



Universidad
de Alcalá

WEB TESTING TOOLS: LOAD, STRESS & PERFORMANCE

TG3 –Grupo T8

Desarrollo de Tecnologías
Emergentes

Pedro Ignacio Santiago (Coordinador)

Sandra Antoral

Bogdan Ionut Ciobanu

Hilera González José Ramón

Universidad de Alcalá

Contenido

1	Autores del trabajo, planificación y entrega.....	3
1.1	Autores	3
1.2	Planificación	3
1.3	Entrega.....	3
2	Requisitos del prototipo a implementar	4
2.1	Requisitos funcionales.....	4
2.2	Otros requisitos	5
3	Criterios de comparación en la implementación.....	5
3.1	Criterio 1: Facilidad de instalación	5
3.2	Criterio 2: Idioma	5
3.3	Criterio 3: Asistente de uso.....	5
3.4	Criterio 4: Diseño de interfaz	5
3.5	Criterio 5: Intuitividad	6
3.6	Criterio 6: Utilización de recursos	6
3.7	Criterio 7: Parámetros a analizar	6
3.8	Criterio 8: Generación de informes	6
3.9	Criterio 9: Requisitos previos a la instalación.....	6
3.10	Criterio 10: Ayudas y tutoriales	6
3.11	Criterio 11: Tiempo invertido en realizar una prueba desde cero	7
4	Informe de la prueba con WebLoad.....	7
4.1	Diseño de la prueba.....	7
4.2	Documentación de construcción de las pruebas.....	8
4.3	Documentación de la configuración de la consola de testeo.....	11
4.4	Documentación del análisis de resultados	13
5	Informe de la prueba con LoadRunner.....	15
5.1	Diseño de la prueba.....	15
5.2	Documentación de construcción de las pruebas.....	15
5.3	Documentación de la configuración de la consola de testeo.....	21
5.4	Documentación del análisis de resultados	25
6	Comparación de las dos implementaciones.....	28
6.1	Evaluación de los criterios en la implementación usando WebLoad	28
6.2	Evaluación de los criterios en la implementación usando LoadRunner.....	28
7	Comparación de la implementación de las tecnologías.....	29
8	Conclusiones	29

1 Autores del trabajo, planificación y entrega

1.1 Autores

El grupo 8 del turno de tarde (T8) está formado por:

- Pedro Ignacio Santiago (Coordinador).
- Bogdan Ionut Ciobanu
- Sandra Antoral

La comparación se realizará entre WebLOAD y LoadRunner, dos herramientas empresariales de testeo de rendimiento, carga y estrés, en aplicaciones y servicios web.

1.2 Planificación

En el siguiente enlace se puede visualizar la planificación de tareas por puntos del trabajo, organizado y adjudicado a cada uno de los miembros del grupo.

<https://www.tomsplanner.es/public/tg2-t8>

El peso de este trabajo en la calificación total de la asignatura es de un 10%, por tanto, requiere de una dedicación de 15 horas del total de 150 horas de la asignatura.

Existen 6 grupos de tareas que se corresponden a los 6 apartados del trabajo con sus correspondientes subtareas. También, están adjudicadas al trabajo realizado por cada componente del grupo, así como de las horas realizadas por cada uno de nosotros.

Se aprecia que la suma de los trabajos realizados en los apartados da un total de 15 horas por persona, siendo el total de horas invertidas en este trabajo de 45.

1.3 Entrega

En el siguiente link, se redirigirá al repositorio de GitHub del grupo T8 donde se podrán encontrar los archivos a calificar con la siguiente nomenclatura:

<https://github.com/PedroISP/TG2>

- Informe del trabajo: con el nombre TG3_final.docx
- Presentación del trabajo: TG3_final.pptx
- Prototipos obtenidos de la implementación de cada una de las tecnologías (Incluye todos los archivos necesarios para la instalación y uso de cada prototipo):
 - o PrototipoWebLoad.zip
 - o PrototipoLoadRunner.zip

2 Requisitos del prototipo a implementar

Partimos de la premisa de que nuestra tecnología no parte de ningún desarrollo ni necesita de ningún diseño para ponerla en funcionamiento.

Las dos suites de aplicaciones para testear servicios, aplicaciones y sistemas web cumplen con un montón de funcionalidades que nos permiten en ambos casos realizar casi cualquier tarea relacionada con la investigación.

En conclusión los requisitos funcionales y los criterios que propondremos a lo largo del trabajo, se centran más en aspectos no tan técnicos como pueden ser la facilidad de uso, instalación y prerequisites, tiempos de desarrollo de las pruebas a realizar, tiempos de ejecución de las pruebas y algunos más que detallaremos a continuación.

El objetivo del proyecto es comparar la implementación de un mismo prototipo de sistema utilizando dos tecnologías diferentes (A y B).

Es importante cumplimentar este apartado antes de empezar a implementar el prototipo de cada tecnología, porque ambos prototipos deben cumplir los requisitos que se establezcan en este apartado. Si se van a crear dos equipos de trabajo, uno para cada prototipo, el contenido de este apartado es lo que han de compartir ambos equipos como punto de partida.

Cuanto más detallados sean los requisitos, mayor será la precisión en la comparación que se realizará al final del trabajo. Se trata de conseguir dos prototipos con igual funcionalidad, pero utilizando diferentes tecnologías.

Se puede dar libertad a los equipos de desarrollo en cuanto al diseño, pero la funcionalidad debe ser lo más parecida posible. Por ejemplo, no es necesario que los colores utilizados en las pantallas sean exactamente los mismos en ambos prototipos, a no ser que los miembros del grupo lo hayan decidido así, en cuyo caso, esos detalles de colores deben incluirse en el catálogo de requisitos, para que ambos equipos los cumplan.

2.1 Requisitos funcionales

En la siguiente tabla se indicará el catálogo de requisitos funcionales del sistema.

REQ.	DESCRIPCIÓN
RF01	Facilidad de instalación
RF02	Idioma
RF03	Asistente de uso
RF04	Diseño de interfaz
RF05	Intuitividad
RF06	Utilización de recursos
RF07	Parámetros a analizar
RF08	Generación de informes

2.2 Otros requisitos

En la siguiente tabla se indicará el catálogo de requisitos no funcionales del sistema.

REQ.	DESCRIPCIÓN
R01	Requisitos previos a la instalación
R02	Ayudas y tutoriales
R03	Tiempo invertido en realizar una prueba desde cero

3 Criterios de comparación en la implementación

En el trabajo TG2 se definieron criterios de comparación de las dos tecnologías a nivel teórico.

En este trabajo se definen criterios para la comparación de la implementación de las tecnologías en la construcción del prototipo de sistema de ejemplo, cuyos requisitos son los establecidos en el apartado 2.

3.1 Criterio 1: Facilidad de instalación

- Nombre del criterio: Facilidad de instalación de todos los componentes del sistema.
- Descripción: Facilidad que ofrece el instalador de la aplicación a la hora de realizar una instalación predeterminada y básica para realizar una prueba normal.
- Tipo de valor: Escala de 1 a 10.

3.2 Criterio 2: Idioma

- Nombre del criterio: Idioma
- Descripción: Lengua en la que está desarrollada la aplicación
- Tipo de valor: Cadena de caracteres

3.3 Criterio 3: Asistente de uso

- Nombre del criterio: Asistente de uso como guía para utilizar la aplicación
- Descripción: Magnitud que nos permite evaluar la relación facilidad-uso de los asistentes para la generación de pruebas
- Tipo de valor: Escala de 1 a 10

3.4 Criterio 4: Diseño de interfaz

- Nombre del criterio: Diseño de la interfaz
- Descripción: Magnitud que nos permite evaluar el diseño de la interfaz para modificar, insertar y borrar datos y parámetros para la ejecución de pruebas
- Tipo de valor: Escala de 1 a 10

3.5 Criterio 5: Intuitividad

- Nombre del criterio: Intuitividad
- Descripción: Magnitud que nos permite evaluar el grado de diseño que ofrece la aplicación para guiarnos fácilmente por la interfaz
- Tipo de valor: Escala de 1 a 10

3.6 Criterio 6: Utilización de recursos

- Nombre del criterio: Utilización de recursos del sistema
- Descripción: Nos permitirá diferenciar las dos herramientas en función del impacto que tienen sobre el sistema sobre el que se ejecutan las pruebas.
- Tipo de valor: Enumerado [Alto-Bajo]

3.7 Criterio 7: Parámetros a analizar

- Nombre del criterio: Parámetros a analizar
- Descripción: Facilidad de elección sobre una serie de parámetros a analizar referentes a la prueba a testear.
- Tipo de valor: Escala de 1 a 10

3.8 Criterio 8: Generación de informes

- Nombre del criterio: Generación de informes
- Descripción: Magnitud que nos permite evaluar la capacidad de generar informes y el estilo y morfología de estos así como, el número de formatos posibles.
- Tipo de valor: Escala de 1 a 10

3.9 Criterio 9: Requisitos previos a la instalación

- Nombre del criterio: Requisitos previos a la instalación
- Descripción: Programas y características necesarias del sistema previas a la instalación de las aplicaciones
- Tipo de valor: Cadena de caracteres

3.10 Criterio 10: Ayudas y tutoriales

- Nombre del criterio: Ayudas y tutoriales
- Descripción: Magnitud que evalúa la relación calidad-cantidad de los tutoriales accesibles on-line
- Tipo de valor: Magnitud de 1 a 10

3.11 Criterio 11: Tiempo invertido en realizar una prueba desde cero

- Nombre del criterio: Tiempo invertido
- Descripción: Tiempo invertido en realizar una prueba desde que se abre el programa hasta que se extraen las analíticas en un archivo de lectura.
- Tipo de valor: Minutos

4 Informe de la prueba con WebLoad

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando WebLoad, cuyos requisitos funcionales se enumeraron en el apartado 2.

Antes de comenzar se requiere una preparación previa a la instalación para el correcto funcionamiento del programa.

- Es necesaria la apertura de los puertos 9000 y 9010 en los canales TCP y UDP en el firewall de Windows
- La instalación del gestor de bases de datos PostgreSQL y su administrador gráfico pgAdmin III. Dicho gestor es necesario para el almacenamiento y posterior análisis de los resultados de las pruebas.

En nuestro caso ambos prerrequisitos se han realizado a mano siguiendo los tutoriales de instalación y realizando configuraciones manualmente ya que el autoejecutable que viene con el programa no generaba automáticamente la base de datos en postgres.

4.1 Diseño de la prueba

La prueba a realizar se realizará sobre la página:

- <http://www.webloadmpstore.com/index.php>

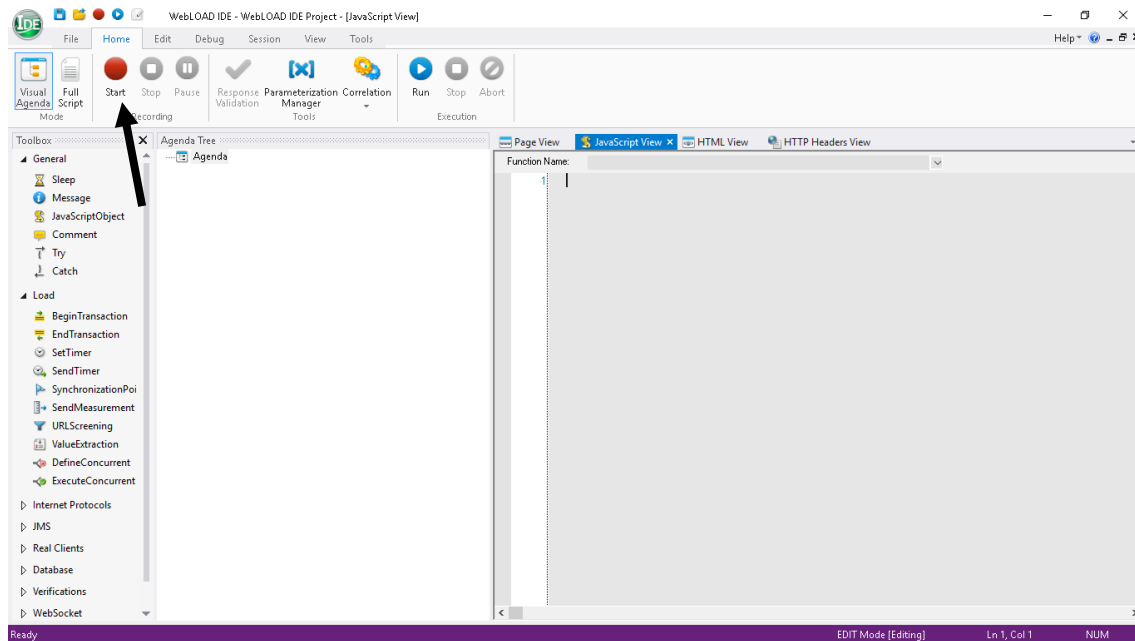
En la cual realizaremos el siguiente recorrido:

1. Ingresar en <http://webloadmpstore.com/>
2. Hacer login
 - a. User: demo
 - b. Pass: demo
3. Hacer búsqueda, arriba a la derecha, "CONSOLE"
4. Click en "WebLOAD Console Demo"
5. Click en "CART" (Carro de la compra)
6. Borrar producto

4.2 Documentación de construcción de las pruebas

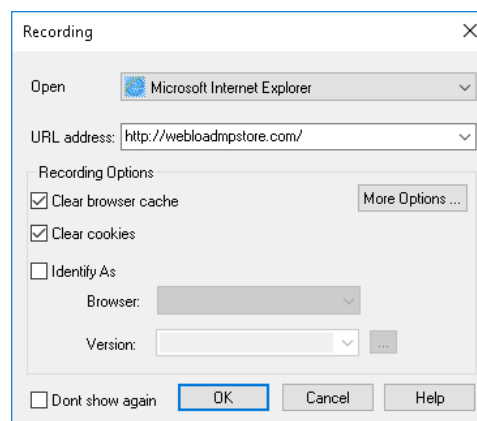
Iniciamos el IDE de WebLoad donde generaremos la “agenda” como se denomina técnicamente para este programa. Esta agenda está formada por un script en lenguaje JavaScript donde se recoge toda la información que se produce al ejecutar una navegación por una página web.

Creamos un nuevo proyecto y nos aparecerá la siguiente ventana



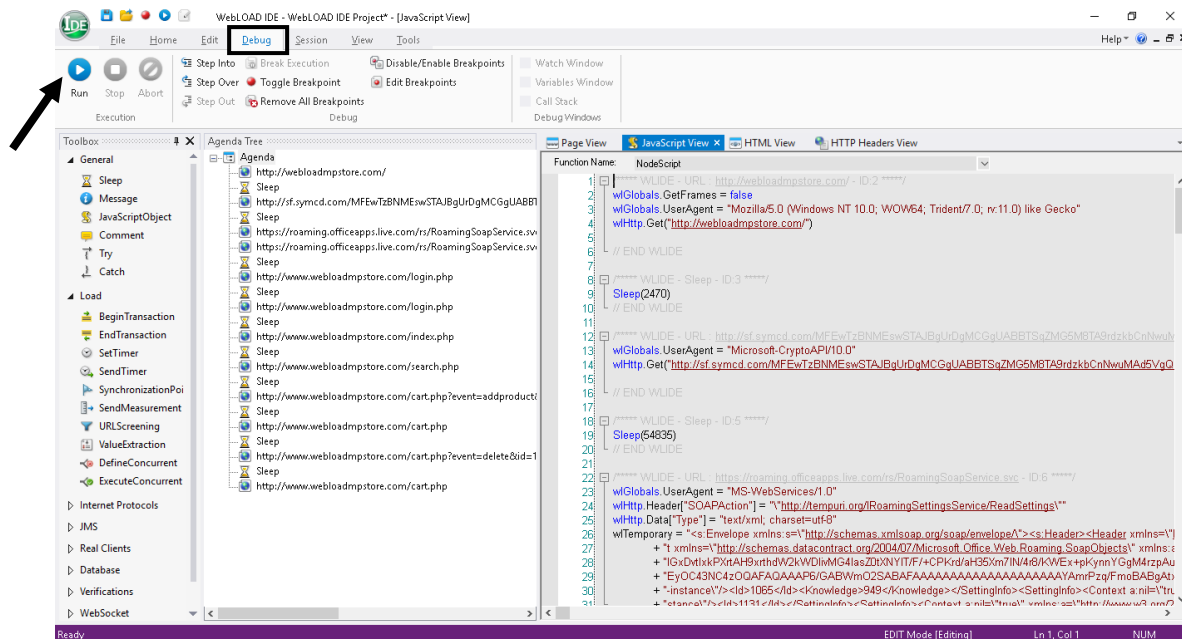
1 Interfaz principal WebLoad

Pulsaremos el botón Start y comenzaremos a configurar la prueba anteriormente descrita:



2 Configuración inicio de Agenda

Una vez realizado el recorrido por el navegador web, pausaremos la grabación de la agenda en la ventana emergente. Tras este paso aparecerá un dialogo que nos preguntará si queremos realizar una correlación entre los valores dinámicos que envía el servidor. Pulsamos sobre el botón de generación de correlación automática, señalamos todos los puntos de la lista y pulsamos okay.

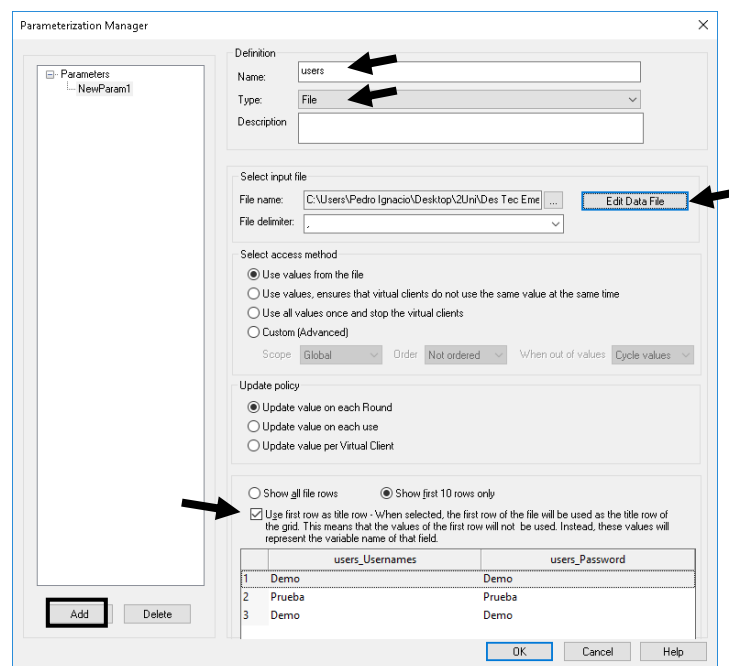


3 Agenda generada

En la siguiente pantalla se muestra el script global generado además de otras 3 pestañas que te permiten visualizar el aspecto de cada página visitada en el registro de la agenda así como su código html y las cabeceras de http.

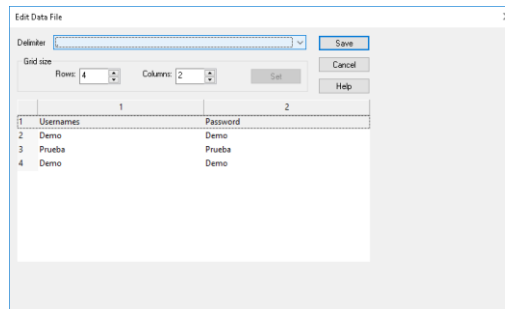
El siguiente paso es la realización de la depuración del código que se inicia en la pestaña “Debug” haciendo click en el botón “Run”.

Lo siguiente que haremos será editar el script para poder extraer de un archivo los nombres de usuario y contraseña de forma aleatoria. En la pestaña “Session” hacemos clic sobre el botón “Start Edit Mode” que nos permitirá realizar cambios sobre el código. Seguidamente pasamos a la pestaña “Home” y pulsamos sobre “Parameterization Manager”. Añadimos un nuevo parámetro de nombre “users”.



4 Gestor de parametrización

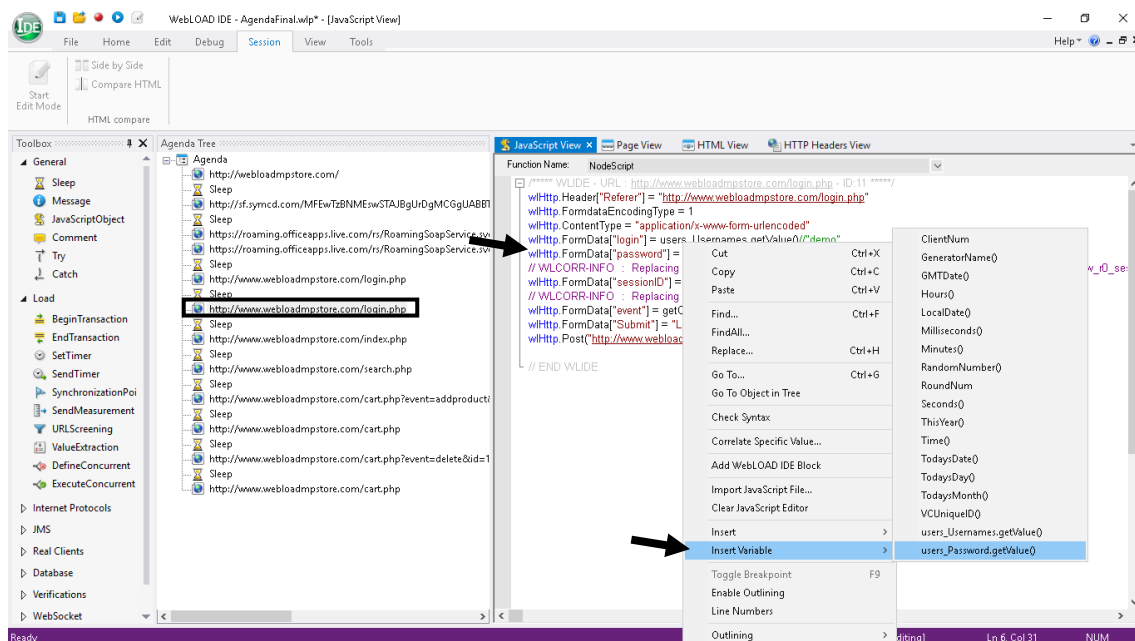
Seleccionamos que el tipo de parámetros sea a partir de un archivo. En ese momento aparecerá una ventana de configuración en la cual haremos clic sobre “Edit Data File” y nos aparecerá la siguiente ventana, que deberemos configurar así.



5 Tabla de usuarios y contraseñas

Por último seleccionamos la casilla de verificación para que la primera fila sea la cabecera de la tabla.

Volvemos a seleccionar la pestaña “Session”. Pinchamos en la visualización de javascript y buscamos en el árbol de la agenda la línea de código que pertenece al login



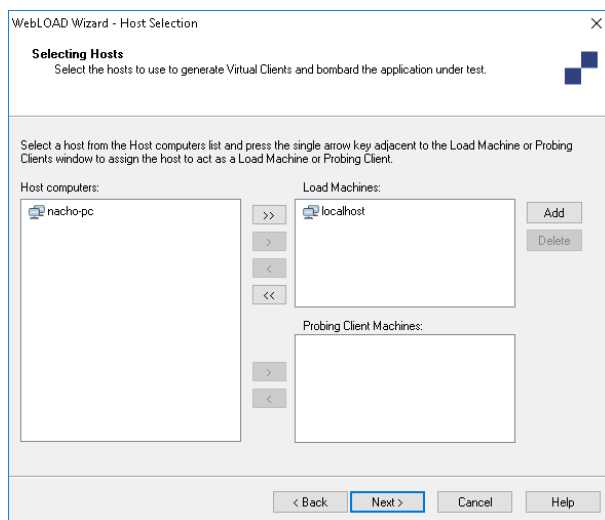
6 Extraer usuarios por código

En el código nos situamos sobre la asignación del usuario y contraseña y pulsamos botón derecho “insert Variable” y seleccionamos las dos funciones que nos devolverán las variables a partir del archivo.

Por último, pulsamos en la pestaña Tools y abrimos “WebLoad Console”

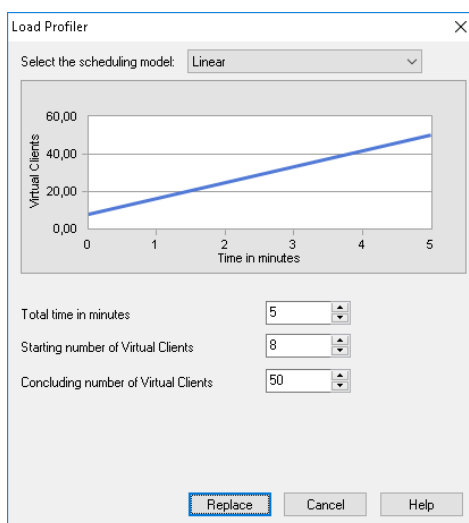
4.3 Documentación de la configuración de la consola de testeo

Una vez tengamos la agenda bien editada pasamos a la consola para generar la carga y comenzar el testeo. La primera pantalla del asistente nos pregunta el origen del archivo que queremos testear.



7 Generador de carga

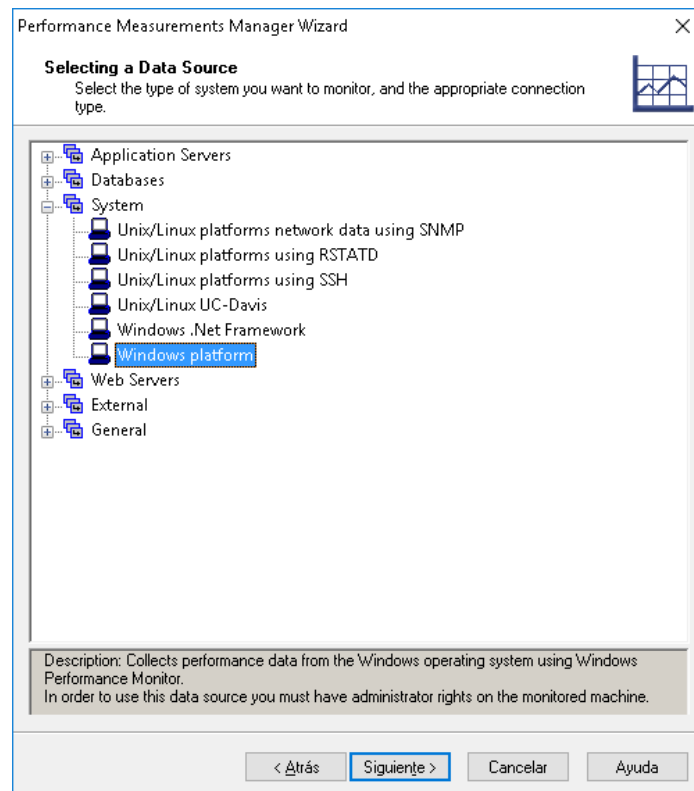
En la siguiente pantalla elegimos la máquina que será la encargada de generar la carga de datos. La siguiente configuración seleccionamos sobre “Colective Scheduling” y pulsamos siguiente.



8 Configuración de carga sobre test

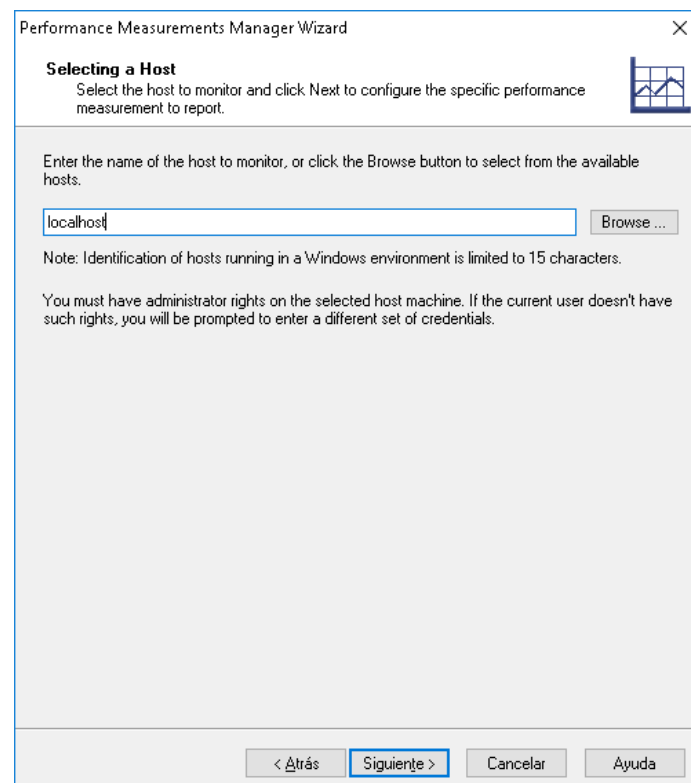
En la siguiente pantalla se seleccionan las celdas y pulsamos sobre el botón de la derecha “Load Profiler” y configuramos tal cual aparece en la captura anterior. Dicha ventana nos permite elegir la cantidad de carga en el tiempo con diferentes esquemas, además de poder configurar el tiempo de ejecución del test y el número de usuarios iniciales y finales.

La pantalla siguiente nos permite poner en marcha monitores de ejecución para obtener datos. Nosotros lo utilizaremos para obtener métricas del sistema. Pinchamos en “Add Monitors”. Nos aparecerá otro cuadro de dialogo donde tendremos que pinchar en la barra superior sobre “Add Data Source”.



9 Monitorización del sistema

En el asistente seleccionamos “Windows Plataforma”. Seguimos aceptando y en la configuración desde donde van a recibir los datos escribimos “localhost”



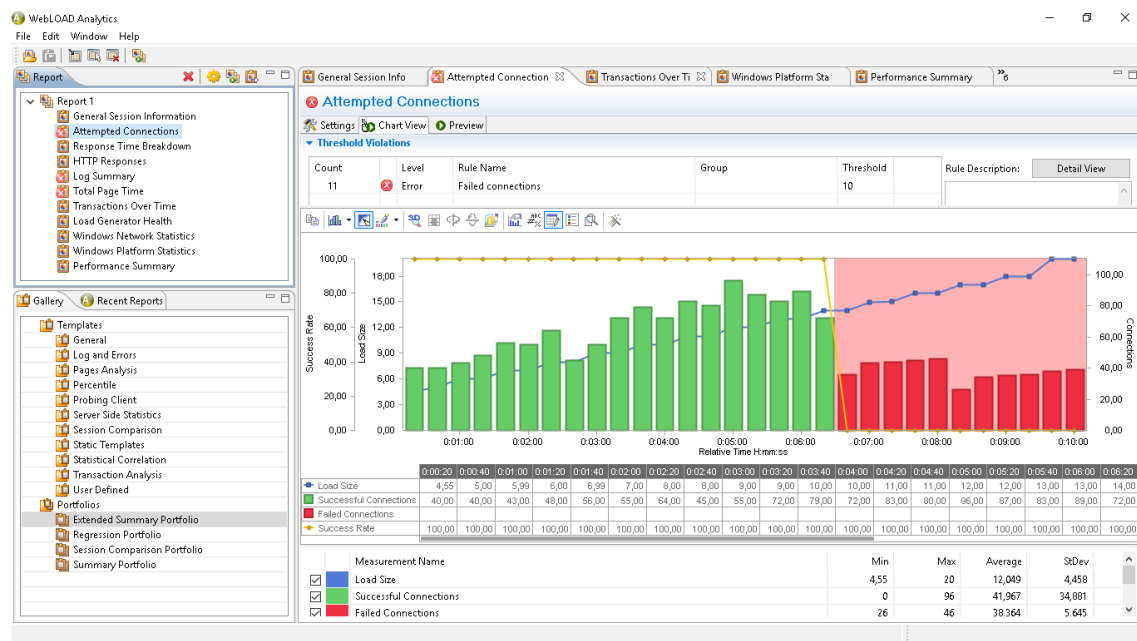
10 Fuente de datos

Pulsamos siguiente hasta finalizar la configuración que nos devolverá al cuadro inicial donde pulsaremos “Close & Update”. En el último punto deshabilitaremos la opción de iniciar automáticamente tras cerrar el asistente.

Para comenzar el testeo y la carga, pulsaremos el botón play de la barra de navegación. A partir de ahí el programa comenzara a ejecutar las pruebas de carga durante 5 minutos

4.4 Documentación del análisis de resultados

En este punto de la prueba solo queda visualizar los datos en forma de tablas y gráficas y recibirlo por pantalla.



11 Interfaz Analytics

WebLoad genera de forma predeterminada un conjunto de informes sobre los datos obtenidos como:

- Información general de la sesión
- Intento de conexiones
- Desglose de tiempo de respuesta
- Respuestas de HTTP
- Resumen de registro
- Tiempo de respuesta
- Transacciones en el tiempo
- Estado del generador de carga
- Estadísticas de la red de Windows
- Estadísticas de la plataforma Windows
- Resumen de ejecución

Estos informes son organizados y ordenados y es posible su extracción en formato de archivos de lectura como:

- Word
- PDF
- Excel WorkBook
- Excel
- HTML
- Formato de texto enriquecido
- Open document text
- Formato CSV

En el anexo se puede comprobar el reportaje emitido.

5 Informe de la prueba con LoadRunner

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología B, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

Para realizar las pruebas con LoadRunner, a diferencia de WebLoad, no hay que hacer ningún paso adicional después de instalar el programa con el instalador que nos provee HP.

LoadRunner, trae como opcional la posibilidad de instalar una base de datos y ejecutar una web de ejemplo en la que se pueden reservar vuelos. Esto es para realizar las pruebas y practicar sobre una web de prueba. En nuestro caso, no lo instalaremos ya que haremos las pruebas sobre una web totalmente online. Por lo tanto, como hemos mencionado al principio, después de instalar LoadRunner con su instalador, no hay que hacer ningún paso adicional.

5.1 Diseño de la prueba

La prueba a realizar se realizará sobre la página:

- <http://www.webloadmpstore.com/index.php>

En la cual realizaremos el siguiente recorrido:

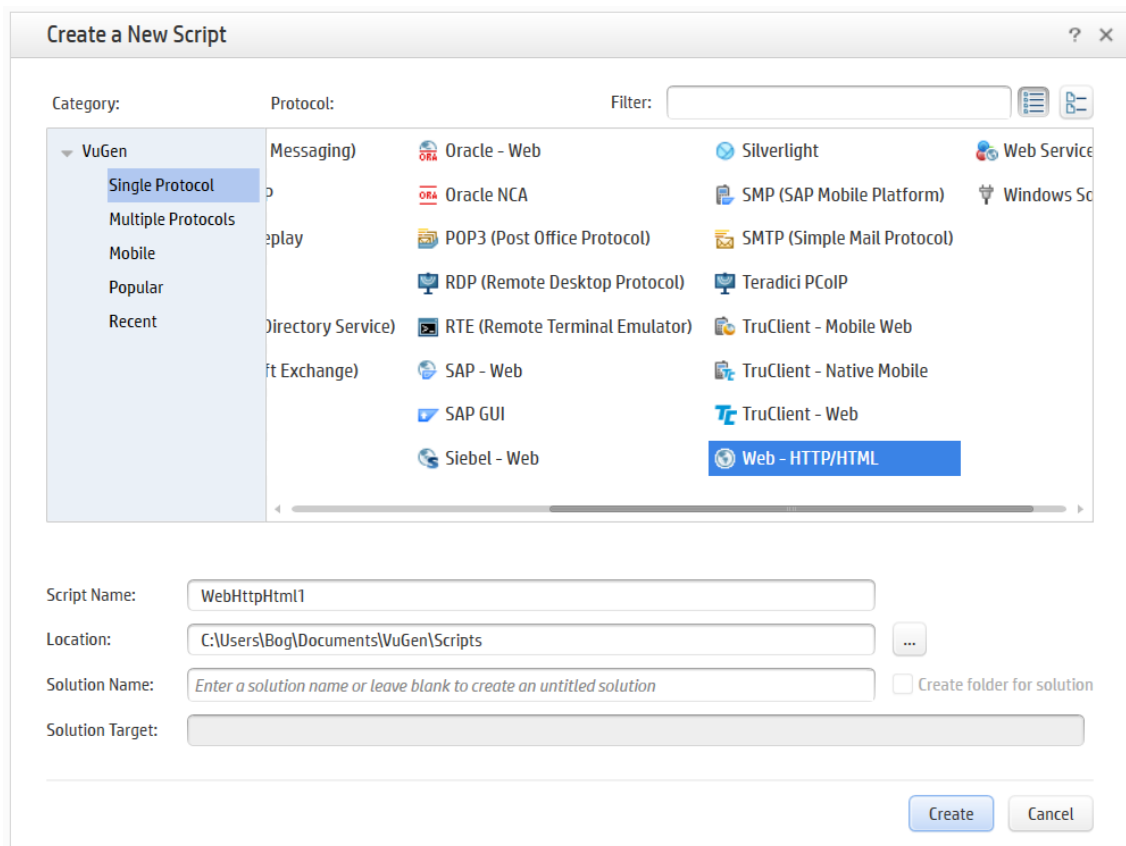
7. Ingresar en <http://webloadmpstore.com/>
8. Hacer login
 - a. User: demo
 - b. Pass: demo
9. Hacer búsqueda, arriba a la derecha, "CONSOLE"
10. Click en "WebLOADConsole Demo"
11. Click en "CART" (Carro de la compra)
12. Borrar producto

5.2 Documentación de construcción de las pruebas

Para comenzar, hay que saber que LoadRunner tiene 3 aplicaciones separadas, la primera llamada "Virtual User Generator", la segunda llamada "Controller" y la tercera llamada "Analysis".

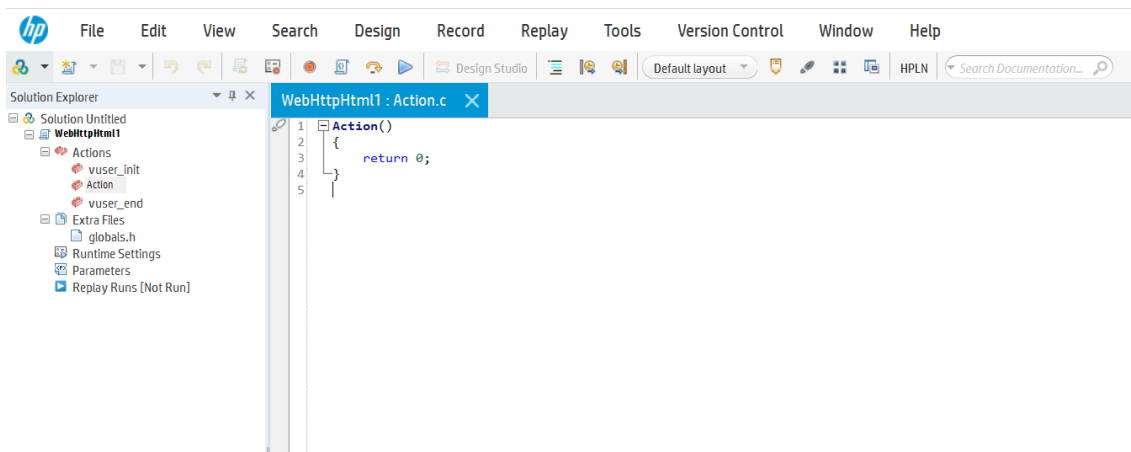
La construcción de la prueba comienza con la primera aplicación "Virtual User Generator", en la que diseñaremos el recorrido que realizara por la web el usuario virtual.

Abrimos la aplicación y una vez dentro nos dirigimos a la esquina izquierda superior, y pinchamos en "File>New Script and Solution".



En la pantalla superior, tenemos que seleccionar el protocolo utilizado, en nuestro caso se trata de una web simple, así que elegimos "Web - HTTP/HTML". Podemos darle también un nombre específico al script o dejarlo tal cual esta. La ruta en "Location" es mejor dejarlo como aparece por defecto ya que el segundo programa, como veremos más adelante, buscara estos scripts en esa ruta.

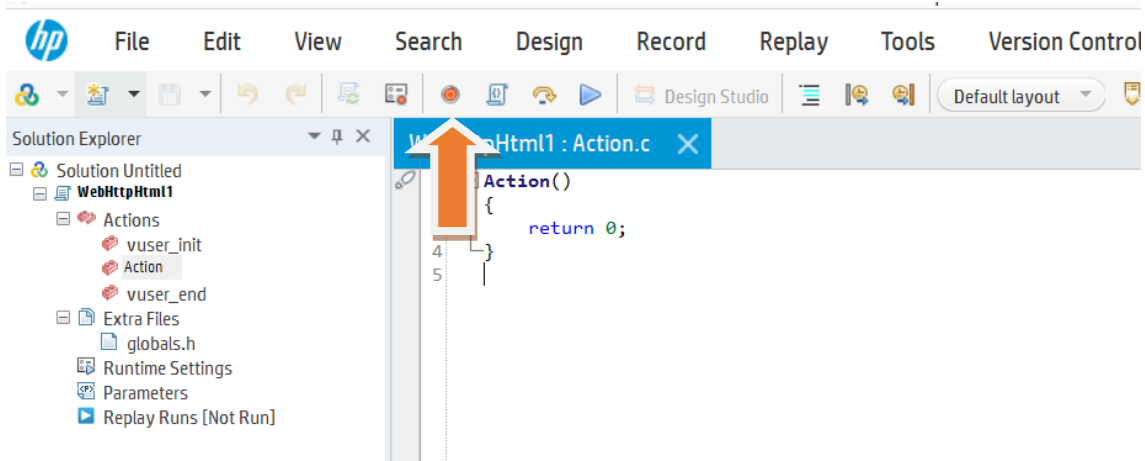
Una vez le demos a Create tendremos lo siguiente:



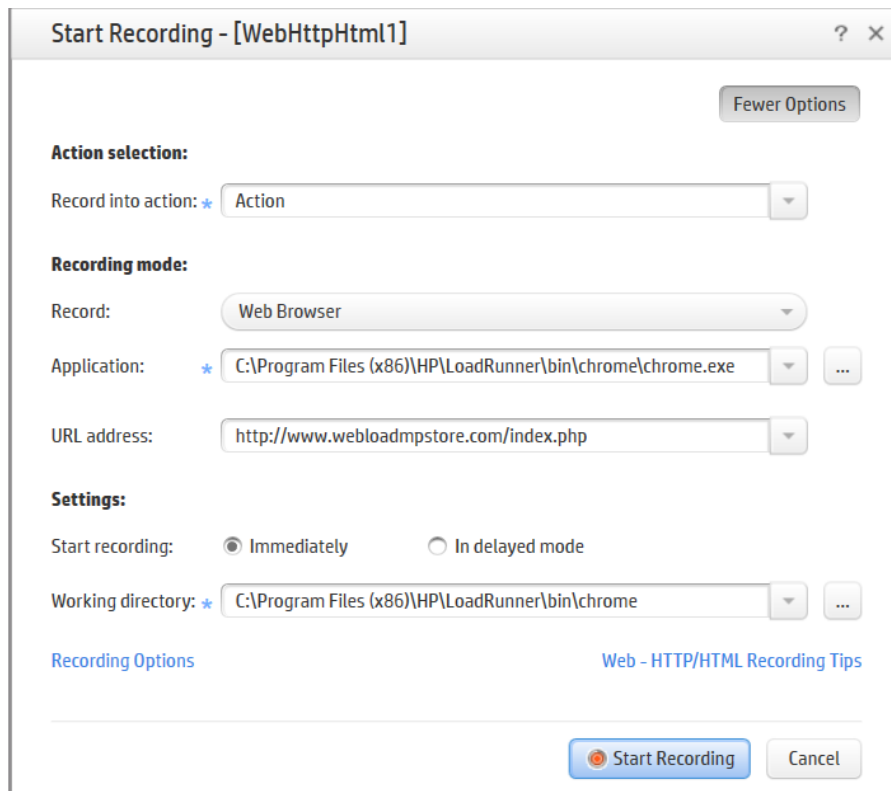
Este apartado corresponde al script que indicara al usuario virtual que hacer. Como vemos al principio estará totalmente vacío.

Como podemos ver en el desplegable de la izquierda hay bastantes archivos, aunque el más importante es "Action" que corresponde a que hará el usuario virtual. Otra parte importante del desplegable es "Runtime Settings", en este apartado podremos indicar cuantas iteraciones queremos que realice el usuario de este script aunque en nuestro caso no lo tocaremos y lo dejaremos por defecto en 1.

El siguiente paso, es grabar el script, si no estamos ya en el apartado "Action", le damos doble click en el desplegable de la izquierda y una vez abierto podemos seleccionar del menú superior la opción "Record > Record" o bien darle al siguiente botón rojo (se señala con una flecha para que se vea claramente):



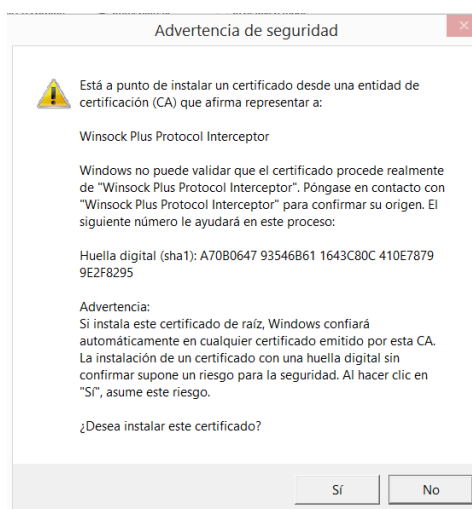
Una vez le demos a dicho botón, se nos abrirá una pestaña de configuración que deberemos configurar como la siguiente (para este ejemplo claro):



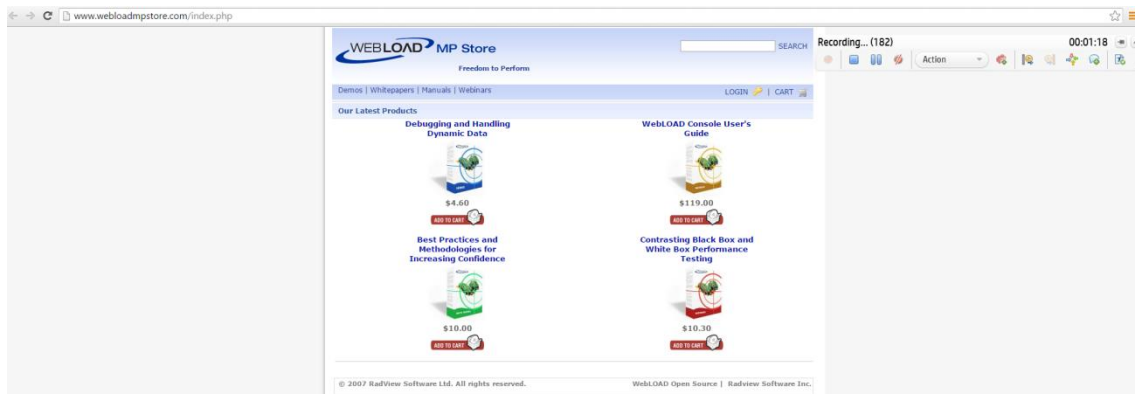
Como podemos ver, en "Application" aparece una ruta hacia chrome.exe, esto es debido a que por defecto el navegador predeterminado que se nos abrirá para grabar el recorrido del usuario es Internet Explorer pero, si damos a los tres puntitos que están al lado "..." podemos buscar dentro de la carpeta de LoadRunner otros navegadores como son por ejemplo Firefox o en este caso Chrome.

URL Address es la dirección a la web a la que se realizara el testeo, cuando le demos al botón "Start Recording" dicha web se nos abrirá en el navegador que hemos elegido en el paso anterior.

El apartado de "Settings" lo dejaremos tal y como viene por defecto y pulsamos el botón "Start Recording".



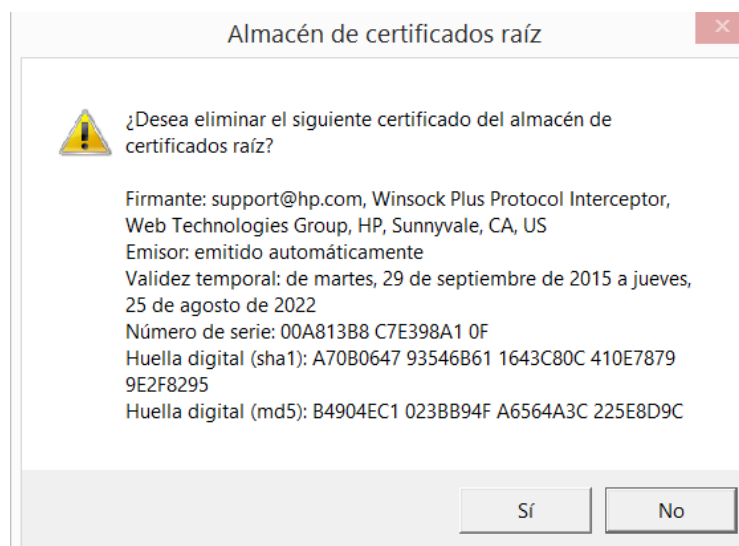
Se nos preguntara si queremos instalar un certificado al cual tenemos que decirle que sí, cuando acabemos la prueba, se nos preguntara si queremos borrar el certificado así que no habrá ningún problema, no se quedara en el ordenador.



Como podemos ver en la captura superior, cuando arranca el programa, tendremos la web a la que queremos hacerle la prueba y a la derecha nos aparecerá una barra flotante que podemos mover a nuestro antojo. Una vez se abra la página como en la captura superior, tendremos que realizar el recorrido por el cual pasara nuestro usuario virtual. Todo se quedara grabado de forma automática, el tiempo entre acciones, también es tenido en cuenta, por ejemplo, si tardamos varios segundos en entrar a un enlace el usuario virtual también lo hará.

Cuando acabemos con el recorrido del usuario, en la barra flotante tendremos que darle al botón de stop (segundo botón empezando por la izquierda de la barra flotante).

Una vez parado, se nos preguntara si queremos eliminar el certificado que hemos instalado antes, le indicaremos que si:




Una vez realizado lo anterior, se nos habrá generado el script del usuario virtual:

```
1 Action()
2 {
3
4     web_url("index.php",
5         "URL=http://www.webloadmpstore.com/index.php",
6         "Resources=0",
7         "RecContentType=text/html",
8         "Referer=",
9         "Snapshot=t3.inf",
10        "Mode=HTML",
11        EXTRARES,
12        "Url=/media/images/border_bg_1.gif", ENDITEM,
13        "Url=/media/images/header_bg.gif", ENDITEM,
14        "Url=/media/images/th.gif", ENDITEM,
15        "Url=https://translate.googleapis.com/translate_a/l?client=chrome&hl=es&key=dumytoken&alpha=1", "Referer=", ENDITEM,
16        LAST);
17
18    web_set_sockets_option("SSL_VERSION", "2&3");
19
20    web_url("crx",
21        "URL=http://clients2.google.com/service/update2/crx?os=win&arch=x64&nacl_arch=x86-64&prod=chromiumcrx&prodchannel=&prodversion=46.0.2490.88&lang=es&x=id&3Dnmw",
22        "Resources=0",
23        "RecContentType=text/xml",
24        "Referer=",
25        "Snapshot=t4.inf",
26        "Mode=HTML",
27        EXTRARES,
28        "Url=https://r8--s-n-w51luxa-h5q6.gvt1.com/crx/blebs/QnAAAC6zw@qH2D7tnXe8Z7rUJP365VNIyEHZEyx887CNu8-6Jc6yJZgxojkhTexB0E4rQ44m485ex2EftUGu580UBf58DEVEK=801xsr",
29        "Url=https://clients2.googleusercontent.com/crx/blebs/QnAAAC6zw@qH2D7tnXe8Z7rUJP2Eiatf_dMDEdu7k2IvH9Za_eevz2K1-kUn7xv8-XfnyOIThiFhPe8pF0-rAlYrwd8ImoAyZFKg",
30        "Url=https://ssl.gstatic.com/safebrowsing/csd/client_model_v5_ext_variation_0.pb", "Referer=", ENDITEM,
31        "Url=https://ssl.gstatic.com/safebrowsing/csd/client_model_v5_ext_variation_0.pb", "Referer=", ENDITEM,
32        "Url=https://www.google.com/searchdomaincheck?format=domain&type=chrome", "Referer=", ENDITEM,
33        "Url=https://chrome.google.com/webstore/inlineinstall/detail/efaidnbmnmbicaipecgcplefindmkai", "Referer=", ENDITEM,
34        LAST);
35
36    In_think_time(15);
37
38    web_url("login.php",
39        "URL=http://www.webloadmpstore.com/login.php",
40        "Resources=0",
41        LAST);
42 }
```


En este paso, tendremos que comprobar que el script generado por el programa funciona correctamente, por lo que en la barra superior le damos al botón "Replay" o en el menú superior "Replay>Run", también nos valdría pulsar "F5" en nuestro teclado.

Esto lo que hará será ejecutar el usuario virtual para ver que funciona correctamente, si todo está bien, tendrá que salirnos la siguiente pantalla:

Script Name: PruebaLoadRunner

 **Script Passed**

Elapsed time: 00:14.984 mm:ss.ms **Started at:** 03/05/2017 16:59:52 **Think time:** 00:00.000 mm:ss.ms
Ended at: 03/05/2017 17:00:07 **Wasted time:** 00:00.000 mm:ss.ms

 [Open NV Analytics Report](#)

For additional details, enable the collection of replay statistics.

Choose:

Tools > Options > Scripting > Replay (select "Collect replay statistics")

These statistics include transaction breakdown information, connections, and various HTTP-related items.

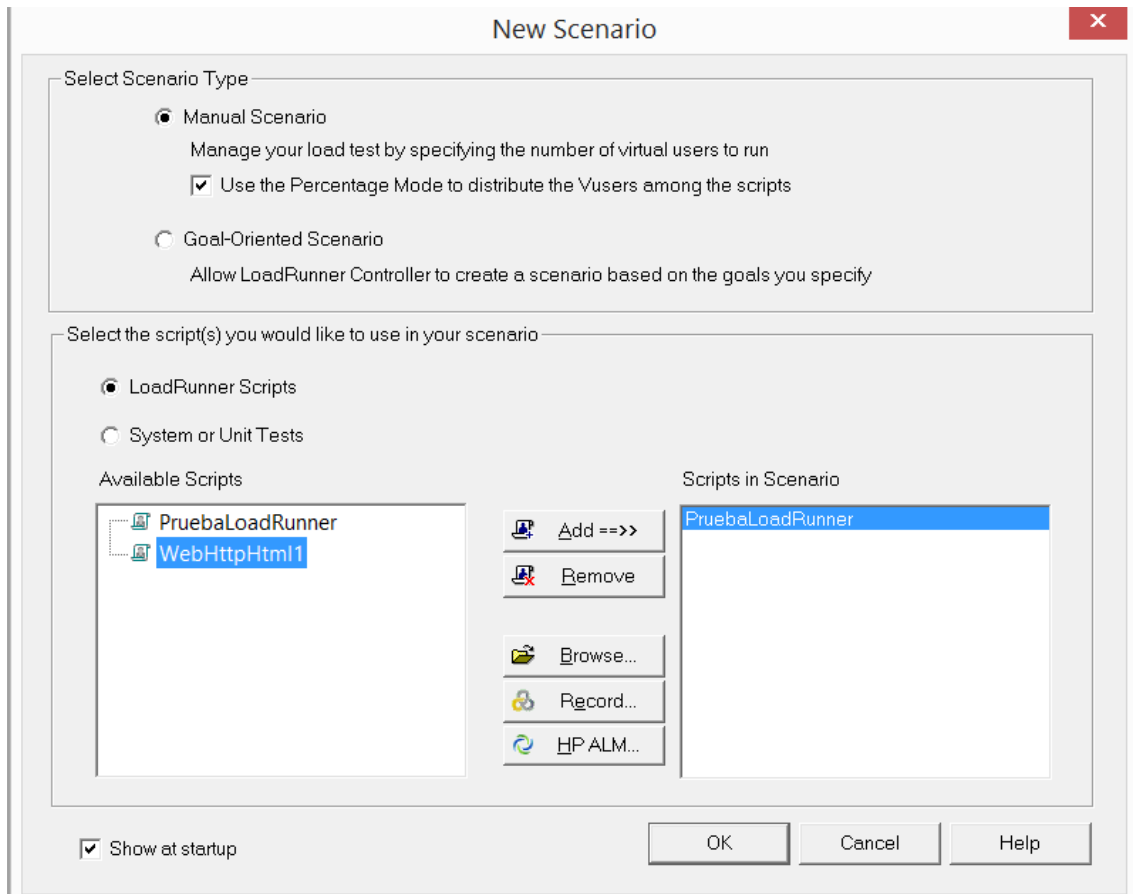
Note: Enabling this option may affect the replay performance.

Si todo está bien, y el script no ha fallado, guardamos y ya hemos acabado con esta parte del programa. Si por el contrario el script ha fallado, podremos editar el script manualmente en busca del fallo o bien volver a grabar el script desde el principio como hemos indicado antes.

5.3 Documentación de la configuración de la consola de testeo

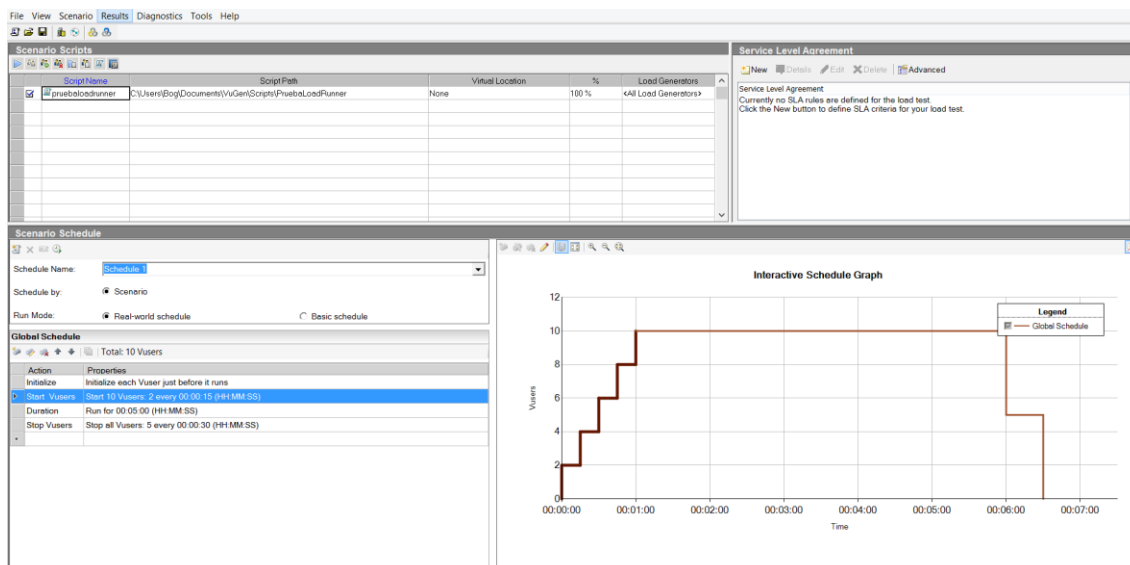
Para comenzar con este apartado, tenemos que abrir la segunda aplicación que nos provee el programa LoadRunner. Se trata de la aplicación "Controller".

Una vez abierto, tendremos que crear un nuevo escenario, eligiendo el script que hemos creado anteriormente para los usuarios virtuales:



Y lo añadimos en "Scripts in Scenario". Pulsamos OK.

Se nos abrirá la siguiente pantalla:



En esta pantalla, que es básicamente el controlador que se encargara de generar carga sobre la web con usuarios virtuales, en la parte inferior izquierda tendremos que configurarlo de la siguiente manera:

The screenshot shows the 'Scenario Schedule' panel with the following configuration:

- Schedule Name:** Schedule 1
- Schedule by:** Scenario
- Run Mode:** Real-world schedule
- Global Schedule:** Total: 50 Users

Action	Properties
Initialize	Initialize each Vuser just before it runs
Start Vusers	Start 50 Vusers: 9 every 00:01:00 (HH:MM:SS)
Duration	Run for 00:05:00 (HH:MM:SS)
Stop Vusers	Stop all Vusers: 5 every 00:00:30 (HH:MM:SS)

Como podemos ver, el "Run Mode" se configura en la opción "Real-world schedule", esto significa que los usuarios, accederán al sistema de forma progresiva, y se irán también de forma progresiva, una prueba muy realista. En cambio, la opción "Basic Schedule" lo que hace es introducir a todos los usuarios virtuales de golpe y después sacarlos de golpe, algo que no podríamos considerar como una prueba de lo que en la realidad ocurrirá aunque sí que nos serviría para hacer pruebas de carga y ver cuántos usuarios simultáneos aguantaría el sistema.

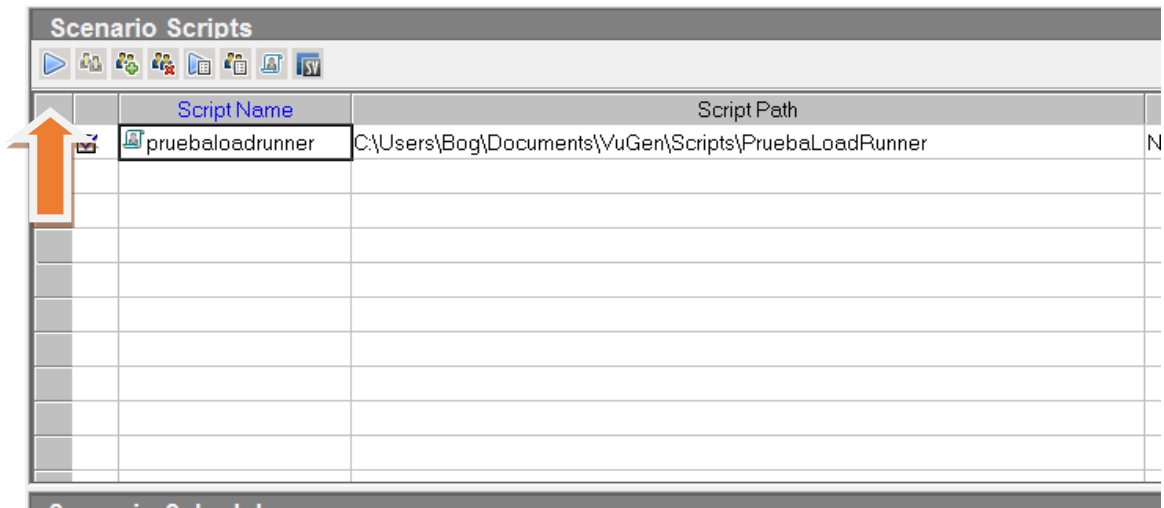
Para cambiar cualquier opción que está en la columna "Properties", solo tendremos que hacer doble click sobre ella y se nos abrirá un cuadro de dialogo donde podremos editarlo.

Como podemos observar, se inicializaran 50 usuarios progresivamente, 9 usuarios cada minuto. La prueba durara 5 minutos y se pararan 5 usuarios cada 30 segundos.

Estos parámetros han sido establecidos así, para hacer la prueba lo más parecida posible a la realizada con WebLoad.

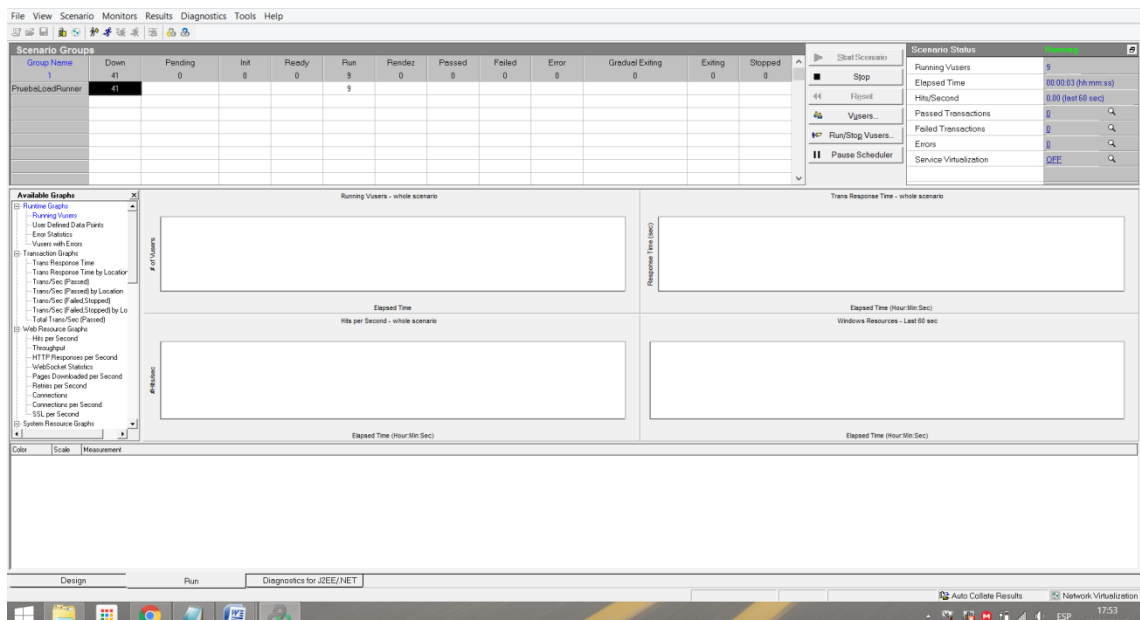
Una vez que tengamos configurado lo anterior, podemos proceder a realizar la prueba.

Lo único que tendremos que hacer es darle al botón del "Play" que tenemos en la esquina izquierda superior:



Una vez realizado lo anterior, tendremos que esperar a que los usuarios virtuales acaben.











Como veremos, se nos abrirá otra vista en la que veremos cómo se van ejecutando poco a poco los usuarios virtuales y si hay errores o no.



En esta pestaña, también podemos ver gráficos de las operaciones que están sucedientes en tiempo real con los usuarios virtuales. Además de las 4 gráficas que vienen por defecto, en el desplegable de la izquierda hay muchos más que podremos añadir.

Como podemos ver, la prueba en total ha tardado unos 15 minutos, 5 minutos para introducir los usuarios progresivamente, 5 minutos dando toda su potencia con los 50 usuarios simultáneos y por ultimo otros 5 minutos finalizando progresivamente cada usuario.

Este sería el resultado de la prueba sin entrar en el aspecto del análisis aun:

 <u>S</u> tart Scenario  Stop  Reset  Vusers...  Run/Stop Vusers...	Scenario Status	Down	
	Running Vusers	0	
	Elapsed Time	00:15:04 (hh:mm:ss)	
	Hits/Second	3.93 (last 60 sec)	
	Passed Transactions	251 	
	Failed Transactions	1002 	
	Errors	1002 	
	Service Virtualization	OFF 	

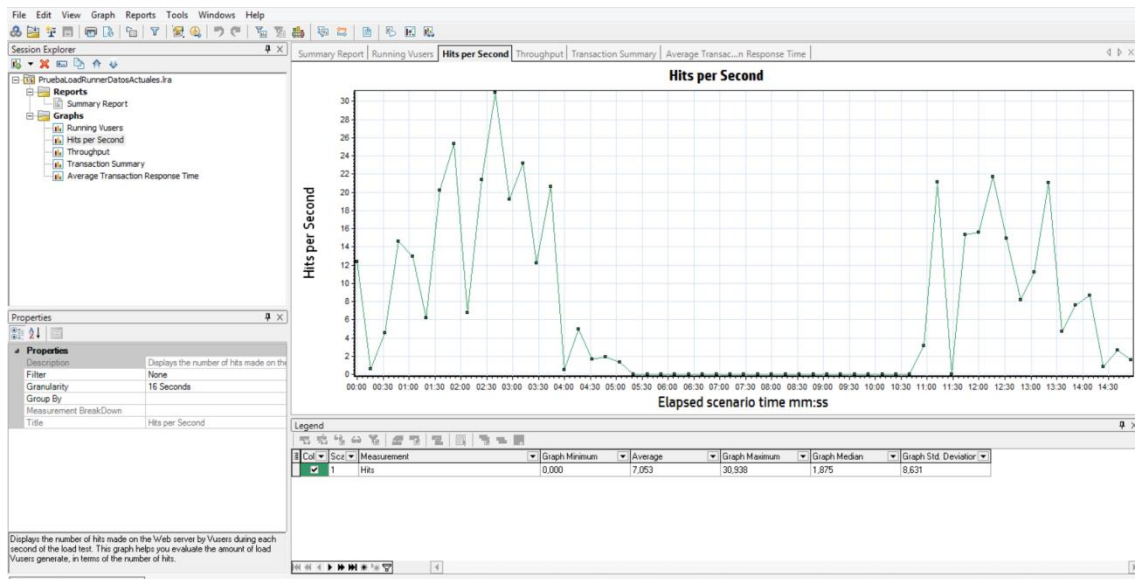
Como podemos ver, han fallado 1002 transacciones, esto es debido a que a partir de los 30 usuarios, el sistema no ha aguantado más, y no se han podido realizar más transacciones de los usuarios virtuales. Con 50 usuarios virtuales activos, todo han sido peticiones fallidas. Cuando se han ido sacando los usuarios, en los últimos 5 minutos, el sistema pudo recibir peticiones otra vez y se realizaron algunas mas de forma satisfactoria.

En este caso, podemos afirmar a groso modo, que el sistema, no soporta 50 usuarios simultáneos ni de lejos.

Por último, para analizar los datos generados en ese apartado, en la barra de tareas tendremos que seleccionar "Results>Analyze results". Esto, nos abrirá el tercer programa llamado Analysis que veremos a continuación.

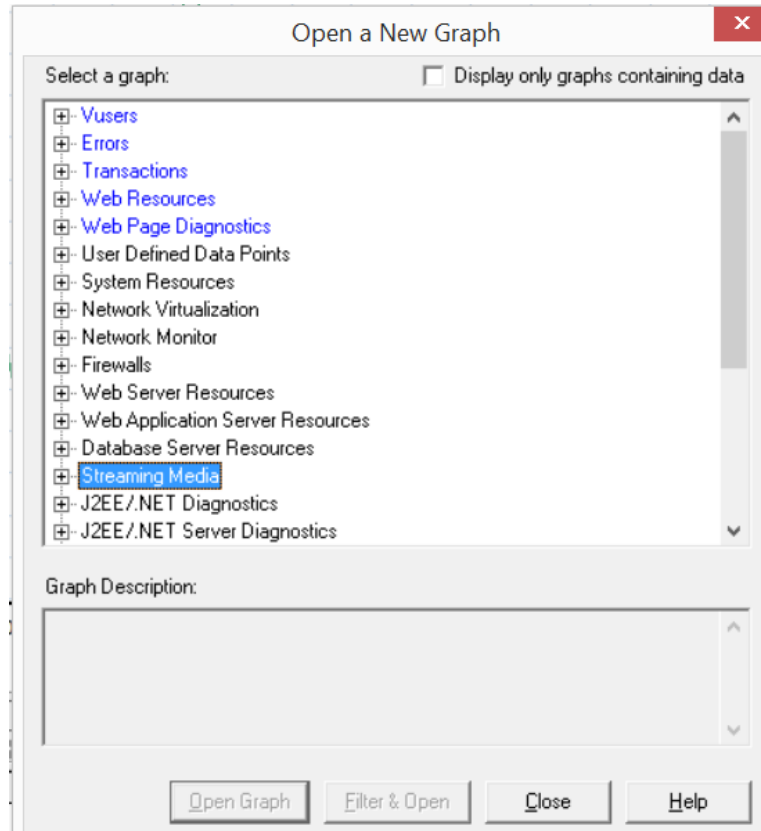
5.4 Documentación del análisis de resultados

En este apartado, solo queda visualizar los datos obtenidos del testeo realizado:



Por defecto, solo nos aparecerá un resumen general y 5 gráficas, como se puede observar en la imagen superior, aunque, podemos añadir de forma adicional mas gráficas:

En la barra de herramientas superior, pulsaremos en "Graph>Add New Graph..." y nos mostrara lo siguiente:



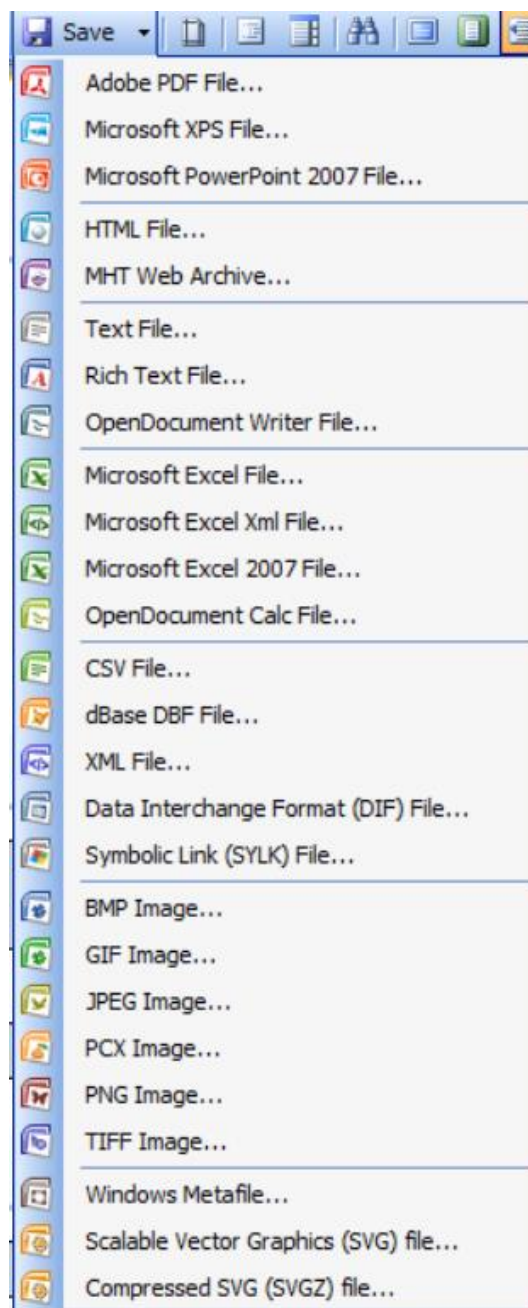
Cada una de las opciones mostradas arriba, es un desplegable con muchísimos gráficos dentro de cada una, es probable que este marcada la opción "Display only graphs containing data" con lo cual, no aparecerán todos los gráficos. Recordemos que al principio, cuando diseñamos la prueba para los usuarios virtuales, seleccionamos el protocolo que queríamos testear, en nuestro caso se trataba de HTTP/HTML simple, con lo cual, la mayoría de las gráficas que podemos ver en la imagen superior, no tendrían datos para este protocolo. Solo los desplegables marcados en azul tienen datos resultantes de nuestro testeo.

Aun mencionado lo anterior, tenemos muchas gráficas disponibles, entre las que se encuentran las siguientes:

- Vusers ejecutandose
- Resumen de los Vusers
- Estadísticas de errores
- Errores por segundo
- Errores totales por segundo
- Promedio del tiempo de respuesta de la operación.
- Operaciones por segundo
- Operaciones totales por segundo
- Resumen de las operaciones
- Resumen del rendimiento de la operación
- Tiempo de respuesta de la operación bajo carga
- Tiempo de respuesta de la operación por ubicación.
- Peticiones por segundo
- Rendimiento
- Conexiones
- Conexiones por segundo
- Diagnóstico de la página Web

Este reporte puede ser exportado en formato HTML o en formato de reporte normal. El formato HTML es básicamente como una web que podemos ver de forma local con todos los gráficos que hayamos añadido.

El reporte normal puede tener cualquiera de las siguientes extensiones:



En el anexo de este documento, se añade también el reporte exportado en formato PDF para su posterior revisión.

6 Comparación de las dos implementaciones

A continuación, a lo largo de este apartado, se van a realizar unas tablas esquemáticas de cada una de las tecnologías analizadas. En cada una de ellas se evaluarán cada uno de los criterios que se han definido y especificado en el apartado 3.

6.1 Evaluación de los criterios en la implementación usando WebLoad

CRITERIO	EVALUACIÓN
Criterio 1	9
Criterio 2	Solo está disponible en Inglés
Criterio 3	8
Criterio 4	6
Criterio 5	6
Criterio 6	Alto
Criterio 7	9
Criterio 8	9
Criterio 9	No requiere de ningún requisito previo. Además está disponible para los sistemas operativos de Windows o Linux.
Criterio 10	8
Criterio 11	10 minutos

6.2 Evaluación de los criterios en la implementación usando LoadRunner

CRITERIO	EVALUACIÓN
Criterio 1	10
Criterio 2	Inglés (Aunque cuenta con paquete de idiomas para traducirlo en Francés, Alemán, Italiano, Coreano, Japonés, Ruso y Español.)
Criterio 3	8
Criterio 4	7
Criterio 5	8
Criterio 6	Alto
Criterio 7	5
Criterio 8	10
Criterio 9	Ningún requisito previo.
Criterio 10	10
Criterio 11	60 minutos

7 Comparación de la implementación de las tecnologías

CRITERIOS	WebLoad	LoadRunner	COMENTARIOS
Criterio 1	9	10	
Criterio 2	Inglés	Inglés	LoadRunner dispone de un paquete de idiomas en francés, alemán, japonés, coreano, ruso, italiano y español.
Criterio 3	8	8	Ambos son fácil de usar
Criterio 4	6	7	
Criterio 5	6	8	
Criterio 6	Alto	Alto	Ambos consumen muchos recursos de la maquina en la que se está realizando las pruebas.
Criterio 7	9	5	
Criterio 8	9	10	
Criterio 9	-----	-----	Ninguno requiere de requisitos previos a la instalación.
Criterio 10	8	10	
Criterio 11	10 minutos	60 minutos	

8 Conclusiones

En cuanto a las conclusiones que se pueden sacar después de haber testeado las dos herramientas y hacer una pequeña evaluación en los apartados anteriores, hemos podido llegar a la conclusión de que ambas herramientas son de fácil instalación, pero sobre todo, algo muy importante, ninguna requiere de unos requisitos previos para poder iniciar la instalación para su posterior uso.

Sí que es cierto, que si miramos la tabla de comparativa de ambas, en el criterio 1, vemos que WebLoad frente a LoadRunner, es más fácil de instalarla, aunque apenas hay diferencia, ya que en una escala del 1 al 10, la primera tiene un 9 y la segunda un 10, por lo que se puede decir que en cuanto a este criterio son prácticamente iguales.

Ambas están diseñadas en inglés, por lo que la aplicación se encuentra en inglés, lo único, que LoadRunner, presenta una ventaja frente a WebLoad, y es que permite descargarte e instalar desde su página web oficial, un paquete de idiomas que te permite la opción de poder trabajar con la herramienta en 8 idiomas, que son el francés, inglés, ruso, japonés, coreano, italiano, alemán y español. Por lo que para nosotros nos ha resultado una gran ventaja a la hora de trabajar con LoadRunner, ya que nos ha permitido poder trabajar en nuestra lengua natal.

Una de las desventajas que nos presentan las dos herramientas de pruebas, es el elevado consumo de los recursos de nuestra maquina cuando se inicia una prueba con sus respectivos análisis.

En cuanto al sistema de ayudas y tutoriales, hemos llegado a la conclusión de que WebLoad está algo más floja en este aspecto, ya que tiene un valor de 8 en una escala del 1 al 10 y LoadRunner en cambio un 10. Nuestra propia conclusión al

respecto sobre esta diferencia de valores y porque una tiene menos ayudas y tutoriales y nos ha costado más encontrar información que nos pudiera ser útil para poder realizar un prueba con éxito, se debe a que por ejemplo WebLoad es una herramienta que no está asociada a ninguna empresa o arca, en cambio LoadRunner, es una herramienta de HP, por lo que cuenta con más documentación e información útil para poder llevar a cabo una prueba de forma óptima.

En cuanto al criterio de facilidad de uso, ambas han obtenido un valor exactamente igual, un 8. Por lo que podemos decir que son más bien fáciles de usar. En cambio, en cuanto al diseño de interfaz e intuitividad, LoadRunner destaca frente a WebLoad, aunque no por mucha diferencia, pero sí que es cierto que parece tener una interfaz más amigable LoadRunner.

Por último, y más importante en cuanto a las pruebas de análisis, los parámetros a analizar son más en WebLoad frente a LoadRunner, ya que esta última se queda muy justa en cuanto al estudio que hace de los parámetros para obtener la información generada a través de los informes. WebLoad ha obtenido una puntuación de 9, mientras que LoadRunner un 5, por lo que queda patente que en este sentido, se queda muy justa esta herramienta y si se quiere hacer un examen más exhaustivo y meticuloso de nuestra página web, no nos bastaría con analizar con dicha herramienta. En cambio, en cuanto a la generación de informes que nos generan ambas es excelente, ya que LoadRunner ha obtenido una valoración de 10 puntos y WebLoad 9 puntos. Y en cuanto a los tiempos de duración de la realización de una prueba completa, sí que existe una gran diferencia entre ambas. Siendo WebLoad mucho más rápida con un tiempo total de 10 minutos, frente a LoadRunner con 60 minutos, es decir una hora.

Tras esta larga y exhaustiva comparativa, bajo nuestra experiencia, y tras haber trabajado del mismo modo con ambas, nosotros si tuviéramos que elegir alguna de ellas, nos quedaríamos con WebLoad por dos grandes motivos principalmente:

1. La cantidad de parámetros a analizar son muchísimo más elevados que la otra herramienta, permitiendo de este modo generar un análisis y unos informes más exactos, reales y fiables sobre el estudio que estamos realizando sobre nuestra página.
2. La velocidad en la que es capaz de realizar un análisis completo. Es mucho más rápida y eficiente que la otra herramienta.

A pesar de no tener un paquete de idiomas, de disponer de menos tutoriales y ayudas para comprender el buen funcionamiento del programa y ser un poco menos intuitiva o tener una interfaz un poco menos amigable, creemos que los dos criterios citados anteriormente pesan más que estos pequeños detalles.