

## Capítulo 6. Pruebas

### 6.1. Tipos de Pruebas de Software

Aunque no hay una clasificación oficial o formal acerca de los diversos tipos de pruebas de software, existen dos vertientes fundamentales:

- Pruebas de tipo Caja Negra (*Black Box testing*): cuando una aplicación es probada usando su interfaz externa, generalmente la GUI [Katz-Lichtenstein, 2003].
- Pruebas de tipo Caja Blanca (*White Box testing*): cuando una aplicación es probada desde dentro, usando su lógica aplicativa [Katz-Lichtenstein, 2003].

Una prueba de tipo Caja Negra se lleva a cabo sin tener conocimiento de la estructura/funcionamiento interno del sistema, de ahí su nombre. Quien realiza la prueba sólo conoce las entradas apropiadas que deberá recibir la aplicación, así como las correspondientes salidas, sin llegar a saber cómo es que se realiza este proceso [Koudinya, 2003].

Por la otra parte, la prueba de tipo Caja Blanca utiliza datos para realizar la tarea derivados de un análisis directo del código a ser probado; a diferencia de la prueba de tipo Caja Negra, se necesita conocimiento específico del código para analizar los resultados [Webopedia, 2006].

Algunas de las otras clasificaciones que se hacen acerca de las pruebas, incluyen las siguientes:

- de unidad (*unit testing*);
- de módulos;
- de estrés;

- de carga;
- de rendimiento.

Existen muchas otras más, y de entre todas éstas, varias no tiene una definición estándar, por lo cual no se profundizará en el tema.

## **6.2. Tipos de Pruebas Realizadas**

Para probar el software desarrollado en esta Tesis, se utilizó el paradigma de las pruebas de tipo Caja Negra. Específicamente, se implementaron pruebas de rendimiento, ya que el objetivo buscado era asegurar que el sistema fuera capaz de manejar una carga determinada y considerable de trabajo (en base al contexto real en que funcionaría éste) y, a la vez, mantener un buen tiempo de respuesta, lo cual coincide con lo mencionado por ChandraMohan Lingam acerca de este tipo de pruebas en su artículo *Performance Testing and Tuning* [Lingam, 2004].

## **6.3. Pruebas para Aplicaciones Basadas en JavaServer Faces**

Antes de mencionar las herramientas utilizadas, debe destacarse que realizar el tipo de pruebas mencionado anteriormente en aplicaciones basadas en JavaServer Faces, es bastante complejo, principalmente debido a la siguiente razón que se describirá brevemente:

- JSF es un framework dirigido por eventos, por lo cual no se puede realizar un simple *submit* de la página en cuestión para realizar el *request*, sino que se debe llevar a cabo una acción (clic en un botón, link, etc.) en un componente con

*listeners* asociados para que se dispare el evento (*Action* o *Value Change*, véase capítulo sobre JSF) y así procesar las acciones especificadas; en otras palabras, se necesita hacer clic en un botón para navegar a la página correspondiente.

Debido a lo mencionado, no cualquier herramienta puede servir para probar una aplicación basada en JSF, lo que requirió una búsqueda considerable, llegando a la conclusión de que eran necesarias dos de éstas.

También debe señalarse que son muy escasas las fuentes que realicen pruebas sobre JSF, dificultando aún más su implementación.

## **6.4. Herramientas Utilizadas**

Para realizar las pruebas se utilizaron dos herramientas, que por medio de su combinación pudieron cumplir con lo requerido.

Por una parte se utilizó *The Grinder 3*, un framework libre en Java para realizar pruebas de carga haciendo uso de una consola de aplicación gráfica [Aston, 2005]; como su descripción lo indica, esta herramienta se utilizó para mantener ocupado al sistema con trabajo que realizar, para así capturar los tiempos. Esta versión además hace uso del poderoso lenguaje de *script* Jython, lo que permite a cualquier código Java ser encapsulado como una prueba [Aston, 2005], otra funcionalidad muy importante tomada en cuenta.

Para complementar a *The Grinder*, se utilizó *jWebUnit*, otro framework libre en Java que facilita la creación de pruebas para aplicaciones Web. Esta herramienta proporciona un API de alto nivel para navegar a través de una aplicación Web; así mismo ofrece un conjunto de aserciones que permiten verificar que la aplicación se está ejecutando correctamente; esto incluye navegación vía links, definir las entradas de los formularios y

enviarlos, validar tablas de contenidos, etc. [jWebUnit]. jWebUnit se eligió debido a que son clases Java las que realizan las pruebas, permitiendo así encapsularlas a través de Jhyton para ser utilizadas por *The Grinder*. Así mismo, como se mencionó líneas arriba, con *jWebUnit* se pueden simular acciones de usuario, como realizar un clic en un botón, resolviendo así la problemática de JSF mencionada en el apartado anterior.

El funcionamiento de las herramientas utilizadas no será documentado, ya que queda fuera del alcance de esta Tesis, además de que ambas son complejas de utilizar y por lo tanto de describir.

## **6.5. Pruebas Realizadas**

Las pruebas realizadas fueron planeadas para identificar posibles problemas en cuanto al desempeño de casos de uso. Estos fueron elegidos principalmente de acuerdo a su demanda; es decir, se probaron aquellos que serán ejecutados por un número considerable de usuarios al mismo tiempo, pudiendo poner en riesgo el desempeño y tiempo de repuesta del sistema.

Tomando en cuenta el funcionamiento del sistema, los casos de uso con que éste cuenta y el contexto en el que va a funcionar, se llegó a la conclusión que dos casos cubren el perfil descrito en el párrafo anterior.

Las pruebas fueron realizadas de manera local en una computadora personal (PC) con procesador Pentium 4 a 3.4 Ghz. y 1 Gb. de memoria RAM. Los resultados de las pruebas pueden variar de acuerdo al contexto en que se realicen éstas.

A continuación se presentan y describen las pruebas realizadas.

### 6.5.1. Caso 1: Registrar congreso

Debido a las funcionalidades del Sistema Central, éste no será accedido por muchos usuarios ni realizará una gran cantidad de tareas simultáneas. Su carga de trabajo será baja, tomando en cuenta que los congresos tienen un ciclo de vida finito, reduciendo así las operaciones a realizar. El único Caso de Uso que llama la atención es el de registrar congresos, puesto que éste es realizado por organizadores de los eventos, pudiendo mandar peticiones varios de ellos a la misma vez.

Debido a lo señalado, se realizó una prueba de rendimiento sobre el caso de uso mencionado, tomando como base el envío de 300 peticiones de registros. Los resultados obtenidos por la consola de *The Grinder* se muestran en la Figura 6.5.1.

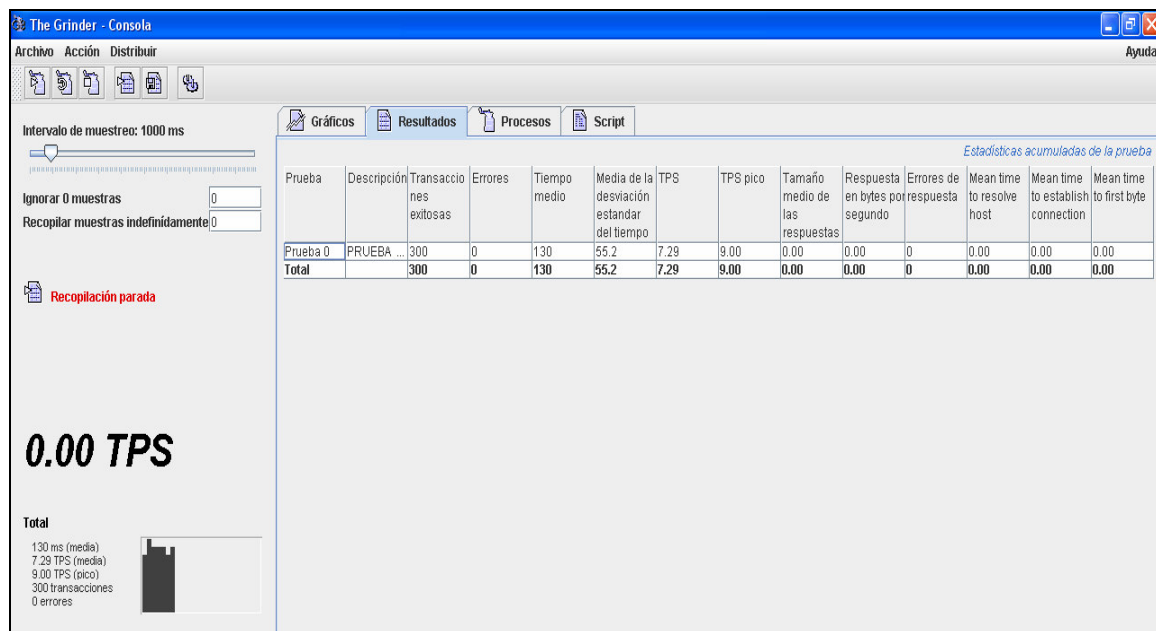


Figura 6.5.1 Resultados Obtenidos

Los resultados sobresalientes que arroja la prueba son:

- Transacciones exitosas: número total de iteraciones de la prueba que fueron ejecutadas exitosamente durante la ejecución de la prueba.
- Tiempo medio: tiempo medio tomado para ejecutar la prueba y recibir una respuesta completa del servidor/aplicación objetivo; en milisegundos.
- Media de la desviación estándar del tiempo: media de la desviación estándar del tiempo tomado para ejecutar la prueba y recibir una respuesta completa del servidor/aplicación objetivo; en milisegundos.
- TPS: transacciones por segundo; el número promedio de iteraciones de la prueba que corrieron exitosamente en un intervalo de un segundo.
- TPS pico: máximas transacciones por segundo; número máximo de iteraciones de la prueba que corrieron exitosamente en un intervalo de un segundo.

Como se observa, el tiempo medio arrojado por la prueba es bastante aceptable a pesar del número de usuarios simulados, por lo cual este caso de uso no presentará problema alguno.

### **6.5.2. Caso 2: Prerregistrar participante**

Para las Aplicaciones Web, el caso de uso que se ejecutará repetida y simultáneamente será definitivamente el prerregistro de participantes. Sin embargo, se ha tomado en cuenta que para este caso de uso, puede afectar la configuración de la aplicación, ya que una configuración simple puede variar bastante de una configuración completa, ya que en esta última se realizan más accesos a la base de datos. Por esta razón se han realizado dos pruebas.

Para el primer caso, la configuración de la aplicación es de tal manera que la página del prerregistro queda como en la Figura 6.5.2.

### Datos del Participante

NOMBRE(S):	<input type="text"/>
APELLIDO PATERNO:	<input type="text"/>
APELLIDO MATERNO:	<input type="text"/>
PROFESIÓN:	<input type="text"/>
GRADO ACADÉMICO:	Estudiante <input type="button" value="v"/>
INSTRUCIÓN / EMPRESA:	<input type="text"/>
E-MAIL:	<input type="text"/>
CALLE:	<input type="text"/>
NÚMERO:	<input type="text"/>
COLONIA:	<input type="text"/>
CÓDIGO POSTAL:	<input type="text"/>
CIUDAD:	<input type="text"/>
ESTADO:	<input type="text"/>
PAÍS:	<input type="text"/>
TELÉFONO:	<input type="text"/>
CELULAR:	<input type="text"/>

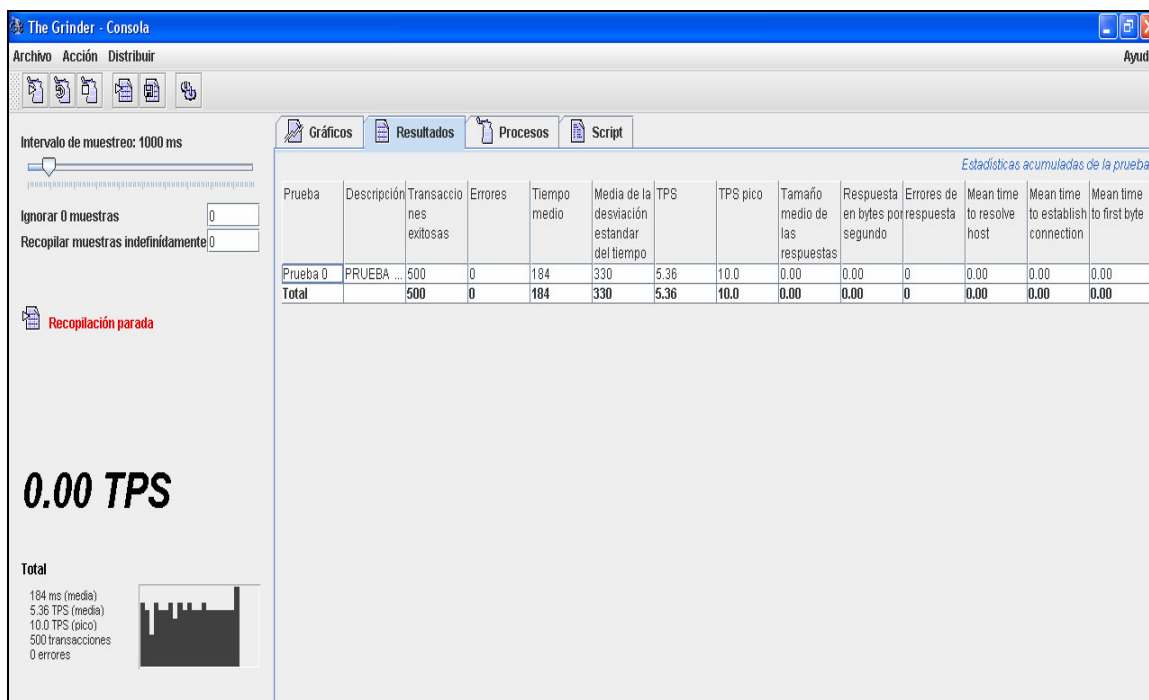
### Rol

ROL:	Asistente <input type="button" value="v"/>
------	--

### Costos

Figura 6.5.2 Página Simple para Prerregistro

Para este primer caso, se simularon 500 usuarios accediendo la aplicación. Los resultados obtenidos se muestran en la Figura 6.5.3.



**Figura 6.5.3** Resultados Obtenidos para Prerregistro Simple

Como se puede observar, el tiempo medio obtenido, 184 milisegundos, es bastante bueno, sin embargo se trata de una aplicación simple, con pocos accesos a la base de datos.

Para el segundo caso se utilizó una aplicación más compleja, cuya página de prerregistro se muestra en la Figura 6.5.4.

Para este caso también se simularon 500 usuarios; los resultados obtenidos se muestran en la Figura 6.5.5.



**Datos del Participante**

NOMBRE(S): \*

APELLIDO PATERNO:

APELLIDO MATERNO:

PROFESIÓN:

GRADO ACADÉMICO:

INSTIRUCIÓN / EMPRESA:

E-MAIL:

CALLE:

NÚMERO:

COLONIA:

CÓDIGO POSTAL:

CIUDAD:

ESTADO:

PAÍS:

TELÉFONO:

CELULAR:

**Rol**

ROL:

**Costos**

TIPO DE PAGO:

IMPORTE:

**Comprobante Fiscal**

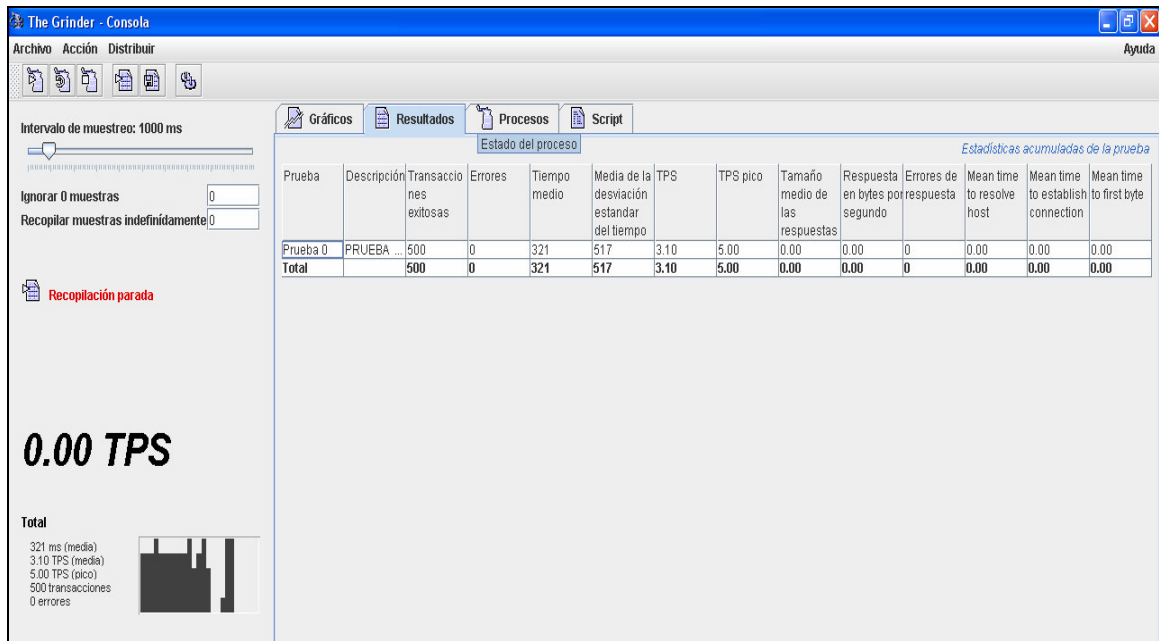
DESEA COMPROBANTE FISCAL:

**Tutoriales**

TUTORIAL 2: Desc Tutorial 2

TUTORIAL 1: Desc Tutorial 1

Figura 6.5.4 Página Compleja para Prerregistro



**Figura 6.5.5** Resultados Obtenidos para Prerregistro Complejo

A pesar de que aumentó 137 milisegundos el tiempo medio, para un total de 321, respecto al Prerregistro simple, sigue siendo un buen tiempo a razón de los 500 usuarios.