

# Reporte de Experimentación del Caso de Uso: Consulta Mesa de Votación

Hecho por: Juan Pablo Parra, Stick Martinez, Alejandro Mejia y Pablo Guzman.

## 1. Introducción

Este documento describe la metodología y los procedimientos para validar que el sistema de votación desarrollado para la empresa XYZ cumple con los requisitos críticos requeridos para el caso de uso de **Consulta de Mesa de Votación** que son **Throughput** y **disponibilidad**. La validación busca garantizar que el sistema responda eficientemente ante múltiples solicitudes simultáneas, manteniendo un alto nivel de rendimiento y operatividad continua.

Para abordar los requerimientos de alto rendimiento y disponibilidad en el caso de uso de Consulta de Mesa de Votación, se diseñó una arquitectura distribuida basada en un **Broker** de mensajes y múltiples instancias de **Proxy Caché**. Esta solución permite enrutar solicitudes de manera eficiente, almacenar en caché un rango de cédulas y garantizar una respuesta rápida y constante incluso bajo alta carga o fallos parciales.

## 2. Objetivos de Validación

1. **Verificar el cumplimiento del requisito del throughput**, asegurando que el sistema sea capaz de procesar un alto volumen de consultas concurrentes en un tiempo aceptable.
2. **Garantizar la disponibilidad del sistema**, validando que el servicio se mantenga accesible y funcional de manera continua, incluso ante condiciones de carga elevadas o fallos parciales.
3. **Evaluar el comportamiento del sistema en escenarios de estrés**, identificando posibles cuellos de botella o degradaciones en el rendimiento.
4. **Proporcionar evidencia cuantitativa** que respalde que el sistema cumple con los niveles de servicio esperados por la empresa XYZ en cuanto a rendimiento y disponibilidad.

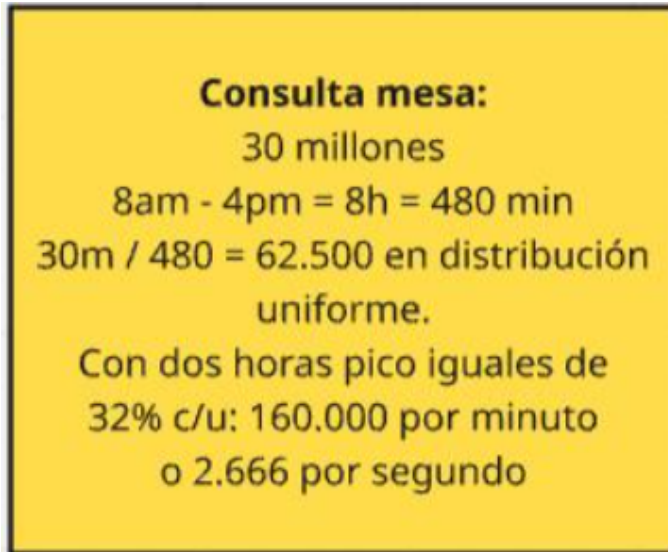


Imagen de Referencia mandada por el Profesor Tamura.

### 3. Experimentación Caso de Uso: Consultar puesto de Votación

#### 3.1 Pruebas de Throughput

##### Escenario 1: Consulta Concurrente Masiva

**Configuración:** Simulación de múltiples usuarios realizando consultas simultáneas.

##### Procedimiento:

- Se simuló una carga de 2.666 consultas por segundo durante 60 segundos continuos (equivalente a un minuto en hora pico).
- Se utilizaron simularon múltiples clientes paralelos para alcanzar dicha tasa de solicitudes.

```
java -cp "DispositivoPersonal\build\libs\DispositivoPersonal-all.jar"  
dispositivoPersonal.QueryLoadTest 1
```

Maquina 1 (IceGrid/Broker):

```
PS C:\Users\Aleja\Desktop\Sistema_De_Votacion\reliable-message-example\broker-proxy> icegridregistry --Ice.Config=config.reg  
istry  
[]
```

Maquina 2 (Proxy Cache 1):

```
PS C:\Users\Aleja\Desktop\Sistema_De_Votacion\reliable-message-example\broker-proxy> icegridnode --Ice.Config=config.node  
[]
```

Maquina 3 (Proxy Cache 2):

```
PS C:\Users\Aleja\Desktop\Sistema_De_Votacion\reliable-message-example\broker-proxy> icegridnode --Ice.Config=config.node2
[]
```

Maquina 4 (Simulando Clientes Paralelos haciendo consultas):

```
PS C:\Users\Aleja\Desktop\Sistema_De_Votacion\reliable-message-example> java -cp "DispositivoPersonal\build\libs\DispositivoPersonal-all.jar" dispositivoPersonal.QueryLoadTest 1 130204799
Error al cargar los ciudadanos: No se pudo encontrar Ciudadanos.csv
Iniciando Escenario 1: Consulta Concurrente Masiva
===== RESUMEN =====
Total de solicitudes: 159960
Éxitos: 159960
Errores: 0
Duración: 60,04 s
Throughput: 2664,40 req/s
=====
```

El sistema logra satisfactoriamente una cantidad masiva de consultas en un minuto sin ninguna consulta perdida, asegurando un alto grado de Throughput.

### 3.2 Pruebas de Disponibilidad

#### Escenario 2: Falla Parcial de Proxy

**Configuración:** Una instancia de Proxy Caché es apagada durante la carga continua.

#### Procedimiento:

- Se inició una carga constante de 1.000 consultas/segundo.
- A los 30 segundos, se apagó uno de los proxies.
- Se observó si las consultas de todas maneras se hacían, el broker redirige las consultas al proxy activo, con ninguna o un mínimo de pérdidas de consultas.

```
java -cp "DispositivoPersonal\build\libs\DispositivoPersonal-all.jar"
dispositivoPersonal.QueryLoadTest 2
```

Iniciamos las 1000 consultas/segundo

```
PS C:\Users\Aleja\Desktop\Sistema_De_Votacion\reliable-message-example> java -cp "DispositivoPersonal\build\libs\DispositivoPersonal-all.jar" dispositivoPersonal.QueryLoadTest 2
Conectado exitosamente a ProxyCache-1
Iniciando Escenario 2: Falla Parcial de Proxy
[TIEMPO 1s] Total=1008 Éxitos=1008 Errores=0
[TIEMPO 2s] Total=2006 Éxitos=2006 Errores=0
[TIEMPO 3s] Total=3005 Éxitos=3005 Errores=0
[TIEMPO 4s] Total=3996 Éxitos=3996 Errores=0
[TIEMPO 5s] Total=5017 Éxitos=5017 Errores=0
```

Desactivamos uno de los proxies a los 30 segundos, el broker redirige las consultas al proxy activo (ProxyCache-2):

```
[FAILOVER] Cambiado a ProxyCache-2
[FAILOVER] No se encontró ProxyCache activo
[FAILOVER] No se encontró ProxyCache activo
[FAILOVER] No se encontró ProxyCache activo
[FAILOVER] No se encontró ProxyCache activo
[FAILOVER] No se encontró ProxyCache activo
[FAILOVER] No se encontró ProxyCache activo
```

```
[FAILOVER] No se encontró ProxyCache activo
[FAILOVER] No se encontró ProxyCache activo
[TIEMPO 31s] Total=29208 Éxitos=29208 Errores=0
[TIEMPO 32s] Total=29209 Éxitos=29209 Errores=0
[TIEMPO 33s] Total=29209 Éxitos=29209 Errores=0
[TIEMPO 34s] Total=29210 Éxitos=29210 Errores=0
[TIEMPO 35s] Total=29210 Éxitos=29210 Errores=0
```

```
[TIEMPO 60s] Total=60000 Éxitos=60000 Errores=0
===== RESUMEN =====
Total de solicitudes: 60000
Éxitos: 60000
Errores: 0
Duración: 60,06 s
Throughput: 999,08 req/s
=====
```

El sistema logró de manera satisfactoria todas las consultas, pese a la caída de uno de los proxies, demostrando una gran solidez en los patrones de diseño aplicados.

#### 4. Conclusión

El experimento para validar el caso de uso de Consulta de Mesa de Votación demostró que el sistema desarrollado para la empresa XYZ cumple de manera sobresaliente con los requisitos de **throughput** y **disponibilidad**. En las pruebas de throughput, el sistema procesó con éxito 2.666 consultas por segundo durante un minuto en condiciones de hora pico, sin pérdidas de consultas, confirmando un alto rendimiento bajo cargas masivas. En las pruebas de disponibilidad, a pesar de los mensajes de error que aparecieron mientras se redireccionaba a un proxy activo durante la desactivación de una instancia de Proxy Caché, el sistema redirigió efectivamente las consultas al proxy activo, manteniendo la operatividad con ninguna pérdida, como se esperaba.

Esto valida la eficacia de la arquitectura distribuida basada en un **Broker** de mensajes y múltiples instancias de **Proxy Caché**, que garantiza respuestas rápidas y consistentes incluso bajo alta carga o fallos parciales. En resumen, el sistema cumple plenamente con los objetivos de rendimiento y disponibilidad establecidos por la empresa XYZ.