ST0263 "Tópicos Especiales En Telemática"

Universidad EAFIT. 2021-1

Proyecto N°1: MiddleWare

*Katherin Valencia Correa, Santiago Bedoya Cano, Ricardo Jaramillo Velasquez*

Correo-e: [kvalenciac@eafit.edu.co](mailto:kvalenciac@eafit.edu.co),  [sbedoyac1@eafit.edu.co](mailto:sbedoyac1@eafit.edu.co),  [rjaram20@eafit.edu.co](mailto:rjaram20@eafit.edu.co)

# **INTRODUCCIÓN**

Este documento recopila todos los pasos que se llevaron a cabo para diseñar e implementar un middleware de mensajería (MOM) en Python que le permita a los clientes enviar y recibir mensajes de datos. Se implementa la appColas en donde se tienen los clientes (publishers) y los servidores (subscribers), se permite en registro de nuevos usuarios y la administración de los mensajes a las colas.

# **DESARROLLO**

Tenemos el Middleware en el cual se verifica que los usuarios estén autenticados para poder establecer la conexión, de no estar registrados saldrá un mensaje de autentificación fallida.

El server se encarga de mostrar las subscripciones a las diferentes colas por parte de los diferentes usuarios registrados además de permitirle a un usuario subscrito a una cola consumir los mensajes de esta y finalmente visualizar la totalidad de los mensajes consumidos por un usuario. Se manejan fallos como digitar un comando no valido para ser interpretado por el servidor o querer consumir más mensajes de una cola vacía.

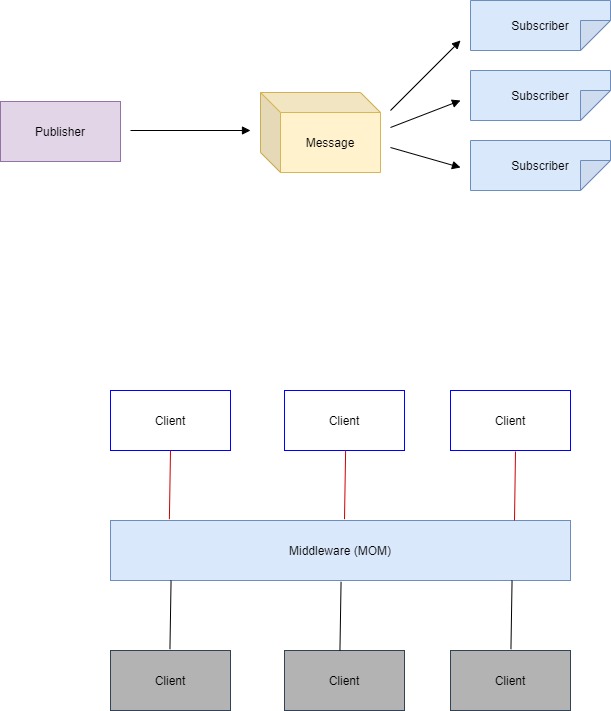
El cliente hace el rol de publisher el cual puede crear una cola o enviar mensajes a las colas que ya están creadas, se maneja error al momento de querer añadir un mensaje a una cola inexistente.

De igual manera la aplicación permite a un usuario ingresar o registrarse en caso de no estarlo, para ello necesita digitar una contraseña esto con el fin de realizar la debida autentificación, en caso de querer registrar un usuario con un username existente se mostrará un mensaje de error. Al momento de querer ingresar al sistema se le solicitará al usuario ingresar su correspondiente contraseña, una vez la autentificación se haya verificado se le mostrará al usuario las colas disponibles y este tendrá la opción de suscribirse a la o a las colas que el desee, en caso tal de subscribirse a una cola a la cual ya se encuentra subscrito se le mostrará un mensaje de error, además de tener el control de errores para cuando se pretenda hacer una subscripción a una cola inexistente o ingresar al sistema con un username no registrado.

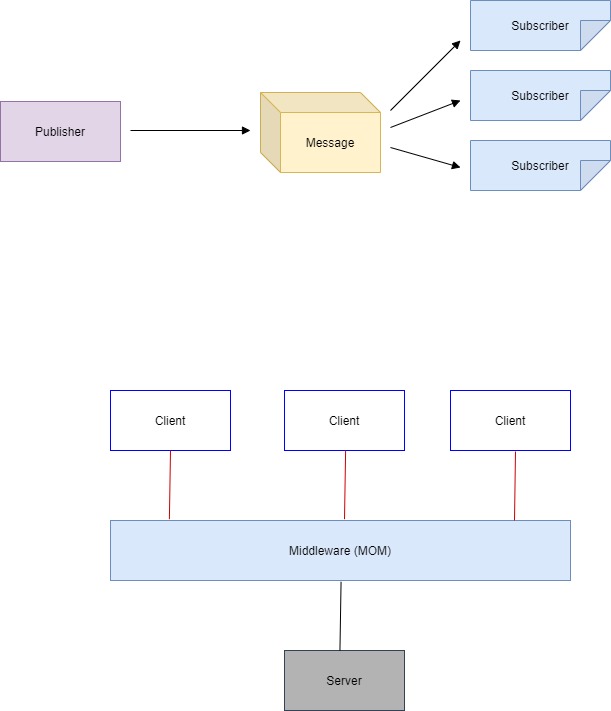
Todo este desarrollo es posible gracias a la utilización de las librerías flask y middleware, disponibles para su uso en Python.

# **ARQUITECTURA**

Modelo Publisher/Subscribers



*Figura 1. Arquitectura Publisher/Subscriber.*



*Figura 2. Conexión entre el middleware, el servidor y los clientes.*

# **DESPLIEGUE EN AWS**

Se realizó el despliegue de todo el sistema solución en una maquina virtual de AWS Educate, se subieron los archivos a una instancia EC2 con nombre Middleware sobre una subnet publica, los archivos de la solución se subieron a través de scp.

# **CONCLUSIONES**

* El middleware permite la comunicación y la administración de datos en aplicaciones distribuidas, sirve como un orquestador entre el servidor y los clientes, pero una de las desventajas de los sistemas es que el MOM es una comunicación asíncrona, los sistemas asíncronos pueden presentar sobrecarga de la red y el servidor de procesamiento puede tornarse un poco lento para la entrega de los mensajes.
* Las ventajas de implementar un MOM son la posibilidad del envío de mensajes de manera asincrónica, que permite la flexibilidad, interoperabilidad y puede ser portable.