

# DESARROLLO DE UN PAQUETE DE SEGUIMIENTO DE USUARIOS PARA DOTLRN

Autores:

David Ortega (doa@tid.es), Telefónica I+D

Pablo Arozarena (pabloa@tid.es), Telefónica I+D

Rocael Hernández (roc@galileo.edu), Universidad Galileo

*Organizaciones: Telefónica I+D y Universidad Galileo (Proyecto E-LANE)*

Palabras clave: E-LANE, dotLRN, OpenACS, user-tracking, Awstats, e-learnig, @lis, tele-enseñanza

El proyecto E-LANE [1] pertenece al programa @lis [2], de la Comisión Europea. Este programa aborda proyectos de demostración de aplicaciones de las tecnologías de la información en el ámbito Latinoamericano, con el objetivo de probar su utilidad para la sociedad de esta región. En el marco de este programa, E-LANE se centra en el desarrollo y demostración de plataformas y metodologías de e-learning (tele-enseñanza). Para ello, una de las líneas de actividad del proyecto se centra en la utilización y ampliación de la plataforma de e-learning dotLRN [3], construyendo distintas herramientas destinadas a mejorarla. En el presente artículo, se explicarán los conceptos fundamentales relacionados con el desarrollo de una de estas herramientas, llamada user-tracking, capaz de recopilar y mostrar información sobre el comportamiento de los usuarios dentro de la plataforma. Con ella se pretende que, tanto profesores como administradores del sistema, comprendan cómo se está utilizando el mismo y así sean capaces de mejorar el proceso de aprendizaje. De manera adicional los alumnos tendrán acceso a un conjunto limitado de estadísticas sobre los cursos y comunidades a las que pertenecen.

## 1 INTRODUCCIÓN

La principal motivación para abordar el desarrollo de esta herramienta surge de la constatación de las limitaciones de la plataforma dotLRN para proporcionar de forma cómoda información fidedigna sobre el uso de la misma. Dicha información es de gran importancia tanto desde el punto de vista de la administración del sistema, especialmente en aquellas instancias con un elevado número de usuarios, como para comprender mejor el funcionamiento y uso de la plataforma y poder abordar acciones encaminadas a mejorar los contenidos didácticos así como la misma plataforma. Una de las principales restricciones de diseño que se presenta a la hora de abordar el desarrollo de un sistema de seguimiento de usuarios (en adelante lo llamaremos user-tracking), es el intento de

minimizar al máximo el uso adicional de recursos que pueda repercutir negativamente en el rendimiento general del sistema.

En el caso concreto de dotLRN, toda la información del sistema se almacena en la base de datos del mismo. Es por eso, que la primera intención sería almacenar la información relacionada con el seguimiento de usuarios dentro de la base de datos. Quizás esta solución sería la que proporcionase información más fidedigna pero, sin embargo, la utilización de la base de datos sería constante, ya que cada vez que un usuario interactuase con el sistema, habría que almacenar en ella las acciones realizadas. En la actualidad, dotLRN se utiliza en entornos con gran cantidad de usuarios por lo que esta solución, repercutiría de forma muy negativa en las prestaciones del sistema.

Asimismo, hay que tener en cuenta que cada vez que se realiza un petición http al servidor WEB que

utiliza la plataforma (AOL server), ésta es almacenada en un fichero de log. Por tanto, la información almacenada en los ficheros de log del servidor WEB puede constituir una importante fuente de datos para el sistema de user-tracking.

Recapitulando, desde el punto de vista del proyecto, existen en dotLRN dos importantes fuentes de datos: Una, la base de datos del sistema, en la que se puede encontrar información sobre los objetos que se han creado dentro del mismo, incluyendo su tipo, fecha de creación, usuario que lo ha creado, etc.

Otra, los ficheros de log del servidor WEB, en la que se almacenan las peticiones http que realizan los usuarios al sistema, recogiendo: la fecha completa de petición, la url solicitada, el tipo de navegador utilizado, la dirección IP desde la que se ha realizado la petición, etc.

Como se verá más adelante, la herramienta desarrollada utilizará ambas fuentes de datos para conseguir los objetivos planteados.

## 2 REQUISITOS

La principal funcionalidad que ha de proporcionar la herramienta es la de ser capaz de capturar los aspectos más relevantes relacionados con la interacción de los usuarios con el sistema. En base a ellos, se realizará la generación de distintos tipos de informes, que se describen a continuación, a grandes rasgos.

En primer lugar, el sistema será capaz de generar informes sobre el funcionamiento general del sitio WEB. En ellos, se mostrará información sobre la visitas que han realizado los usuarios, generando informes que analicen las horas en las que se visita el sitio con más frecuencia y los días de la semana y del mes en los que se producen. Igualmente se analizará la duración de las mismas. De esta forma, los administradores del sistema conocerán los momentos del día, semana o mes, en los que se produce una mayor carga dentro del sistema, pudiendo actuar en consecuencia.

Otro importante grupo de informes, se centrará en los contenidos que los usuarios han introducido o han consultado en el sistema, ya sean mensajes en los foros, noticias, ficheros, faqs, encuestas, etc. Gracias a estos informes, los profesores podrán conocer mejor tanto el grado de participación de sus alumnos, como qué elementos han sido mejor aceptados por ellos, obteniendo de esta forma una importante retroalimentación para la organización posterior de cursos basados en los mismos contenidos.

Todos los informes se podrán generar acotando los datos de entrada a los mismos. De esta forma, se podrá mostrar información sobre:

- el comportamiento de todo el sitio WEB, recopilando información útil para los administradores del sistema
- el comportamiento de todos los usuarios de una o varias clases, destinada por ejemplo, a ser analizada por los profesores de las mismas o personas responsables del departamento que las imparte.
- el comportamiento de algunos usuarios en una o más clases (o incluso en todo el sistema), igualmente útil para los profesores encargados de impartir las clases e incluso para los administradores del mismo.

Asimismo, la herramienta tendrá que ser capaz de gestionar correctamente los permisos de acceso a los distintos informes, de forma que la información sólo pueda ser consultada por los usuarios autorizados para hacerlo. En principio, cada usuario será capaz de ver información sobre su propio comportamiento dentro del sistema. Además, los profesores podrán solicitar informes relacionados con las clases que impartan y, por último, los administradores serán capaces de solicitar la generación de todos los informes disponibles.

Tabla 1: Fuentes de datos

Tipo de informe	Fuente de datos
Estadísticas de acceso (mensuales, semanales, diarias...)	Ficheros de log
Visitas de usuarios	Ficheros de log
Páginas visitadas	Ficheros de log
Elementos (foros, mensajes, noticias, ficheros...) visitados por los usuarios	Ficheros de log
Contribuciones (foros, mensajes, noticias, ficheros...) de los usuarios	Base de datos
Usuarios registrados en el sistema	Base de datos
Número de clases y comunidades	Base de datos

En la tabla anterior, se enumeran los principales tipos de informes que se podrán realizar con la herramienta, indicando de dónde se obtendrán los datos para generarlos (base de datos o ficheros de

log). Cada uno de estos tipos puede comprender la realización de varios informes distintos. Así, dentro de los informes sobre las contribuciones de los usuarios al sistema, se podrán generar informes particulares para cada tipo de contribución distinta, es decir, un informe para ver las noticias publicadas, otro para ver los mensajes en los foros, otro para los ficheros subidos, ...

### 3 DESCRIPCIÓN DE LA HERRAMIENTA DESARROLLADA

#### 3.1 Obtención de Datos

Tras estudiar los requisitos del paquete de *user-tracking*, se puede apreciar que los datos necesarios se encuentran repartidos entre la base de datos y los ficheros de log. Toda la información guardada en la base de datos es fácil de obtener y fácil también de integrar dentro de la estructura de dotLRN.

La mayor dificultad surge con el análisis de los datos que se almacenan en los ficheros de log del servidor WEB. Para procesar esos ficheros, se ha decidido utilizar un analizador de logs externo que será integrado en dotLRN.

En las siguientes secciones se explica en detalle la problemática asociada con el análisis de logs y cómo ha sido resuelta.

#### 3.2 Análisis de logs

Una típica línea de log generada por el servidor WEB AOL server tiene el siguiente aspecto:

```
10.95.74.90 - - [15/Feb/2005:14:55:33 +0100] "GET /dotlrn/classes/uno/prueba/prueba1/lorsm/delivery/view/p3/ch01.html HTTP/1.1" 200 14876 "http://10.95.74.92:8000/dotlrn/classes/uno/prueba/prueba1/lorsm/delivery/view/p3/index.html?imsitem%5fid=6515" "Mozilla/5.0 (Windows; U; Windows NT 5.1; es-ES; rv:1.7.5) Gecko/20041108 Firefox/1.0"
```

No obstante, para poder recoger la información necesaria para generar los distintos tipos de informes mencionados en el apartado anterior, es necesario enriquecer esas líneas de log, añadiendo información sobre el usuario que realiza la petición, así como de la comunidad/clase dentro de la cuál se está realizando la misma. De esta forma, una línea de log típica quedaría así:

```
LogFormat= "%host %other %other %timeI %methodurl %code %bytesd %refererquot %uaquot %lognamequot %extraI"
```

El penúltimo campo es el identificador de usuario. Será cero si la petición realizada no corresponde con ningún usuario y un entero positivo en caso contrario.

El último campo es el identificador de la comunidad. Para tenerlo en cuenta se reconoce como un parámetro extra (*%extraI*), sobre el que, en función del analizador de logs seleccionado, se pueden realizar análisis adicionales no generados directamente por el mismo.

Para añadir a los ficheros de log de dotLRN información adicional sobre el usuario que realiza la petición y la comunidad sobre la que la realiza, hace falta implementar un filtro. Para ello es suficiente añadir lo siguiente a algún fichero del tipo *\*-procs.tcl* de dotLRN.

```
ns_register_filter preauth GET "*" log_user_id
ns_register_filter preauth POST "*" log_user_id

ad_proc -public log_user_id {auth conn} {
    Log user_id
} {
    set user_id [ad_conn user_id]
    set community_id [dotlrn_community::get_community_id]
    if {[empty_string_p $community_id]} {
        set community_id 0
    }
    ns_set put [ns_conn headers] X-User-Id "$user_id\" \"community_id=$community_id"
    return "filter_ok"
}
```

Dentro del paquete de *user-tracking* se ha incluido este código dentro del fichero *tcl/filter.tcl*.

Además también es necesario modificar el fichero de configuración de dotLRN (*/etc/config.tcl*) para que incluya la sección *X-User-Id* en los logs. Para eso hay que añadir la siguiente línea dentro de la sección *"ns\_section ns/server/\${server}/module/nslog"*.

```
ns_param extendedheaders X-User-Id
```

Después de examinar distintas herramientas para realizar el análisis de los ficheros de log, se ha seleccionado *Awstats* [4]. Se trata de una herramienta de código abierto implementada en Perl, muy bien desarrollada, con una estructuración muy clara y unos comentarios correctos. Los motivos que han llevado a seleccionar *Awstats* son:

- su potencia y flexibilidad para realizar análisis de logs
- la facilidad de ampliarla modificando su código fuente.

El funcionamiento de Awstats es el siguiente:

1. Lee el fichero de configuración para obtener la ubicación de los ficheros de log, y todos los parámetros de configuración.
2. Lee los ficheros de log, y va analizando línea por línea:
  - Coge los datos que se le han pedido a través del fichero de configuración, (url visitada, fecha, hora, dirección IP, usuario, etc.)
  - Almacena los datos obtenidos en vectores
3. Guarda los datos obtenidos en un fichero de texto estructurado
4. Por último lee esos ficheros de texto y genera informes en HTML de forma dinámica.

### 3.2.1 Problemas detectados

Awstats ofrece la posibilidad de añadir ciertos informes extras de manera que el análisis realizado consiga ofrecer la mayoría de los datos especificados en los requisitos aunque, como la mayoría de las herramientas de análisis de logs, se centra principalmente en un uso básicamente orientado a la administración del sitio, mostrando un gran número de estadísticas temporales y horarias, y dando prioridad a los datos de utilización y consumo de ancho de banda del sitio, sobre los objetos visitados, y los usuarios que lo han hecho.

La obtención de datos filtrados por usuarios plantea mayores problemas ya que la herramienta no permite filtrar los ficheros de log en función de este parámetro. Se plantean dos posibilidades:

- Filtrar los ficheros de log antes de que los use Awstats
- Modificar Awstats para que filtre los ficheros internamente

El hecho de realizar un filtrado previo de los ficheros de log implicaría realizar dos lecturas de estos ficheros (que son bastante grandes): una primera para filtrar, y una segunda por parte de Awstats.

Esta solución si bien es muy sencilla, tiene claros problemas de rendimiento, por lo que se ha decidido modificar el código de Awstats.

### 3.2.2 Modificaciones realizadas en Awstats

Una vez estudiado el esquema general de funcionamiento de Awstats, se puede apreciar que una posible solución al problema planteado en el apartado anterior radica en añadir un filtro en el punto en el que se analiza cada línea de log.

De esta forma, dentro del código de awstats se ha añadido el parámetro *onlyusers*, que se puede pasar al script bien en el fichero de configuración, o bien por línea de comandos. El valor de este parámetro puede ser uno o varios identificadores de usuario, de forma que se puedan obtener informes de un solo usuario o de varios usuarios agregados.

Aprovechando la potencia de análisis de cadenas y expresiones regulares de perl, se ha añadido además la opción de poder filtrar los usuarios por expresiones regulares, de forma que se permita seleccionar grupos de usuarios que cumplan con un determinado patrón y realizar informes sobre ellos. Para ello es suficiente con poner la expresión regular buscada dentro del código REGEX[ ... ] y pasárselo al parámetro *onlyusers*.

Asimismo, se ha modificado la herramienta para que sea capaz también de generar informes independientes de cada comunidad o clase presentes en el sistema. Para ello se ha seguido el mismo esquema de funcionamiento utilizado para filtrar usuarios, definiendo el parámetro *onlycoms*, cuyo funcionamiento es similar al de *onlyusers*, con la única diferencia de que lo que recibe este parámetro son identificadores de comunidades o clases y no de usuarios.

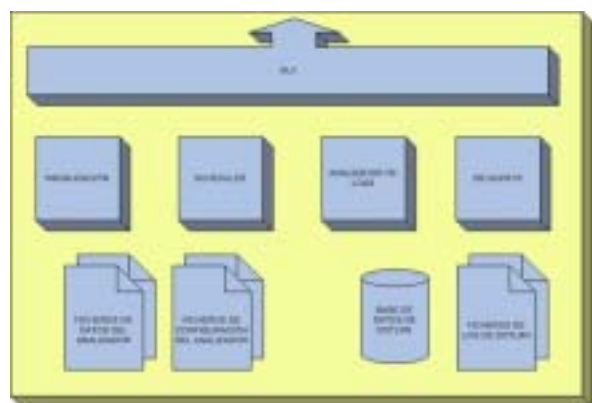


Figura 1: Arquitectura de la herramienta

### 3.3 Arquitectura del sistema

A alto nivel, se puede descomponer la arquitectura de la herramienta de *user-tracking* desarrollada en los siguientes módulos (ver figura 1)

- **GUI:** Pares de páginas dinámicas tcl/adp que permiten al usuario elegir los análisis a realizar, y muestran los informes solicitados. De igual forma que se realiza en el resto de los paquetes de dotLRN, dentro de este módulo se efectúa toda la comprobación de permisos, que indica qué informes pueden ver los usuarios, en función de si son administradores del sistema, profesores, alumnos, etc...
- **INICIALIZACIÓN:** Se encarga de todo lo relacionado con el proceso de inicialización del paquete, una vez que éste ha sido instalado y configurado. Crea, entre otras cosas, la estructura de directorios del paquete y todos los ficheros de configuración que necesita el analizador de logs. Asimismo, activa los filtros encargados del enriquecimiento de los ficheros de log.
- **SCHEDULER:** Se encarga de la gestión de tareas periódicas, como la actualización de los ficheros de datos generados por awstats.
- **AWSTATS:** Compuesto por awstats y los ficheros de configuración necesarios para su utilización.
- **DB QUERYS:** Conjunto de queries predefinidas que se realizan sobre la base de datos de dotLRN para obtener los datos necesarios.
- **FICHEROS DE CONFIGURACIÓN DE AWSTATS:** Predefinidos en el paquete básico y autoconfigurados en el proceso de instalación/inicialización de la herramienta.
- **FICHEROS DE DATOS:** Generados por awstats, recogen la información relevante obtenida de los logs.
- **FICHEROS DE LOG Y BASES DE DATOS:** Fuentes de datos para la herramienta.

### 3.4 Estructura de la herramienta

La herramienta de seguimiento de usuarios, está compuesta por tres paquetes de dotLRN, que se describen a continuación:

- DotLRN-user-tracking: Applet utilizado para integrar la herramienta en dotLRN.
- User-tracking-portlet: Portlet, utilizado igualmente para integrar la herramienta en dotLRN (ver figura 2).
- *User-tracking*: Paquete que implementa toda la funcionalidad de la herramienta.

El hecho de que se utilice un analizador externo para realizar el análisis de los ficheros de log, provoca que en este último paquete se añadan unos cuantos directorios extra, que no se adecuan al estándar que siguen la mayoría de los paquetes de dotLRN.

La siguiente tabla muestra los directorios añadidos dentro del paquete de *user-tracking*:

Tabla2: Directorios adicionales

Ruta	Función
User-tracking /config	<p>Contiene los ficheros de configuración utilizados por el analizador externo:</p> <ul style="list-style-type: none"> <li>• awstatsDirecto.class.conf → Parámetros para realizar informes sobre comunidades/clases</li> <li>• awstatsDirecto.user.conf → Parámetros para realizar informes sobre usuarios</li> <li>• awstatsDirecto.site.conf → Parámetros para realizar informes sobre todo el sistema.</li> <li>• AllowedUrls.conf → Lista con las direcciones que pueden acceder al script por medio de cgi's (no utilizado en la R3).</li> </ul>
User-tracking /tools	<p>Contiene el script logresolvemerge.pl, utilizado por el analizador de logs, para procesar todos los ficheros de logs presentes en el servidor.</p>
User-tracking /www/awstats	<p>Contiene el analizador de logs propiamente dicho, así como todos los ficheros que éste necesita.</p>

Además, se crea automáticamente un nuevo directorio que cuelga de la carpeta en la que el servidor WEB almacena sus ficheros de logs. Dentro de este directorio, se almacenan los ficheros de datos generados por el analizador, que no son sino resúmenes de la interacción de los usuarios con el sistema, y que son utilizados por la herramienta para generar los informes solicitados.

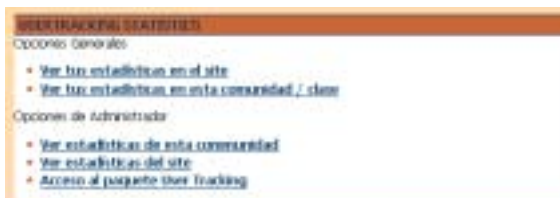


Figura 2: Portlet de user-tracking



comunidades o clases. En todos ellos se podrá seleccionar también los usuarios sobre los que se realiza el informe. En relación a esto, en la figura 8 se muestra la pantalla a través de la cual se seleccionarían las comunidades sobre las que se realizaría un informe.



Figura 8: Pantalla de selección de comunidades

## 5 CONCLUSIONES

Por motivos de escalabilidad, el uso de un analizador de log constituye una excelente herramienta, no costosa en consumo de recursos, para analizar las visitas y el uso de los diferentes objetos (elementos) de dotLRN.

Sin embargo, no toda la información necesaria está disponible en los ficheros de log del servidor WEB, siendo preciso en algunos casos acceder a la información almacenada en la base de datos del sistema. Dado que el acceso a la base de datos puede plantear problemas de rendimiento a la aplicación, sobre todo si requiere almacenar información adicional para el proceso de seguimiento de usuarios, hace falta estudiar detalladamente en cada caso cuál es la mejor opción y llegar a un compromiso entre la funcionalidad requerida y las prestaciones del sistema.

Todas estas consideraciones se han tenido en cuenta a la hora de diseñar el presente paquete de seguimiento de usuarios, obteniendo como resultado una herramienta flexible para la creación de informes sin comprometer en exceso las prestaciones del sistema. Se ha comprobado en laboratorio que el sistema es capaz de responder a las peticiones de generación de estadísticas en un tiempo relativamente corto (unos pocos segundos) y, además, se han incluido herramientas que permiten programar el análisis automático de datos en horas de baja carga del sistema.

Actualmente el paquete *user-tracking*, en su primera versión, está siendo utilizado en las distintas instalaciones del proyecto E-LANE, así como en algunas de las universidades que colaboran en el mismo. Es de esperar que, como resultado de su uso en entornos reales, se obtenga una valiosa retroalimentación que permita mejorar la herramienta en futuras versiones.

Adicionalmente, con la presencia de esta herramienta se abre la puerta a desarrollos de adaptación automática como, por ejemplo, material que se ofrece o deja de ofrecer automáticamente dependiendo de su utilidad, obtenida a partir de las estadísticas de acceso.

El paquete de seguimiento de usuarios está disponible en:

[cvs.openacs.org/openacs-4/user-tracking](http://cvs.openacs.org/openacs-4/user-tracking).

### Referencias

- [1] E-Lane, <http://www.e-lane.org/>
- [2] Programa @lis, [http://europa.eu.int/comm/europeaid/projects/alis/index\\_es.htm](http://europa.eu.int/comm/europeaid/projects/alis/index_es.htm)
- [3] dotLRN, <http://dotlrn.org/>
- [4] Awstats, <http://awstats.sourceforge.net/>.