

*Vigila el acceso  
a tus datos*

► **Rendimiento**

<b>1.</b>	<b><u>INTRODUCCIÓN</u></b>	<b><u>3</u></b>
<b>2.</b>	<b><u>CONFIGURACIÓN</u></b>	<b><u>4</u></b>
<b>3.</b>	<b><u>RESULTADOS</u></b>	<b><u>5</u></b>
	3.1 Tiempo de Ejecución	6
	3.2 Tiempo de Ejecución sin registro	7
	3.3 Uso de CPU	8
	3.4 Memoria	9
<b>4.</b>	<b><u>CONCLUSIONES</u></b>	<b><u>10</u></b>
<b>5.</b>	<b><u>REFERENCIAS</u></b>	<b><u>11</u></b>

## 1. INTRODUCCIÓN

*LogLOPD* resuelve el problema que plantea cumplir la LOPD en las aplicaciones basadas en la plataforma J2EE, de forma que sea transparente a los desarrolladores, y con una configuración sencilla que no incremente la complejidad del desarrollo.

La intención de este documento es evaluar el rendimiento de *LogLOPD*. Para ello se han realizado diferentes pruebas sobre aplicaciones Web en diferentes servidores de aplicaciones.

La configuración utilizada ha sido la basada en la intercepción a nivel DataSource, ya que es el modo de utilización más aconsejable por su sencillez y funcionamiento.

Los fundamentos en los que se basa esta comparativa han consistido en comparar el rendimiento de la aplicación utilizando *LogLOPD* contra el rendimiento que se obtiene sin utilizarlo.

Al final de este documento se explican las conclusiones de este estudio.

## 2. CONFIGURACIÓN

Todos los test han sido ejecutados en un entorno automático utilizando el siguiente software y hardware.

El hardware y software ha sido configurado en un entorno de servidor que simule la actividad de un entorno real de producción.

- Software
  - JMeter 2.2 en modo gráfico sobre J2SDK 1.4.2\_04 con la configuración de JVM por defecto.
  - Web servers (con la configuración de JVM por defecto):
    - Tomcat 5.0.28 y SUN's JRE, 1.4.2\_04
    - Websphere 6.0.1 y Websphere's JRE v6
- Hardware
  - Intel Pentium 4 Dual Core CPU 3.2 GHz
  - 1 GB DDR
  - LAN 100 Mb/s

### 3. RESULTADOS

Para garantizar que los resultados obtenidos son válidos los test han sido ejecutados cuando el tiempo de respuesta (Base de Datos – Servidor de aplicaciones) es estable.

Los resultados obtenidos en un entorno sin la solución integrada han sido considerados como la base de comparación para calcular la diferencia de rendimientos entre utilizar *LogLOPD* y no utilizarlo.

Los tiempos de comparación se han basado en ejecutar 10 peticiones diferentes sobre una aplicación Web, repitiendo este proceso 100 veces automáticamente gracias al uso de *JMeter*.

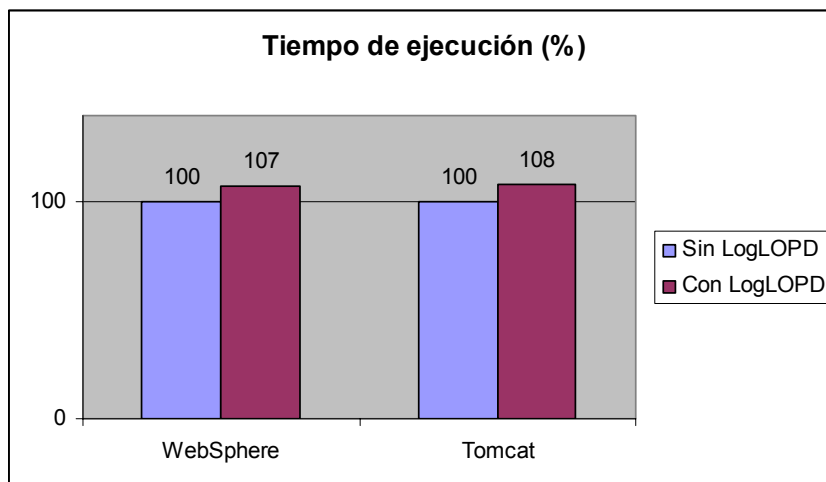
Variable de rendimiento	Definición	Método de cálculo	Configuración LogLOPD
<b>Tiempo de ejecución (%)</b>	Tiempo (ms) que se ha tardado en realizar las consultas a base de datos en las 1000 peticiones	Calculado por el uso de JMeter	Logging sólo de consultas/Sin logging
<b>CPU (%)</b>	Tiempo de uso de CPU	Calculado a través de las utilidades de los servidores de aplicaciones	Logging de consultas y de resultados
<b>Memoria (%)</b>	Cantidad de memoria requerida por el servidor de aplicaciones	Calculado a través de las utilidades de los servidores de aplicaciones	Logging de consultas y de resultados

### 3.1 Tiempo de Ejecución

De cada petición sobre la aplicación Web se han considerado únicamente los tiempos de acceso a base de datos, comprobando cuanto tardan los métodos de las clases *DAO* (Data Access Object, en una arquitectura de tres capas se corresponden con el nivel de acceso a datos).

De esta forma, en esta comparativa sólo se tiene en cuenta el tiempo de acceso a base de datos en el caso de no utilizar *LogLOPD*, y utilizando *LogLOPD* el tiempo de acceso a base de datos y el registro de las consultas en un fichero.

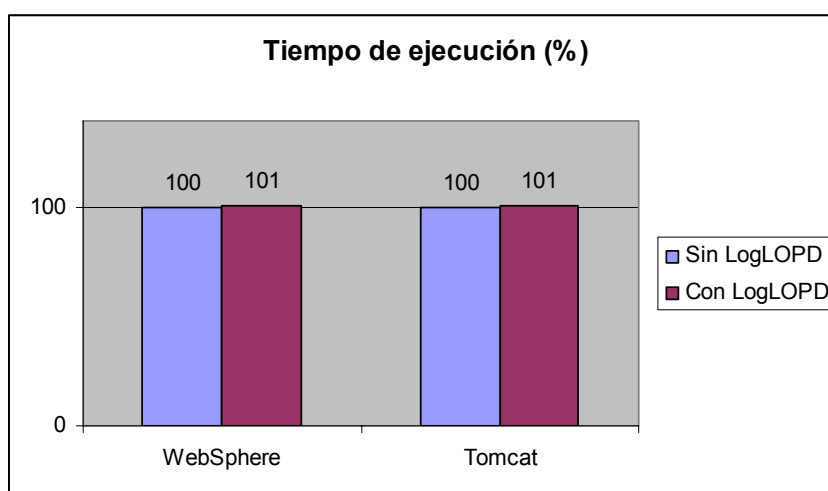
Cabe resaltar que en este test el logging sólo es a nivel de consultas, y no a nivel de los resultados devueltos por las consultas.



Los resultados de las pruebas se muestran sobre un porcentaje del 100% considerado en el caso de no utilizar *LogLOPD*, suponiendo un incremento entorno al 7%.

### 3.2 Tiempo de Ejecución sin registro

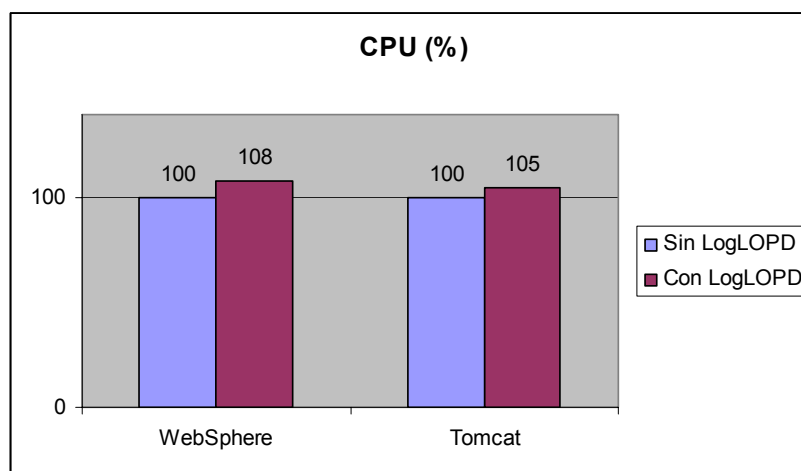
Al igual que en el apartado anterior, en el gráfico que se muestra a continuación se muestra en porcentaje el tiempo de ejecución que añade *LogLOPD* al rendimiento de la aplicación Web sin utilizarlo. Sin embargo, en este apartado el funcionamiento de *LogLOPD* se ha configurado para que no registre las consultas ni los resultados, utilizando un módulo logger que no realiza ninguna acción.



Como se puede observar, en este caso los resultados muestran que la inclusión de *LogLOPD* supone un incremento de sólo el 1% en el tiempo de ejecución, verificando el hecho de que prácticamente la totalidad de la pérdida de rendimiento es debida a la escritura en los registros (la actividad del logger; ya sea escribir en fichero, en base de datos, etc...).

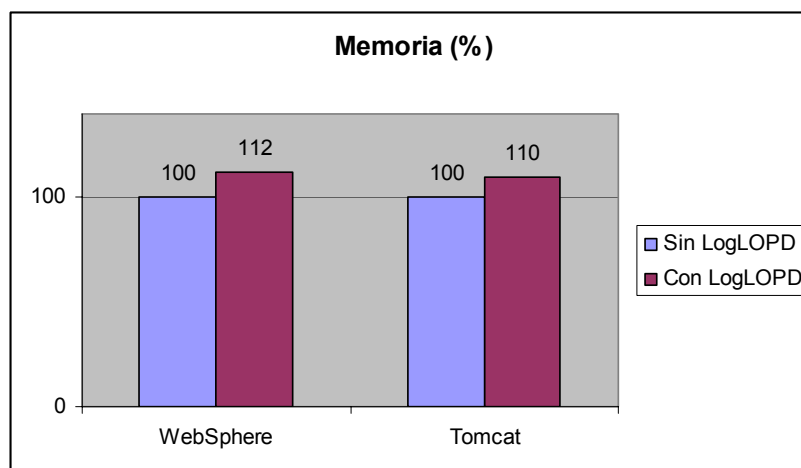
### 3.3 Uso de CPU

En el gráfico inferior se muestra el porcentaje de uso de CPU que se ha obtenido en la ejecución de las pruebas, partiendo de un uso del 100% en el caso de no utilizar *LogLOPD*. En esta prueba el uso de *LogLOPD* se ha configurado exhaustivo, registrando las consultas junto a los resultados devueltos por dichas consultas, es decir, en el peor de los casos.



### 3.4 Memoria

El gráfico inferior muestra el porcentaje de uso de memoria obtenido en las pruebas de los servidores. Para este caso, la configuración de *LogLOPD* también ha sido exhaustiva, registrando las consultas y los resultados.



## 4. CONCLUSIONES

A tenor de lo que muestran los resultados la penalización que supone la inclusión de *LogLOPD* es prácticamente despreciable por los siguientes motivos:

- En el primer resultado obtenido en tiempo de ejecución se tiene en cuenta el tiempo de logging, es decir; la escritura en fichero es una tarea que cuesta tiempo, y que en este caso, sólo afecta a la inclusión de *LogLOPD*. Sin embargo, se ha de tener en cuenta que este tiempo es absolutamente necesario, se incluya *LogLOPD* u otros mecanismos, para registrar las consultas.
- *LogLOPD* sólo penaliza la primera vez que se ejecuta una consulta siempre que la aplicación se base en la interfaz *PreparedStatement*. Esto es así porque se incluye una caché en la que guarda el análisis de las consultas, por lo que cada consulta sólo se analiza una vez, en el resto de casos su resultado se obtiene de la caché con una penalización despreciable de valor constante.
- El incremento de tiempo de ejecución por la inclusión de *LogLOPD* es del 1% si no se considera el uso de módulo de logging. Pero además, en este caso se ha de tener en cuenta que esta penalización va reduciendo su valor porcentual conforme se realizan más peticiones a la aplicación Web, ya que las primeras peticiones son las que van a penalizar el rendimiento de la aplicación por el tiempo de análisis de la consulta. A partir de ese momento, el rendimiento de la aplicación es prácticamente el mismo que sin utilizar *LogLOPD*.
- El uso real de memoria es despreciable, ya que las consultas se guardan en formato texto ocupando 2 bytes por carácter. Sin embargo, la penalización en el uso de memoria también viene determinada por la cantidad de resultados devueltos por las consultas (como se muestra en la última prueba), ya que son guardadas en objetos en memoria hasta que se registran mediante el logger.

Por todo ello, el uso de *LogLOPD* está justificado sin obtener más que una mínima pérdida de rendimiento en comparación con no utilizarlo, y justificando esta pérdida de rendimiento como algo de obligatorio cumplimiento.

## 5. REFERENCIAS

- [1]. Apache Jmeter <http://jakarta.apache.org/jmeter>
- [2]. Spring FrameWork <http://www.springframework.org>
- [3]. Apache Tomcat <http://tomcat.apache.org>
- [4]. IBM WebSphere <http://www-306.ibm.com/software/websphere>