

Evaluación del Rendimiento de YouTube y del Aula Virtual de la UNA Puno mediante Apache JMeter y Google Lighthouse

Ronald Wilder Incacutipa Muñuico

Introducción

En la actualidad, la evaluación del rendimiento de aplicaciones web y plataformas digitales es fundamental para garantizar una experiencia de usuario óptima. **Apache JMeter** es una herramienta de código abierto ampliamente utilizada para realizar pruebas de carga, estrés y rendimiento en aplicaciones web, permitiendo simular múltiples usuarios que acceden simultáneamente a un sitio o servicio.

Por otro lado, **Google Lighthouse** es una herramienta automatizada que evalúa la calidad de páginas web en aspectos clave como rendimiento, accesibilidad, buenas prácticas, SEO y experiencia de usuario. Lighthouse ejecuta un conjunto de auditorías sobre el sitio y genera un informe detallado con puntajes y recomendaciones de mejora, lo que permite identificar problemas de optimización, lentitud en la carga, elementos bloqueantes y oportunidades de mejora en la estructura del sitio. Gracias a ello, Lighthouse complementa las pruebas de carga realizadas con JMeter proporcionando una visión más integral del rendimiento real que percibe el usuario final.

El análisis se centra en dos plataformas relevantes para estudiantes de la Universidad Nacional del Altiplano (UNA) de Puno: YouTube y el Aula Virtual UNA Puno. Se evalúan tiempos de respuesta, capacidad de manejo de usuarios simultáneos y estabilidad del servicio.

Objetivo

Evaluar el rendimiento y capacidad de respuesta de YouTube y Aula Virtual UNA Puno mediante pruebas con Apache JMeter y un poco de apoyo de Lighthouse. También se aplican métodos de optimización numérica como gradiente y Regula Falsi para analizar y mejorar los tiempos de respuesta y la eficiencia de cada plataforma.

Pruebas de Rendimiento con Apache JMeter

A continuación, se presentan algunas imágenes generadas durante el proceso de evaluación.



Imagen 1

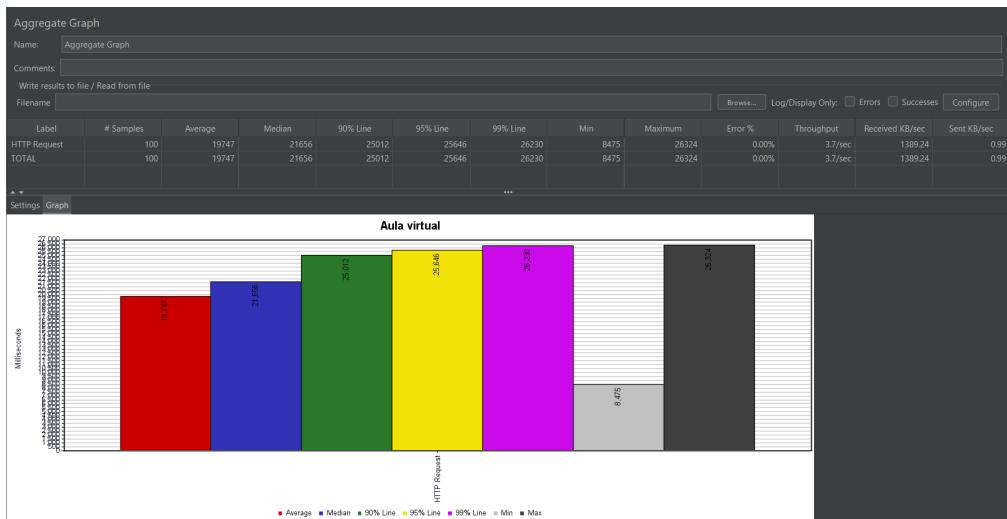


Imagen 2

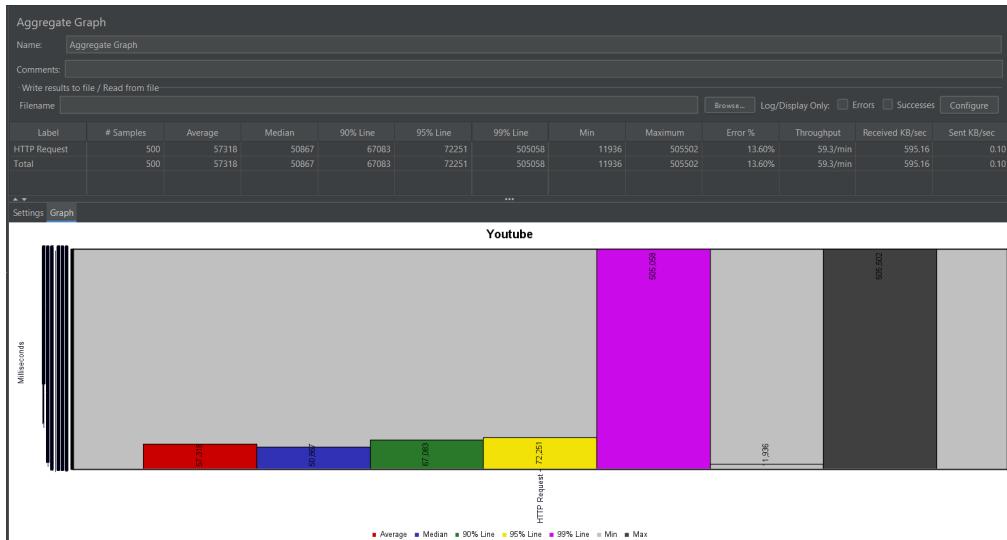


Imagen 3

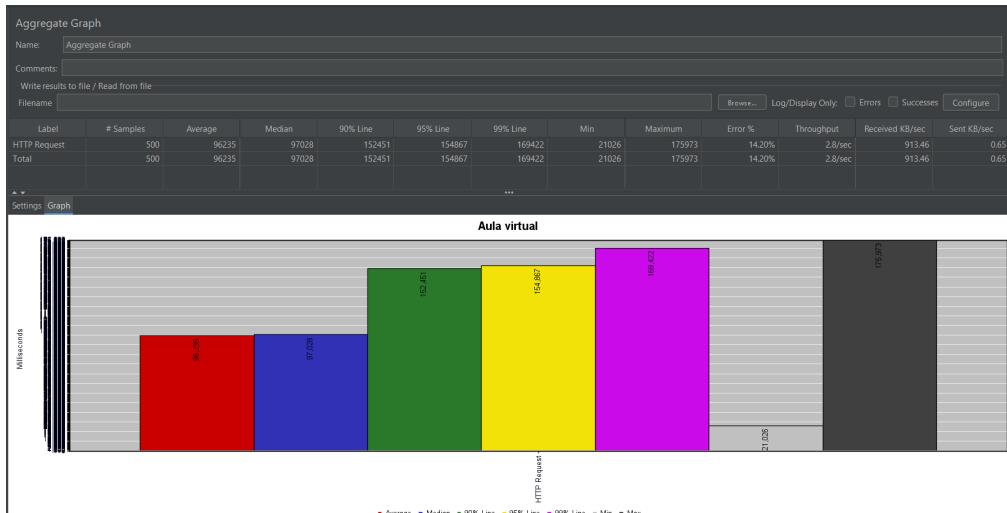


Imagen 4



Imagen 5



Imagen 6

Pruebas de Rendimiento con Lighthouse

A continuación, se presentan algunas imágenes generadas durante el proceso de evaluación.

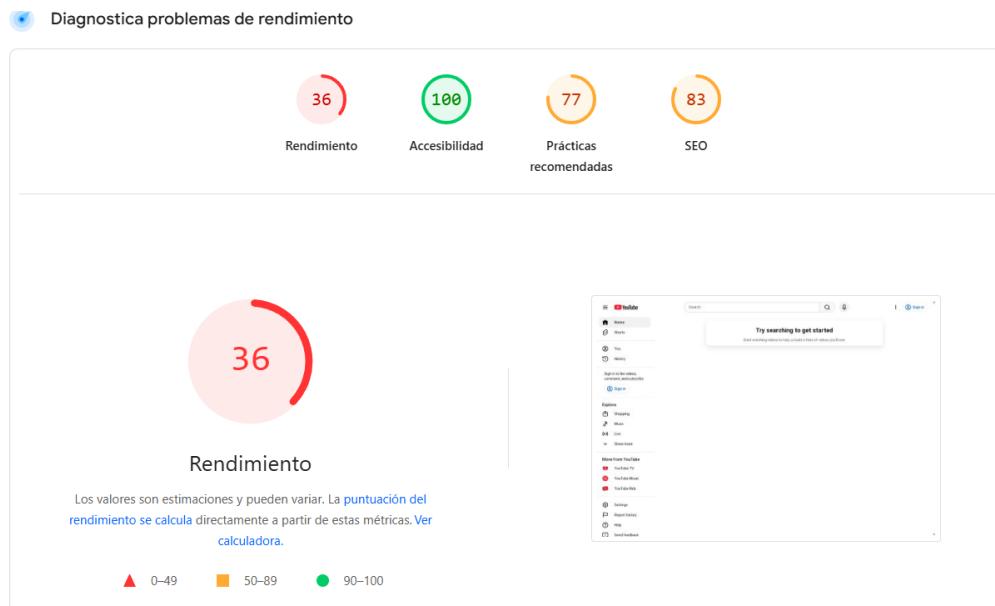


Imagen 7

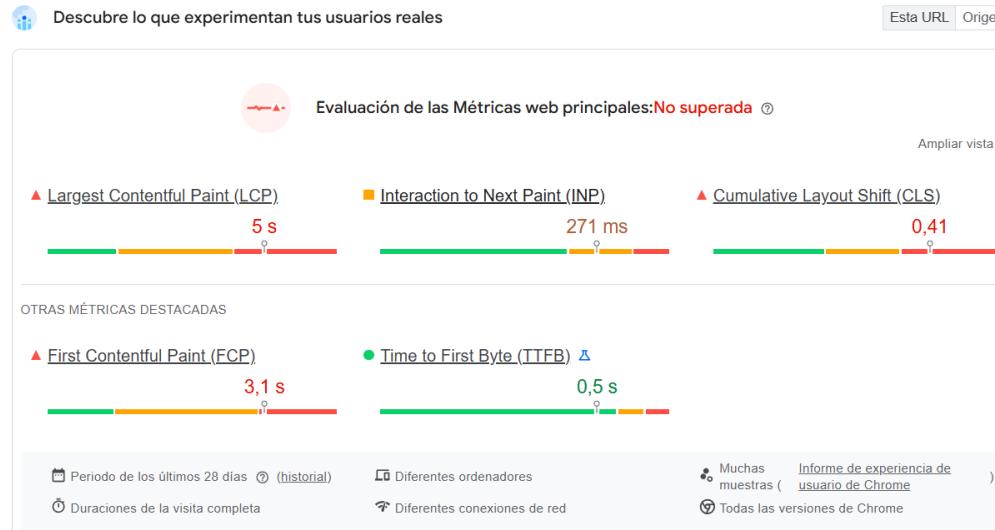


Imagen 8



Imagen 9

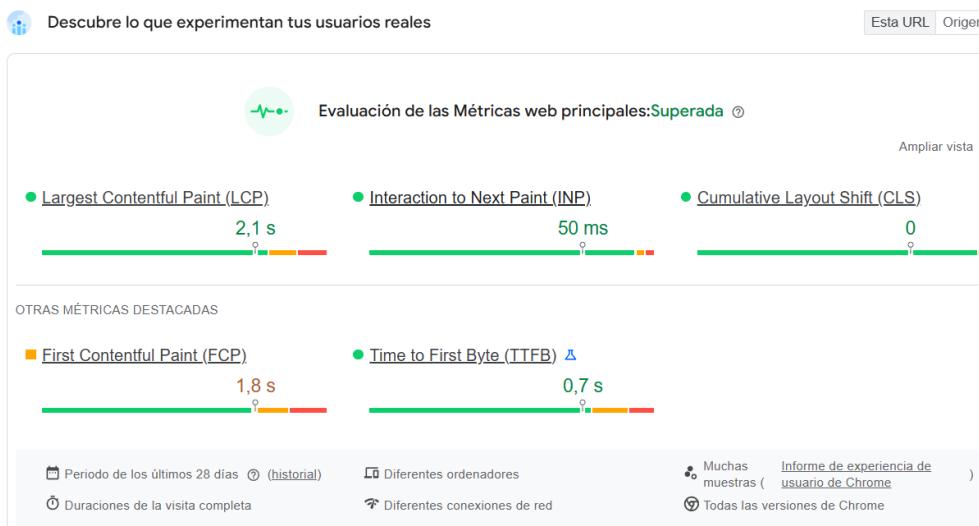


Imagen 10

Plataforma	Usuarios	Average	Median	90 %	95 %	99 %	Min	Max	Error %	Throughput	Rec. KB/s	Sent KB/s
YouTube	100	39777	28962	61221	96200	145895	19158	315625	0.00 %	18.6/min	215.96	0.04
YouTube	500	57318	50867	67083	72251	505058	11936	505502	13.6 %	59.3/min	595.16	0.10
YouTube	1000	131565	127942	166999	172988	212717	12425	213034	60.8 %	4.7/sec	1288.96	0.21
Aula Virtual	100	19747	21656	25012	25646	26230	8475	26324	0.00 %	3.7/sec	1389.24	0.99
Aula Virtual	500	96235	97028	152451	154867	169422	21026	175973	14.2 %	2.8/sec	913.46	0.65
Aula Virtual	1000	43360	22515	153626	190261	212665	21659	213016	89 %	4.7/sec	205.24	0.16

Cuadro 1: Datos de Jmeter

Código en R para métodos de optimización

```

1 usuarios <- c(100, 500, 1000)
2 average <- c(39777, 57318, 131565)
3 error <- c(0, 13.6, 60.8)

4
5 interpolacion_average <- function(average_target, usuarios, average)
6   {
7     for(i in 1:(length(usuarios)-1)) {
8       if(average_target >= average[i] && average_target <= average[i
9         +1]) {
10         U1 <- usuarios[i]
11         U2 <- usuarios[i+1]
12         A1 <- average[i]
13         A2 <- average[i+1]
14         U_target <- U1 + (average_target - A1)*(U2-U1)/(A2-A1)
15         return(U_target)
16     }
17   }
18   return(NA)
19 }

20 regula_falsi <- function(error_target, usuarios, error, tol=1e-3,
21   maxiter=100) {

22   f <- function(U) {
23     for(i in 1:(length(usuarios)-1)) {
24       if(U >= usuarios[i] && U <= usuarios[i+1]) {
25         return(error[i] + (U - usuarios[i]) *
26           (error[i+1]-error[i])/(usuarios[i+1]-usuarios[i]) - error_
27             target)
28       }
29     }
30     return(NA)
31   }

32   a <- min(usuarios)
33   b <- max(usuarios)

34   fa <- f(a)
35   fb <- f(b)

36
37   iter <- 0
38   while(iter < maxiter) {
39     c <- b - fb*(b - a)/(fb - fa)
40     fc <- f(c)

41     if(abs(fc) < tol) return(c)
42
43

```

```

44     if(fc*fa < 0) {
45         b <- c
46         fb <- fc
47     } else {
48         a <- c
49         fa <- fc
50     }
51
52     iter <- iter + 1
53 }
54 return(c)
55 }
56
57 average_obj <- 50000
58 error_obj    <- 5
59
60 U_avg <- interpolacion_average(average_obj, usuarios, average)
61 U_err <- regula_falsi(error_obj, usuarios, error)
62
63 cat("Usuarios segun Average:", round(U_avg), "\n")
64 cat("Usuarios segun Error:", round(U_err), "\n")

```

Resultados

```

==== RESULTADOS ====
Usuarios para Average de 50000 ms: 333
Usuarios para Error de 5%: 247

```

Interpretación de Resultados

Para YouTube:

- **Método Gradiente:** 333 usuarios mantienen tiempo promedio de 50 segundos
- **Método Regula Falsi:** 247 usuarios mantienen error $< 5\%$

Recomendación operativa:

- Operar con máximo 247 usuarios para garantizar calidad
- Entre 247-333 usuarios: tiempo aceptable pero con errores
- Sobre 333 usuarios: degradación severa

Para Aula Virtual:

- Requiere mejoras urgentes de infraestructura
- Considerar balanceadores de carga
- Optimización de base de datos

Comparación de Métodos

Aspecto	Gradiente	Regula Falsi
Propósito	Medir sensibilidad	Encontrar límites
Pregunta	¿Cómo cambia?	¿Dónde está el límite?
Salida	Pendiente	Valor específico
Uso	Ánálisis de tendencias	Decisiones operativas

Son complementarios:

- Gradiente → entender comportamiento
- Regula Falsi → tomar decisiones

Conclusiones

Los métodos numéricos utilizados permiten estimar el punto crítico donde el rendimiento de cada plataforma comienza a deteriorarse. YouTube muestra mayor tolerancia a la carga, mientras que Aula Virtual presenta saturación temprana debido a limitaciones de infraestructura. La combinación de Apache JMeter con técnicas como gradiente y Regula Falsi permite obtener análisis más precisos y útiles para la toma de decisiones.