**INFORME DE PRUEBAS DE CARGA**

**{d.nombreProyecto}**

***Fecha de inicio:*** *{d.fechaInicio}*

***Fecha de finalización:*** *{d.fechaFin}*

**EQUIPO:**

{d.equipo[i].nombre}

{d.equipo[i+1]}

# 

# 1. INTRODUCCIÓN

En todo sistema desarrollado, como parte del ciclo de vida de desarrollo de software SLDC, este debe ser probado mediante situaciones que simulen la generación de demandas virtuales, con el objetivo de ver y analizar el comportamiento del sistema con diferentes niveles de carga y tráfico, lo cual es conocido como pruebas de carga.

Realizar las pruebas de carga permite que el sistema pueda ofrecer una mejor experiencia por parte de los usuarios. Las métricas obtenidas realizan un aporte en la toma de decisiones respecto a las necesidades de infraestructura y optimización en el rendimiento.

La herramienta K6, permite realizar diferentes pruebas de performance, la cual ofrece la opción de escribir el código de las pruebas en lenguaje Javascript, configurar escenarios de ejecución e integración con herramientas de observabilidad.

# 2. TIPO DE PRUEBA

De acuerdo al análisis realizado en el proyecto “{d.nombreProyecto}”, se identificaron zzzz API’s de uso frecuente de acuerdo a la lógica de negocio:

* aaaaaa
* bbbbbb

**Especificaciones técnicas de los equipos**

Las pruebas de carga se realizaron en un ambiente de pruebas, a continuación  se describen las características de los servidores.

| Instancia | CPU | RAM | DISCO | Número de réplicas |
| --- | --- | --- | --- | --- |
| gestion-reclamos-backend-load-bolivia-a-tu-servicio-deploybzng2 | 1 CORE | 256 Mb | Estático | 1 |

El código desarrollado para las pruebas de carga y la configuración de escenarios se encuentra en:

<https://gitlab.agetic.gob.bo/agetic/ciudadania/gestion/-/tree/main/performance/notificador_avisos>

En el repositorio se encuentra la configuración de 2 escenarios:

* Pruebas de carga,evalúa el rendimiento del sistema respecto a usuarios concurrentes y peticiones por segundo. (5 test)
* Pruebas de estrés, evalúa el rendimiento y comportamiento del sistema bajo condiciones extremas. (1 test)

# 3. ANÁLISIS DE RESULTADOS

Producto de la ejecución de los scripts de pruebas de carga con K6 se obtuvieron los siguientes resultados en los escenarios descritos anteriormente:

## {#d.pruebas}APIS PRINCIPALES

## {.nombre}

{.url}

### Pruebas de carga

#### Escenario {d.pruebas[i].escenarios[j].orden}



{d.pruebas[i].escenarios[j].imagen;image}

{d.pruebas[i].escenarios[j].vus} / {d.pruebas[i].escenarios[j].tiempo}

**Análisis.**

Al término de la ejecución de las pruebas de carga, las métricas más importantes son:

**iterations.** {d.pruebas[i].escenarios[j].iteraciones}

**http\_reqs.** {d.pruebas[i].escenarios[j].httpReqs}

**http\_req\_duration.** {d.pruebas[i].escenarios[j].httpReqDuration}

**http\_req\_blocked.** {d.pruebas[i].escenarios[j].httpReqBlocked}

#### Escenario {d.pruebas[i].escenarios[j+1].orden}

1. {d.pruebas[i].escenarios[j+1].imagen;image}
2. {d.pruebas[i].escenarios[j+1].vus} / {d.pruebas[i].escenarios[j+1].tiempo}

**Análisis.**

Al término de la ejecución de las pruebas de carga, las métricas más importantes son:

**iterations.** {d.pruebas[i].escenarios[j+1].iteraciones}

**http\_reqs.** {d.pruebas[i].escenarios[j+1].httpReqs}

**http\_req\_duration.** {d.pruebas[i].escenarios[j+1].httpReqDuration}

**http\_req\_blocked.** {d.pruebas[i].escenarios[j+1].httpReqBlocked}