '2024-05-23 23:41:00.963680'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 3.315591298365689, 'tiempo': '', 'hora': '2024-05-23 23:41:04.592253'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 3.315591298365689, 'tiempo': '', 'hora': '2024-05-23 23:41:04.592253'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.8418370926827576, 'tiempo': '', 'hora': '2024-05-23 23:41:05.034419'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.8418370926827576, 'tiempo': '', 'hora': '2024-05-23 23:41:05.034419'}		
			Muestra de humedad recibida: {'tipo': 'humedad', 'valor': 0.8283937248484565, 'tiempo': '', 'hora': '2024-05-23 23:41:05.158022'}		

```
Promedio de temperatura (2024-05-23 00:05:26.468508): 20.081235586718346 grados Celsius
Promedio de temperatura (2024-05-23 00:05:26.473507): 29.884552106482293 grados Celsius
Promedio de temperatura (2024-05-23 00:05:26.477506): -4.187148851366733 grados Celsius
Promedio de temperatura (2024-05-23 00:05:26.481506): 34.20228712082435 grados Celsius
Promedio de temperatura (2024-05-23 00:05:26.485506): 18.86228373658367 grados Celsius
Promedio de temperatura (2024-05-23 00:05:26.489507): 25.79186103684328 grados Celsius
```

```
Muestra enviada al Proxy.
Muestra enviada al Proxy.
Muestra enviada al Proxy.
Muestra enviada al Proxy.
Muestra enviada al Proxy.
```

Conclusión

El desarrollo e implementación de un sistema distribuido para el monitoreo y control de condiciones ambientales en zonas boscosas representa un paso significativo hacia la

prevención y mitigación de incendios forestales. A lo largo de este proyecto, se ha logrado diseñar una arquitectura distribuida que integra tecnologías como Edge, Fog y Cloud computing para garantizar la eficiencia, resiliencia y escalabilidad del sistema. También se realizó diferentes diagramas que permiten mostrar el funcionamiento del programa y el papel que juega.

Uno de los aspectos más destacados de este proyecto es la utilización de patrones de comunicación síncronos y asíncronos, así como la implementación de estrategias de tolerancia a fallas para garantizar la continuidad operativa del sistema. La arquitectura propuesta ha demostrado su capacidad para manejar la detección temprana de anomalías en las condiciones ambientales, como cambios en la temperatura, presencia de humo y niveles de humedad.

Durante la fase de implementación, se han utilizado diversas herramientas de evaluación de rendimiento para medir variables críticas como la latencia de procesamiento, la utilización de recursos del sistema y la calidad de las mediciones. Estos análisis han proporcionado información valiosa sobre el funcionamiento del sistema en condiciones normales y bajo cargas de trabajo variables.