

# Diseño del Experimento - Validación Sistema de Votaciones ZeroC ICE

## Validación Experimental de Confiabilidad y Unicidad en Sistema de Votaciones Distribuido

**Responsables:** Equipo de Desarrollo - Empresa XYZ

**Cliente:** Registraduría Nacional

**Módulo a Evaluar:** Sistema de transmisión y consolidación de votos mediante ZeroC ICE

**Contexto:** Validación pre-producción para garantizar integridad electoral

**Equipo de desarrollo:**

- Juan Sebastian Diaz
- Julian Mendoza
- Geoffrey Pasaje
- Juan Jose Angarita
- Raul Quigua

**Configuración Experimental:**

- **Infraestructura:** IASLab - 28 workers (x104m01 a x104m28)
- **Tecnología:** ZeroC ICE middleware
- **Carga de Prueba:** 500,000 requests distribuidos
- **Arquitectura:** 1 master + 28 workers distribuidos

---

## Introducción

El presente documento establece el diseño experimental para validar los componentes críticos del sistema de votaciones electrónicas desarrollado para la Registraduría Nacional. El sistema debe garantizar dos requisitos fundamentales: **confiabilidad absoluta** (100% de votos registrados sin pérdida) y **unicidad estricta** (prevención total de duplicación de votos).

El sistema utiliza arquitectura distribuida basada en ZeroC ICE, donde múltiples estaciones de votación (workers) transmiten votos a un servidor central (master) que consolida y valida cada transacción. La criticidad del sistema requiere validación experimental exhaustiva antes del despliegue en procesos electorales reales.

La relevancia de este experimento radica en que cualquier falla en confiabilidad o unicidad compromete directamente la legitimidad democrática del proceso electoral, haciendo indispensable la certificación científica del sistema mediante pruebas controladas.

---

## Objetivo del Experimento

**Objetivo General:** Validar experimentalmente que el sistema de votaciones distribuido cumple con los requisitos críticos de confiabilidad y unicidad bajo condiciones de carga real.

**Objetivos Específicos:**

1. Comprobar que el 100% de los votos transmitidos son registrados correctamente por el sistema
2. Verificar que el sistema detecta y rechaza el 100% de intentos de duplicación de votos
3. Evaluar el comportamiento del sistema bajo carga distribuida de 500,000 transacciones
4. Medir la capacidad de procesamiento concurrente con 28 workers simultáneos

## Planteamiento de Hipótesis

### Hipótesis de Confiabilidad

- **H<sub>0</sub> (Nula):** El sistema presenta pérdida de votos  $\geq 0.01\%$  durante transmisión distribuida
- **H<sub>1</sub> (Alternativa):** El sistema registra el 100% de votos transmitidos sin pérdida ( $< 0.01\%$ )

### Hipótesis de Unicidad

- **H<sub>0</sub> (Nula):** El sistema acepta  $\geq 1$  voto duplicado durante pruebas de concurrencia
- **H<sub>1</sub> (Alternativa):** El sistema rechaza el 100% de votos duplicados independientemente de la concurrencia

### Hipótesis de Rendimiento

- **H<sub>0</sub> (Nula):** El throughput del sistema es  $< 200$  votos/segundo bajo carga distribuida
  - **H<sub>1</sub> (Alternativa):** El sistema mantiene throughput  $\geq 200$  votos/segundo con 28 workers
- 

## Metodología de Evaluación

### Métricas Primarias

#### Tasa de Registro de Votos (TRV)

$$\text{TRV} = (\text{Votos Registrados Exitosamente} / \text{Total Votos Enviados}) \times 100$$

- **Criterio de Aceptación:** TRV = 100%
- **Umbral de Fallo:** TRV  $< 99.99\%$

#### Tasa de Detección de Duplicados (TDD)

$$\text{TDD} = (\text{Duplicados Detectados} / \text{Duplicados Enviados}) \times 100$$

- **Criterio de Aceptación:** TDD = 100%
- **Umbral de Fallo:** TDD  $< 100\%$  (cero tolerancia)

#### Throughput de Procesamiento (TPP)

$$\text{TPP} = \text{Total Votos Procesados} / \text{Tiempo Total de Ejecución (votos/segundo)}$$

- **Criterio de Aceptación:** TPP  $\geq 200$  votos/segundo
- **Umbral de Fallo:** TPP  $< 150$  votos/segundo

#### Tiempo de Respuesta Promedio (TRP)

$$\text{TRP} = \Sigma(\text{Timestamp\_ACK} - \text{Timestamp\_Envío}) / \text{Número de Transacciones}$$

- **Criterio de Aceptación:** TRP  $\leq 2$  segundos
  - **Umbral de Fallo:** TRP  $> 5$  segundos
- 

## Identificación de Variables

## Variables Independientes (Manipuladas)

- **Número de workers activos:** {10, 28}
- **Carga de votos por worker:** {10k, 17.8k, 25k}
- **Patrón de IDs:** {únicos, duplicados, mixtos}
- **Timing de envío:** {sincronizado, asíncrono}

## Variables Dependientes (Observadas)

- **Votos registrados exitosamente** (conteo absoluto)
- **Votos duplicados detectados** (conteo absoluto)
- **Tiempo de respuesta por transacción** (milisegundos)
- **Throughput del sistema** (votos/segundo)
- **Utilización de recursos** (CPU, memoria, red)

## Variables Controladas (Constantes)

- **Hardware:** Hosts x104m01-x104m28 del IASLab
- **Software:** ZeroC ICE versión específica
- **Configuración de red:** Misma subred, sin latencia artificial
- **Estructura de votos:** Formato estándar (ID, Mesa, Elector, Candidato, Timestamp)
- **Método de comunicación:** ZeroC ICE middleware exclusivamente

## Escenarios Experimentales

### Contexto Real de Votación Colombiana:

- **Votantes registrados:** 30 millones
- **Participación esperada:** 20 millones de votos (66.7%)
- **Horario electoral:** 8:00 AM - 4:00 PM (8 horas = 480 minutos)
- **Carga base uniforme:** 41,666 votos/minuto (694 votos/segundo)
- **Picos electorales:** 2 horas con 32% c/u = 106,666 votos/minuto (1,777 votos/segundo)
- **Consultas de mesa:** 62,500/minuto uniforme, picos de 160,000/minuto (2,666/segundo)

### Escalamiento Experimental (500k requests con 28 workers):

- **Factor de escala:** 1:40 respecto al volumen real
- **Carga por worker:** ~17,857 requests (equivalente a ~714k votos reales)
- **Throughput objetivo:** 200-300 votos/segundo (8-12k votos/segundo escalado)

Escenario	Workers	Votos/Worker	Tipo de Falla Inducida	Variables Observadas	Resultado Esperado	Equivalencia Real
E1-Baseline-Uniforme	28	17,857	Ninguna	TRV, TPP, TRP	TRV=100%, TPP≥200, TRP≤2s	Carga electoral uniforme
E2-Pico-Electoral	28	6,250	Ráfaga 5x velocidad	TPP, latencia, estabilidad	TPP≥1000, estabilidad	Pico 32% (1,777/seg real)
E3-Concurrencia-Crítica	28	1	ID duplicado simultáneo	TDD, consistencia	TDD=100%, solo 1 aceptado	Voto duplicado masivo

<b>E4-Carga-Sostenida</b>	28	10,000	8 horas continuas	Degradación, memoria	Sin degradación	Jornada electoral completa
<b>E5-Saturación-Duplicados</b>	28	3,571	50% duplicados	TDD, TPP, recursos	TDD=100%, TPP≥150	Intento fraude masivo
<b>E6-Fallo-Infraestructura</b>	28→20→28	17,857	Desconexión 8 workers	Recuperación, TRV	Recuperación < 30s	Falla centro de cómputo
<b>E7-Consulta-Intensiva</b>	28	25,000	Solo consultas	TPP-consulta, cache	TPP≥300 consultas/seg	Verificación masiva mesa
<b>E8-Cierre-Electoral</b>	28	8,928	Ráfaga final 10x	Buffer overflow, cola	Sin pérdida votos	Último minuto 4:00 PM

#### Validación de Escalabilidad:

- **E1:** Simula carga electoral estándar distribuida
- **E2:** Replica picos reales del 32% de votación concentrada
- **E3:** Valida unicidad bajo máxima concurrencia
- **E4:** Prueba resistencia durante jornada completa
- **E5:** Evalúa detección masiva de intentos fraudulentos
- **E6:** Simula fallas de infraestructura crítica
- **E7:** Valida capacidad de consultas simultáneas de resultados
- **E8:** Prueba el crítico momento de cierre electoral

## Recolección y Análisis de Datos

### Fuentes de Datos

#### Logs del Sistema

- **Master:** `/logs/master_TIMESTAMP.log`
  - Registro de votos recibidos
  - Detección de duplicados
  - Tiempos de procesamiento
- **Workers:** `/logs/worker_x104mXX_TIMESTAMP.log`
  - Votos enviados con timestamp
  - ACKs recibidos
  - Errores de conexión

#### Métricas de Sistema

- **Utilización de recursos:** CPU, memoria, red (cada 5 segundos)
- **Conexiones ZeroC ICE:** Estado de conectividad por worker
- **Latencia de red:** Ping times entre workers y master

#### Formato de Registro

```
[TIMESTAMP] [WORKER_ID] [VOTE_ID] [ACTION] [RESULT] [RESPONSE_TIME]
2024-02-15 10:30:15.123 x104m01 000001 SEND_VOTE SUCCESS 0.156s
2024-02-15 10:30:15.279 MASTER 000001 RECEIVE_VOTE REGISTERED 0.156s
```

## Procedimiento de Análisis

1. **Agregación de datos:** Consolidación de logs por escenario
  2. **Cálculo de métricas:** Aplicación de fórmulas TRV, TDD, TPP, TRP
  3. **Análisis estadístico:** Intervalos de confianza, desviación estándar
  4. **Validación de hipótesis:** Pruebas t-student para significancia
  5. **Generación de reportes:** Gráficos de rendimiento y tablas de resultados
- 

## Conclusión

El diseño experimental presentado establece un marco científico riguroso para validar los aspectos críticos del sistema de votaciones distribuido. Mediante la ejecución de 6 escenarios experimentales controlados con la infraestructura real del IASLab (28 workers, 500k transacciones), se obtendrá evidencia empírica sobre:

**Confiabilidad:** La capacidad del sistema para registrar el 100% de votos sin pérdida bajo condiciones de carga distribuida real.

**Unicidad:** La efectividad del mecanismo de detección de duplicados bajo concurrencia extrema de 28 workers simultáneos.

**Escalabilidad:** El comportamiento del throughput y tiempo de respuesta con la carga operacional prevista.

La validación de las hipótesis planteadas proporcionará la certificación científica necesaria para garantizar la confianza ciudadana en el sistema electoral. Los criterios de aceptación establecidos (TRV=100%, TDD=100%, TPP≥200) representan estándares de excelencia apropiados para la criticidad del sistema.

El experimento generará evidencia cuantificable y auditable que respalde la decisión de despliegue del sistema en procesos electorales reales, cumpliendo con los más altos estándares de calidad y transparencia democrática.