

**UNIVERSIDAD MARIANO GALVEZ  
CENTRO UNIVERSITARIO RETALHULEU  
INGENIERIA EN SISTEMAS DE INFORMACIÓN Y  
CIENCIAS DE LA COMPUTACIÓN  
BASES DE DATOS  
ING. SANTOS**



**TAREA:  
PRUEBAS DE RENDIMIENTO**

**INTEGRANTES:  
ANGEL ANTONIO GONZÁLEZ RAMÍREZ  
EDUARDO ESTEBAN BARRIOS LEAL  
CESAR AUGUSTO TUCH PACAJÓJ  
GABRIEL ADOLFO RODRIGUEZ MORALES**

**2890-14-9238  
2890-14-12296  
2890-14-17950  
2890-10-5339**

## **Pruebas de rendimiento del software**

---

En la ingeniería del software, las **pruebas de rendimiento** son las pruebas que se realizan, desde una perspectiva, para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo. También puede servir para validar y verificar otros atributos de la calidad del sistema, tales como la escalabilidad, fiabilidad y uso de los recursos. Las pruebas de rendimiento son un subconjunto de la ingeniería de pruebas, una práctica informática que se esfuerza por mejorar el rendimiento, englobándose en el diseño y la arquitectura de un sistema, antes incluso del esfuerzo inicial de la codificación.

### **TIPOS DE PRUEBAS DE RENDIMIENTO**

#### **Pruebas de carga**

Este es el tipo más sencillo de pruebas de rendimiento. Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación. Si la base de datos, el servidor de aplicaciones, etc.. También se monitorizan, entonces esta prueba puede mostrar el cuello de botella en la aplicación.

#### **Prueba de estrés**

Esta prueba se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

#### **Prueba de estabilidad (soak testing)**

Esta prueba normalmente se hace para determinar si la aplicación puede aguantar una carga esperada continuada. Generalmente esta prueba se realiza para determinar si hay alguna fuga de memoria en la aplicación.

#### **Pruebas de picos (spike testing)**

La prueba de picos, como el nombre sugiere, trata de observar el comportamiento del sistema variando el número de usuarios, tanto cuando bajan, como cuando tiene cambios drásticos en su carga. Esta prueba se recomienda que sea realizada con un software automatizado que permita realizar cambios en el número de usuarios mientras que los administradores llevan un registro de los valores a ser monitorizados.

## **BENCHMARK JMETER Y SQL-SERVER**

**Grupo Salidas**

**Base de Datos: ComputerDB**

**Tablas: Salidas, SalidaDetalle, Clientes, Productos, Marcas, Pais, Departamento, Municipio**

### **Análisis de Rendimiento**

**Enunciado 1: Reporte de Clientes, debe mostrar el pais, departamento, municipio, nombre del cliente, nit, teléfono, fecha última venta, monto de última venta, jpromedio de venta en dinero y promedio de venta en número de productos.**

-----

### **RESULTADOS:**

#### **PLAN DE PRUEBA CON 5 USUARIOS**

##### **Resolución Enunciado 1:**

**1). Consultas SQL sin Joins dentro de un procedimiento almacenado 5 usuarios.**

**Rendimiento en tiempo: 1,229/minuto**

**Error: 0%**

**Rendimiento General: 1,5/min**

**kb por segundo: 0,04**

**2). Consultas SQL con Joins dentro de un procedimiento almacenado 5 usuarios.**

**Rendimiento en tiempo: 0,396/minuto**

**Error: 0%**

**Rendimiento General: 6,7/min**

**kb por segundo: 0,17**

**3). Consultas SQL con Cursores dentro de un procedimiento almacenado 5 usuarios.**

**Rendimiento en tiempo: 1,259/minuto**

**Error: 52,94%**

**Rendimiento General: 1,4/min  
kb por segundo: 0,00**

---

## **PLAN DE PRUEBA CON 10 USUARIOS**

**Resolución Enunciado 1:**

**1). Consultas SQL sin Joins dentro de un procedimiento almacenado 10 usuarios.**

**Rendimiento en tiempo: 1,128/minuto  
Error: 0%  
Rendimiento General: 1,4/min  
kb por segundo: 0,03**

**2). Consultas SQL con Joins dentro de un procedimiento almacenado 10 usuarios.**

**Rendimiento en tiempo: 0,48/minuto  
Error: 0%  
Rendimiento General: 1,2/min  
kb por segundo: 0,03**

**3). Consultas SQL con Cursores dentro de un procedimiento almacenado 10 usuarios.**

**Rendimiento en tiempo: 1,287/minuto  
Error: 47,37%  
Rendimiento General: 1,3/min  
kb por segundo: 0,00**

---

## **PLAN DE PRUEBA CON 15 USUARIOS**

**Resolución Enunciado 1:**

**1). Consultas SQL sin Joins dentro de un procedimiento almacenado 15 usuarios.**

**Rendimiento en tiempo: 1,097/minuto  
Error: 0%  
Rendimiento General: 1,3/min  
kb por segundo: 0,03**

**2). Consultas SQL con Joins dentro de un procedimiento almacenado 15 usuarios.**

**Rendimiento en tiempo: 0,577/minuto**

**Error: 0%**

**Rendimiento General: 1,1/min**

**kb por segundo: 0,03**

**3). Consultas SQL con Cursores dentro de un procedimiento almacenado 15 usuarios.**

**Rendimiento en tiempo: 1,278/minuto**

**Error: 40,91%**

**Rendimiento General: 1,3/min**

**kb por segundo: 0,00**

---

## **PLAN DE PRUEBA CON 20 USUARIOS**

**Resolución Enunciado 1:**

**1). Consultas SQL sin Joins dentro de un procedimiento almacenado 20 usuarios.**

**Rendimiento en tiempo: 1,248/minuto**

**Error: 0%**

**Rendimiento General: 1,4/min**

**kb por segundo: 0,04**

**2). Consultas SQL con Joins dentro de un procedimiento almacenado 20 usuarios.**

**Rendimiento en tiempo: 0,784/minuto**

**Error: 0%**

**Rendimiento General: 1,5/min**

**kb por segundo: 0,04**

**3). Consultas SQL con Cursores dentro de un procedimiento almacenado 20 usuarios.**

**Rendimiento en tiempo: 1,44/minuto**

**Error: 34,62%**

Rendimiento General: 1,4/min  
kb por segundo: 0,00

---

## PLAN DE PRUEBA CON 25 USUARIOS

### Resolución Enunciado 1:

1). Consultas SQL sin Joins dentro de un procedimiento almacenado 25 usuarios.

Rendimiento en tiempo: 1,452/minuto  
Error: 0%  
Rendimiento General: 1,6/min  
kb por segundo: 0,04

2). Consultas SQL con Joins dentro de un procedimiento almacenado 25 usuarios.

Rendimiento en tiempo: 1,016/minuto  
Error: 0%  
Rendimiento General: 1,9/min  
kb por segundo: 0,05

3). Consultas SQL con Cursores dentro de un procedimiento almacenado 25 usuarios.

Rendimiento en tiempo: 1,611/minuto  
Error: 29,03%  
Rendimiento General: 1,5/min  
kb por segundo: 0,00

---

---

Enunciado 2: Reporte de Ventas de Productos, mostrar el nombre del producto, el recuento de ventas en unidades, la suma de ventas en quetzales, precio promedio y el cliente que mas compra

---

---

## PLAN DE PRUEBA CON 5 USUARIOS

## RESULTADOS:

**Resolución Enunciado 2:**

**1). Consultas SQL sin Joins dentro de un procedimiento almacenado 5 usuarios.**

**Rendimiento en tiempo: 1,349/minuto**

**Error: 0%**

**Rendimiento General: 1,5/min**

**kb por segundo: 0,04**

**2). Consultas SQL con Joins dentro de un procedimiento almacenado 5 usuarios.**

**Rendimiento en tiempo: 0,964/minuto**

**Error: 0%**

**Rendimiento General: 1,6/min**

**kb por segundo: 0,04**

**3). Consultas SQL con Cursores dentro de un procedimiento almacenado 5 usuarios.**

**Rendimiento en tiempo: 1,428/minuto**

**Error: 28,12%**

**Rendimiento General: 1,4/min**

**kb por segundo: 0,00**

---

**PLAN DE PRUEBA CON 10 USUARIOS**

**Resolución Enunciado 2:**

**1). Consultas SQL sin Joins dentro de un procedimiento almacenado 10 usuarios.**

**Rendimiento en tiempo: 1,314/minuto**

**Error: 0%**

**Rendimiento General: 1,4/min**

**kb por segundo: 0,03**

**2). Consultas SQL con Joins dentro de un procedimiento almacenado 10 usuarios.**

**Rendimiento en tiempo: 0,961/minuto**

**Error: 0%**

**Rendimiento General: 1,5/min**

**kb por segundo: 0,03**

**3). Consultas SQL con Cursores dentro de un procedimiento almacenado 10 usuarios.**

**Rendimiento en tiempo: 1,441/minuto**

**Error: 26,47%**

**Rendimiento General: 1,4/min**

**kb por segundo: 0,00**

---

**PLAN DE PRUEBA CON 15 USUARIOS**

**Resolución Enunciado 2:**

**1). Consultas SQL sin Joins dentro de un procedimiento almacenado 15 usuarios.**

**Rendimiento en tiempo: 1,381/minuto**

**Error: 0%**

**Rendimiento General: 1,5/min**

**kb por segundo: 0,03**

**2). Consultas SQL con Joins dentro de un procedimiento almacenado 15 usuarios.**

**Rendimiento en tiempo: 1,052/minuto**

**Error: 0%**

**Rendimiento General: 1,6/min**

**kb por segundo: 0,03**

**3). Consultas SQL con Cursores dentro de un procedimiento almacenado 15 usuarios.**

**Rendimiento en tiempo: 1,52/minuto**

**Error: 24,32%**

**Rendimiento General: 1,5/min**

**kb por segundo: 0,00**



---

## **PLAN DE PRUEBA CON 20 USUARIOS**

### **Resolución Enunciado 2:**

**1). Consultas SQL sin Joins dentro de un procedimiento almacenado 20 usuarios.**

**Rendimiento en tiempo: 1,508/minuto**

**Error: 0%**

**Rendimiento General: 1,6/min**

**kb por segundo: 0,03**

**2). Consultas SQL con Joins dentro de un procedimiento almacenado 20 usuarios.**

**Rendimiento en tiempo: 1,193/minuto**

**Error: 0%**

**Rendimiento General: 1,8/min**

**kb por segundo: 0,03**

**3). Consultas SQL con Cursores dentro de un procedimiento almacenado 20 usuarios.**

**Rendimiento en tiempo: 1,643/minuto**

**Error: 21,95%**

**Rendimiento General: 1,6/min**

**kb por segundo: 0,00**

---

## **PLAN DE PRUEBA CON 25 USUARIOS**

### **Resolución Enunciado 2:**

**1). Consultas SQL sin Joins dentro de un procedimiento almacenado 25 usuarios.**

**Rendimiento en tiempo: 1,662/minuto**

**Error: 0%**

**Rendimiento General: 1,8/min**

**kb por segundo: 0,03**

**2). Consultas SQL con Joins dentro de un procedimiento almacenado 25 usuarios.**

**Rendimiento en tiempo: 1,362/minuto**

**Error: 0%**

**Rendimiento General: 2,1/min**

**kb por segundo: 0,03**

**3). Consultas SQL con Cursores dentro de un procedimiento almacenado 25 usuarios.**

**Rendimiento en tiempo: 1,79/minuto**

**Error: 19,57%**

**Rendimiento General: 1,8/min**

**kb por segundo: 0,00**

## COMPARACIÓN DE RESULTADOS CON AUTORES Y LIBROS

### 1. Pruebas funcionales y de rendimiento con JMeter

**Autores:** Pablo Olmos

**Localización:** Sólo programadores, ISSN 1134-4792, Nº 128, 2005, págs. 50-56

**Idioma:** español

**Resumen** Toda aplicación pasa por una fase de pruebas. JMeter es una utilidad escrita en Java que sirve para realizar pruebas de rendimiento y de funcionalidad sobre aplicaciones tipo cliente/servidor escritas en cualquier lenguaje.

2. Según Microsoft Developer Network, la especificación de rendimiento, como mínimo, debería responder a las siguientes preguntas:

- ¿Cuál es el alcance, en detalle, de la prueba de rendimiento? ¿Qué subsistemas, interfaces, componentes, etc están dentro y fuera del ámbito de ejecución de esta prueba?
- Para las interfaces de usuario involucradas, ¿Cual es el número de usuarios concurrentes que se esperan para cada uno (especificando picos y medias)?
- ¿Cuál es la estructura objetivo del sistema (hardware, especificandor todos los servidores de red y configuraciones de dispositivo)?
- ¿Cuál es la distribución del volumen de trabajo de la aplicación para cada componente? (por ejemplo: 20% login, 40% buscando, 30% seleccionando elemento, 10% comprando).
- ¿Cual es la distribución del trabajo del sistema? [Las cargas de trabajo múltiples pueden ser simuladas en una sola prueba de eficacia] (por ejemplo: 30% del volumen de trabajo para A, 20% del volumen de trabajo para B, 50% del volumen de trabajo para C)
- ¿Cuáles son los requisitos de tiempo para cada uno y para todos los procesos por lotes (especificando picos y medias)?

3. Según Javier Medina en su libro “PRUEBAS DE RENDIMIENTO TIC”, nos dice que el tiempo de tiempo que tarda un sistema de información en atender una petición de usuario, incluye el tiempo de transmisión, el tiempo de latencia, y el tiempo de procesamiento. Esto último hace referencia al hardware y la cantidad usuarios que intentan acceder a la información al mismo tiempo.

4. Según el libro – JUnit in action, Petar Tahchiev, Felipe Leme, Vincent Massol, Gary Gregory. \*\* Explica JUnit y después pruebas unitarias, mocks, stubs. JUnit no tiene por qué usarse solo para pruebas unitarias, y TDD ya que las pruebas unitarias son responsabilidad de los desarrolladores. Esto indica que como analizadores de bases de datos debemos someter los sistemas a las distintas pruebas de rendimiento que existen para conocer el rendimiento de

nuestras consultar y así poder saber cuál es la más óptima para manejar dentro de nuestros sistemas.

5. Según **Silberschatz – Korth** en su libro **Fundamentos de Bases de Datos 5a Ed.**- Las estructuras índice se denominan caminos de acceso, ya que proporcionan un camino a través del cual se pueden localizar y acceder a los datos. En el Capítulo se señaló la eficiencia de leer los registros del archivo en un orden próximo al orden físico. Esto quiere decir que al momento de la lectura de datos otra buena práctica es hacer la búsqueda de información muy similar a como se encuentren almacenados los datos mediante el orden físico dentro de la base de datos, esto permitirá optimar la base de datos y reducir tiempo de espera dentro de la misma.