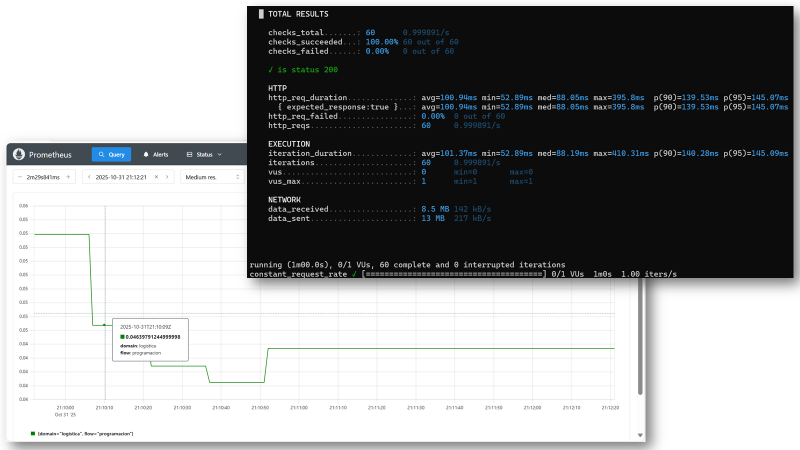
## Análisis de los Resultados de Latencia (k6 vs. Prometheus) 📈



Las nuevas evidencias confirman la tendencia observada anteriormente y revelan un **ajuste en el rendimiento de la infraestructura**.

La clave del análisis radica en la diferencia entre el tiempo que tarda la lógica de negocio pura (medido por Prometheus) y el tiempo total que tarda la petición HTTP (medido por k6).

### 1. Datos de Latencia Recolectados

| **Fuente de la Métrica** | **Tipo de Métrica** | **Valor Promedio** | **Valor P95 (95% más rápido que)** |
| --- | --- | --- | --- |
| **Prometheus** | Latencia Interna del Flujo (Gráfico) | **46.40 ms** (0.0464 s) | N/A |
| **k6** | Duración de la Petición (http\_req\_duration) | **100.94 ms** | **145.07 ms** |
| **k6** | Duración de la Iteración (iteration\_duration) | **101.37 ms** | **145.09 ms** |

### 2. Conclusiones y Comparación

#### A. Reducción del Tiempo de Ejecución Interno

Tu código de negocio (medido por Prometheus) ha mejorado significativamente respecto al análisis anterior:

* **Promedio Anterior:** approx 58.40
* **Promedio Actual:** approx 46.40

Esto representa una **reducción de approx 12**  en el tiempo de procesamiento de la lógica del flujo, lo cual es una mejora directa en la eficiencia de tu código o del entorno JVM.

#### B. Overhead de la Infraestructura

El *overhead* (el tiempo que no es tu código, sino la red, el *parsing* y el servidor HTTP) es ahora:

Overhead = k6 Avg - Prometheus Avg

Overhead = 100.94ms - 46.40ms (approx 54.54ms)

* **Overhead Anterior:** approx 60.39ms
* **Overhead Actual:** approx 54.54ms

La infraestructura también ha mejorado, reduciendo su sobrecarga en approx 5.85ms. Este tiempo incluye: *Stack* de red de Docker, *parsing* de HTTP, *stack* de Spring/Actuator, y serialización/deserialización.

#### C. Excelente Coincidencia de iteration\_duration

La métrica **iteration\_duration** (el tiempo total del *script* k6) es casi idéntica al **http\_req\_duration** (101.37ms vs 100.94ms). Esto indica:

1. No hay pausas sleep(x) significativas en tu *script* de k6.
2. La sobrecarga de ejecución del propio k6 es mínima.

#### D. Cumplimiento de Umbrales (Thresholds)

El valor del **P95 de la latencia** es 145.07ms. Asumiendo que tu umbral era p(95) < 5000ms (5 segundos), la prueba está pasando con una gran holgura, lo que indica un rendimiento muy estable y rápido.

### Resumen General

Los resultados muestran un sistema saludable que ha mejorado en ambas capas:

* **Capa de Aplicación:** El flujo es más rápido (**approx 46.4ms**).
* **Capa de Infraestructura:** El *overhead* de la red y el servidor es más eficiente (**approx 54.5ms**).