

**Juan David Cruz**  
**Juan Sebastian Martinez**  
**Juliana Montes**  
**Daniel Ordoñez**

## **Arquitectura y Diseño de Software**

### **Sistema de Alerta de Tsunamis**

De acuerdo a la continuación del experimento, a continuación se presentan los resultados de las pruebas de desempeño y escalabilidad realizadas sobre la aplicación una vez construida su capa de persistencia. Los resultados se compararon con aquellos obtenidos en la pre experimentación con el fin de determinar los efectos de la inclusión de serialización en una base de datos.

Para la persistencia, se escogió el uso de JPA dado que presenta mayor facilidad de manejo de entidades en la base de datos, además, el modelo propuesto en el diseño consta de entidades desacopladas y sin relaciones de mayor complejidad que puedan dificultar la consulta de datos o la actualización de los mismos. Con este supuesto, se espera que los atributos de calidad no se vean comprometidos en mayor medida con este esquema.

Vale la pena resaltar que para asegurar una buena escalabilidad y desempeño de la aplicación se trató de diseñar un sistema desacoplado y con entidades sencillas, cercanas a sencillos POJO sin relaciones complejas; todo esto con el fin de favorecer los procesos transaccionales en la base de datos.

Por otra parte, la solución construida a partir de la arquitectura JAX-RS permite manejar Beans sin estado que aceleran las consultas de servicios, especialmente a la hora de agregar mediciones de diferentes sensores concurrentes; además, las experiencias previas con esta arquitectura demuestran que es lo suficientemente robusta para manejar la carga esperada en la aplicación (mediciones de hasta 4000 sensores).

### **Pre-experimentación**

**Problemática:** El monitoreo de la zona costera con el propósito de realizar la detección temprana de un tsunami es vital para la seguridad de las personas que habitan en estos lugares. Dado que el monitoreo debe ser suficiente, se debe procesar eficientemente una gran cantidad de información proveniente de varios sensores, todo con una periodicidad definida y la respuesta debe ser lo suficientemente temprana para evitar un desastre. Con esto se plantea entonces poder procesar información de alrededor de 4000 sensores con una respuesta inferior a 1 minuto. Adicionalmente, En el momento de la generación de un evento sísmico, existen varios tipos de alertas que se pueden generar en cuanto al comportamiento de las olas. Estas alertas sirven para informar a la comunidad sobre las acciones correspondientes a tomar, pero para evitar desastres, es necesario que los boletines de alerta se generen lo más pronto

posible y lo más acertado posible para que la comunidad esté informada oportunamente sobre el estado de la situación. Con esta problemática, se define entonces un tiempo de generación de boletines de máximo 500 milisegundos, después de detectar un evento sísmico.

Con esto en mente, la adición de una capa de persistencia y manejo de una base de datos puede resultar en una afectación de estos criterios de desempeño y escalabilidad, por lo cual se debe hacer un adecuado manejo de estas transacciones y entidades para asegurar el menor impacto posible en relación a la experimentación anterior, donde no se tenía esta capa.

**Objetivo del experimento:** La hipótesis de escalabilidad y desempeño planteada en la problemática se pondrá a prueba con pruebas de carga; con el objetivo de determinar aquellas características clave del sistema que determinan su comportamiento alrededor de estos criterios (tiempos máximos de respuesta en pruebas incrementales para registro de mediciones de sensores y generación de boletines).

**Descripción del experimento:** Para el experimento a realizar, se plantea utilizar la herramienta JMeter para determinar las características de tiempos de respuesta, Throughput y porcentaje de error de la aplicación, para el caso de registro de mediciones de sensores y generación de boletines de información, lo cual incluye la recepción y serialización del evento sísmico correspondiente. Con esto en mente, se plantea realizar para estos casos (registro de sensores y generación de boletines) una prueba de carga incrementar hasta de 4000 threads para caracterizar el sistema.

**Artefactos a construir:** Los únicos elementos de software a utilizar son la herramienta JMeter y el sistema construido (archivo .war a desplegar). No se construyeron artefactos adicionales.

**Recursos de la experimentación:** El software de pruebas a utilizar es JMeter 2.1.3, mientras que el software construido se realizó en NetBeans IDE 8.1 y se desplegó en el servidor Glassfish 4.1.1 en una máquina virtual Windows 7 con 8GB de memoria y procesador Intel Xeon de 2.4GHz y 2 núcleos.

**Resultados esperados:** De acuerdo a las pruebas a realizar, se espera que todos los tiempos máximos de respuesta sean inferiores a 500 milisegundos en la generación de boletines de información, y menores a 1 minuto en el registro de mediciones de sensores para una ventana de 1 minuto, como se establece en los requerimientos de escalabilidad y desempeño para este experimento.

**Duración y etapas:** El experimento constará de 2 pruebas de carga incrementales en varias etapas; donde se consignarán los tiempos de respuesta de la aplicación, Throughput y porcentaje de error tanto para las pruebas de generación de boletines como las de registros de medición de sensores. Las etapas consistirán en el incremento desde 100 hasta 4000 threads consumiendo los servicios de la aplicación, todo con un ramp-up de 60 segundos y loop count

de 1 para todas las etapas. En ningún caso se espera obtener un tiempo de experimentación a los 5 minutos

## Post-experimentación

### Resultados obtenidos

De acuerdo al criterio de escalabilidad, se realizaron pruebas incrementales de 100, 200, 400, 800, 1600, 3200 y 4000 solicitudes de registro de medición de sensores. Todo de acuerdo al ramp-up definido anteriormente. Los resultados se muestran en la siguiente serie de capturas:

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	100	21	18	64	6,64	0,00%	1,7/sec
Total	100	21	18	64	6,64	0,00%	1,7/sec

#### 100 Mediciones

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	200	18	16	77	4,37	0,00%	3,3/sec
Total	200	18	16	77	4,37	0,00%	3,3/sec

#### 200 Mediciones

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	400	17	14	75	5,28	0,00%	6,7/sec
Total	400	17	14	75	5,28	0,00%	6,7/sec

#### 400 Mediciones

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	800	16	13	68	4,77	0,00%	13,3/sec
Total	800	16	13	68	4,77	0,00%	13,3/sec

#### 800 Mediciones

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	1600	17	7	421	27,49	0,00%	26,4/sec
Total	1600	17	7	421	27,49	0,00%	26,4/sec

#### 1600 Mediciones

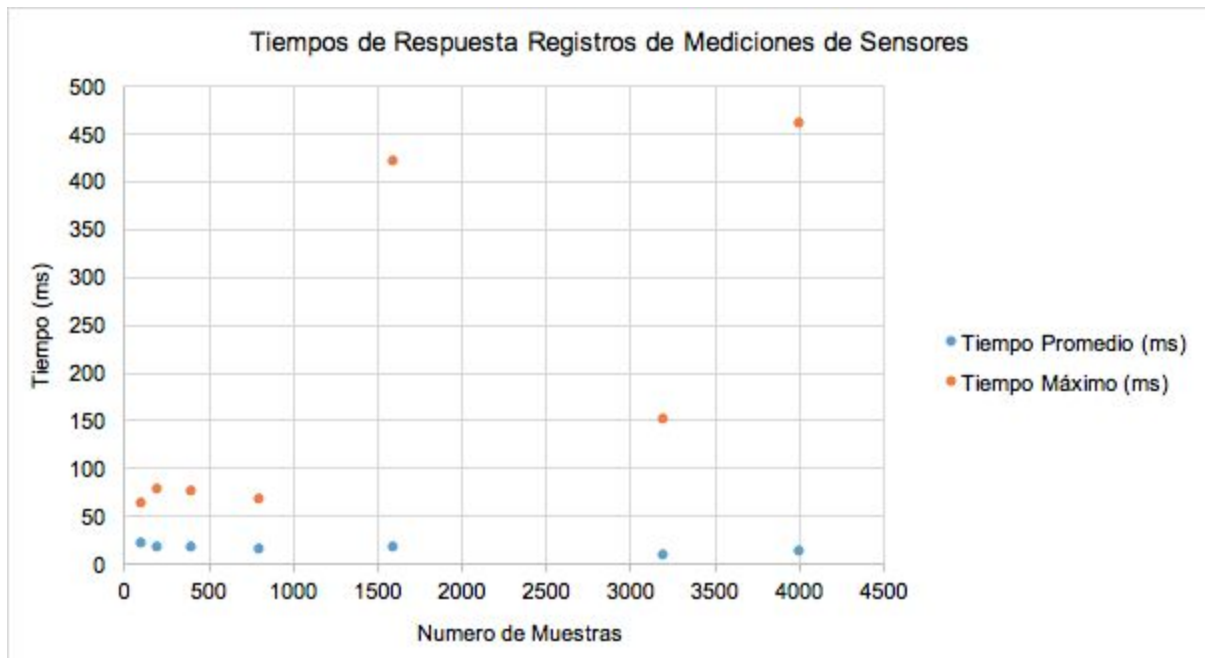
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	3200	10	5	151	7,28	0,00%	52,1/sec
Total	3200	10	5	151	7,28	0,00%	52,1/sec

#### 3200 Mediciones

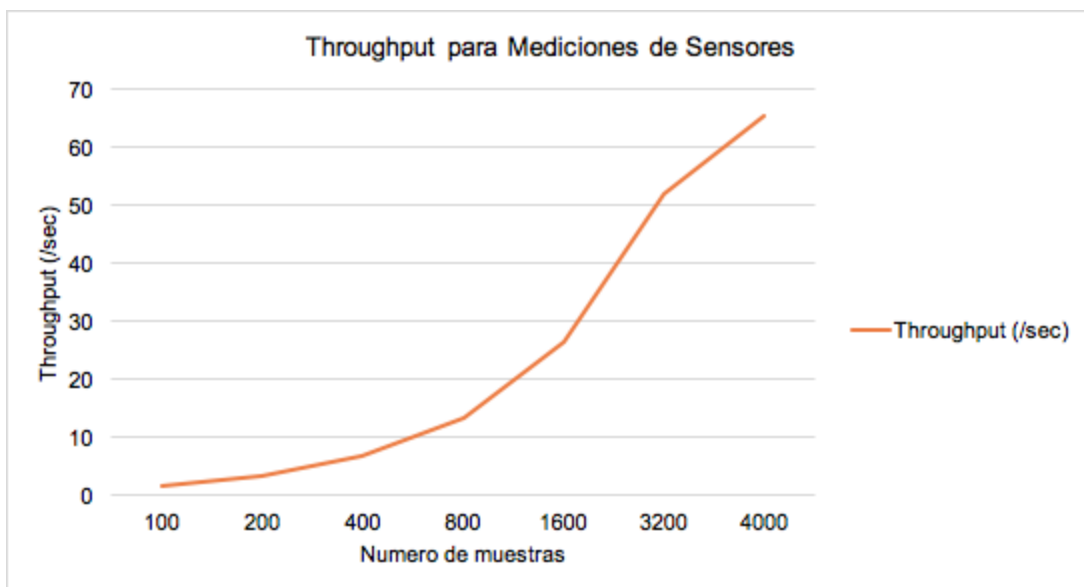
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	4000	13	5	460	23,71	0,00%	65,4/sec
Total	4000	13	5	460	23,71	0,00%	65,4/sec

#### 4000 Mediciones

De acuerdo a estos resultados, se obtuvieron los siguientes comportamientos de tiempos y Throughput para el manejo de altas cargas de sensores:



Comportamiento de tiempos de respuesta para mediciones de sensores.



Comportamiento de Throughput para mediciones de sensores.

Para el caso de la generación de boletines, se realizó el mismo procedimiento anterior, teniendo en cuenta que un boletín se genera a partir de la aparición de un evento sísmico; los resultados fueron:

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	100	59	55	106	9,36	0,00%	1,7/sec
Total	100	59	55	106	9,36	0,00%	1,7/sec

#### Generación de 100 boletines

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	200	58	53	107	7,63	0,00%	3,3/sec
Total	200	58	53	107	7,63	0,00%	3,3/sec

#### Generación de 200 boletines

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	400	57	24	153	9,39	0,00%	6,7/sec
Total	400	57	24	153	9,39	0,00%	6,7/sec

#### Generación de 400 boletines

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	800	47	31	130	9,40	0,00%	13,2/sec
Total	800	47	31	130	9,40	0,00%	13,2/sec

#### Generación de 800 boletines

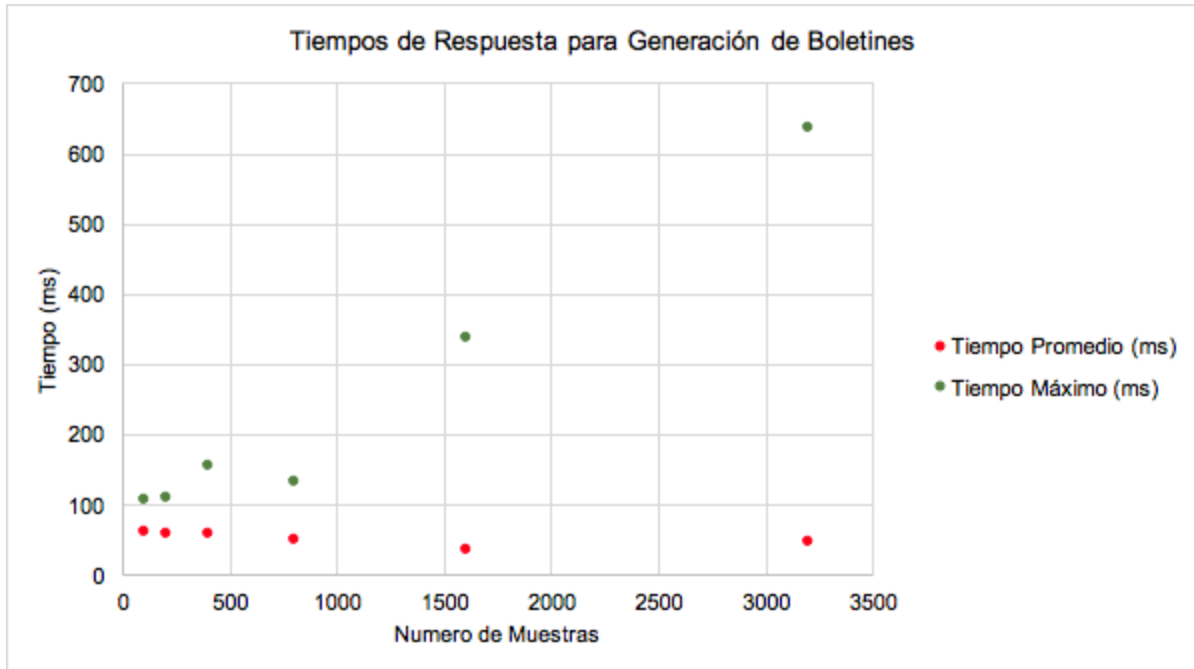
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	1600	33	22	337	20,11	0,00%	26,4/sec
Total	1600	33	22	337	20,11	0,00%	26,4/sec

#### Generación de 1600 boletines

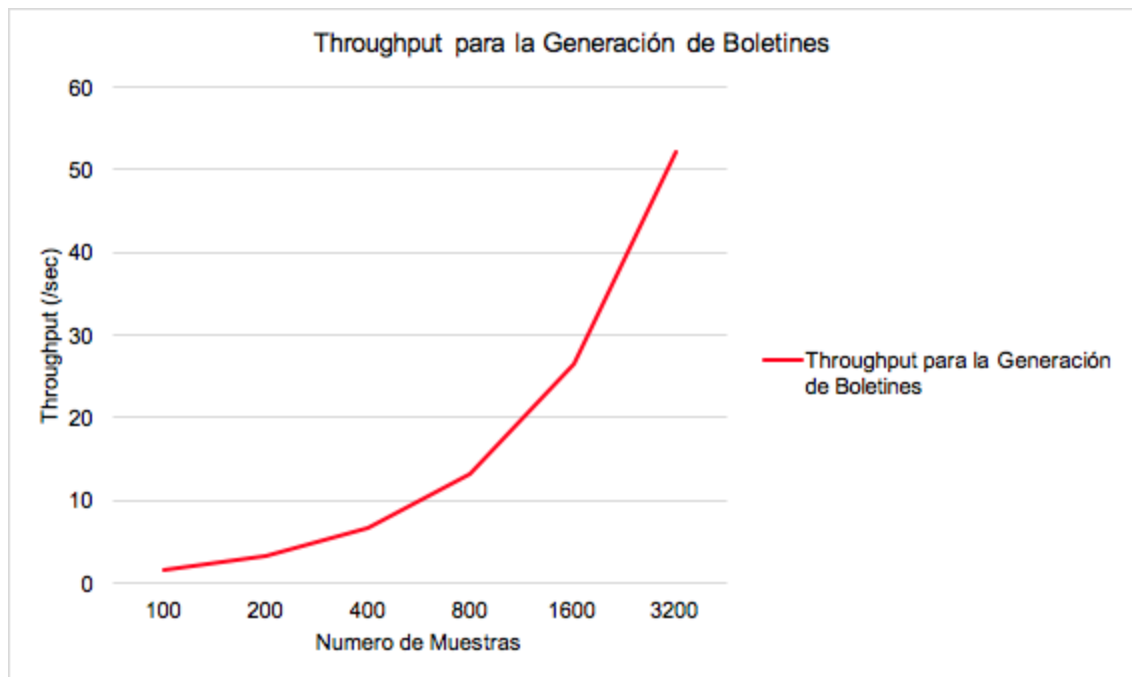
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento
HTTP Requ...	3200	45	21	635	50,97	0,00%	52,2/sec
Total	3200	45	21	635	50,97	0,00%	52,2/sec

#### Generación de 3200 boletines

Para esto, se vio que el sistema llegó a su límite (tiempo máximo superior a 500 ms) en la generación de 3200 boletines de acuerdo a peticiones concurrentes de la misma magnitud. Más adelante se explicará este hecho y el porqué esto no indica que el sistema no cumpla con el escenario de calidad. El comportamiento de los tiempos y Throughput para esto se muestra en las siguientes figuras.



Comportamientos de los tiempos de respuesta para la generación de boletines



Comportamiento de Throughput para la generación de boletines

**Duración real:** Cada experimento tomó entre 3 y 5 minutos en realizarse, lo cual no superó lo estimado debido a que no se contaba con pruebas bastante complejas para su realización.

**Artefactos contruidos:** Los únicos artefactos contruidos fueron los escenarios en JMeter y el archivo .war de despliegue. No se llevó a cabo ninguna otra construcción pues las herramientas disponibles fueron suficientes para modelar el escenario con todas sus cualidades establecidas y poder llevar a cabo las pruebas necesarias.

**Análisis y comparación con resultados anteriores:** De acuerdo con los resultados obtenidos, observando el comportamiento de los tiempos de respuesta del procesamiento de mediciones de sensores; es posible concluir que los escenarios de calidad se cumplen en cuanto a una buena escalabilidad y desempeño, pues todas las peticiones fueron procesadas efectivamente con bajos tiempos máximos de respuesta; siempre por debajo del límite planteado de un minuto para una ventana de un minuto de peticiones. Adicionalmente, se vio que los tiempos eran fluctuantes; ya que para el caso de 1600 peticiones y 4000 peticiones, se obtuvieron resultados fuera del comportamiento regularmente creciente de los demás datos. Sin embargo, ningún valor sobrepasó el límite impuesto por el escenario de calidad.

En comparación con la experimentación anterior, se vio como en los casos incrementales la latencia empeoró en un costo alrededor de los 200 milisegundos en el peor caso, si bien es un cambio considerable, todo esto era de esperar dadas las costosas operaciones de serialización en la base de datos, lo cual es un costo inevitable. Sin embargo, los resultados estuvieron dentro de los lineamientos aceptables y la arquitectura respondió bien en cuanto a su escalabilidad. Adicionalmente, el Throughput presentó un comportamiento incremental acorde a la carga, lo cual implica que el sistema tuvo las capacidades para procesar las peticiones en las unidades de tiempo suficientes para no generar errores.

Para el caso de las pruebas de desempeño, medidas a partir de la latencia en la generación de boletines, se vio que el límite de 500 milisegundos fue sobrepasado en las 3200 peticiones. En este caso, para el análisis es necesario entender la forma de operación del sistema en cuanto a la generación de boletines.

Primero, un boletín generado solamente se da cuando se recibe un evento sísmico; cuando esto sucede, el sistema procede a realizar la lógica de negocio correspondiente e iniciar un thread monitor que se ocupará de generar boletines cada 5 minutos como es estipulado. Así, en el contexto de la prueba, 3200 solicitudes de boletines generan 3200 procesos de monitoreo de fondo en la aplicación, que es un costo bastante grande. Sin embargo, esta cantidad de solicitudes implicarían en la vida real (dado el ramp-up) 53 reportes sísmicos cada segundo; lo cual no corresponde con el escenario de operación real del sistema. Es por esto que en la experimentación anterior se realizó esta prueba con 1 solicitud, sin embargo, para determinar los efectos de la persistencia y para descubrir los límites de la aplicación se realizó esta prueba incremental.

De acuerdo a los resultados, se vio que la latencia máxima de 500 milisegundos en la generación de un reporte tiene un límite un poco después de 1600 solicitudes, en donde se registró una latencia máxima de 337 milisegundos; además, el comportamiento del Throughput

es bastante prometedor, ya que representa una capacidad incremental de procesamiento suficiente para la carga empleada. Si se compara con el resultado de la experimentación anterior donde se obtuvo una latencia máxima de 68 milisegundos, para este caso, un solo boletín se acerca a los 100 milisegundos en generarse, lo cual es explicado por los altos costos del manejo de la base de datos, pues esto implica el proceso de serialización del evento sísmico que genera el boletín, la lógica de negocio a través de los escenarios pre modelados serializados en la base de datos y la serialización del boletín. Esto es la operación más compleja de la lógica. Sin embargo, hasta 1600 solicitudes se cumple el escenario de calidad, lo cual es un caso hipotético debido a su irrealidad, lo cual permite concluir que en la realidad el sistema se desempeñaría en óptimas condiciones de desempeño.

**Conclusiones:** A partir del proceso de experimentación realizado, fue posible determinar el comportamiento del sistema luego de la inclusión de la capa de persistencia JPA. Si bien fue claro desde un principio que el desempeño y la escalabilidad iban a ser afectados, mediante el experimento se pudo determinar en qué magnitud se dio este cambio. Como lo demostraron los resultados, la afectación no fue suficiente como para incumplir los escenarios de calidad propuestos, todo bajo los marcos de operación real del sistema frente a mediciones de sensores y apariciones de eventos sísmicos en las zonas costeras del país.