Facultad de Ingeniería
Escuela de Ingeniería Civil Informática

DESARROLLO DE APLICACIONES WEB

Por

Marcelo Esteban Verdugo Reyes
Nicolás Albornoz Reyes

Tarea 2
Mayo 2015

Resumen

El siguiente trabajo tiene como objetivo adquirir conocimientos acerca de el desarrollo de aplicaciones web, por lo cual, en esta tarea se trabajó con el lenguaje PHP, HTML, Mysql, un framework de PHP CodeIgniter y con la herramienta Jmeter

En este trabajo se realizaron tres consultas a una base de datos, las que posteriormente se mejoraron, bajo diferentes maneras, las cuales se verán en este documento.
Las mejoras se analizaron realizando pruebas de carga con la herramienta Jmeter.

Índice general

Índice de tablas

Índice de figuras

0.1. Introducción

0.1.1. PHP

La sigla PHP nació como **Personal Home Page**, sin embargo hoy en día se está vinculando a **Hyper Pre-Processor**.
Este lengujae fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP.

PHP es el código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. PHP fue uno de los primeros lenguajes que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. Dentro de los cambios que ha incluido PHP, es que ahora también cuenta con una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes.

PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

PHP se considera uno de los lenguajes más flexibles debido a que puede ser usado en la mayoría de los servidores web, lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico, como Facebook, para optar por el mismo como tecnología de servidor.

0.1.2. HTML

HTML, siglas de HyperText Markup Language **lenguaje de marcas de hipertexto**, hace referencia al lenguaje de marcado para la elaboración de páginas web.

Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, entre otros. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación.

El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta

directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene sólo texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.). Sin embargo, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Normalmente los cambios son aplicados mediante parches de actualización automática (Firefox, Chrome) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial (Internet Explorer).

Un navegador no actualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple. Es por ello que ciertos navegadores aún mantienen la capacidad de interpretar páginas web de versiones HTML anteriores. Por estas razones, aún existen diferencias entre distintos navegadores y versiones al interpretar una misma página web.

0.1.3. MySql

MySQL cuyas siglas en inglés significa **My Structured Query Language o Lenguaje** es un lenguaje de consulta estructurado de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

MySQL es usado por muchos sitios web grandes y populares, como Wikipedia, Google (aunque no para búsquedas), Facebook, Twitter, Flickr, y YouTube.

0.1.4. CodeIgniter

CodeIgniter es un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. *Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas*.

También hay que destacar que CodeIgniter es más rápido que muchos otros entornos. Incluso en una discusión sobre entornos de desarrollo con PHP, Rasmus Lerdorf, el creador de PHP, expresó que le gustaba CodeIgniter *porque es rápido, ligero y parece poco un entorno*.

Este framework se basa principalmente en el popular Modelo-Vista-Controlador. Este modelo, se divide en tres partes, debidamente separadas una de otra. En donde el controlador se encarga de responder a los eventos y es el responsable de enviar peticiones o ”tareas” al modelo(Por ejemplo, editarun registro o un documento en una base de datos).
El modelo se encarga de representar la información con la cúal el sistema opera, gestiona todos los accesos a dicha información solicitados desde el controlador. Las que posteriormente son mostradas en la vista.

0.1.5. Jmeter

JMeter es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web.

JMeter puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, Servicios web, JMS, HTTP y conexiones TCP genéricas.

Mientras que JMeter es clasificado como una herramienta de ”generación de carga”, no es una descripción completa de la herramienta. JMeter soporta aserciones para asegurarse que los datos recibidos son correctos, por cookies de hilos, configuración de variables y una variedad de reportes.

0.1.6. Base de Datos utilizada

La base de datos utilizada para realizar la tarea fue la base de datos cuyo nombre es ”**employees**”.

Esta base de datos cuenta con 6 tablas que son las siguientes:

- **departaments**: Cuenta con dos atributos, estos son dept_no y dept_name, y existen 9 estancias de esta tabla.

- **dept_emp**: Esta tabla no cuenta con instancias, sin embargo nos sirve para obtener una asociación entre employees y departaments, ya que cuenta con los atributos emp_no, dept_no, from_date y to_date.

- **dept_manager**: Esta tabla cuenta con 24 registros la cual contiene dept_no, emp_no, from_date, to_date.

- **employees**: Esta tabla aproximádamente un total de 300,695 registros y tiene los siguientes atributos: emp_no, birth_date,first_name, last_name gender, hire_date.

- **salaries**: Esta tabla cuenta con una cantidad de aproximada de 2,844,513 registros y los siguientes atributos: emp_no, salary, from_date, to_date.

- **titles**: Esta tabla cuenta con aproximádamente un total de 443,506 registros y cuenta con los siguientes atributos: emp_no, title, from_date, to_date.

0.2. Definición de la tarea

En la siguiente tarea se solicita crear tres peticiones o query a un servidor desde una interfáz web. Estas peticiones deben ser de gran esfuerzo, es decir, se debe procesar una gran cantidad de datos desde la base de datos.

Posteriormente realizadas estas query se deben realizar 5 iteraciones para omptimizar estas solicitudes. Estas iteraciones buscan reducir los tiempos de espera del usuario que solicita la gran cantidad de datos a la base de datos.

Finalmente realizadas las iteraciones que buscan otpimizar las solicitudes. Se debe realizar una pruebas de carga con la herramienta Jmeter. Estas pruebas buscan estudiar el comportamiento de estas mismas consultas optimizadas pero suponiendo que varios usuarios accederán a ellas en un tiempo simultaneo. Cabe mencionar que las pruebas se deben realizar en un entorno ”limpio”, es decir, suponer que el dispositivo del usuario no se ve afectado por otros programas o por otros problemas de rendimiento.

0.3. Consultas Básicas

En un comienzo se realizó una interfaz web simple, la cual cuenta con 3 botones, en donde cada botón al presionarlo genera una consulta a la base de datos **”employees”** como se muestra en las siguientes figuras 0.3.1, 0.3.2, 0.3.3.
Cabe mencionar que cada una de estas query se realiza de forma inmediata luego de presionar el botón, y no de la forma en que trabaja CodeIgniter (Modelo-Vista-Controlador).
Además previo a realizar cada consulta se realizan las siguientes acciones:

```php
$mysqli = new mysqli("localhost", "root", "", "employees");
if ($mysqli->connect_errno) {
    echo "Fallo al contenctar a MySQL: (" . $mysqli->connect_errno . ") " . $mysqli->connect_error;
}
```

Figura 1: Esta figura muestra las acciones a realizar previo a llevar a cabo una query la cual se repite para las 3 consulas.

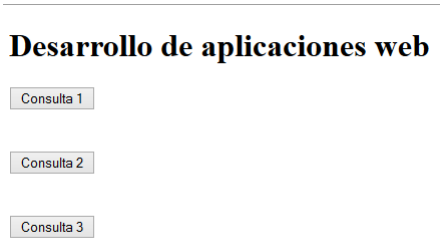

Figura 2: Figura de la interfáz de inicio de la página web de consultas.

0.3.1. Consulta 1

El usuario al presionar el botón ”Consulta 1” se realiza la siguiente query:

```
$query = $this->db->query('SELECT * FROM employees where gender = "M" OR first_name = "Mary"');
```

Figura 3: Query consulta 1 iteración 1.

Esta consulta solicita a la base de datos el ”emp_no” o número de empleado, de los cuales su genero es ”M” o masculino, O también el número de empleados cuyo nombre sea ”Mary”.

Los resultados entregados por el servidor son los que se muestran en la figura 0.3.1:

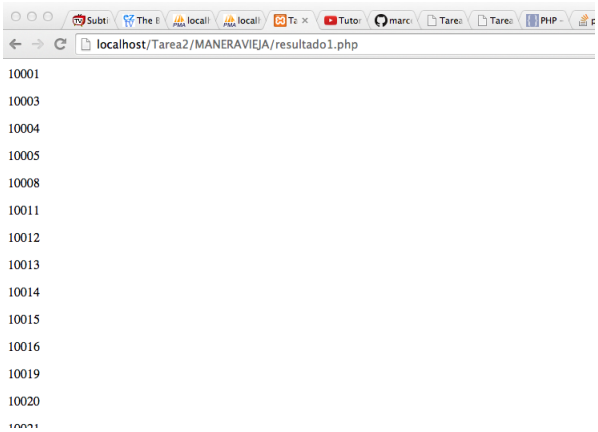

Figura 4: Figura que muestra el resultado obtenido desde el servidor, en donde se muestran algunos números de empleados.

0.3.2. Consulta 2

El usuario al presionar el botón ”Consulta 2” se realiza la siguiente query:

```
$query = $this->db->query('SELECT emp_no, salary FROM salaries where salary ="88208" OR salary = "40000"');
```

Esta consulta solicita a la base de datos el ”emp_no” o número de empleado, de los cuales su genero es ”M” o masculino, O también el número de empleados cuyo nombre sea ”Mary”.

Los resultados entregados por el servidor son los que se muestran en la figura 0.3.2:

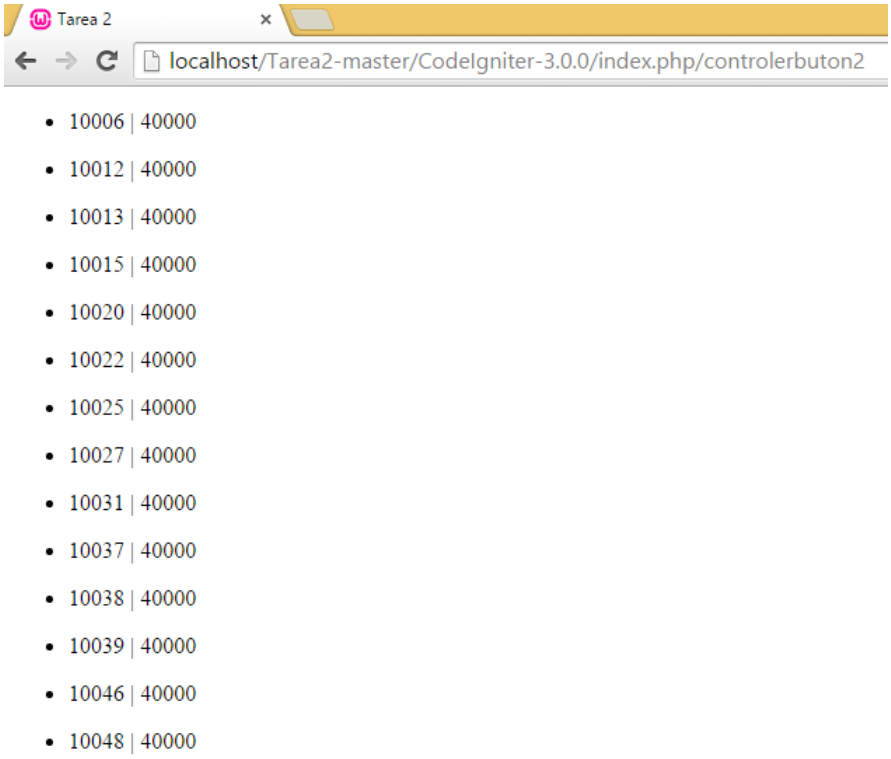

Figura 5: Figura que muestra el resultado obtenido desde el servidor, en donde se muestran algunos números de empleados.

0.3.3. Consulta 3

El usuario al presionar el botón ”Consulta 3” se realiza la siguiente query:

```
$query = $this->db->query("SELECT a.first_name, a.last_name, s.salary, d.dept_name, t.title FROM employees
    a, salaries s, titles t, departments d, dept_emp e WHERE a.emp_no=s.emp_no and a.emp_no=e.emp_no and e.dept_no=d.dept_no
    and s.to_date between str_to_date ('1900-01-01','%Y-%m-%d') and str_to_date ('2000-01-01','%Y-%m-%d') and t.emp_no=a.emp_no
    and t.title='Senior Engineer' group by a.last_name");
```

Esta consulta solicita a 5 tablas de la base de datos, los campos first_name, last_name, salary, dept_name y title.

Los resultados entregados por el servidor son los que se muestran en la figura 0.3.3:

Figura 6: Figura que muestra el resultado obtenido desde el servidor.

0.4. Pruebas de carga

En la siguiente sección se muestran los resultados de las pruebas de carga realizadas con la herramienta Jmeter, para realizar estas pruebas se levantó un servidor utilizando Virtual Box con el fin de obtener resultados en un ambiente controlado, libre de otros procesos o programas que puedan interferir en la memoria de nuestro equipo. [1]

La maquina virtualizada tiene las siguientes carácteristicas:

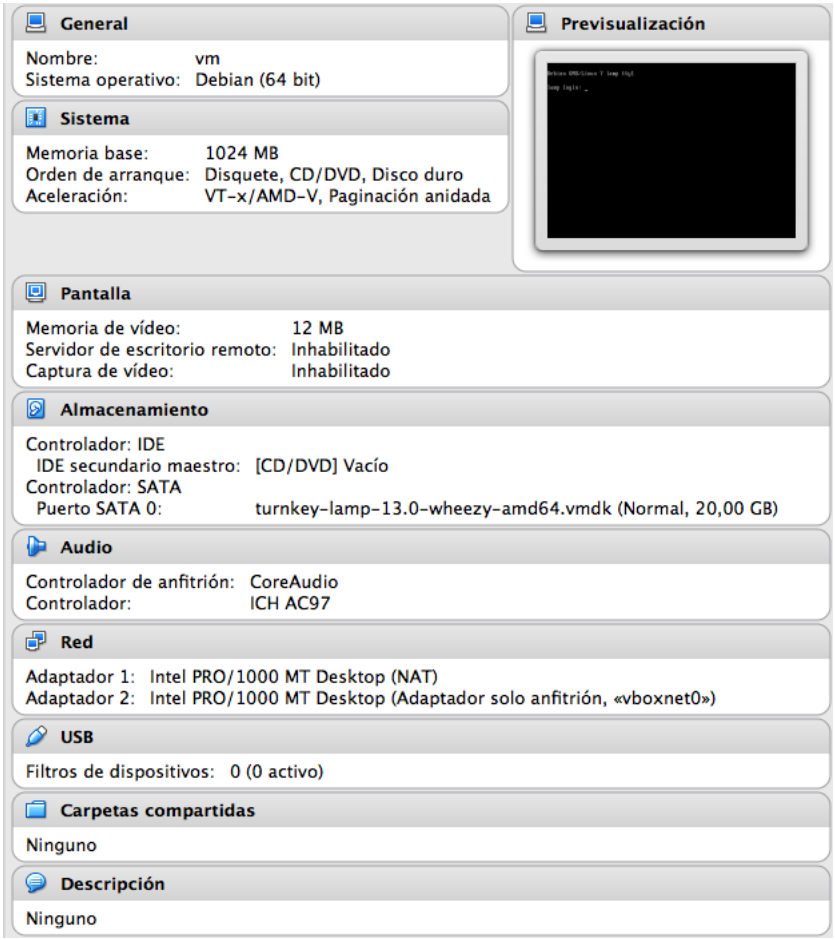

Figura 7: Figura que muestra las carácteristicas de la maquina virtual utilizada.

0.4.1. Iteración 1

Se realizan las pruebas de carga a la primera iteración.

Pruebas de carga a la iteración 1

La prueba se configuro bajo las siguientes condiciones

- Número de hilos: 20
- Periodo de subida: 10 seg.
- Ciclos: 1

Los resultados obtenidos para las consultas fueron los siguientes:

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec
lte 1 consulta 1	20	18353	3793	21673	4054,91	0,00%	43,4/min	317,46
ite 1 consulta 2	20	17808	4203	20569	3917,18	0,00%	43,9/min	186,38
ite 1 consulta 3	20	17	14	21	2,50	0,00%	2,1/sec	44,41
Total	60	12060	14	21673	9118,84	0,00%	52,5/min	208,60

Figura 8: Tabla resumen iteración 1

0.4.2. Iteración 2 (Mejorando las consultas MySql)

Luego de realizar las respectivas pruebas de cargas, se optó por modificar las query MySql, en donde:

- Consulta 1: En la consulta 1, se solicita al servidor todos los atributos de la tabla 'employees' ya que se realiza de la forma "SELECT * FROME employees" y donde finalmente se muestra solo "emp_no" (Numero de empleado). Por lo cual una mejora a esta query sería solicitar solamente lo que necesitamos, es decir, "emp_no".

- Consulta 2: En esta consulta se utiliza el operador lógico OR, el cual investigando, se puede mejorar utilizando el operador IN, ell cual entrega un mejor performance al momento de realizar una query.

- Consulta 3: Se utiliza INNER JOIN por sobre la comparación con la sentencia WHERE, la cual se espera tenga un mejor rendimiento.

Pruebas de carga a la iteración 2

La prueba se configuro bajo las siguientes condiciones

- Número de hilos: 20
- Periodo de subida: 10 seg.
- Ciclos: 1

Los resultados obtenidos son los siguientes:

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec
ite 2 consulta 1	20	11028	2330	13669	3054,38	0,00%	58,8/min	429,51
ite 2 consulta 2	20	16544	4910	19536	3517,00	0,00%	46,3/min	196,85
ite 2 consulta 3	20	18	13	22	2,95	0,00%	2,1/sec	44,37
Total	60	9196	13	19536	7377,44	0,00%	1,0/sec	238,75

Figura 9: Tabla resumen iteración 2

Se puede observar leves mejoras frente a la iteración 1, por lo que los supuestos se han cumplido.

0.4.3. Iteración 3 (Utilizando CodeIgniter)

Debido a que este Framework ofrece la utilización del Modelo - Vista - Controlador, nos parece pertinente utilizar esta arquitectura, por lo cual se adapta completamente la forma en la que se se escribe el código de este proyecto.

La interfáz web sigue siendo la misma, sin embargo, esta interfáz es manjeda por un controlador, el cual se encarga de cargarla.

Posteriormente cuando el usuario presiona un botón para llevar a cabo las consultas, es un controlador el encargado de realizar la acción, donde cada botón es manejado independientemente. Posterior a realizar esta acción es el controlador el que solicita al modelo realizar una query al servidor. El modelo realiza la query y envía los datos nuevamente al controlador, para que este se los envíe a la vista para que estos datos puedan ser mostrados al usuario.

Usando CodeIgniter además nos ahorramos realizar las conexiones a la base de datos, ya que internamente se configura la base de datos que se utilizará.

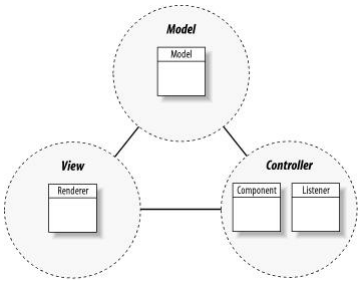

Figura 10: Esta imagen ejemplifica como funciona el Modelo - Vista - Controlador.

Pruebas de carga realizadas a la iteración 3

La prueba se configuro bajo las siguientes condiciones

- Número de hilos: 20
- Periodo de subida: 10 seg.
- Ciclos: 1

Los resultados no son los esperados y las pruebas no logran tener exito, por lo que se decide cambiar las condiciones de las pruebas, a las siguientes:

- Número de hilos: 10
- Periodo de subida: 10 seg.
- Ciclos: 1

Los resultados obtenidos son los siguientes:

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec
ite 3 consulta 1	10	14674	5720	17015	3257,56	0,00%	25,5/min	200,44
ite 3 consulta 2	10	13629	11949	14683	802,59	0,00%	28,6/min	126,37
ite 3 consulta 3	10	31	27	39	3,76	0,00%	1,1/sec	23,64
Total	30	9445	27	17015	6945,67	0,00%	29,2/min	123,21

Figura 11: Tabla resumen iteración 3

Se concluye de estas pruebas que la implementación con CodeIgniter no presenta mejoras frente a las iteraciones anteriores. De todas formas la siguiente iteración se planteara como mejora utilizar la opción de cacheado de CogeIgniter.

0.4.4. Iteración 4 (Utilizando la opción de Cachè en CodeIgniter))

Como además CodeIgniter ofrece la utilización de Cachè, las consultas obendrá un mejor desempeño, ya que luego de realizar una de las query, los datos estarán ”Cacheados” por lo cual, las posteriores consultas deberían resultar ser más rápidas.

Pruebas de carga realizadas a la iteración 4

La prueba se configuro bajo las siguientes condiciones

- Número de hilos: 10
- Periodo de subida: 10 seg.
- Ciclos: 1

Los resultados obtenidos son los siguientes:

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec
ite 4 consulta 1	10	13727	4546	15875	3274,98	0,00%	26,4/min	207,57
ite 4 consulta 2	10	12404	9673	14015	1493,63	0,00%	29,4/min	129,91
ite 4 consulta 3	10	36	27	42	4,22	0,00%	1,1/sec	23,44
Total	30	8722	27	15875	6506,71	0,00%	29,9/min	125,98

Figura 12: Tabla resumen iteración 4

Se observa leve mejora frente a la iteración 3, aún así las pruebas no reflejan mejoras frente a la iteración 2, que es hasta el momento la cual ha presentado un mejor rendimiento.

0.4.5. Iteración 5 (Paginación de los resultados de las query)

En esta iteración se ha optado por entregar el set de resultados separados por páginas. Utilizando peticiones asincronas al servidor se puede lograr que el rendimiento mejore.

Para las peticiones asincronas se ha utilizado javascript, así se le entrega cierta carga al cliente para la paginación.

Pruebas de carga realizadas a la iteración 5

La prueba se configuro bajo las siguientes condiciones

- Número de hilos: 20
- Periodo de subida: 10 seg.
- Ciclos: 1

Los resultados obtenidos son los siguientes:

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec
ite 5 consulta 1 - pag 1	20	24	15	45	7,95	0,00%	2,1/sec	6,05
ite 5 consulta 1 - pag 2	20	23	12	36	6,54	0,00%	2,1/sec	6,01
ite 5 consulta 1 - pag 3	20	24	12	38	6,96	0,00%	2,1/sec	6,04
ite 5 consulta 1 - pag 4	20	25	12	35	7,37	0,00%	2,1/sec	6,03
ite 5 consulta 1 - pag 5	20	26	14	35	7,29	0,00%	2,1/sec	6,05
ite 5 consulta 1 - pag 6	20	24	12	32	5,47	0,00%	2,1/sec	6,08
ite 5 consulta 1 - pag 7	20	26	14	67	11,18	0,00%	2,1/sec	6,14
ite 5 consulta 1 - pag 8	20	29	13	74	15,08	0,00%	2,1/sec	6,05
ite 5 consulta 1 - pag 9	20	29	12	60	13,93	0,00%	2,1/sec	6,11
ite 5 consulta 1 - pag 10	20	21	11	63	12,11	0,00%	2,1/sec	6,20
ite 5 consulta 2 - pag 1	20	42	19	92	21,43	0,00%	2,1/sec	6,82
ite 5 consulta 2 - pag 2	20	35	14	75	17,71	0,00%	2,1/sec	6,81
ite 5 consulta 2 - pag 3	20	33	13	63	11,58	0,00%	2,1/sec	6,83
ite 5 consulta 2 - pag 4	20	32	13	50	9,46	0,00%	2,1/sec	6,82
ite 5 consulta 2 - pag 5	20	31	16	50	8,83	0,00%	2,1/sec	6,84
ite 5 consulta 2 - pag 6	20	32	15	58	9,74	0,00%	2,1/sec	6,86
ite 5 consulta 2 - pag 7	20	33	13	82	15,62	0,00%	2,1/sec	6,96
ite 5 consulta 2 - pag 8	20	34	17	67	12,15	0,00%	2,1/sec	6,94
ite 5 consulta 2 - pag 9	20	36	14	133	24,55	0,00%	2,1/sec	6,99
ite 5 consulta 2 - pag 10	20	28	13	108	20,52	0,00%	2,2/sec	7,09
ite 5 consulta 3 - pag 1	20	83054	79999	87373	2290,33	0,00%	13,4/min	2,98
ite 5 consulta 3 - pag 2	20	0	0	9	1,92	100,00%	8,8/sec	17,77
Total	440	3802	0	87373	17301,15	4,55%	2,3/sec	7,96

Figura 13: Tabla resumen iteración 5

En esta iteración se ha alcanzo buenos indices de rendimiento, mejores que los obtenidos en las iteraciones anteriores. Por lo que se concluye que el ”cuello de botella” de las consultas era la cantidad de datos a transferir, y no el tiempo que toma en realizar un petición al servidor de base de datos. Esta última iteración ha solucionado este punto y genera la mejor performance de las 3 consultas.

Bibliografía

[1] Pedro Hernandez. Introducción a jmeter. 2008. Último acceso: 22 de mayo del 2015.