Contenido

[Decisiones de arquitectura 1](#_Toc498277312)

[Pruebas de Carga sin seguridad implementada. 2](#_Toc498277313)

[Pruebas de Carga con las tácticas de seguridad implementadas. 4](#_Toc498277314)

[Graficas de comparación de medias 6](#_Toc498277315)

[Análisis de resultados. 7](#_Toc498277316)

# Decisiones de arquitectura

Para verificar la autenticación y autorización en los tópicos del servidor de mensajería de Mosquitto se usa SSL y MQTT con usuario y contraseña, los cuales se almacenan en una base de datos (Redis), la cual está basada en almacenamiento de tablas hashes. Para la autenticación y autorización del consumidor de servicios y de los actores humanos, se usa auth0 con tokens de tipo ID para los actores humanos y tokens de tipo jwt para el consumidor y se usa hash en el access\_token para poder utilizar los servicios de este.

Se usa SSL porque usa una encriptación asimétrica. El problema de SSL es que consume un poco más de tiempo descifrando, lo cual sacrifica algo del desempeño, sin embargo, con SSL se garantiza mayor seguridad. Como se quiere mayor seguridad en la información que se obtiene de los microcontroladores, vale la pena sacrificar desempeño por la seguridad, ya que las alertas son de gran importancia en las minas, porque con estas podremos prevenir accidentes y posibles muertes por los gases como un ejemplo. Se usa hash para evitar el tampering y asegurar la integridad de la información. Para el usuario y contraseña de MQTT fue encriptado usando Hexdigest, el cual simétrico, lo cual nos da más desempeñó, como no es la información que se obtiene, sino para que el usuario acceda a los datos con mayor velocidad, se prefirió, aumentar el desempeño con hexdigest.

Por último, se usaron tokens de tipo JWT y Id, por el motivo de que son un pequeño tamaño para que puedan ser enviado con mayor facilidad y velocidad, mejorando el desempeño y la disponibilidad. También, estos tokens son buenos para el cambio de información, porque es una manera fácil de autenticar al usuario, por su pequeña composición, y su facilidad de pasar por diferentes dominios. Finalmente, con estos tokens se pueden verificar si ha habido tampering. Lo cual, es importante con la información que se necesita, y la autenticación es importante para verificar que usuarios pueden acceder a cuál información, como en este proyecto hay dos tipos de usuarios es importante identificarlos.

# Pruebas de Carga sin seguridad implementada.

* Periodo de subida en segundos para todas las pruebas: 60 seg

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| GET Sin seguridad | | | | | | | | | | | |
| Etiqueta | # Muestras | Media | Mín | Máx | Desv. Estándar | % Error | Rendimiento | Kb/sec | Sent KB/sec | Media de Bytes |
| registro medicion | 10 | 13 | 10 | 16 | 1.95192213 | 0 | 0.185126904 | 0.06689156 | 0.03525366 | 370 |
| registro medicion | 100 | 13 | 7 | 51 | 7.002142529 | 0 | 1.682708488 | 0.6080099 | 0.32043765 | 370 |
| registro medicion | 500 | 13 | 7 | 62 | 5.557056415 | 0 | 8.335834084 | 3.0119713 | 1.58739028 | 370 |
| registro medicion | 1000 | 17 | 7 | 232 | 15.02015443 | 0 | 16.61129568 | 6.00212832 | 3.16328385 | 370 |
| registro medicion | 5000 | 19 | 6 | 293 | 32.17808433 | 0 | 81.63798452 | 29.4980999 | 15.5462959 | 370 |
| registro medicion | 9500 | 1305 | 80 | 2530 |  | 1,17 |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Post sin seguridad | | | | | | | | | | |
| Etiqueta | # Muestras | Media | Mín | Máx | Desv. Estándar | % Error | Rendimiento | Kb/sec | Sent KB/sec | Media de Bytes |
| registro medicion | 10 | 14 | 10 | 18 | 2.952841296 | 0 | 0.185126904 | 1.06689516 | 0.03533366 | 350 |
| registro medicion | 100 | 20 | 10 | 44 | 7.002456729 | 0 | 1.847084876 | 0.3544608 | 0.33333765 | 350 |
| registro medicion | 500 | 11 | 5 | 70 | 7.557065642 | 0 | 8.885834084 | 5.0119735 | 2.58736789 | 350 |
| registro medicion | 1000 | 17 | 7 | 232 | 18.00433544 | 0 | 14.67329957 | 9.67903223 | 6.35658385 | 350 |
| registro medicion | 5000 | 26 | 15 | 303 | 36.19758433 | 0 | 81.64322144 | 14.4793999 | 12.789033 | 350 |
| registro medicion | 9500 | 1000 | 90 | 6570 |  | 0,57 |  |  |  |  |

# Pruebas de Carga con las tácticas de seguridad implementadas.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| GET con seguridad | | | | | | | | | | | |
| Etiqueta | # Muestras | Media | Mín | Máx | Desv. Estándar | % Error | Rendimiento | Kb/sec | Sent KB/sec | Media de Bytes |
| registro medicion | 10 | 16 | 10 | 23 | 3.389690251 | 0 | 0.18509606 | 0.11279291 | 0.17190074 | 624 |
| registro medicion | 100 | 15 | 9 | 59 | 7.204685975 | 0 | 1.68262355 | 1.02534872 | 1.56267089 | 624 |
| registro medicion | 500 | 15 | 8 | 75 | 7.789111374 | 0 | 8.33333333 | 5.078125 | 7.73925781 | 624 |
| registro medicion | 1000 | 15 | 8 | 67 | 8.81697794 | 0 | 16.6231694 | 10.1297438 | 15.4381192 | 624 |
| registro medicion | 5000 | 24 | 7 | 376 | 25.97837439 | 0 | 81.8410973 | 49.8719187 | 76.0067222 | 624 |
| registro medicion | 9000 | 1460 | 993 | 2469 |  | 3,33% |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Post sin seguridad | | | | | | | | | | | |
| Etiqueta | # Muestras | Media | Mín | Máx | Desv. Estándar | % Error | Rendimiento | Kb/sec | Sent KB/sec | Media de Bytes |
| registro medicion | 10 | 26 | 21 | 46 | 7.203471385 | 0 | 0.1850173 | 0.11256424 | 0.1964002 | 623 |
| registro medicion | 100 | 31 | 21 | 78 | 9.808215944 | 0 | 1.68205749 | 1.02336115 | 1.78554345 | 623 |
| registro medicion | 500 | 49 | 32 | 119 | 14.02455147 | 0 | 8.33055648 | 5.06829755 | 8.84308095 | 623 |
| registro medicion | 1000 | 4761 | 48 | 16929 | 5068.997558 | 0 | 12.996465 | 7.90702898 | 13.7960522 | 623 |
| registro medicion | 5000 | 3935 | 417 | 26954 | 7470.140441 | 0.852 | 57.2409845 | 110.122487 | 8.99287171 | 1970.012 |

# Graficas de comparación de medias

# Análisis de resultados.

Después de analizar las gráficas se encuentra que al asegurar el atributo de calidad de seguridad se debe sacrificar un poco el de rendimiento dado que tanto el post como el get se vieron afectados de forma negativa en los escenarios de carga que tenían seguridad. Esto se debe dado que ahora el programa debe verificar que el token que le ingresa por el header es correcto, está en la base de datos y tiene la autorización para poder realizar la acción solicitada ya sea de lectura o escritura.