



Eusko Jaurlaritzaren Informatika Elkarte
Sociedad Informática del Gobierno Vasco

Procedimiento de evaluación de indicadores en Desarrollo

Fecha: 23/02/2017

Referencia:

EJIE S.A.
Mediterráneo, 14
01010 Vitoria-Gasteiz
Posta-kutxatila / Apartado: 809
01080 Vitoria-Gasteiz
Tel. 945 01 73 00*
Fax. 945 01 73 01
www.ejie.es

Este documento es propiedad de EJIE, S.A. y su contenido es confidencial. Este documento no puede ser reproducido, en su totalidad o parcialmente, ni mostrado a otros, ni utilizado para otros propósitos que los que han originado su entrega, sin el previo permiso escrito de EJIE, S.A.. En el caso de ser entregado en virtud de un contrato, su utilización estará limitada a lo expresamente autorizado en dicho contrato. EJIE, S.A. no podrá ser considerada responsable de eventuales errores u omisiones en la edición del documento.

Control de documentación

Título de documento: OTC_METD. Procedimiento de evaluación de indicadores en Desarrollo.doc

Histórico de versiones

Versión: 2.0

Eliminado como anexo las “metrics definitions” de Sonar, se mantiene solo como referencia. Añadida referencia al manual de tareas ant para generar algunos informes de rendimiento en desarrollo. Añadido cómo poder registrar en el Portal SQA los resultados de rendimiento del entorno de Desarrollo. Se añade HP Diagnostics para la obtención de Indicadores. Se suprime awrreport.

Fecha: 09/02/2012

Versión: 2.1

Fecha: 09/02/2012

Cambios producidos desde la última versión

Se define y revisa en SonarQube el perfil de calidad “OTCEjje” que corresponde con el punto 2 de este documento.

Control de difusión

Responsable:

Aprobado por:

Firma:

Fecha:

Distribución:

Referencias de archivo

Autor: Consultoría de Areas de Conocimiento

Nombre archivo: OTC_METD_vx.y Procedimiento de evaluación de indicadores en Desarrollo.doc

Localización:

Contenido

	Capítulo/sección	Página
1	Introducción	4
1.1	Destinatarios del documento	4
1.2	Estándares de desarrollo y de calidad	4
1.3	Objetivo del documento	4
1.4	Referencias	5
2	Indicadores de calidad análisis estático de código	5
2.1	Indicadores de calidad para código Java	6
2.2	Características de calidad ISO	6
3	Pruebas basadas en requisitos (pruebas definidas en Testlink)	7
3.1	Pruebas de prestaciones	8
3.1.1.	Cumplimiento global indicadores de prestaciones	8
3.1.2.	Indicadores de Servidor Web para evaluación (EVAL-ISW)	9
3.1.3.	Indicadores de Servidor de Aplicaciones para evaluación (EVAL-ISA)	9
3.1.4.	Indicadores de BBDD para evaluación (EVAL-IBD):	10
3.1.5.	Ejecución pruebas prestaciones en Desarrollo	10
3.2	Seguridad : Grado de cumplimiento del owasp top 10 de seguridad	23
3.3	Usabilidad: Grado de cumplimiento evaluación heurística de usabilidad	23
3.4	Accesibilidad	23

1 Introducción

1.1 Destinatarios del documento

Roles Probamet y SQA	Destinatarios
Equipo de Desarrollo y Pruebas	X
OTC	
OTC-EJIE	X
Analista Responsable en EJIE	
Oficina de Evaluación	

1.2 Estándares de desarrollo y de calidad

EJIE establece ArinBide como la metodología de desarrollo de sistemas software y adoptó la norma ISO 9001:2000 como modelo de especificación de requisitos para un sistema de gestión de la calidad. Del mismo modo se han homologado ya las herramientas de desarrollo asociadas al ciclo de vida de los sistemas software que permiten acelerar y sistematizar todo el proceso.

Por otra parte, EJIE también dispone del conjunto de estándares de calidad del producto software que permitirán asegurar con cierta garantía que los aplicativos implantados en el entorno productivo son lo suficientemente robustos y estables para su explotación. Dichos estándares vienen definidos por el modelo de Aseguramiento de la calidad software y la Metodología de Pruebas Probamet:

- El Aseguramiento de la Calidad del Software (Modelo SQA) es un marco para la evaluación de la calidad de los productos generados durante cada fase del ciclo de vida del desarrollo de software y las actividades de pruebas.
- Probamet establece las actividades del proceso de pruebas de desarrollo del software, definiendo cada una de las fases que forman el ciclo de pruebas, las tareas y los entregables de las mismas.

Dentro de estos estándares de calidad, se encuentran el conjunto de indicadores que permiten evaluar la calidad de un proyecto de desarrollo.

En función del Nivel de Aseguramiento de la Calidad (NAC) asignado a un proyecto, se exigirán unos niveles mínimos de calidad para dichos indicadores, la lista de indicadores así como los umbrales de superación de los mismos se encuentran en el documento .

1.3 Objetivo del documento

Es objeto del presente documento:

- Detallar el procedimiento de cálculo y evaluación de los indicadores de calidad software requeridos por EJIE.

Los detalles de las pruebas de prestaciones para Desarrollo se han actualizado a partir del antiguo documento de "Indicadores NAC Desarrollo".

1.4 Referencias

Documento	Código documento	Ubicación
Modelo SQA y Probamet		
[1] Indicadores de calidad software	OTC_MET	\Repositorio base\ModeloSQA\ Indicadores\ OTC_MET_v0.1 Indicadores de calidad software
[2] Metric definitions Sonar	-	http://docs.codehaus.org/display/SONAR/Metric+definitions
[3] Detalle características y atributos de calidad de código estático	OTC_ISO	\Repositorio base\ModeloSQA\ Indicadores\Anexos_indicadores\OTC_ISO_v1.0 Detalle características y atributos de calidad de código estático.doc
[4] Pruebas Prestaciones y Capacidad - Informacion Procesos de Negocio a Probar	OTC_PPN	\Repositorio base\PROBAMET\Plantillas\APB\OTC_PPN_v1.0 Pruebas Prestaciones y Capacidad - Informacion Procesos de Negocio a Probar
Manuales de herramientas de calidad opcionales en Desarrollo		
[5] Manual de usuario Jmeter	-	Consultoría de áreas de conocimiento > Espacio compartido > Herramientas de desarrollo > Manuales de usuario > Jmeter.Manual de usuario v1.2
[6] Manual de usuario Badboy	-	Consultoría de áreas de conocimiento > Espacio compartido > Herramientas de desarrollo > Manuales de usuario > Badboy.Manual de usuario v1.2.doc
[7] Manual de usuario TAW	-	Consultoría de áreas de conocimiento > Espacio compartido > Herramientas de desarrollo > Manuales de usuario > TAW.Manual de usuario v1.0

2 Indicadores de calidad análisis estático de código

Los indicadores de análisis de código se obtienen en Sonar a partir de la reglas definidas por el grupo de Consultoría de Áreas de Conocimiento. Estas reglas se pueden consultar en el propio sonar, están recogidas en el Quality Profile "OTCEjje".

2.1 Indicadores de calidad para código Java

Número de reglas incumplidas

Descripción	Miden la calidad del código fuente de la aplicación.		
Variable Objeto a medir	Descripción	Cálculo	Herramienta origen
Errores prioridad 1 de calidad de código.	Suma de incumplimientos de prioridad "Blocker" de Quality Profile "OTCEjje"	Suma de incumplimientos de prioridad "Blocker" de Quality Profile "OTCEjje" / Líneas de código *100	Sonar Dashboard
Errores prioridad 2 de calidad de código.	Suma de incumplimientos de prioridad "Major" de Quality Profile "OTCEjje"	Suma de incumplimientos de prioridad "Major" de Quality Profile "OTCEjje" / Líneas de código *100	Sonar Dashboard
Errores prioridad 3 de calidad de código.	Suma de incumplimientos de prioridad "Info" de Quality Profile "OTCEjje"	Suma de incumplimientos de prioridad "Info" de Quality Profile "OTCEjje" / Líneas de código *100	Sonar Dashboard
Código duplicado	Porcentaje de líneas de código duplicado obtenido por CPD frente a líneas físicas de código	Líneas de código afectadas por duplicados / líneas físicas *100	Sonar Dashboard

Donde:

Líneas de código (**lines of code**) = Líneas físicas – líneas en blanco – líneas comentadas.

El token para los duplicados está establecido en 200, según configuración EJIE (por defecto en CPD es 100)

2.2 Características de calidad ISO

Los atributos de calidad y las reglas asignadas se detallan en el documento "Detalle características y atributos de calidad de código estático" [3] (ver referencias)

Descripción	Miden los atributos de calidad del código fuente de la aplicación como porcentaje, restando del 100% los incumplimientos ponderados según la prioridad y el tamaño del código.		
Atributo de calidad	Descripción	Cálculo	Herramienta origen

Índice de Calidad Estático	Porcentaje ponderado de cumplimiento de todas las reglas de estático activas (independientemente de a qué atributos de calidad estén asignadas)	$100 - \{\text{incumplimientos ponderados} / \text{Líneas de código}\} * 100$ Es decir este índice NO es una media de los índices de las categorías mantenibilidad, eficiencia, etc, sino se calcula directamente a través de los incumplimientos.	Sonar Dashboard
Mantenibilidad (capacidad del producto software para ser modificado)	Porcentaje ponderado de cumplimiento de las reglas que afectan a la mantenibilidad.	$100 - \{\text{incumplimientos ponderados de las reglas de mantenibilidad} / \text{Líneas de código}\} * 100$	Sonar Dashboard
Eficiencia (capacidad del producto software para proporcionar el rendimiento apropiado, relativo a la cantidad de recursos utilizados, bajo las especificaciones determinadas)	Porcentaje ponderado de cumplimiento de las reglas que afectan a la eficiencia.	$100 - \{\text{incumplimientos ponderados de las reglas de eficiencia} / \text{Líneas de código}\} * 100$	Sonar Dashboard
Portabilidad (capacidad del producto software de ser transferido de un entorno a otro)	Porcentaje ponderado de cumplimiento de las reglas que afectan a la portabilidad.	$100 - \{\text{incumplimientos ponderados de las reglas de portabilidad} / \text{Líneas de código}\} * 100$	Sonar Dashboard
Fiabilidad (capacidad del producto software para mantener un nivel especificado de rendimiento cuando se utiliza en las condiciones especificadas)	Porcentaje ponderado de cumplimiento de las reglas que afectan a la fiabilidad.	$100 - \{\text{incumplimientos ponderados de las reglas de fiabilidad} / \text{Líneas de código}\} * 100$	Sonar Dashboard
Usabilidad (capacidad del producto software de ser entendido, aprendido, utilizado y atractivo al usuario, cuando se utiliza en las condiciones especificadas. Depende del tipo de "uso" que se espera y tipo de "usuario" que utilizará el producto)	Porcentaje ponderado de cumplimiento de las reglas que afectan a la usabilidad.	$100 - \{\text{incumplimientos ponderados de las reglas de usabilidad} / \text{Líneas de código}\} * 100$	Sonar Dashboard

Donde los incumplimientos ponderados (**Weighted violations**) se calculan como:

Suma (incumplimiento 1*peso incumplimiento1+ incumplimiento 2*peso incumplimiento2 +)

3 Pruebas basadas en requisitos (pruebas definidas en Testlink)

Los indicadores que se miden son el **porcentaje de pruebas ejecutadas** y las **incidencias abiertas** en Mantis, lo que da la medida de fallos pendientes de resolver (puesto que cada prueba fallada genera automáticamente una incidencia, si no hay incidencias abiertas, no habrá casos de prueba fallados).

Descripción	Calidad de pruebas unitarias, integración y sistemas. Calidad de los componentes independientes de la aplicación.		
Variable Objeto a medir	Descripción	Cálculo	Herramienta origen
Porcentaje de Pruebas completadas	Número de casos de prueba ejecutadas frente al número de casos de prueba especificados (Testlink)	Número de casos de prueba ejecutadas frente al número de casos de prueba especificados (Testlink)	Testlink
Incidencias abiertas en Mantis de severidad bloqueo o mayor	Suma del número de incidencias no-cerradas de severidad bloqueo y de severidad mayor en Mantis	Suma del número de incidencias no-cerradas de severidad bloqueo y de severidad mayor en Mantis	Mantis
Incidencias abiertas en Mantis de severidad normal	Suma del número de incidencias no-cerradas de severidad normal en Mantis	Suma del número de incidencias no-cerradas de severidad normal en Mantis	Mantis
Incidencias abiertas en Mantis de severidad menor	Suma del número de incidencias no-cerradas de de severidad menor en Mantis	Suma del número de incidencias no-cerradas de de severidad menor en Mantis	Mantis

Para las pruebas unitarias se muestra además, a modo informativo en el Sonar Dashboard, la cobertura de código (**coverage**) conseguida con las pruebas unitarias.

3.1 Pruebas de prestaciones

Nota:

las pruebas de prestaciones para la evaluación de aplicaciones se ejecutan en el entorno de Pruebas.

Se mencionan aquí los indicadores de estas pruebas solo a título informativo, para que en Desarrollo se tenga conocimiento de los criterios sobre las que serán evaluadas.

Es **recomendable** que en Desarrollo se vigilen estas métricas, y que se realicen algunas pruebas de rendimiento en Desarrollo con herramientas como Jmeter + HP Diagnostics (pero no se exigen). Los resultados de estas pruebas no serán idénticos a los del entorno de Pruebas, ni garantizan que se vayan a superar, pero permitirán detectar en fases tempranas posibles problemas o cuellos de botella en el código.

Opcionalmente se podrán registrar los valores obtenidos mediante el Portal SQA, seleccionando el *proyecto para el entorno de Desarrollo* y cumplimentando la "Checklist de Rendimiento" del menú "Pruebas". Si se hace esto, siempre habrá que tener mucho cuidado de que se haya escogido el entorno de desarrollo, cuyos datos son sólo informativos para el equipo de desarrollo y no se tienen en cuenta para la evaluación de la aplicación para el paso a producción.

3.1.1. Cumplimiento global indicadores de prestaciones

Como indicador global de las pruebas de prestaciones se mide si se cumplen o no todos los indicadores definidos – para el entorno de Pruebas.

3.1.2. Indicadores de Servidor Web para evaluación (EVAL-ISW)

Descripción	Miden el rendimiento de la aplicación con respecto a un servidor web.		
Variable Objeto a medir	Descripción	Cálculo	Herramienta origen
Tiempo de resolución de una petición	Entendiendo este, como el tiempo comprendido desde que un usuario realiza una petición al servidor hasta que recibe la respuesta	tiempo comprendido desde que un usuario realiza una petición al servidor hasta que recibe la respuesta	- Jmeter - HP Diagnostics -> Pestaña Server Request - Otras rendimiento
Número de peticiones centralizadas correctas	En % indica el número de peticiones al servidor resueltas correctamente	$100 - \{ \text{Número de peticiones que han dado respuestas 4xx} + \text{Número de peticiones que han dado respuestas 5xx} / \text{Número total de peticiones} \} * 100$	- Jmeter - Otras rendimiento
Peso de una página	Se obtiene el tamaño de una página web, incluyendo también los archivos estáticos como imágenes, css o js.	tamaño de una página web, incluyendo también los archivos estáticos como imágenes, css o js	- Jmeter - Otras rendimiento

3.1.3. Indicadores de Servidor de Aplicaciones para evaluación (EVAL-ISA)

Descripción	Miden el rendimiento de la aplicación con respecto a un servidor de aplicaciones.		
Variable Objeto a medir	Descripción	Cálculo	Herramienta origen
Conexiones a base de datos no cerradas	Conexiones a base de datos no cerradas	Comprobar que no hayan quedado conexiones de BBDD sin cerrar en la herramienta	- HP Diagnostics -> Pestaña Metrics - Otras rendimiento
Porcentaje de memoria libre en la maquina virtual	Porcentaje de memoria libre en la maquina virtual	Comprobar este porcentaje en la herramienta	- HP Diagnostics -> Pestaña Metrics - Otras rendimiento
Número de hilos libres de ejecución	Número de hilos libres de ejecución	Comprobar el número de hilos libres en la herramienta	- HP Diagnostics-> Pestaña Metrics - Otras rendimiento
Porcentaje de CPU utilizada en la maquina.	Porcentaje de CPU utilizada en la maquina.	Comprobar este porcentaje en la herramienta	- HP Diagnostics-> Pestaña Metrics - Otras rendimiento

3.1.4. Indicadores de BBDD para evaluación (EVAL-IBD):

Descripción	Miden el rendimiento de la aplicación en base de datos.		
Variable Objeto a medir	Descripción	Cálculo	Herramienta
Tiempo medio de ejecución SQL	Se calcula el tiempo total en segundos que tarda una consulta desde que comienza su análisis y pasando por su compilación y optimización de consulta.	Comprobar el tiempo medio en la herramienta	- HP Diagnostics-> Pestaña All SQL
Número de objetos de BBDD de la aplicación inválidos	Resultado de la ejecución de la consulta <code>Select count (1) from all_objects where status='INVALID';</code>	Resultado de la ejecución de la consulta Select count (1) from all_objects where status='INVALID';	Oracle
Fragmentación de los índices de la bbdd	Número de índices con más de 4 niveles. Es el resultado de la ejecución de la consulta <code>Select count (1) from all_indexes where blevel>4;</code>	Número de índices con más de 4 niveles. Es el resultado de la ejecución de la consulta Select count (1) from all_indexes where blevel>4;	Oracle

3.1.5. Ejecución pruebas prestaciones en Desarrollo

En este apartado se definen los pasos a seguir para realizar pruebas de prestaciones en Desarrollo mediante Jmeter y BadBoy y las indicaciones oportunas para el análisis de los resultados obtenidos.

Paso 1: Definición de escenarios de prueba y generación de scripts.

Paso 2: Ejecución y obtención de resultados.

Paso 3: Consulta del resultado para su análisis.

Paso 1: Definición de escenarios de prueba y generación de scripts.

Requisitos previos o precondiciones:

Casos de prueba documentados en Testlink. Los procesos de negocio a probar son los mismos en Desarrollo y en Pruebas, y se habrán documentado en Testlink mediante la plantilla Pruebas Prestaciones y Capacidad - Informacion Procesos de Negocio a Probar [4]

Tener instalada la herramienta BadBoy.

Herramientas:

BadBoy 2.0.

Procedimiento:

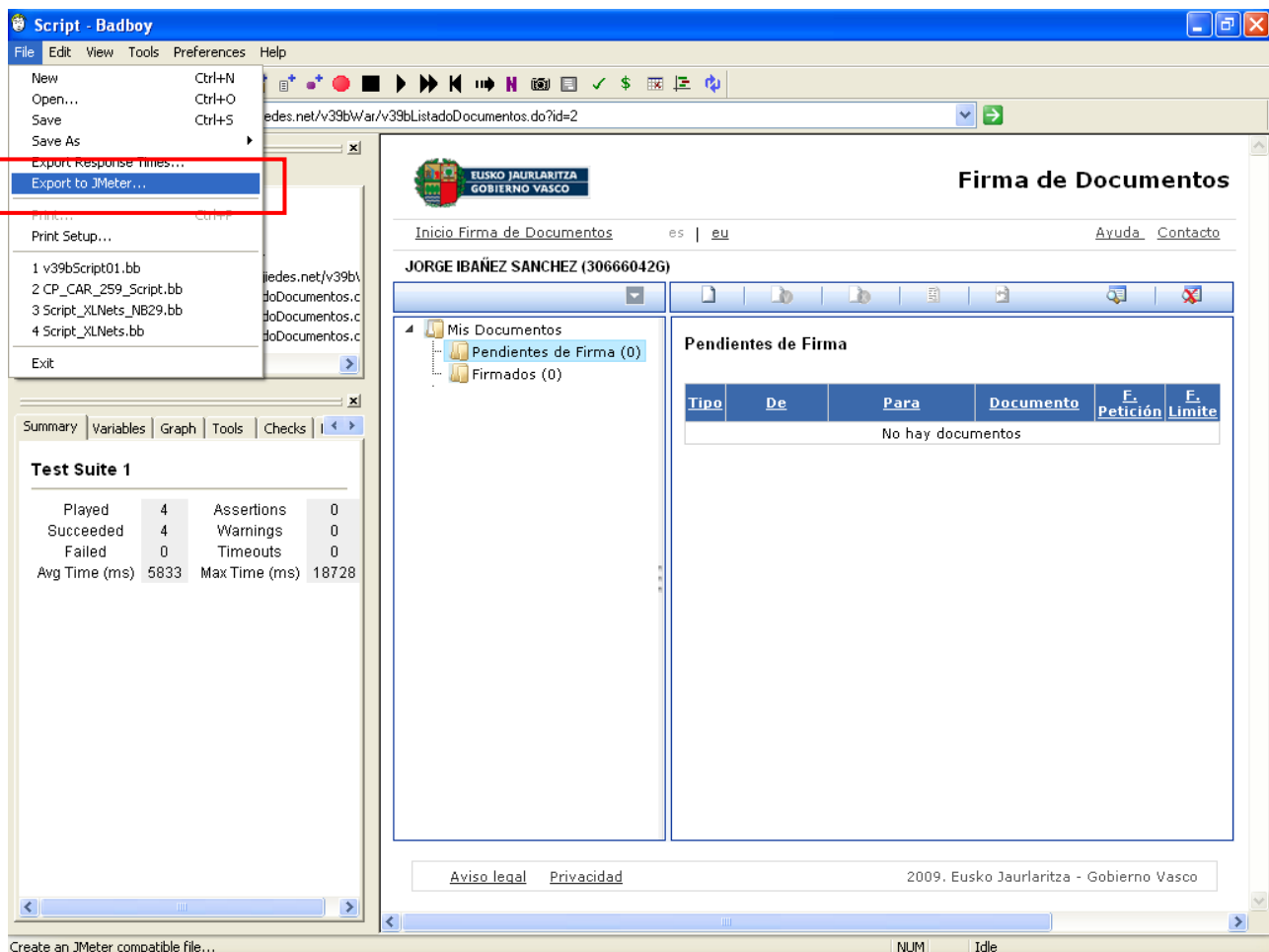
El primer paso antes de realizar las pruebas, es definir los escenarios que se van a contemplar en las mismas, para esto analizaremos la aplicación buscando los casos de uso más habituales, o los más problemáticos, ya sea por el número de usuarios concurrentes o por la complejidad de computación que requieran.

Una vez definidos estos escenarios, se crearán los correspondientes “scripts” que los simulan con la herramienta BadBoy.

Ejecutamos el BadBoy. Inicialmente se ejecuta con el modo de grabación activado. Si nuestra aplicación usa la autenticación de usuario de XINets, procederemos a desactivar el modo de grabación, antes de interactuar con nuestra aplicación.

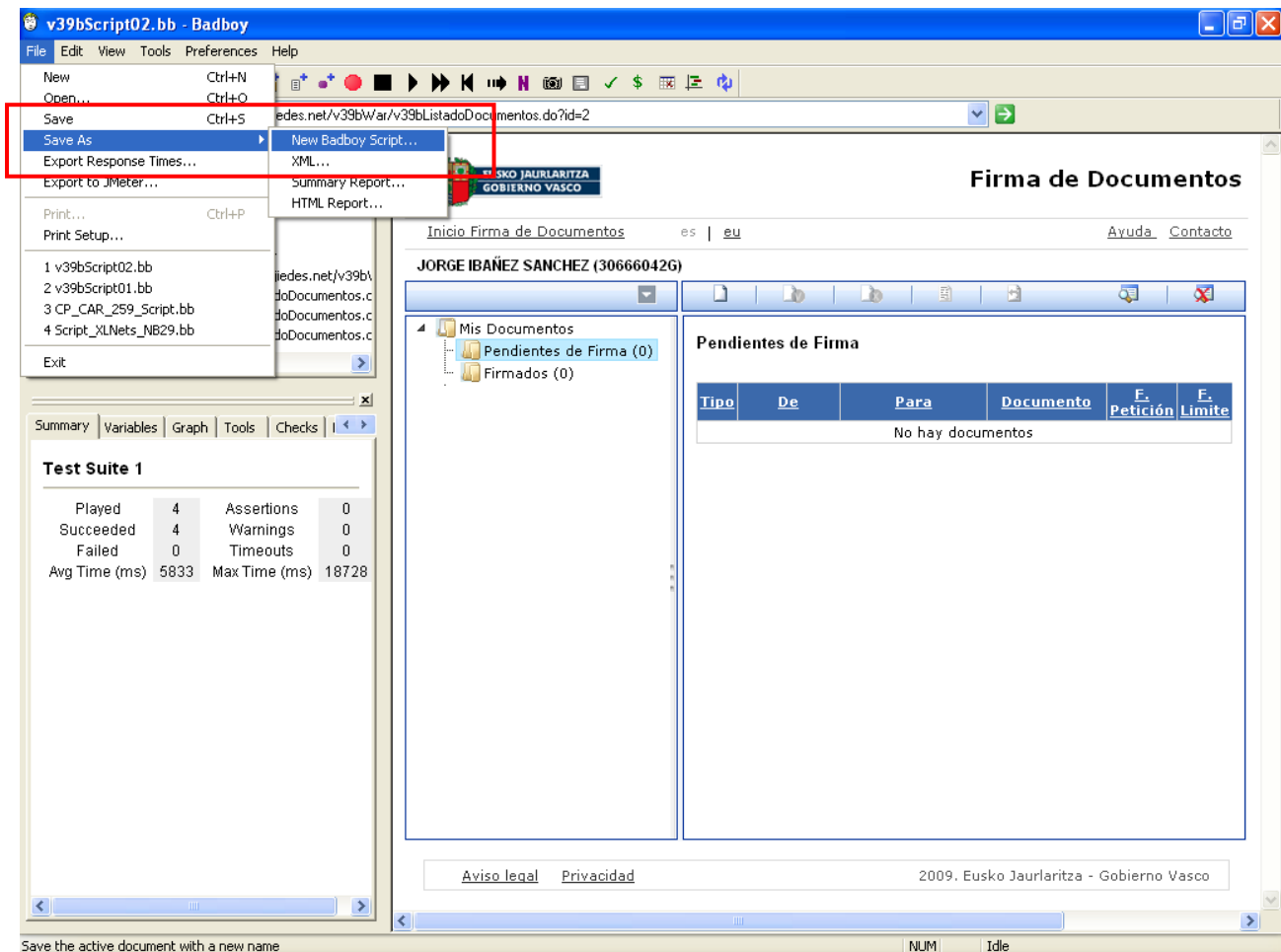
Una vez que nos hemos autenticado en XINets, y estamos en el punto de entrada de nuestra aplicación, procederemos a realizar la grabación, navegando por nuestra aplicación siguiendo los casos definidos.

Una vez finalizados los casos definidos, paramos la grabación y exportamos el script generado como script de JMeter. Para ello, desde Badboy, ir a File -> Export to JMeter:



Damos un nombre al script y aceptamos.

También podemos guardar el “script” con formato de BadBoy, por si más adelante quisiéramos añadir más navegación. Para ello, ir a File -> Save as -> New Badboy Script:



Por cada escenario de prueba de la aplicación, seguir el mismo procedimiento.

Como resultado, habremos generado todos los scripts correspondientes a todos los escenarios definidos.

Paso 2: Ejecución y obtención de resultados.

Requisitos previos o precondiciones:

Paso 1

Herramientas:

- JMeter 2.3.2
- HP Diagnostics

Procedimiento:

1. Configuración de la prueba

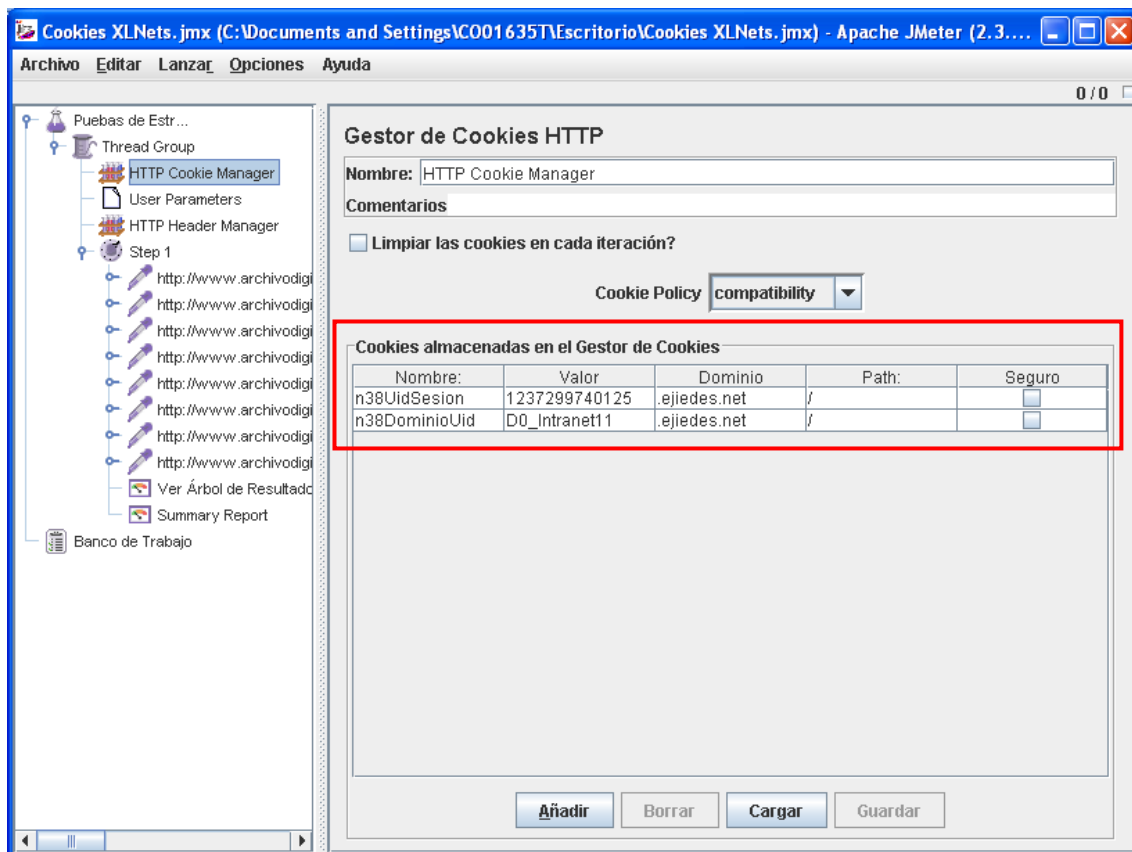
Arrancamos JMeter 2.3.2. Esta herramienta nos permitirá ejecutar las pruebas de prestaciones siguiendo los "scripts" de navegación que hemos creado con la herramienta "BadBoy".

La aplicación tiene dos entidades básicas. Por un lado tenemos el “Test plan” (o plan de pruebas) que representa una prueba y por el otro los “elements” (o elementos) que representan las diferentes etapas o acciones que usaremos para componer las pruebas.

Una vez abierto el “script” con JMeter, en el caso de que la aplicación utilizase autenticación XINets de usuario, se deberán fijar las cookies necesarias en el plan de pruebas. Para ello, nos autenticamos en XINets, con un usuario válido en la aplicación, abrimos (dobleclick) el candado de XINets y cogemos los datos necesarios para configurar nuestra navegación, el “UID de Sesión” y el “Dominio”.



Volvemos a JMeter, y en la opción HTTP Cookie Manager, le damos al botón añadir, para fijar las cookies:



En el caso del ejemplo anterior, la aplicación se encuentra en el siguiente dominio:

<http://www.archivodigital.jakina.ejiedes.net/.....>

Los valores a rellenar para cada cookie serán por tanto:

- **Nombre:** n38UidSesion
- **Valor:** 1237299740125 (obtenido del candado)
- **Nombre:** n38DominioUid
- **Valor:** D0_Intranet11 (obtenido del candado)

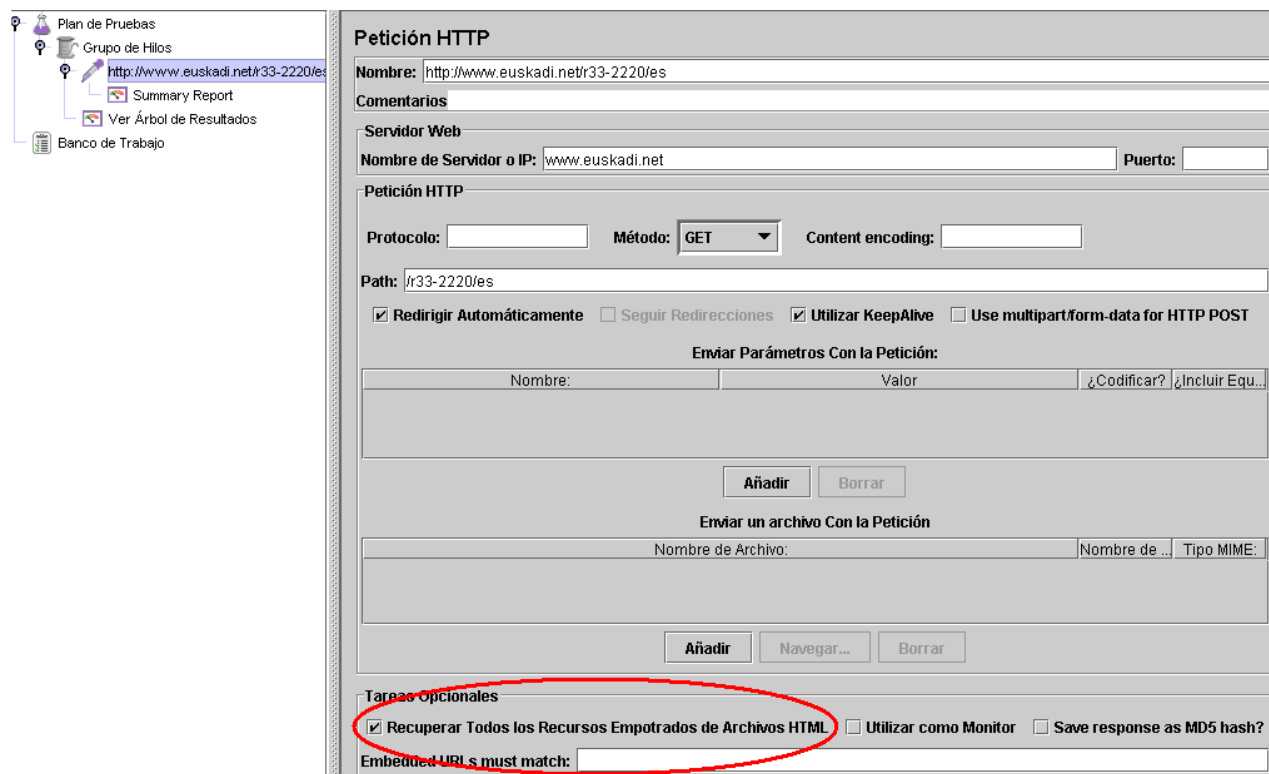
El resto de propiedades se rellenaran para las dos de la siguiente manera:

- **Dominio:** .ejiedes.net (depende del dominio de la aplicación, no olvidar el punto inicial)
- **Path:** "/" (siempre tendrá este valor)

El nombre de las cookies no variará nunca, lo que puede variar son los valores de las cookies "n38UidSesion" y "n38DominioUid", pero solo en el caso de que ejecutemos el plan de pruebas en sitios diferentes.

Habrà que comprobar también, que en **todas** las peticiones HTTP del script, esté marcada la casilla para que descargue los archivos estáticos incluidos en el HTML, como imágenes o ficheros .js entre otros. Sólo de esta manera se conseguirá que los datos obtenidos, como el peso de la página, sean lo más reales

posible. Seleccionar cada petición HTTP y marcar el check “Recuperar Todos los Recursos Empotrados de Archivos HTML”:



Petición HTTP

Nombre:

Comentarios:

Servidor Web

Nombre de Servidor o IP: Puerto:

Petición HTTP

Protocolo: Método: Content encoding:

Path:

☒ Redirigir Automáticamente ☐ Seguir Redirecciones ☒ Utilizar KeepAlive ☐ Use multipart/form-data for HTTP POST

Enviar Parámetros Con la Petición:

Nombre:	Valor	¿Codificar?	¿Incluir Equ...

Enviar un archivo Con la Petición

Nombre de Archivo:	Nombre de ...	Tipo MIME:

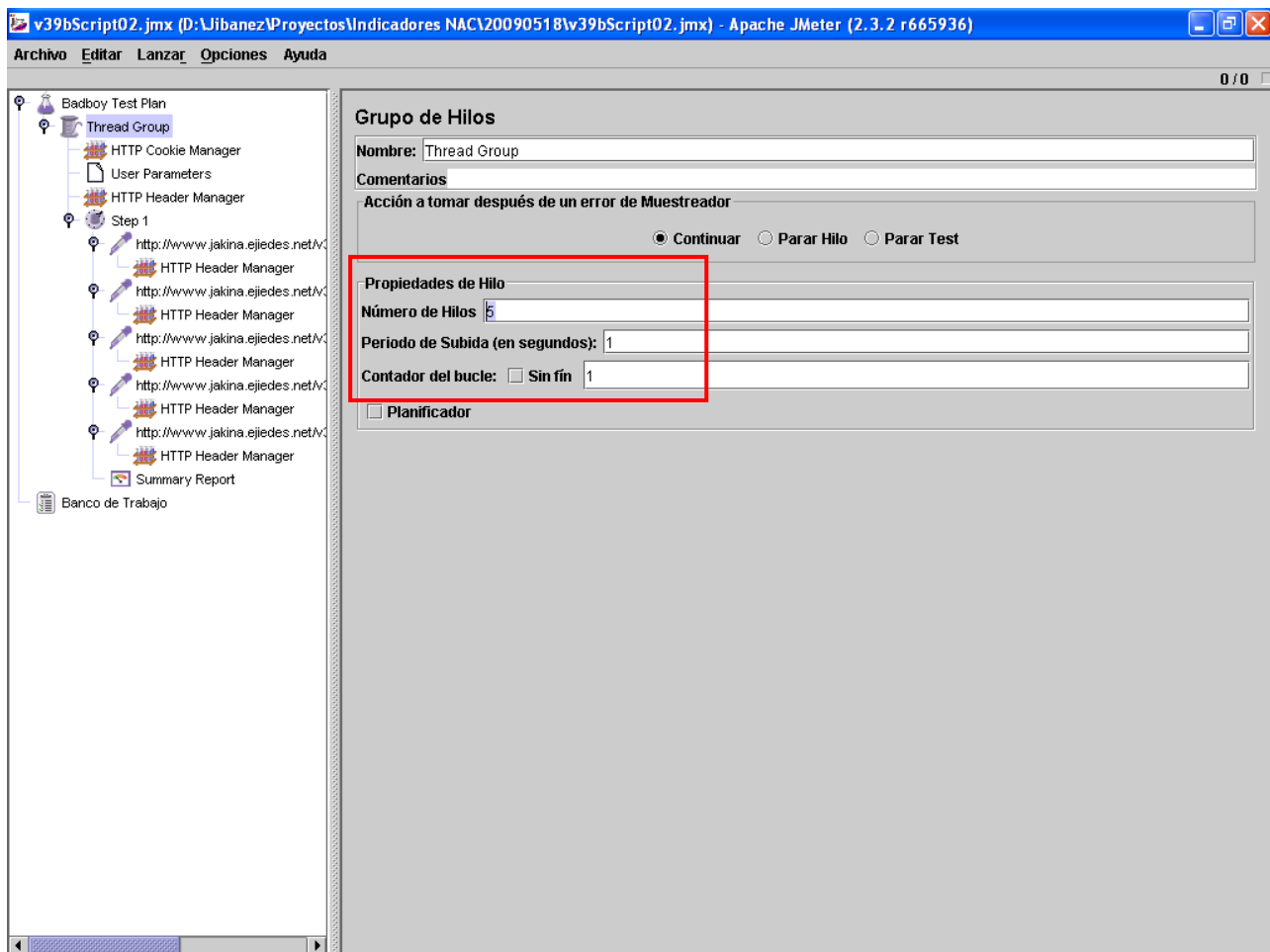
Tareas Opcionales

☒ **Recuperar Todos los Recursos Empotrados de Archivos HTML** ☐ Utilizar como Monitor ☐ Save response as MD5 hash?

Embedded URL's must match:

Realizados estos pasos se procederá a configurar el plan de pruebas con el número de usuarios concurrentes y ciclos definidos.

Para esto se modificará la configuración del Thread Group, indicando en el “Numero de Hilos”, el número de usuarios concurrentes en la aplicación y en el “Contador de bucle”, las veces que se quiere repetir de manera consecutiva el plan de pruebas con ese número de usuarios.



2. Ejecución y obtención de resultados

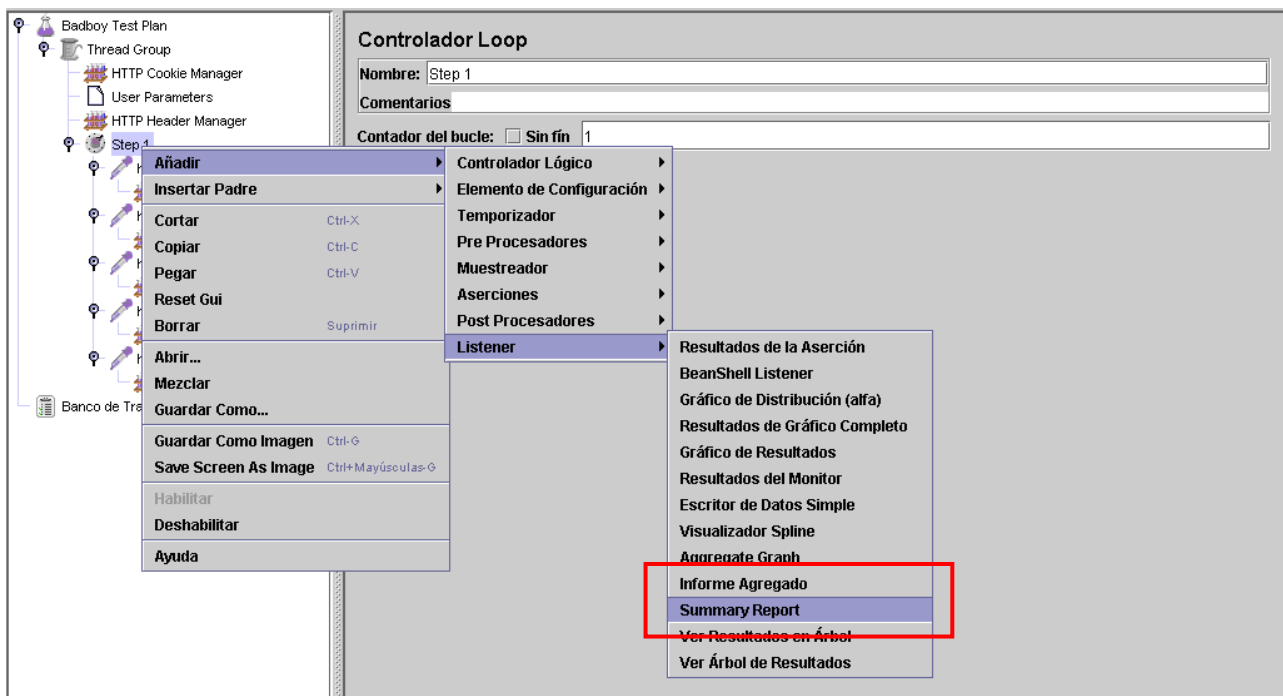
Ejecutar la prueba de prestaciones con Jmeter.
Los resultados de la prueba se obtienen con Jmeter y HP Diagnostics.

HP Diagnostics es una herramienta de profiling que nos permite detectar a nivel de código qué metodos de las aplicaciones y qué sentencias SQL tienen mayores tiempos de respuesta y/o consumen mayores recursos (memoria, cpu).

También nos reporta información sobre numerosas métricas de interés relacionadas con el rendimiento de las aplicaciones, por ejemplo métricas sobre los recursos de weblogic (JDBC, Ejbs, Sesiones Web, JMS) o sobre la JVM (Heap, Garbage collection).

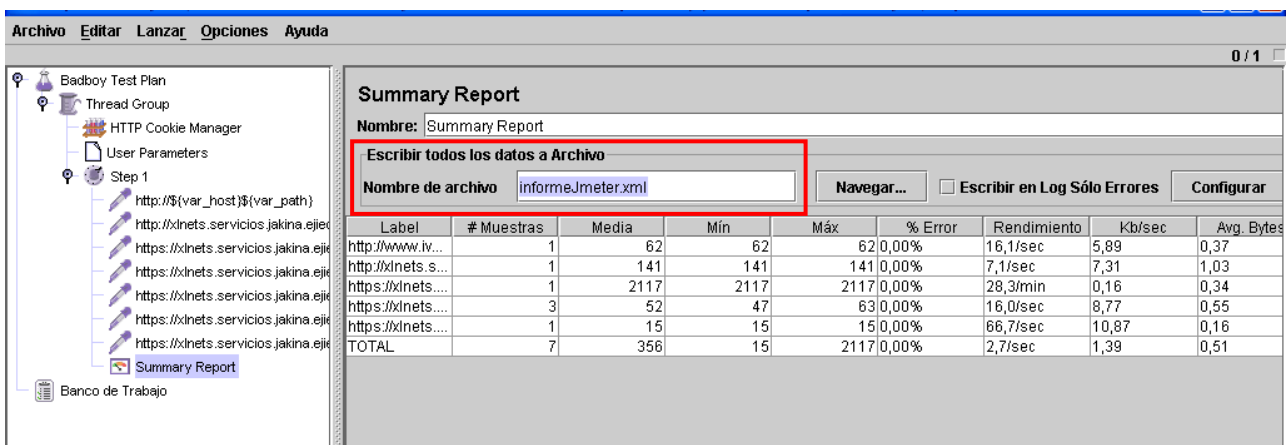
Jmeter

Para recopilar los datos de la prueba, será necesario añadir el informe “**Summary Report**” al plan de pruebas, para ello, seleccionar el Step en el árbol de la izquierda y con el botón secundario: Añadir -> Listener -> Summary Report:

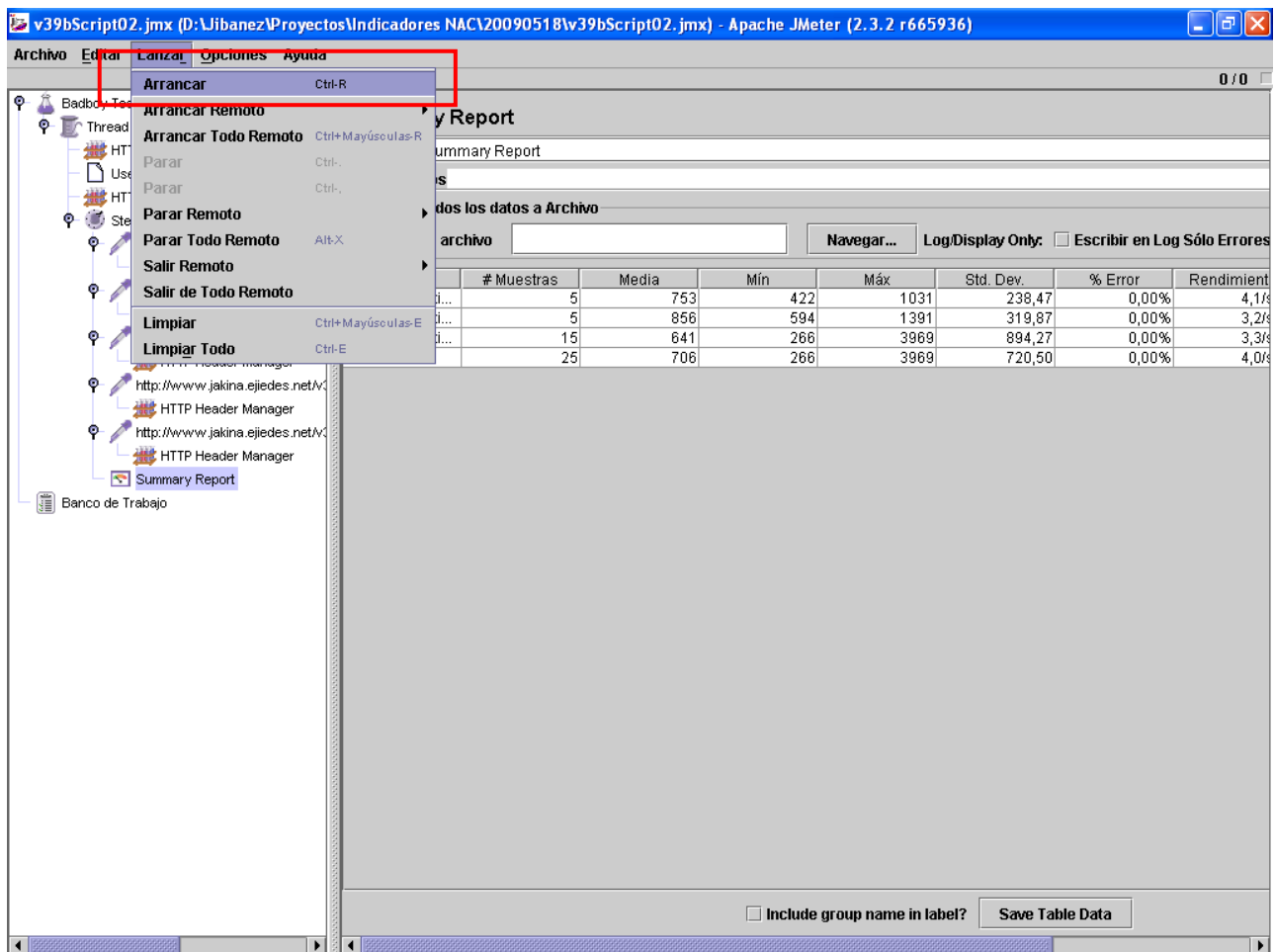


Antes de ejecutar la prueba, hay que configurar JMeter para que genere un informe con el resultado en un archivo XML. Para ello, seleccionar Summary Report -> Rellenar la caja de texto “Nombre de archivo” con la ruta completa y nombre del fichero XML donde se va a generar el informe, por ejemplo:

“C:\Informes\SW\InformeJmeter01.xml”



Para ejecutar la prueba, seleccionamos Summary Report y en el menú superior, Lanzar -> Arrancar:



Cada vez que ejecutemos la prueba los resultados se van acumulando en el informe Summary Report. Si no queremos que se acumulen los resultados hay que borrar el Summary Report y volverlo a generar.

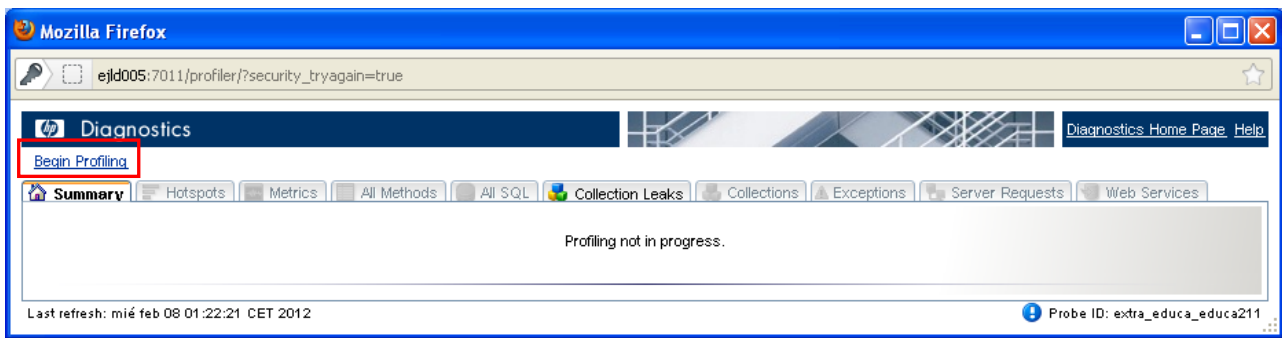
HP Diagnostics

(La carga se genera con Jmeter, ejecutando la prueba como se indica en el apartado anterior).

Antes de ejecutar la prueba con Jmeter, arrancamos HP Diagnostics. Para ello, lanzamos el script de arranque de la sonda (probe) del propio producto y accedemos vía web a la consola a través del puerto correspondiente (<http://host:port>).

En caso de no disponer de script de arranque solicitarlo vía RFC.

Desde la consola iniciamos el proceso de profiling pulsando el enlace Begin profiling:



En este momento podemos generar la carga ejecutando la prueba con Jmeter, según lo indicado en el apartado anterior. Los resultados de la prueba se irán recopilando en las pestañas de la consola.

Paso 3: Consulta del resultado para su análisis.

Dónde consultar los indicadores propuestos con las herramientas.

JMeter

Visualización de los datos con la propia herramienta, Summary Report.

También se puede aplicar al informe (informeJmeter.xml) una plantilla xsl (jmeter-results.xsl) para Jmeter y generar un informe HTML.

En el informe obtenido con la herramienta JMeter se obtendrán los valores de los indicadores:

- **peso medio de las páginas (Indicador ISW)**
- **tiempo medio de resolución de una petición (Indicador ISW)**
- **% de páginas con error** (en este caso % de correctas). **(Indicador ISW)**

Summary

Tests	Failures	Success Rate	Average Time	Min Time	Max Time
128	0	68.75%	983 ms	346 ms	5029 ms

Pages

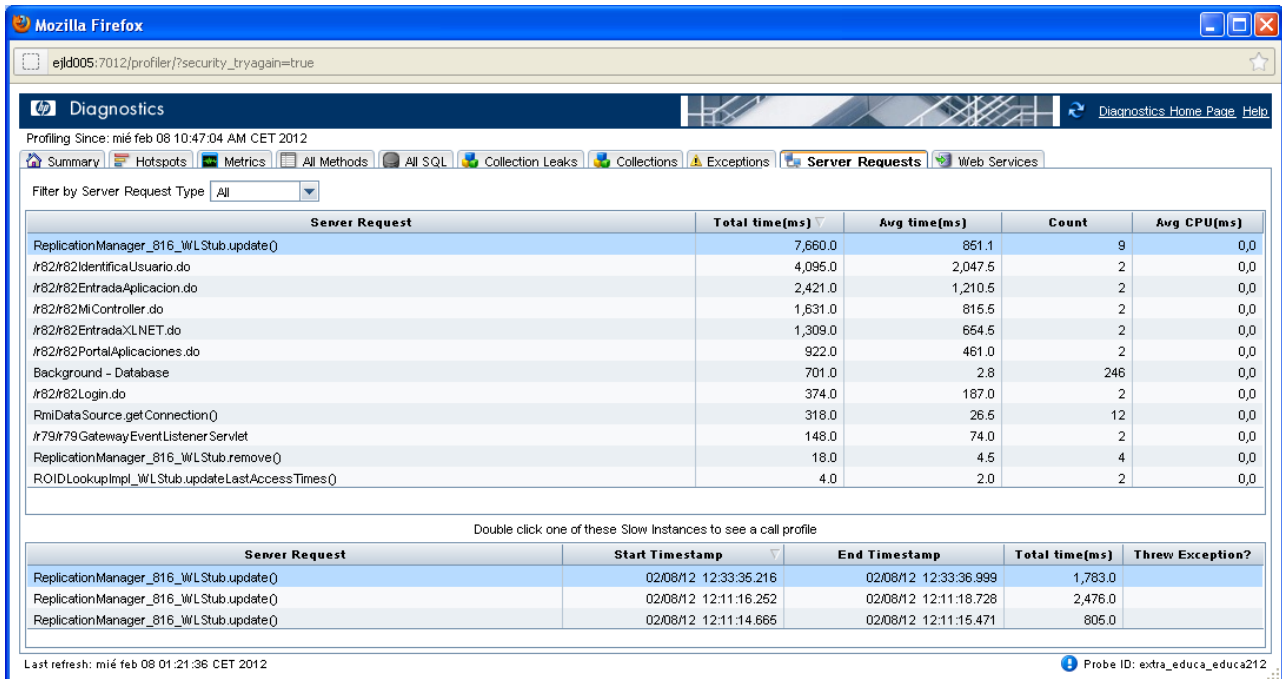
URL	Tests	Failures	% ACIERTO	TIEMPO MEDIO	TAMANO MEDIO	Min Time
			Success Rate	Average Time	Average Bytes	
/xxxIGADWar/comunJSP/xxxInicio.do	16	0	68.75%	1020 ms	4038 bytes	353 ms
/xxxIGADWar/comunJSP/xxxSubmitInicio.do;jsessionid=1-11811630051-1147598386	16	0	68.75%	698 ms	4154 bytes	357 ms
/xxxIGADWar/busquedasJSP/xxxInicializarBusquedaAvanzada.do	16	0	68.75%	928 ms	5021 bytes	366 ms
/xxxIGADWar/busquedasJSP/xxxSubmitBusquedaAvanzada.do	80	0	68.75%	1044 ms	7051 bytes	346 ms

HP Diagnostics

En la consola se obtendrán los valores de los siguientes indicadores:

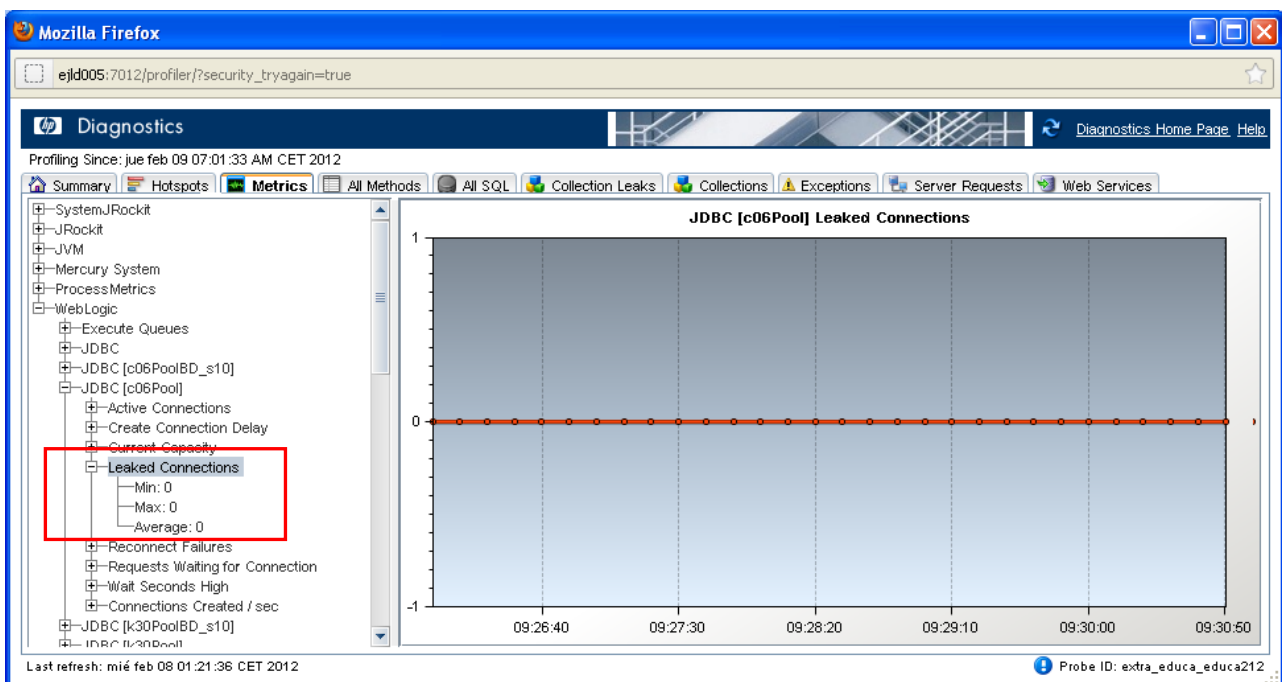
- **tiempo medio de resolución de una petición (Indicador ISW)**

Seleccionar la pestaña Server Requests: Aparecen los mayores tiempos totales y medios de ejecución de peticiones al Servidor.



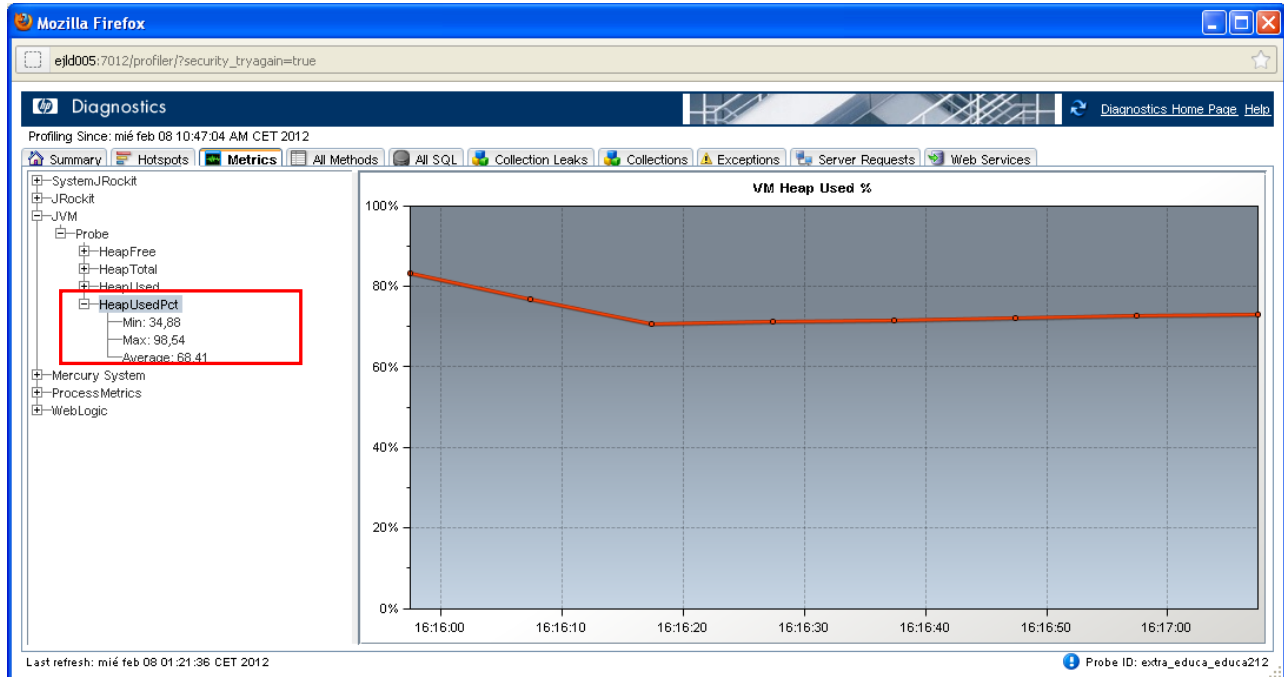
- **conexiones a BBDD no cerradas (Indicador ISA)**

Seleccionar la pestaña Metrics: Expandir el árbol Weblogic y el pool de conexiones JDBC a analizar -> Leaked connections



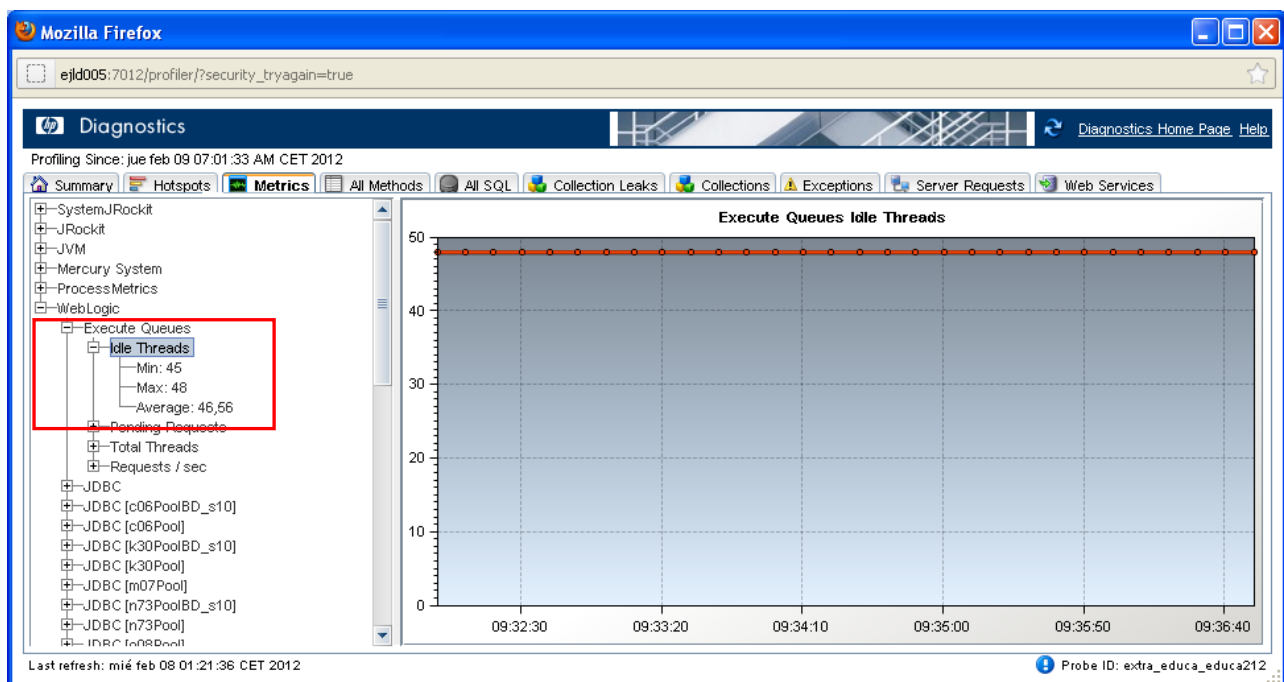
- **memoria libre de la máquina virtual (Indicador ISA)**

Seleccionar la pestaña Metrics: Expandir el árbol JVM -> Probe -> HeapUsedPct (Se muestra el % utilizado, restar 100% al % mostrado).



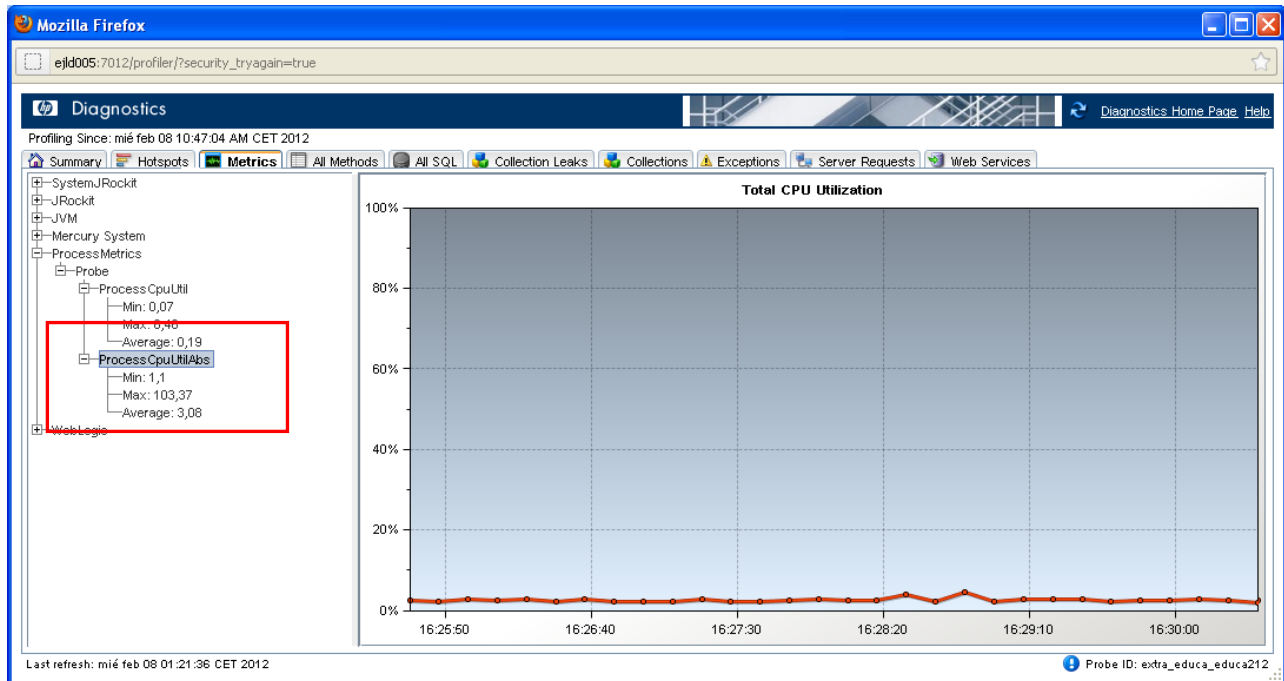
- **hilos libres (Indicador ISA)**

Seleccionar la pestaña Metrics: Expandir el árbol Weblogic y Execute Queues -> Idle threads



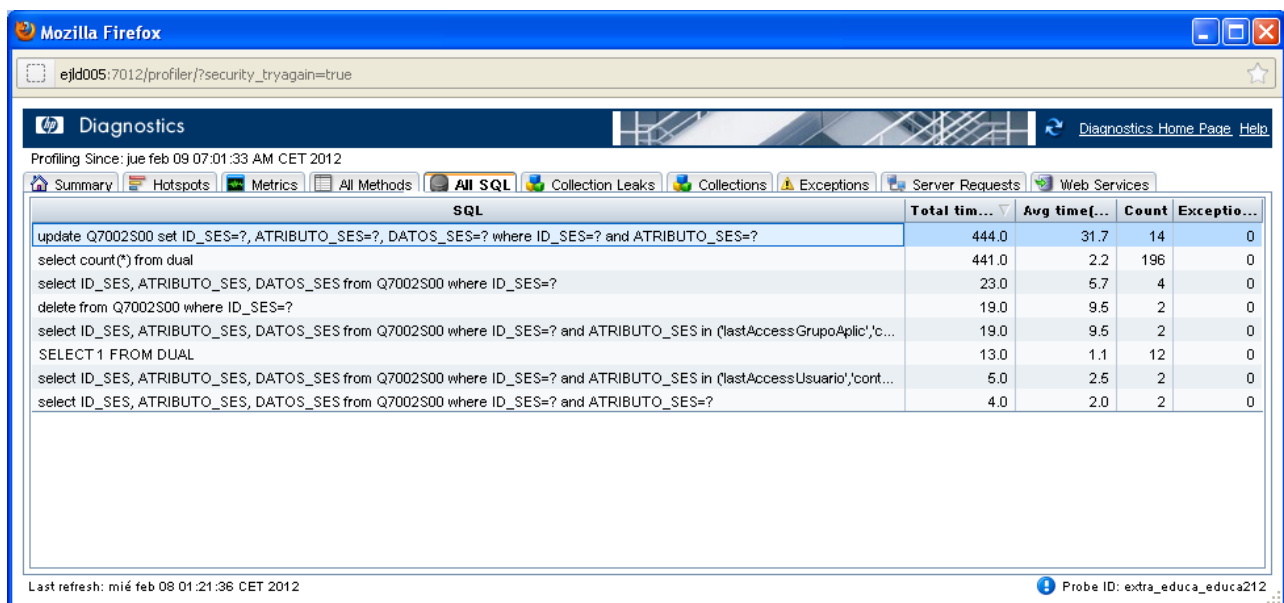
- **% de CPU utilizada (Indicador ISA)**

Seleccionar la pestaña Metrics: Expandir el árbol ProcessMetrics -> Probe -> ProcessCpuUtilAbs



- **tiempo medio de ejecución SQL (Indicador IBD)**

Seleccionar la pestaña All SQL: Aparecen los mayores tiempos totales y medios de ejecución de SQL



Profiling Since: jue feb 09 07:01:33 AM CET 2012

Summary | Hotspots | Metrics | **All SQL** | Collection Leaks | Collections | Exceptions | Server Requests | Web Services

SQL	Total tim...	Avg time(...	Count	Exceptio...
update Q7002S00 set ID_SES=?, ATRIBUTO_SES=?, DATOS_SES=? where ID_SES=? and ATRIBUTO_SES=?	444.0	31.7	14	0
select count(*) from dual	441.0	2.2	196	0
select ID_SES, ATRIBUTO_SES, DATOS_SES from Q7002S00 where ID_SES=?	23.0	5.7	4	0
delete from Q7002S00 where ID_SES=?	19.0	9.5	2	0
select ID_SES, ATRIBUTO_SES, DATOS_SES from Q7002S00 where ID_SES=? and ATRIBUTO_SES in ('lastAccessGrupoAplic','c...	19.0	9.5	2	0
SELECT 1 FROM DUAL	13.0	1.1	12	0
select ID_SES, ATRIBUTO_SES, DATOS_SES from Q7002S00 where ID_SES=? and ATRIBUTO_SES in ('lastAccessUsuario','cont...	5.0	2.5	2	0
select ID_SES, ATRIBUTO_SES, DATOS_SES from Q7002S00 where ID_SES=? and ATRIBUTO_SES=?	4.0	2.0	2	0

Last refresh: mié feb 08 01:21:36 CET 2012

Probe ID: extra_educa_educa212

3.2 Seguridad : Grado de cumplimiento del owasp top 10 de seguridad

El detalle del contenido de las checklist se encuentra en el documento de indicadores: Indicadores de calidad software[1]

3.3 Usabilidad: Grado de cumplimiento evaluación heurística de usabilidad

Este indicador presenta el porcentaje de cumplimiento de los heurísticos de usabilidad partir de la media ponderada de cada una de las cuestiones de usabilidad.

Es un indicador calculado a partir de los datos de la checklist de usabilidad (Portal SQA). El detalle del contenido de las checklist se encuentra en el documento de indicadores: Indicadores de calidad software[1]

3.4 Accesibilidad

El valor del indicador de accesibilidad será el requerido en los requisitos del proyecto. Por tanto, debería estar documentado en Testlink

Una vez realizadas las pruebas de accesibilidad con la herramienta escogida (TAWdis, TAW u otras – ver “Manual de usuario TAW” [7]en el apartado 1.4 Referencias), se obtendrá el nivel de accesibilidad A, AA, o AAAA.