ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

MQTT-Mosquitto

Arquitectura de Computadoras

Alumnos:

Kevin Topón

Andrés Paspuel

Henry Simba

Docente:

Ing. Darwin Omar Alulema Flores

CONTENIDO

Título: MQTT Pub-Sub

1.- Planteamiento del Problema

Desde la llegada del internet de las cosas(IoT), la mayoría de los dispositivos electrónicos o humanos recopilan, procesan y monitorean datos alrededor del mundo así que lo que vamos a realizar es entender cómo es que realmente funciona este tipo de protocolo MQTT .

2.- Objetivos

Objetivo General

Evaluar y familiarizarnos con lo que es el patrón de publicación/suscripción ya que utilizaremos como medio de comunicación el protocolo MQTT.

Objetivos Específicos

- Conocer cómo está estructurada el MQTT y la funcionalidad que tienen los elementos que se encuentran.
- Identificar cuáles podrían ser los posibles inconvenientes al momento de manejar las pruebas de comunicación.
- Entender cómo funciona la teoría pub/sub y los Message broker.

3.- Estado del Arte

- Francesc Moreno Cerdà, en el año 2018, en Barcelona se realizó sobre Demostrador arquitectura publish/subscribe con MQTT y que pretendía mostrar la gran aplicabilidad del protocolo a diferentes tecnologías, softwares, herramientas de desarrollo y lenguajes de programación.
- Vera Martín y Ana María, en el año 2019, en Madrid se ha realizado la implementación de un entorno de control de un sistema Industrial Internet of Things basado en Message Queuing Telemetry Transport con Labview y por resultados se ha obtenido la interacción de distintas suscripciones y publicaciones en paralelo con el Broken MQTT. Esto hace que la librería sea mucho más abierta a su uso con varios dispositivos de forma simultánea.

- Nasi Tantitharanukul, Kitisak Osathanunkul, Kittikorn Hantrakul, Part Pramokchon, and Paween Khoenkaw, en el año 2016, en Tailandia, realizaron sobre MQTT-Topics Management System for Sharing of Open Data.en los resultados de la publicación / suscripción de mensajes livianos protocolo se puede utilizar para compartir datos dinámicos, el intercambio de datos, por ejemplo, datos de monitoreo de inundaciones, datos de terremotos o tráfico los datos a través del protocolo MQTT pueden utilizarse para mejorar la forma de vida humana.

4.- Marco Teórico

MQTT

Según (Tang, Wang, Liu, & Sheng, 2013), menciona que MQTT (Message Queuing Telemetry Transport), es un protocolo de mensajería instantánea de publicación/suscripción basado en intermediarios fue diseñado para ser simple, liviano y fácil de ponerlo en funcionamiento.

El protocolo fue creado por Andy Stanford y Areln Nipper (IBM) en 1999 en sus inicios estaba orientado a crear un protocolo para la pérdida mínima de la batería y el mínimo ancho de banda así como también conectar a grandes cantidades de sensores remotos y dispositivos de control y a ido evolucionando e implementando en el internet de las cosas actualmente el protocolo está implementado en una variedad de sistemas integrados como el de los hospitales utilizado para comunicarse con marcapasos y otros proveedores de equipos médicos, las compañías de petróleo y gas lo utilizan para monitorear oleoductos desde una distancia considerable.

El protocolo MQTT es utilizado en mensajería instantánea y un protocolo de mensajería de publicación/suscripción basado en un agente ligero admitiendo así todas las plataformas y una variedad de lenguajes de programación populares siendo así una solución ideal para enviar mensajes debido a su simplicidad y escalabilidad así utilizado principalmente en el campo de los mensajes móviles.

El agente de mensajes Really Small Message Broker es una agencia de MQTT proporcionada por IBM, es usado para crear fácilmente un servidor de mensajería instantánea altamente eficiente este es el responsable de recibir mensajes del servidor y reenviarlos a dispositivos que estén relacionados con el tópico que se este utilizando, además de esto cada mensaje debe pasar por Message Broker, recibiendo y enviando a un cliente diferente.

Tópicos

En la operación de inserción, el servidor que proporciona información y el cliente que recibe los mensajes, si el servidor envía un mensaje basado en el asunto o el cliente recibe mensajes por asunto, todos pasan por un intermediario de mensajes del servidor proxy de esta manera logran que sus servidores y clientes se integren bien con Message Broker, aprovechando su estabilidad para tratar la cola de mensajes y asi mejorando la estabilidad del sistema.

(Hunkeler & Truong, 2005), argumenta que MQTT siendo un protocolo de Pub/Sub se basa en temas que utiliza cadenas de caracteres para brindar soporte a temas jerárquicos facilitando la suscripción a múltiples temas como por ejemplo un sensor de temperatura ubicado en el piso 2 podria publicar sus datos utilizando el tema jerarquico

"Wsn / sensor / P2 / R248 / temperatura". El carácter de barra diagonal "/" se usa para separar cada parte del tema. Los caracteres comodín se pueden usar para reemplazar cualquier parte del tema, por ejemplo, la cadena "Wsn / sensor / P2 / + / temperatura" podría emplearse para suscribirse a los datos generados por todos los sensores de temperatura en el piso P2. En este ejemplo el personaje "+" se usó como comodín para cualquier patrón en el 4to nivel del tema.

(Stanford & Truong, 2013), señala que el protocolo MQTT tiene varias versiones como MQTT-SN que se adapta a las peculiaridades de un entorno de comunicación inalámbrica. Los enlaces de radio inalámbricos tienen en general tasas de falla más altas que las cableadas debido a su susceptibilidades, desvanecimiento e interferencias de la misma manera que tiene una tasa de transmisión más baja.

Pub/Sub

Pub/Sub es un servicio de mensajería asíncrono que separa los servicios que producen eventos de los que procesan eventos.Puedes usar Pub/Sub como la transferencia y entrega de eventos o de software multimedia orientado a la mensajería para las canalizaciones de

estadísticas de transmisión.Pub/Sub ofrece almacenamiento duradero de mensajes y entrega de mensajes en tiempo real con alta disponibilidad y rendimiento uniforme a gran escala. Los servidores de Pub/Sub se ejecutan en todas las regiones de GOOGLE CLOUD del mundo.

Broker o servidor

¿Qué es?

Se puede definir como un middleware orientado a mensajes, actuando como un agente de transferencia de mensajes, intercambiándolos entre diferentes aplicaciones, pudiendo ser estas aplicaciones: emisores o receptores. Se encarga de traducir los mensajes de los productores a los consumidores. Estos mensajes son elementos que han sido formalmente definidos entre las diferentes aplicaciones que se comunican. También proporciona la validación, transformación y enrutamiento de los mensajes.

Además, actúa como un mediador entre las comunicaciones de las aplicaciones, minimizando el grado de conocimiento entre ellas. De esta forma, se obtiene un efectivo desacoplamiento.

¿Qué nos proporciona?

Asincronicidad

Al usar un sistema MOM, un cliente puede enviar un mensaje a un destino administrado por el proveedor. La llamada invoca los servicios del proveedor para enrutar y entregar el mensaje. Una vez que ha enviado el mensaje, el cliente puede continuar haciendo otro trabajo, con la confianza de que el proveedor conservará el mensaje hasta que un cliente receptor lo recupere, haciendo posible crear un sistema de componentes débilmente acoplados.

Además, la mayoría de los sistemas MOM asíncronos proporcionan almacenamiento persistente para hacer una copia de seguridad de la cola de mensajes. Esto significa que el emisor y el receptor no necesitan conectarse a la red al mismo tiempo, resolviendo así los problemas con la conectividad intermitente. Es decir, si la aplicación del receptor falla por alguna razón, se acumulan en la cola de mensajes para su posterior procesamiento cuando el receptor se reinicia.

Enrutamiento

Existen muchas implementaciones de MOM. Algunas de ellas permiten que el propio bróker de mensajería se haga cargo de la lógica de enrutamiento, mientras que otras

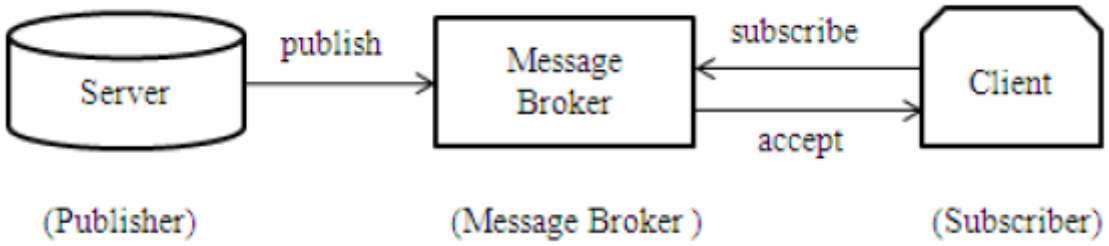
dependen de las aplicaciones del cliente. Proporcionando así la difusión o multidifusión de mensajes.

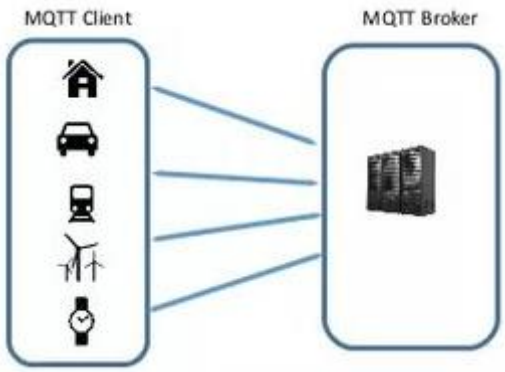
Transformación

En un sistema *middleware* basado en mensajes, el mensaje recibido en el consumidor no necesita ser idéntico al mensaje enviado desde un productor. Estos sistemas tienen inteligencia incorporada pudiendo transformar los mensajes, para que coincidan con los requisitos del consumidor. Una aplicación puede enviar un mensaje en su propio formato nativo y diferentes aplicaciones pueden recibir una copia del mensaje en su propio formato nativo. Muchos sistemas modernos de MOM proporcionan herramientas sofisticadas de transformación de mensajes (o mareo) que permiten especificar reglas de transformación.

5.- Diagramación

Flujo básico de Mensajería Pub/ Sub





6.- Lista de Componentes

Recursos Utilizados
• **Computadora o laptop**
• **Acceso a Internet**
• **Programa Mosquito**

7.- Mapa de variable

1. sub: Mensaje a recibir.
2. pub: Mensaje a enviar.
3. -v: Mostrará el nombre del tema y mensaje.
4. -t y -m: Si está enviando múltiples valores, entonces necesitamos poner la cadena completa entre comillas.

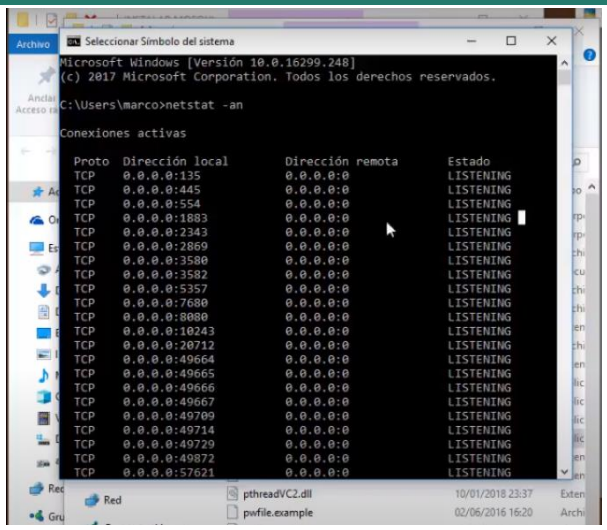
8.- Descripción código fuente

Tópicos

1. mosquitto_sub -t "topico" -v:
    a. En esta parte se escribe así porque va a ser el que reciba el mensaje para imprimir en pantalla.
2. mosquitto_pub -t "topico" -m "on":
    a. En este si escribe la palabra topico porque es en donde se va a imprimir en pantalla además de que se imprimirá con su mensaje. Ejemplo: topico on.

9.- Descripción de pre-requisitos y configuración

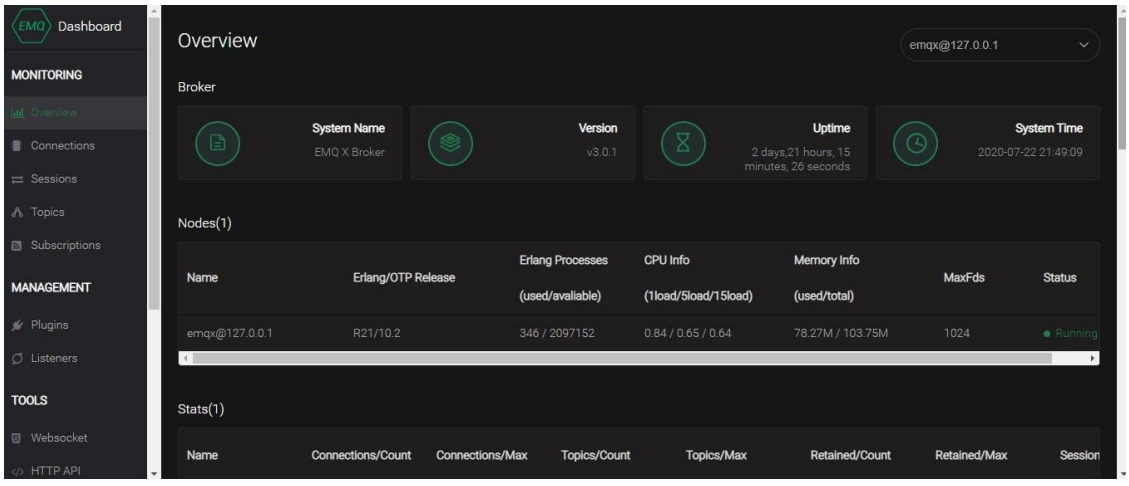
Antes de iniciar con las pruebas de comunicación tenemos que comprobar que el servidor mosquito está escuchando en el puerto 1883 porque es el puesto estándar que escucha el broker.



Posterior a confirmar que está escuchando debemos ir al firewall de windows para habilitar el firewall para que permita conexiones externas con nuestro computador porque vienen desactivados por defecto.

10.- Aportaciones

En este punto tenemos la plataforma de GOOGLE CLOUD en la cual nos permite realizar este tipo de prueba MQTT para hacer pruebas de PUBLICADOR Y SERVIDOR debemos seguir los siguientes pasos cómo tener una cuenta con una tarjeta de crédito y tener ya un nivel muy alto en programación html.



11.- Conclusiones

- El protocolo MQTT se ha alzado como uno de los estándares para aplicaciones IoT tanto comerciales como de ámbito maker.
- Los problemas hallados es de que hubo plataformas que no eran amigables y que debido a ello a veces la comunicación de enviar y receptar un mensaje fallaba, entonces se decidió optar por mosquito que era más entendible al momento de utilizar.
- El broker mosquitto es uno de los más sencillos de utilizar ya que no requiere programación, solo conocer instrucciones básicas para la suscripción y publicación.
- En la actualidad la humanidad se ha visto inmersa con la tecnología y la gran cantidad de datos, el servicio del protocolo MQTT hace que la información además de aumentar sea eficaz en las aplicaciones móviles.

12.- Recomendaciones

1. Indagar el tema debido a que hay plataformas MQTT complicadas en entender, sobre todo como funciona, pero hay fáciles como mosquito que entender su funcionamiento es más transparente en cómo funciona.
2. Ejecutar el Broker mosquito antes de comenzar a ejecutar nuestros comandos en el terminal de CMD por el motivo de que si no se encuentra corriendo de forma adecuada nos presenta errores al momento de insertar nuestras líneas de código.
3. verificar que las instrucciones que pongamos en el terminal de CMD se encuentren ejecutando desde mosquitto además de que la sintaxis que se utiliza debe ser la misma, caso contrario presentará errores como el de que no se encuentra conectado a nuestro broker.

13.- Cronograma

TRABAJO DE INVESTIGACION	16/07/2020	17/07/2020	18/07/2020	19/07/2020	20/07/2020	21/07/2020	22/07/2020
PLANTEAMIENTO DEL PROBLEMA							
OBJETIVOS							
ESTADO DEL ARTE							
MARCO TEÓRICO							
DIAGRAMAS							
LISTA DE COMPONENTES							
MAPA DE VARIABLES							
EXPLICACIÓN DEL CÓDIGO FUENTE							
DESCRIPCIÓN DE PRERREQUISITOS Y CONFIGURACIÓN							
APORTACIONES							
CONCLUSIONES							
RECOMENDACIONES							
CRONOGRAMA							
BIBLIOGRAFÍA							
ANEXOS							
MANUAL DE USUARIO							
HOJAS TÉCNICAS							

14.- Bibliografia

Cerdà, F. M. (2018). Demostrador arquitectura publish/subcribe con MQTT. *Universidad Politecnica de Catalunya*.

Hunkeler, U., & Truong, H. (2005). MQTT-S un protocolo de Pub/Sub para Sensor de redes inálambricas. *Research Laboratory, Suiza*.

MArtin, A. M. (2019). El MQTT implementado en LABVIEW.

Stanford, A., & Truong, H. L. (2013). MQTT para redes de sensores (MQTT-SN). 28.

Tang, K., Wang, Y., Liu, H., & Sheng, Y. (2013). Diseño e implementación del sistema de notificacion push basado en el Protocolo MQTT. *Conferencia internacional sobre ciencias de la información*.

Tantitharanukul, N., & Osathanunkul, K. (2016). MQTT-Criterios de denominación de temas de datos abiertos para ciudades inteligentes. *Semantic Scholar*.