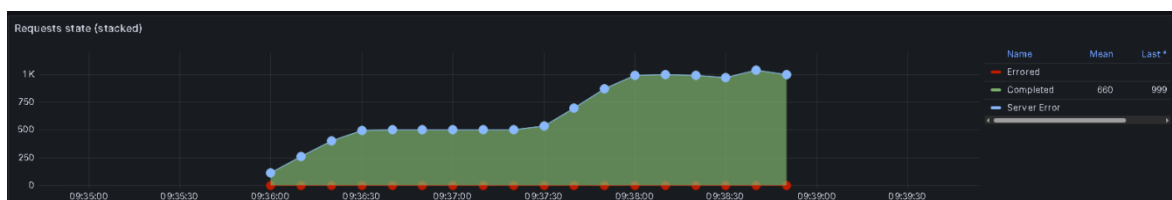


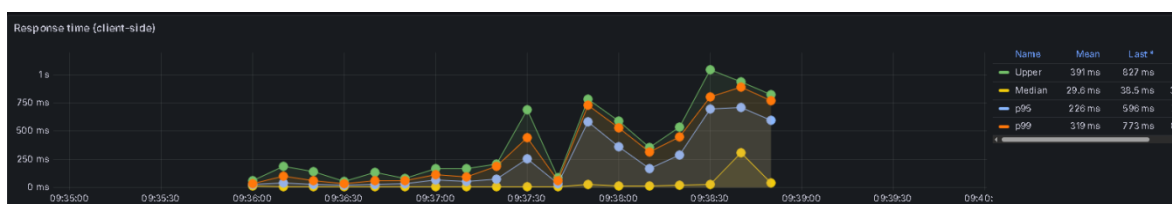
En este documento, se analizan las pruebas realizadas en Artillery, con los gráficos de Grafana. Se revisan las pruebas para cada endpoint, y para las versiones v1 (Original) y v1.1 (Version con mejoras)

Para los GET Accounts, GET Rates y PUT Rates, todas las solicitudes son procesadas correctamente, y no hay solicitudes con errores y todas siguen la misma forma en su gráfico. Ejemplo con GET Accounts:

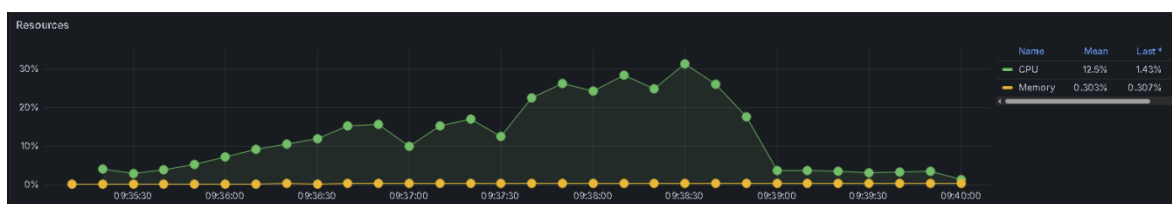


GET_ACCOUNTS_V1

Podemos ver que casi todas las métricas siguen la misma tendencia: Primero tienen un pico, luego se “planchan”, y luego se repite. Los valores rondan entre los 500ms y 1s.

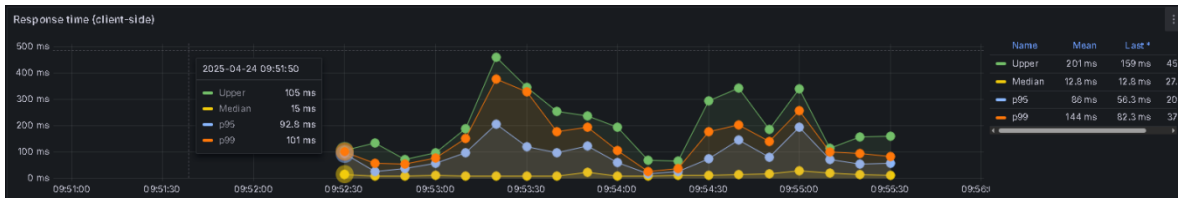


El uso de CPU no es significativo

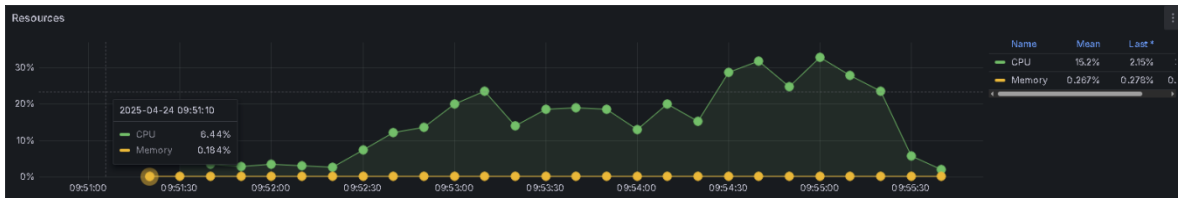


GET_ACCOUNTS_V1.1

A diferencia del otro, en este gráfico vemos que se comienza con grandes picos, se van “planchando”, y luego los picos son mas pequeños, en general. Además si se ve bien, los valores no superan los 500ms, a diferencia del V1 donde los valores podían estar entre los 500ms y 1s

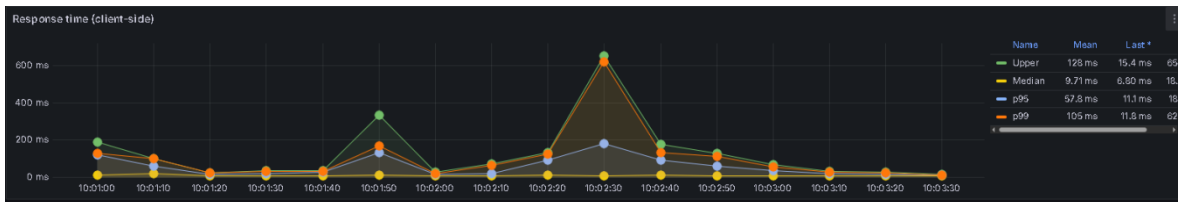


El uso de CPU es similar al de la versión v1.

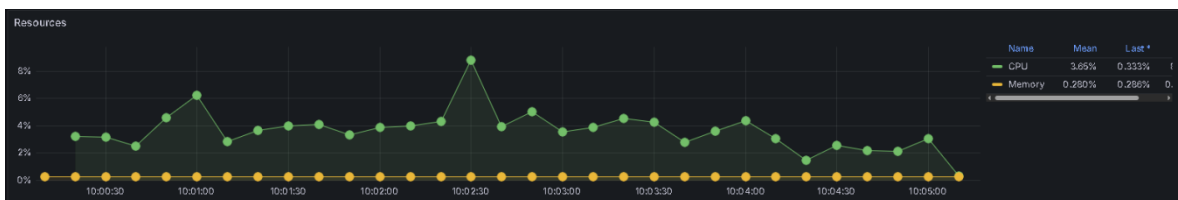


GET RATES V1

Acá se nota que pasa algo parecido que con el GET Accounts v1, con la diferencia de que los picos son esporádicos pero en general el gráfico está bastante plano y no se mueve mucho. Los valores como mucho llegan a casi 400ms y a 600ms.



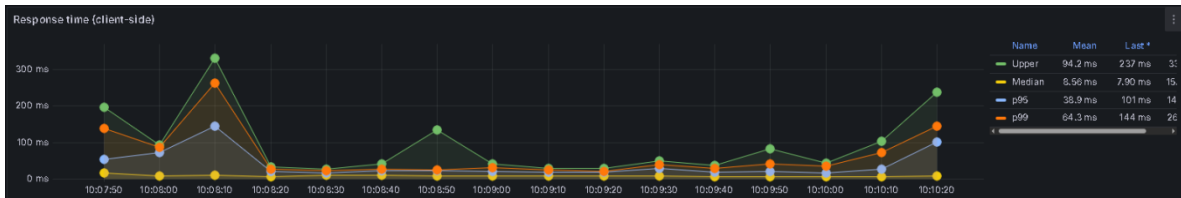
El uso de CPU es muy pequeño, con pocos picos.



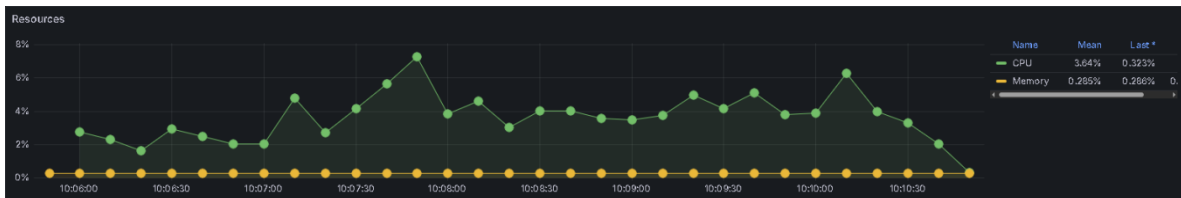
GET RATES V1.1

Vemos algo parecido a lo de GET Accounts v1.1, vemos que los picos decrecen en vez de crecer con el tiempo (En general), y en general el grafico está “planchado”. Además también notamos la

mejora en tiempos, pues el mayor pico no supera los 300ms, mientras que en el gráfico de GET Accounts v1, uno de los picos más pequeños era de 300ms



El uso de CPU tiene picos un poco mas pequeños a comparación del v1.



GET LOG V1

Con un log de este tamaño (989 bytes)

TP 1 - arVault / Get Log

GET `http://{{host}}:{{port}}/{{version}}/log` Send

Params Authorization Headers (6) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (7) Test Results

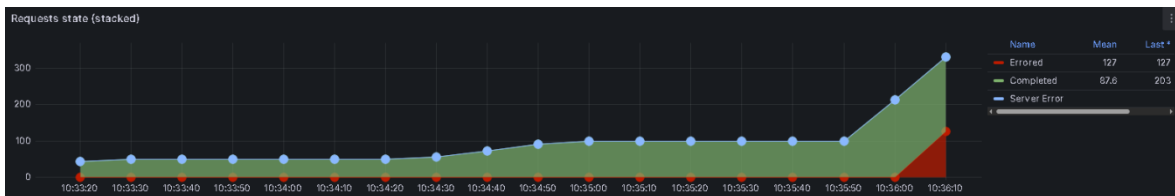
200 OK • 130 ms • 989 B • Save Response

JSON Preview Visualize

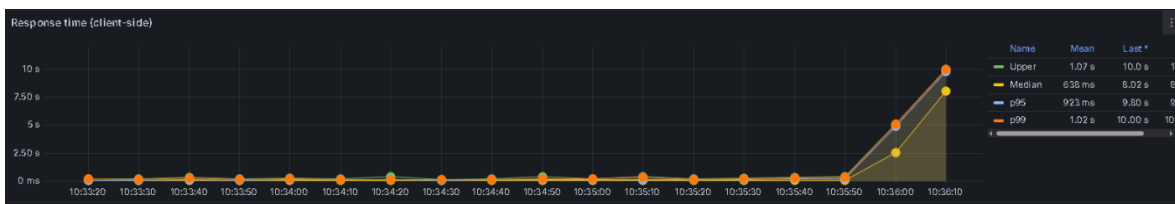
```
1 [
2   {
3     "id": "ZHR5G2hKOHUhtNRuBIAf",
4     "ts": "2025-04-24T02:26:25.615Z",
5     "ok": true,
6     "request": {
7       "baseCurrency": "ARS",
8       "counterCurrency": "USD",
9       "baseAmount": 100,
```

Response Size: 989 B
Headers: 236 B
Body: 753 B
Request Size: 207 B
Headers: 207 B
Body: 0 B

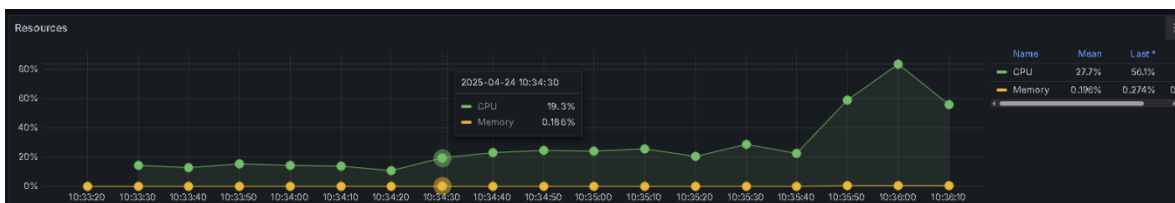
Vemos que empiezan a surgir errores en el tramo final de las pruebas. No es como en los demás GET que se procesa todo correctamente.



Los response time están “planchados” pero aumentan mucho en el tramo final también.



Y el uso de CPU también se hace muy significativo al final



GET LOG V1.1

Con un log de este tamaño

[HTTP](#)
TP 1 - arVaul / **Get Log**

Save

Share

GET

http:// {{host}} : {{port}} / {{version}} /log

Send

Params

Authorization

Headers (6)

Body

Scripts

Tests

Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (7)

Test Results

🕒

200 OK

130 ms

989 B

🌐

📄 Save Response

...

{ } JSON

▶ Preview

🔍 Visualize

▼

1

2

3

4

5

6

7

8

9

[

{

"id": "ZHR5G2hKOHUjtNRuBIAf",

"ts": "2025-04-24T02:26:25.615Z",

"ok": true,

"request": {

"baseCurrency": "ARS",

"counterCurrency": "USD",

"baseAmount": 100,

Response Size

989 B

Headers

236 B

Body

753 B

Request Size

207 B

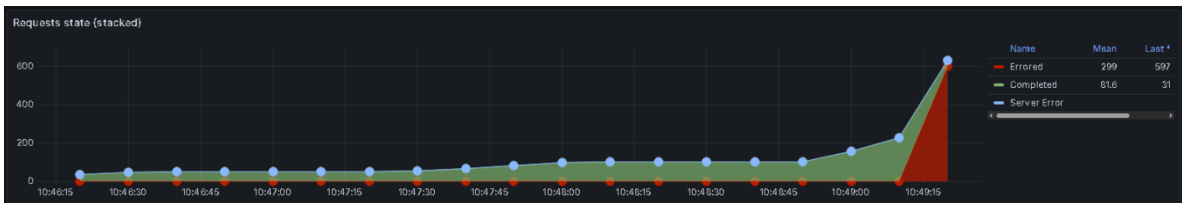
Headers

207 B

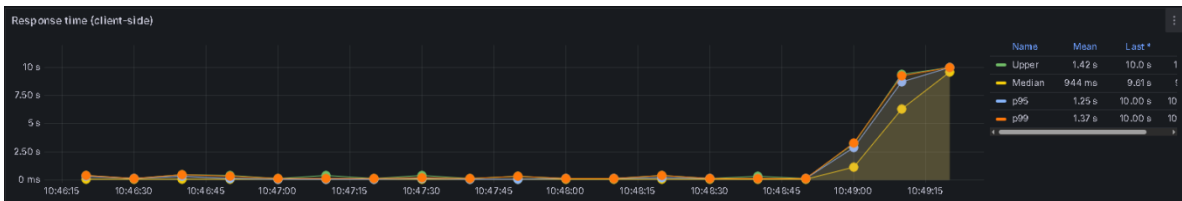
Body

0 B

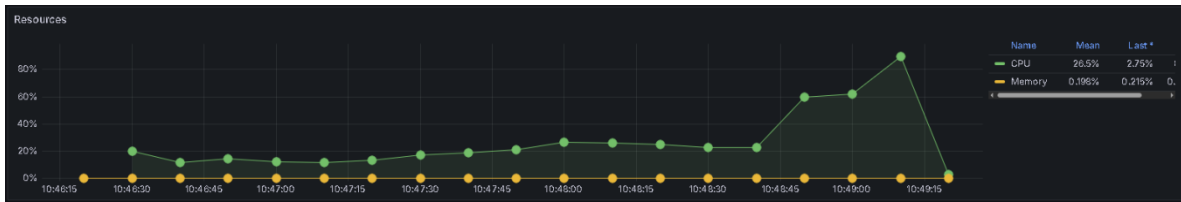
En este gráfico se ve que la tasa de errores es mas alta en esta version



Los tiempos desde el lado del cliente son parecido a los de la v1.

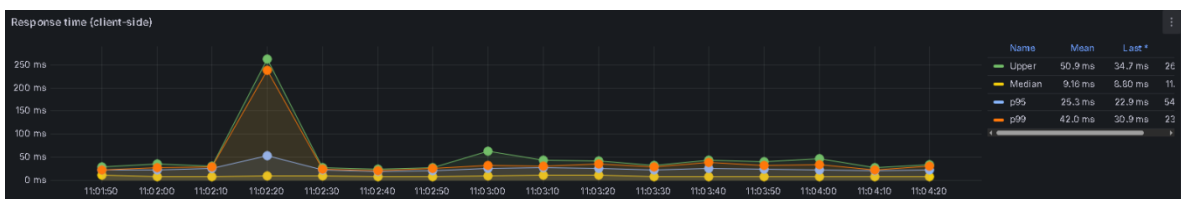


Y el uso de CPU es parecido que en el v1.

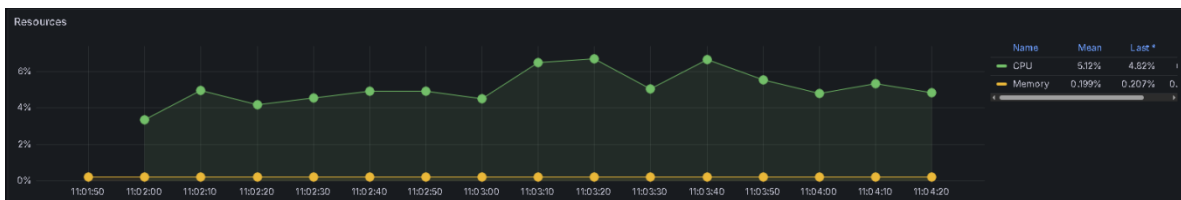


PUT RATES V1

Hay un pico que alcanza aproximadamente los 250ms pero en general el gráfico está mas “planchado”

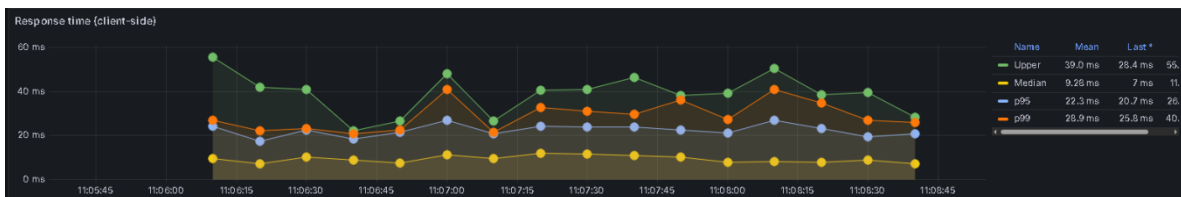


Y el uso de CPU es menor.

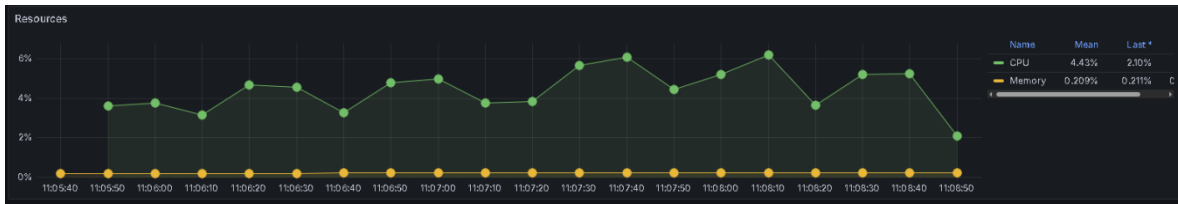


PUT RATES V1.1

A diferencia del v1, este gráfico tiene mas movimiento, pero los tiempos son mucho mejores, el pico está entre los 60ms.

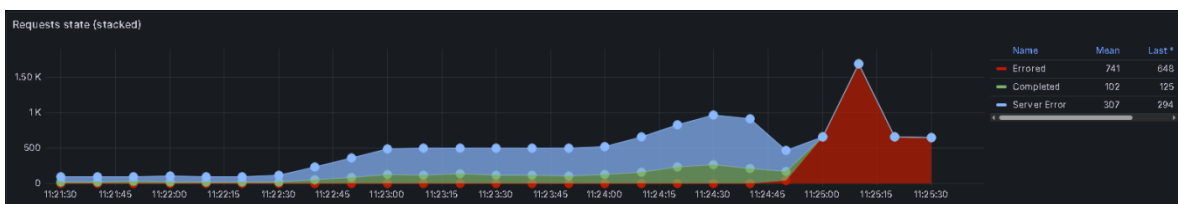


Y el uso de CPU es parecido al de la versión v1.

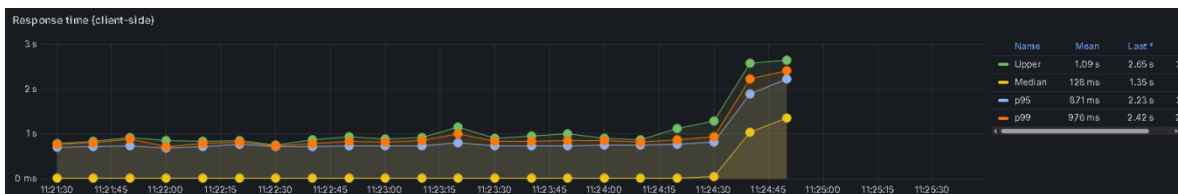


POST EXCHANGE V1

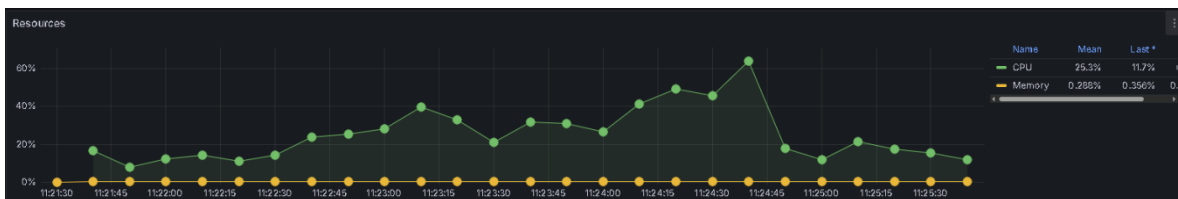
En estas pruebas, la tasa de error al final es muy recurrente, y también en el medio hay muchos errores de servidor.



Los tiempos se mantienen en el segundo, pero en el tramo final aumentan de forma importante.

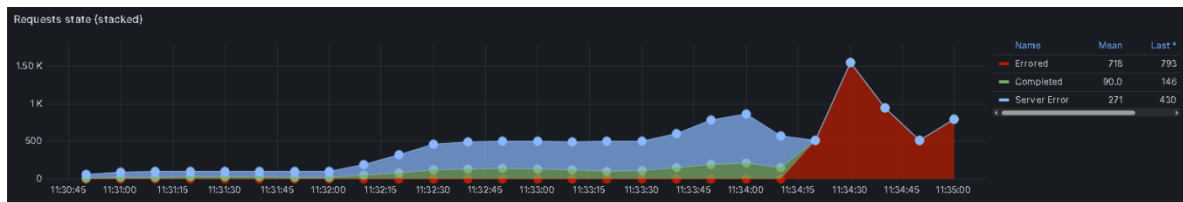


Y el uso de CPU es alto pero se mantiene en niveles excepto por algún pico.

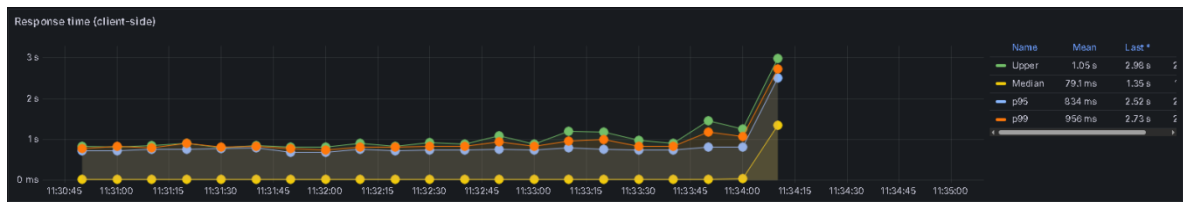


POST EXCHANGE V1.1

La tasa de error no es muy distinta a la de v1



No hay mucha diferencia con el v1 en tiempos



Y el uso de recursos es similar al de v1

