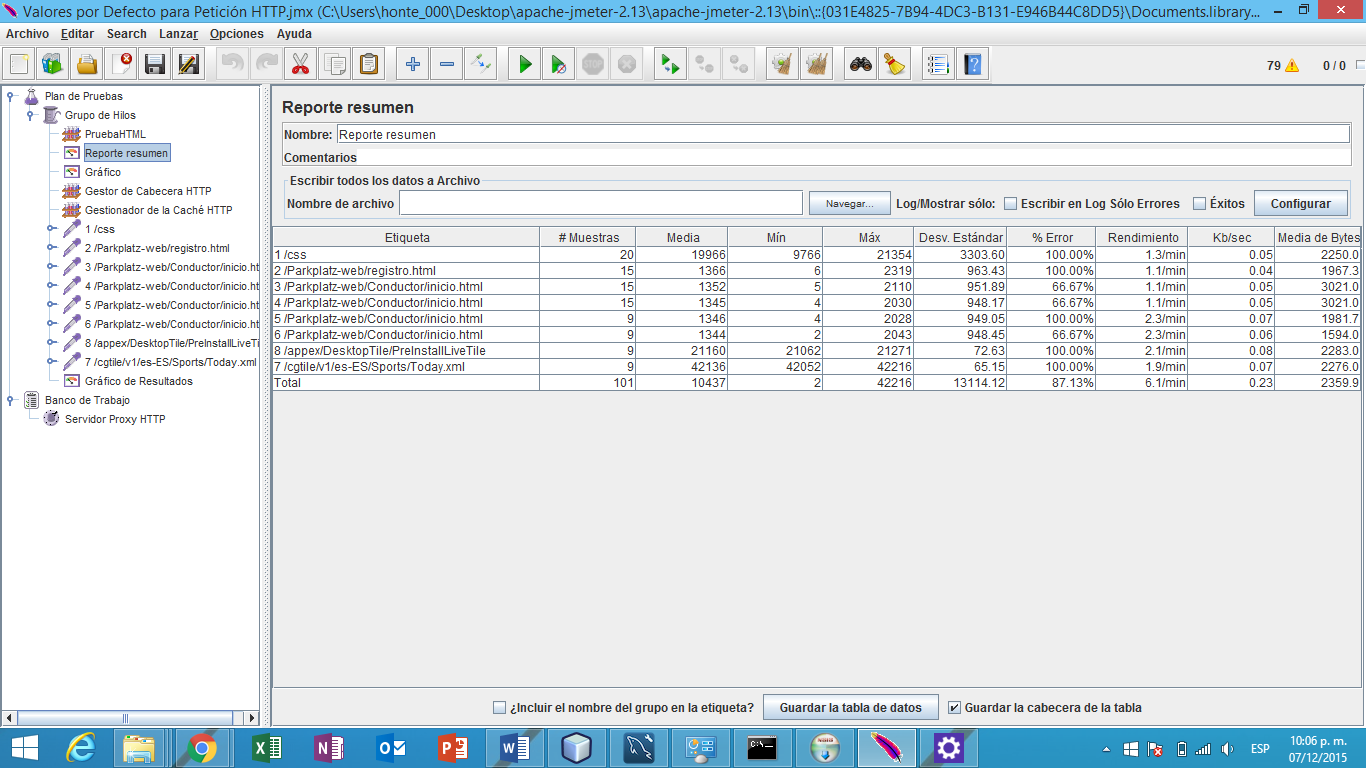
Prueba de Stress

|  |  |
| --- | --- |
| Objetivo de la Prueba: | Verificar que el sistema funciona apropiadamente y sin errores, bajo estas condiciones de stress:   * Memoria baja o no disponible en el servidor. * Máximo número de clientes conectados o simulados (actuales o físicamente posibles) * Múltiples usuarios desempeñando la misma transacción con los mismos datos. * El peor caso de volumen de transacciones (ver pruebas de desempeño).   NOTAS: La meta de las pruebas de stress también es identificar y documentar las condiciones bajo las cuales el sistema FALLA. |
| Descripción de la Prueba: | Las pruebas de stress se proponen encontrar errores debidos a recursos bajos o completitud de recursos. Poca memoria o espacio en disco puede revelar defectos en el sistema que no son aparentes bajo condiciones normales. Otros defectos pueden resultar de incluir recursos compartidos, como bloqueos de base de datos o ancho de banda de la red. Las pruebas de stress identifican la carga máxima que el sistema puede manejar.  El objetivo de esta prueba es investigar el comportamiento del sistema bajo condiciones que sobrecargan sus recursos. No debe confundirse con las pruebas de volumen: un esfuerzo grande es un pico de volumen de datos que se presenta en un corto período de tiempo.  Puesto que la prueba de esfuerzo involucra un elemento de tiempo, no resulta aplicable a muchos programas, por ejemplo, a un compilador o a una rutina de pagos.  Es aplicable, sin embargo, a programas que trabajan bajo cargas variables, interactivos, de tiempo real y de control de proceso.  Aunque muchas pruebas de esfuerzo representan condiciones que el programa encontrará realmente durante su utilización, muchas otras serán en verdad situaciones que nunca ocurrirán en la realidad. Esto no implica, sin embargo, que estas pruebas no sean útiles.  Si se detectan errores durante estas condiciones “imposibles”, la prueba es valiosa porque es de esperar que los mismos errores puedan presentarse en situaciones reales, algo menos exigentes. |
| Criterio de Completitud: | Todas las pruebas planeadas han sido ejecutadas y excedidas sin que el sistema falle. ( O si las condiciones en que el sistema falle ocurren por fuera de las condiciones especificadas). |
| Consideraciones Especiales: | · Producir stress en la red puede requerir herramientas de red para sobrecargarla de tráfico.  · El espacio en disco utilizado para el sistema debe ser reducido temporalmente para limitar el espacio disponible para el crecimiento de la Base de datos.  · Sincronización de varios clientes accediendo simultáneamente los mismos registros. |
| Técnica a utilizar: | JMeter versión 2.13 |

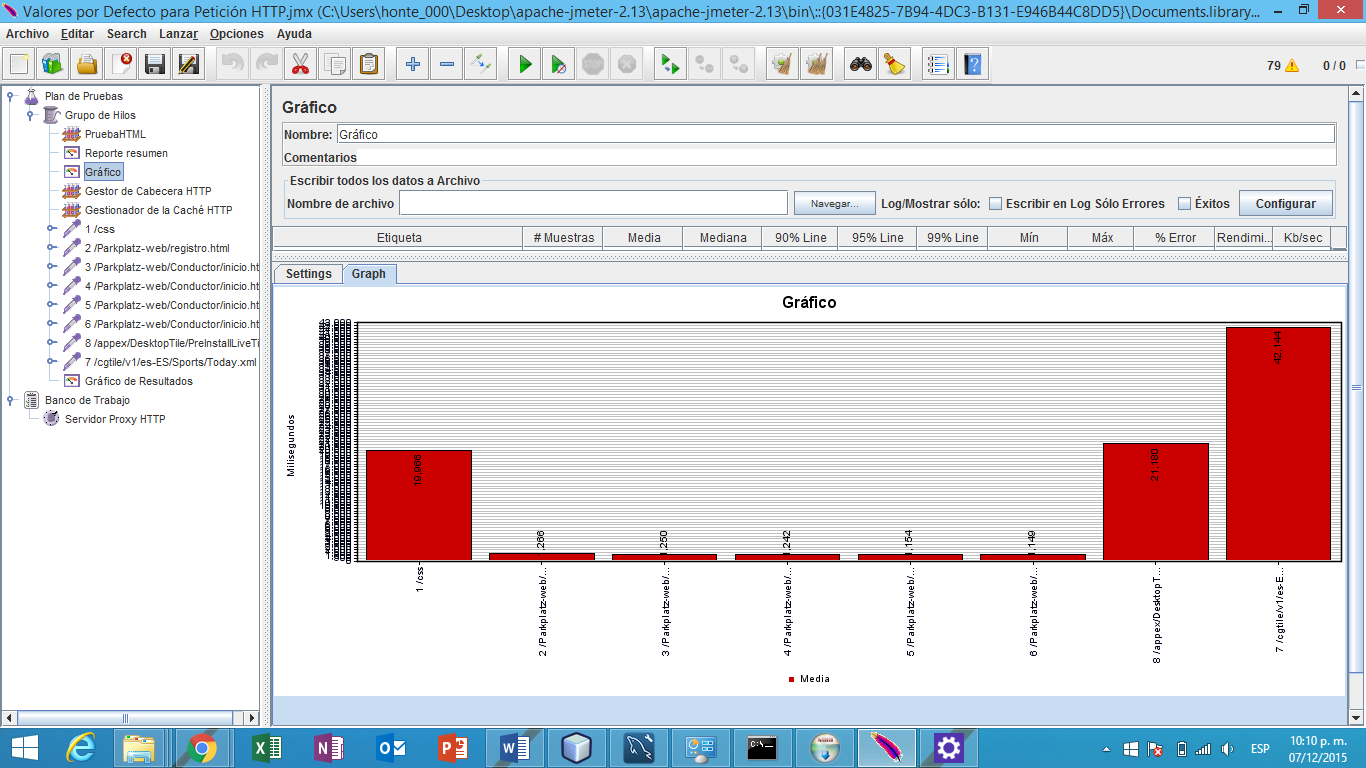
Se le aplico la prueba de estrés al sistema ParkPlatz , el cual consiste en 2 aplicaciones, hasta el momento, base una, la aplicación web, que se dedica a despachar a los administrador y conductores (*Léase Documentación-Alcance para más información)* y, la aplicación de escritorio, que se dedica a la gestión de lugares de aparcamiento así como el registro, y administración de los estacionamientos, a estas 2 aplicaciones se les aplico esta prueba con 3 computadoras esclavas y 1 servidora, en las cuales la servidora le ordena a las computadoras esclavas enviar peticiones a dichas aplicaciones para analizar su comportamiento.

Por ultimo cabe mencionar que esta prueba se le realizo al módulo, conductor y estacionamiento, ya que la aplicación de escritorio probada anteriormente es todo un módulo y la parte del conductor le corresponde a las pruebas de web.

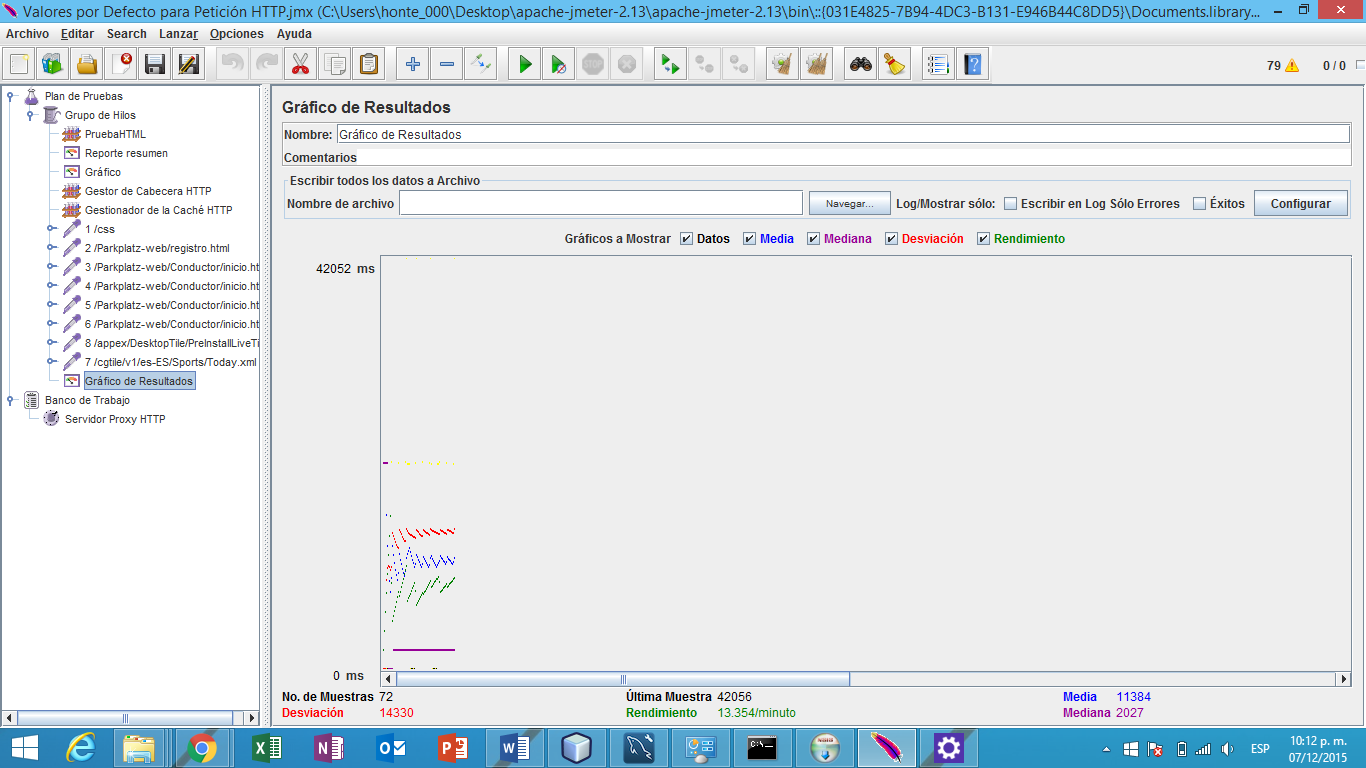


Donde cabe destacar el porcentaje de error de cada página, y hay que notar que los dos últimos dos elementos pertenecen a la aplicación de escritorio.

Por otro parte el reporte grafico quedo de la siguiente forma:



Por último, la gráfica de resultados totales fue la siguiente:



En conclusión, lo máximo que el sistema podría durar bajo estrés, es decir con pocos recursos y mucha demanda, una situación meramente hipotética, es un máximo de 6.1 minutos, con un porcentaje de error total de 87.13.

Cabe destacar que la aplicación con más robustez y más complejidad es la de escritorio a la cual le corresponde el máximo rendimiento bajo estrés de 2.1 minutos con un máximo de 21271 hilos de petición. También en la gráfica cabe destacar que el rendimiento está muy por de cerca de la media, y esto, bajo una situación de estrés es buen indicio de como funcionara el proyecto en situaciones de mucho tráfico de datos.

Aunque un nunca se lleguen a estas situaciones extremistas no hay que desmeritar las pruebas de estrés ya que nos dan una idea de cómo se comporta nuestro sistema en situaciones caóticas en cuanto a flujo de datos con los mínimos recursos posibles y, así poder prevenir futuros errores de sistema.