

Arquitectura de publicación automatizada de contenidos educativos supervisados en Internet.

¹A. Arin, ²E. Magaña, ³J.J. Astrain, ⁴Jesús Villadangos, ⁵J.R González de Mendivil

Departamento de Automática y Computación

Universidad Pública de Navarra

Campus Arrosadia s/n. 31006 Pamplona (Navarra)

Tel: 948-168904, Fax: 948-169281

E-mail: {¹ma10747, ³ja11066}@zurron.upna.es, {²eduardo.magana, ⁴jesusv, ⁵mendivil}@upna.es

Abstract

This paper presents an automated publishing architecture of educational and supervised contents over the Internet. The system makes easier the job of publishing educational courses over the network using specific tools that automates the access control (CGI -- Common Gateway Interface), encrypts the information that goes through the net for not being accessed by strangers (SSL – Secure Sockets Layer), gives an statistical control of the usage of the system, and tutors the course. This project is multiplatform, i.e. all the components that integrate the system are available for any operating system (Windows 9X, Solaris, Linux,...) and it is based on the Web.

1. Introducción

Las necesidades de formación cada vez más especializadas han provocado el auge de métodos de enseñanza alternativos como el uso de CDROMs multimedia e intercambio de información a través de la red Internet.

El presente trabajo trata de facilitar la publicación de cursos en la red mediante el uso de herramientas específicas que automaticen el control de accesos según las condiciones deseadas, encriptación de la información para que no pueda ser accedida por personas ajenas y un control estadístico de su uso.

Este medio facilita que el alumno pueda acceder a la información cuando le convenga. Se pretende aportar una herramienta educativa mediante la cual los alumnos puedan seguir los cursos a través de la red sin sufrir los inconvenientes que pueda suponer el entorno de la Web y sin tener que hacerlo durante unos horarios fijos.

La herramienta desarrollada posee las siguientes características:

- *Cursos fáciles de generar:* El profesor o dueño del curso tendrá una aplicación completa y sencilla de manejar para publicar sus contenidos en Internet.
- *Curso supervisado:* El alumno estará controlado en todo momento por el profesor ya que nuestro sistema mantendrá un registro de todas las peticiones que realice al servidor. También permite el acceso a los contenidos sólo a las personas autenticadas.
- *Curso tutorizado:* Cada alumno podrá exponer sus dudas en cualquier instante mediante correo electrónico. En caso de que

el profesor estuviese disponible, las dudas se resolverían mediante una comunicación on-line.

- *Curso monitorizado mediante estadísticas:* El dueño del curso puede visualizar estadísticas para así poder corregir errores, insuficiencias en los contenidos o para conocer los temas de más éxito.

Este artículo se va a estructurar de la siguiente forma: en la sección dos se presenta una serie de experiencias educativas. En la sección tercera se proporciona una visión global del sistema. En la cuatro se explica el interfaz del usuario con el sistema, mientras que en el apartado cinco se da una explicación de la base de datos. En la siguiente sección se explica el núcleo del sistema. Para finalizar el artículo, en la sección siete se muestra el generador de cursos. Por último se expondrán las conclusiones.

2. Experiencias de sistemas educativos por la red

En la actualidad existen diferentes métodos de seguimiento de cursos a través de Internet. En esta sección se mostrarán las distintas tendencias que se han seguido en diferentes experiencias.

En la Universidad de Amsterdam [1] (Mayo – Julio de 1996) se estudió la posibilidad de ofrecer cursos académicos a través de Internet y revelar dificultades asociadas. Para ello se pidió a los alumnos que estudiaran ciertos temas, que recibían en formato texto, durante cada semana. Luego tomaban parte en una discusión de la materia de la semana con los profesores en la clase virtual.

Otra experiencia interesante se montó entre el Swiss Federal Institute of Technology of Zürich y el Swiss Federal Institute of Technology of Laussane (1995) [2]. Se buscaba obtener experiencia en temas

como servicios soportados por la red, equipamiento utilizado y la tele-enseñanza como un concepto didáctico.

En la experiencia realizada en la Universidad de Stuttgart (1997) [3] se hace hincapié en las limitaciones de la Web, tales como que la interacción entre el usuario y el servidor de Web es sin estado, el efecto denominado “perdido en el espacio” debido a la inconsistencia de los hyperlink, y la dificultad de localizar algunos recursos.

También se han desarrollado otros sistemas de telecomunicación orientados al mundo de la enseñanza como pueden ser las siguientes herramientas:

- Sala ETSIT.
- ISABEL [4].
- Mbone [5].

La mayor parte de estas últimas herramientas son de ámbito universitario aunque se podrían utilizar en ámbitos de enseñanza básica. El único inconveniente es que necesitan una infraestructura de comunicación que no es muy habitual en los centros de enseñanza básica, como por ejemplo una antena parabólica para la sala ETSIT.

3. Arquitectura del sistema.

En la siguiente figura (Fig.1) se puede apreciar la arquitectura básica del sistema de supervisión de cursos.

En el lado del cliente se opta por utilizar un interfaz Web para seguir los cursos en lugar de desarrollar una herramienta propia. Con dicha herramienta propia se podría tener un sistema más a la medida de las necesidades pero presenta el inconveniente de tener que incluirla en todos los ordenadores. En cambio, los navegadores están tan extendidos que cualquiera tiene acceso a ellos, incluso vienen incluidos con el propio sistema operativo.

En el lado del servidor se opta por utilizar un servidor Web, una base de datos y un CGI principal, cuya labor será la de controlar el acceso a los contenidos, evitando que accedan a la información personas no autenticadas.

Por último, se incluye una herramienta visual que genere los cursos de una forma sencilla y transparente.

En esta visión global del sistema se diferencian cuatro partes muy claras:

- Generador de cursos.
- Proveedor de servicio.
- Base de datos.
- Interfaz de usuario.

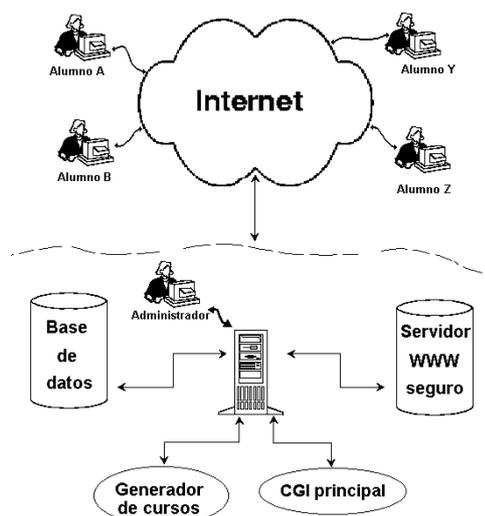


Fig 1. Visión global de la arquitectura del sistema.

La relación entre los distintos elementos del sistema es la que se representa en Fig 2.

El generador de cursos es la herramienta que utilizará el creador del curso para introducir dicho curso en nuestro sistema. El generar el curso será el primer paso a realizar. Una vez generado el curso, el control de dicho curso pasa a ser del servidor de cursos.

La herramienta de generación de cursos confiere al sistema un interfaz gráfico mediante la cual el administrador puede generar cursos de una forma transparente, facilitándole así el trabajo ostensiblemente. De esta manera, evitamos que el administrador o persona encargada de introducir los cursos tenga conocimientos del funcionamiento interno o a bajo nivel del programa.

El generador de cursos copia las páginas que conforman cada página en un directorio del servidor de Web para hacerlos accesibles a través del servidor Web. Además renombra las páginas, cambiando su nombre original por un identificador numérico para un mejor manejo de la base de datos, y reformatea los links entre páginas debido al renombrado de las páginas, para así controlar las páginas y los accesos de los usuarios.

El proveedor del sistema está constituido por un servidor Web, cookies como sistema para llevar el estado de la conexión cliente/servidor, un programa

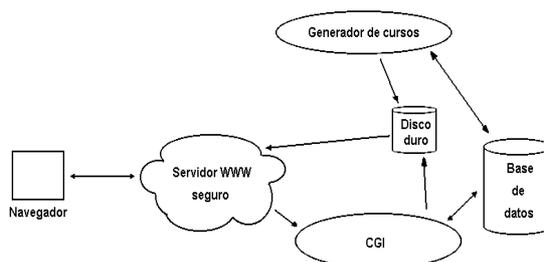


Fig 2. Relación entre los distintos elementos del sistema.

CGI principal, dos CGI's auxiliares y el protocolo SSL para la encriptación de los datos.

El servidor Web será el encargado de servir los cursos en última instancia al cliente o alumno. Este servidor es multidifusión.

El CGI principal junto a las cookies serán los que permitan tener un control de los accesos de los distintos alumnos a los diferentes curso y permitan realizar un seguimiento de la evolución de dichos alumnos. Los dos CGI's auxiliares se utilizan para realizar la labor de inscripción de alumnos de una forma automática y la presentación de una página personalizada que presenta un listado de los cursos en los que un alumno está matriculado actualmente.

El protocolo SSL es el que confiere cierta seguridad a los datos enviados en todas las conexiones existentes entre el cliente y el servidor. Este protocolo es el único protocolo de seguridad estándar incluido en los navegadores.

En definitiva, el módulo del proveedor de servicios es el encargado de servir los ficheros Web a los alumnos de una forma segura y supervisada manteniendo en cada momento un histórico de los accesos a cada uno de los ficheros de los distintos cursos ofrecidos.

La base de datos es la herramienta en la que se basa el supervisor de cursos como sistema de almacenamiento de toda la información generada por el acceso de los alumnos a los cursos. Para almacenar dichos datos se ha generado una estructura específica de tablas que facilitará la inclusión de nuevos cursos, alumnos, páginas, etc.

La base de datos almacena información para luego poder llevar un registro exhaustivo de los accesos de los alumnos al sistema, controlar el número de usuarios activos en cada instante y tener un control de los contenidos.

4. Interfaz de usuario

4.1. El navegador

El interfaz de usuario para el alumno en este sistema es el navegador de Web.

El navegador es una aplicación que hace de cliente en una conexión cliente/servidor de Internet. Sirve para visualizar páginas HTML, todo tipo de elementos multimedia como son las imágenes, vídeo, sonido, vídeo en tiempo real, macromedia, etc. y también para enviar correo electrónico.

4.2. Herramienta de comunicación on-line

La herramienta de comunicación on-line se generó con el objetivo de ofrecer a los alumnos una

forma más inmediata de comunicación con el tutor o profesor del curso que estén siguiendo.

Mediante la herramienta de comunicación on-line, el profesor está recibiendo simultáneamente las preguntas de los alumnos y el alumno recibe de la misma manera la respuesta del profesor. Esta funcionalidad del sistema está programada en Java por ser multiplataforma y además tener posibilidad de un entorno gráfico.

La conexión va sobre TCP ya que una de las características de la herramienta es que no debe haber pérdidas de información en la transmisión. TCP garantiza la fiabilidad de la conexión. El único inconveniente de que la conexión vaya sobre TCP es el retardo implícito que conlleva cada una de las retransmisiones que se dan debido a errores de los paquetes en la red.

Para facilitar la conexión tanto al alumno como al profesor, hay una agenda, que se puede visualizar accediendo a uno de los menús de la aplicación, en la que aparece la lista de cursos tutorizados con los puertos a los cuales se debe conectar el alumno y la máquina donde está el profesor. Al tutor la agenda le sirve de ayuda, a la hora de crear el tutorial, para ponerse de acuerdo en el puerto de conexión.

5. Base de datos

5.1. Introducción

La base de datos se utiliza como herramienta de almacenamiento de datos de distinto carácter:

- Datos personales de los alumnos.
- Información de los ficheros que conforman los cursos ofrecidos.
- Accesos a los distintos ficheros de los cursos.
- Estado de cada una de las conexiones activas que hay en el momento.
- Requisitos de cada uno de los cursos o ficheros que lo conforman.

Con toda esta información almacenada en diferentes tablas de una base de datos, el sistema es capaz de realizar las siguientes tareas:

- Autenticar los usuarios con login y password.
- Mantener estado de la conexión.
- Monitorizar las estadísticas de acceso a cursos por parte de cada alumno.
- Generar cursos de una forma sencilla y transparente para el propietario del curso.

5.2. ¿Por qué MySQL?

En la actualidad hay muchos servidores de bases de datos, tanto gratuitos como comerciales. Como ejemplo están Access, mSQL, Oracle, Dbase, etc.

Las razones por la que se elige MySQL son su velocidad, la robustez y su facilidad de uso. Se pueden utilizar en un entorno con hasta 40 bases de datos que contienen 10.000 tablas, muchas de las cuales pueden tener hasta millones de filas. Además, MySQL tiene un avanzado aunque no estandarizado sistema de seguridad/privilegios. La función básica de estos privilegios es la de asociar un nombre usuario a unos privilegios de *select*, *insert*, *update* y *delete* en una base de datos.

El sistema de privilegios de MySQL asegura que cada usuario pueda realizar los cambios que se supone debe hacer. El sistema administra unos privilegios de acuerdo con qué usuario se conecte, de qué máquina lo esté haciendo y a qué tabla este intentando acceder.

5.3. API C de acceso

El código de este API se distribuye junto con el servidor de MySQL. Está incluido en la librería `mysqlclient` y permite que programas en C accedan a la base de datos [7].

La conexión que se establece entre el programa y el servidor de la base de datos depende de si el servidor de la base de datos está en la misma máquina en la que se ejecuta el programa en C, o si, por el contrario, está en una máquina remota.

En la primera situación la arquitectura de la conexión es la que aparece en Fig 3. En el caso de que la base de datos esté en una máquina remota la arquitectura de la conexión tendría el aspecto que se muestra en Fig 4.

Para que el programa en C se conecte al servidor de la base de datos se debe crear una conexión TCP (Transmission Control Protocol).

5.4. API Java de acceso

El código de este API se distribuye aparte de la distribución de MySQL. Hay unos cuantos drivers distintos en el mercado [8].

El driver de JDBC es un API Java mediante el cual se comunican los programas en Java y el servidor de la base de datos que hay en una máquina remota o local. Al driver de Java le resulta indiferente que el programa Java y la base de datos estén en una misma máquina o estén conectadas de forma remota. En ambos casos establece una conexión TCP.

La arquitectura de la conexión que se establece entre el programa Java y el servidor de la base de datos tiene la estructura mostrada en Fig 5.

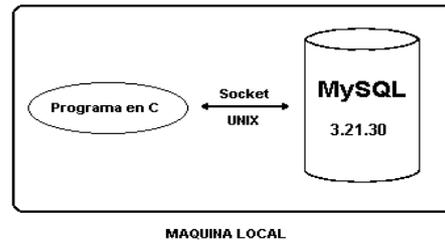


Fig 3. Arquitectura de la conexión dentro de la misma máquina.

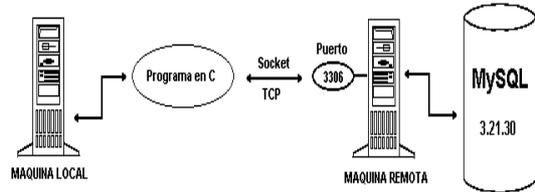


Fig 4. Arquitectura de interconexión entre el programa C y la base de datos.

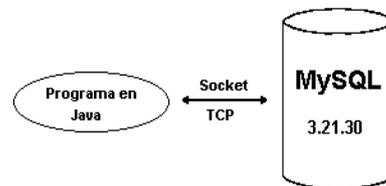


Fig 5. Arquitectura de la conexión TCP entre el programa Java y el servidor de la base de datos.

Para establecer la conexión primeramente se debe cargar el driver en la máquina virtual de Java donde esté situado el programa. Una vez cargado el driver correctamente, se establece la conexión TCP entre el programa Java y el servidor de la base de datos donde se le pasan los parámetros de máquina, puerto, usuario y password de usuario.

5.5. Diseño de la base de datos.

La base de datos que se ha generado para la supervisión total de los cursos se denomina *courses*.

Hay dos tipos de tabla, unas las que hacen referencia a la información referente a los alumnos y el resto son las que hacen referencia a los cursos y sus contenidos. Ambos tipos de tablas se interconectan mediante la tabla de *historicos* y *matriculas* que se detallarán más adelante. En Fig 6. se puede ver la relación de campos entre las diferentes tablas.

5.6. Tablas de la base de datos

En las siguientes subsecciones se van a explicar la estructura y razón de ser de las distintas tablas.

5.6.1. Tabla de alumnos

En esta primera tabla se mantiene un registro de la información personal de los alumnos matriculados en cualquiera de los cursos ofrecidos. Se utiliza para las siguientes funciones:

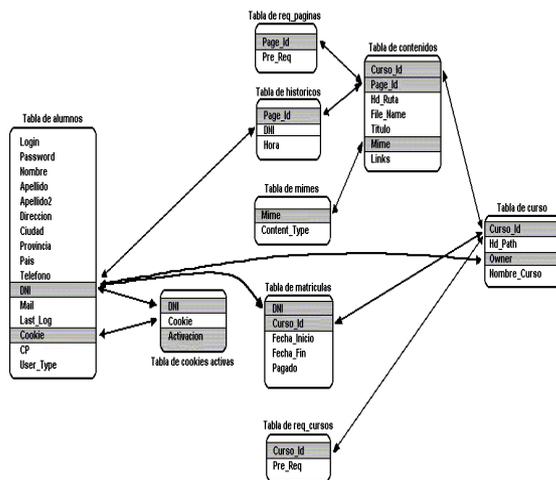


Fig 6. Relación entre los diferentes campos de las tablas que conforman la base de datos.

- Autenticar accesos.
- Almacenar datos personales.
- Mantener registro de las cookies activas.

El DNI se utiliza para referenciar a los alumnos aquí definidos en las tablas de *historicos*, *cookies activas*, *matriculas* y *curso* con el nombre de Owner. Esta tabla se modifica cuando un alumno se inscribe en algún curso. Para ello debe rellenar el formulario que tiene en la página Web principal del servidor de cursos.

Si una persona intenta acceder a un curso y no está dado de alta en esta tabla no podrá acceder al curso.

Los campos *Login* y *Password* se necesitan para la autenticación de usuarios. El campo *Password* se guarda encriptada. Por otro lado, el campo *Cookie* vale para mantener información del estado de la conexión ya que cada conexión tendrá un valor distinto.

5.6.2. Tabla de matriculas

En esta tabla se registran todos los alumnos matriculados en cada curso. Se usa para que el alumno sólo pueda acceder a los cursos en los que está matriculado.

5.6.3. Tabla de curso

En dicha tabla se mantiene un registro de todos los cursos ofrecidos. Se utiliza para registrar al dueño de un curso y así ser él el único con permisos para actualizar, modificar requisitos, borrar o introducir nuevos contenidos

5.6.4. Tabla de cookies activas

Aquí se mantiene un registro de todas las conexiones válidas que hay activas en ese instante. Se utiliza para que el servidor controle que usuarios están conectados en cada momento y mantener el estado.

El campo *Activacion* es necesario para poder eliminar la cookie activa cuando se crea que ya ha caducado o el alumno a cerrado la conexión. Por precaución se mantiene la cookie en la tabla durante un día entero.

5.6.5. Tabla de contenidos

En esta tabla se guarda registro de la información referente a cada fichero que forma parte de los diferentes cursos. Se utiliza en:

- El acceso a las páginas de los cursos por el supervisor de cursos.
- La inserción, actualización, borrado y modificación de requisitos de las páginas.

El campo *Page_Id* es el identificador del fichero en la base de datos. Este identificador es asignado para poder realizar la tarea de renombrado de páginas y así conseguir un mapeo entre el “nombre original del fichero” y “número asignado al fichero”.

5.6.6. Tabla de req_cursos

En esta tabla se mantiene un registro de los requisitos de este curso. Se utiliza para comprobar si el alumno ha cumplido con los requisitos específicos de cada curso.

5.6.7. Tabla de req_paginas

En esta otra tabla se tiene un registro de los requisitos de este curso. Se utiliza para comprobar si el alumno ha cumplido con los requisitos específicos de cada curso.

5.6.8. Tabla de mimes

En esta tabla se registran las Mimes que procesa el servidor Web. Se utiliza para enviar al cliente el tipo de ficheros que debe esperar.

El campo *Mime* da información de la extensión de los ficheros mientras que el campo *Content_Type* informa sobre el Content-Type asociado a ese tipo de ficheros en nuestro servidor de Web.

5.6.9. Tabla de historicos

Por último, en esta tabla se mantiene un registro de quién y cuándo han accedido a cada página de los distintos cursos que se ofrecen. Se utiliza básicamente para generar las gráficas con las estadísticas de cada curso, alumno, etc.

6. Proveedor de servicio

Esta parte del sistema es en la que más elementos del sistema se ven involucrados.

El proveedor del sistema está constituido por los siguientes componentes:

- Un servidor Web.
- Cookies como sistema para llevar el estado de la conexión cliente/servidor.

- Un programa CGI principal y dos CGIs auxiliares.
- El protocolo SSL para la encriptación de los datos.

6.1. El servidor Web

Las funciones que se requiere que realice el servidor Web son:

- Servir eficientemente las peticiones para poder atender el mayor número de peticiones concurrentes con el menor uso de hardware.
- Ser multitarea para así poder operar con más de una petición a la vez.
- Responder a errores que ocurran con respuestas que tengan sentido en el contexto de lo ocurrido. Por ejemplo, si un cliente realiza una petición de una página que no existe, el servidor deberá enviar un código "404", que viene definido en la especificación de HTTP como "página no existente". [9]
- Ofrecer diferentes formatos. Desde un punto de vista más técnico, un cliente puede preferir imágenes JPG frente a imágenes GIF, o TIFF. O puede preferir texto en formato vdi frente a formato PostScript.

6.2. Servidor con o sin estado: Cookies

La interacción entre el usuario y el servidor de Web se basa en un protocolo HTTP de peticiones/respuestas sin estado. No se mantiene ningún contexto entre las diferentes peticiones realizadas por el mismo usuario.

Para el sistema de supervisión de cursos es fundamental el hecho de guardar información sobre qué alumno está conectado en cada momento e información de esta conexión. Para mantener este estado de la conexión hay tres métodos factibles:

- Cookies.
- Identificador de sesión del protocolo SSL (Secure Sockets Layer).
- Métodos GET/POST de los CGIs.

La opción elegida ha sido la de las cookies por sencillez, debido a que el servidor Apache tiene un módulo que administra las cookies. Las cookies son simplemente cabeceras de HTTP que consisten en strings de texto que se introducen en la memoria del navegador [10]. En el caso de Apache, este string es un número que lo genera el servidor Web para cada conexión y la devuelve con cada respuesta de petición. El cliente envía al servidor este identificador cada vez que realiza una petición. De este modo, podemos discernir de quién proviene cada una de las peticiones.

6.3. CGIs

El CGI es un interfaz con el servidor Web que permite extender las funcionalidades del servidor

[11]. Utilizando CGIs, se puede interactuar con el usuario que accede al servidor.

Como CGI es un interfaz común (*common interface*) nadie está restringido a un lenguaje de programación específico. Se puede utilizar cualquier lenguaje de programación que permita imprimir en la salida estándar, leer de la entrada estándar y leer las variables de entorno. La mayoría de los lenguajes de programación cumplen dichos requisitos por lo que podemos programar casi en cualquier lenguaje de programación existente. Para elegir un lenguaje específico hay que tener en cuenta las prioridades de cada administrador del servidor. Hay que sopesar la mayor velocidad y eficiencia de algunos programas frente a la facilidad de programación de otros. Los lenguajes de programación de CGI más habituales son C y Perl aunque también están Tcl/Tk y por qué no Java.

Para el sistema lo que realmente es necesario es que el CGI sea lo más rápido posible por lo que se elegirá el lenguaje de programación por su velocidad. Para comprobar la velocidad de cada lenguaje se va a realizar una prueba en la que se volcará en pantalla un fichero en bloques de tamaño variable. Los resultados obtenidos en dicha prueba son los mostrados en la Fig 8.

En todo momento el más rápido ha sido el lenguaje de programación C por lo que se escoge C como lenguaje de programación.

Para poner en marcha el sistema de supervisión de cursos se ha tenido que realizar tres CGIs independientes. Cada uno es el responsable de llevar a cabo distintas tareas:

- `inscribir.cgi`: Inscripción de los alumnos en los distintos cursos.
- `login.cgi`: Creación de una página dinámica de acceso a los distintos cursos en los que este matriculado el alumno.
- `curso.cgi`: Supervisión de las peticiones de los alumnos.

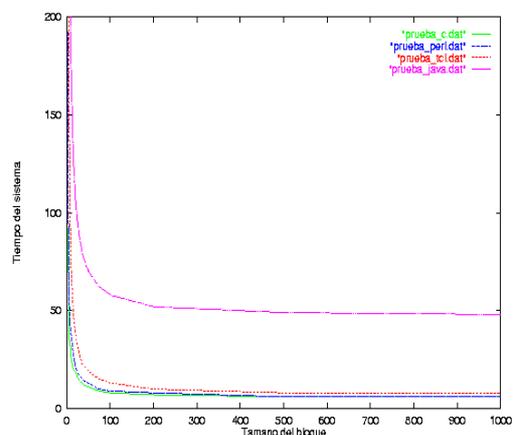


Fig. 8 Prueba de velocidad de acceso a ficheros mediante los distintos lenguajes de programación.

En las siguientes subsecciones se va a realizar un estudio más detallado de la labor y funcionamiento de los tres CGIs.

6.3.1. Inscripción de alumnos

Con este CGI lo que se busca es tener un interfaz cómodo y accesible a cualquiera para insertar entradas de alumnos. Por ello para que un alumno sea dado de alta en un curso debe rellenar un formulario.

6.3.2. Página dinámica con los cursos matriculados

La finalidad de este CGI es la de tener un modo de generar una página de forma dinámica en la que aparezca en cada caso el listado de los cursos en los cuales está matriculado el alumno en cuestión. Una vez generada la página el alumno ya puede acceder al contenido de los cursos.

6.3.3. Supervisor del curso

El objetivo de este CGI es el de controlar y supervisar los accesos al curso por parte de cada uno de los alumnos matriculados. La petición de una página sigue unos pasos muy específicos por el sistema de supervisión del curso (Véase Fig. 9).

6.4. SSL

El protocolo SSL [12] intenta ofrecer un mecanismo práctico, a nivel de aplicación, para la comunicación segura a través de Internet entre el cliente y el servidor.

Es una característica del servidor Web necesaria en el caso de comercio electrónico, acceso a datos confidenciales, etc.

En nuestro sistema se ha incluido SSL al servidor Apache por posibles necesidades de transacciones económicas como pago de los distintos curso. Con SSL, lo que se busca es tener un cierto nivel de confidencialidad de la información de los cursos.

El protocolo SSL ofrece una conexión segura, encriptación de datos, certificación de servidor y cliente.

Para que una conexión se considere segura, dicha conexión debe cumplir tres propiedades:

- La conexión es privada.
- La identidad se puede autentificar usando criptografía asimétrica o de clave pública.
- La conexión es fidedigna.

El protocolo SSL carga al servidor con la encriptación y eso supone un gasto de recursos y un tiempo de procesado. Lo que se busca con los servidores *https* es que la información enviada por Internet esté segura de ataques pero el cliente a su vez busca que el hecho de que la información sea segura no le suponga a él un gran retardo en el envío de la información.

Ante esta situación, se debe demostrar de una forma muy sencilla que el cliente no va a ver ralentizado su servicio de páginas. Para ello se ha realizado un experimento en la que se hacen peticiones de ficheros de tamaño variable, desde 1 Kbyte a 100 Kbytes, tanto al servidor de *http* como al de *https* y se ve la diferencia de tiempos de espera del servicio en el lado del cliente (Véase Fig 10.). Ambos servidores corren sobre la misma máquina y el cliente va a ser una máquina remota. Para que las medidas tomadas no fuesen falseadas por el estado de la red, se realizaron simultáneamente las peticiones a un servidor y a otro desde dos clientes distintos pero con el mismo sistema operativo y misma configuración hardware.

Como se observa en la figura, el servidor Web con el protocolo de SSL es más. En el lado del cliente la diferencia en el tiempo de servicio entre los dos servicios es más notable cuanto mayor es el fichero, hasta un 30% en el caso de estar transmitiendo ficheros de 100 K. Este retardo es bastante menos perceptible en la transmisión de ficheros de 4 a 20 K, un 10%. Por las características del tamaño de ficheros que se manejan se puede concluir que los servidores que sirven con *https* sufren poca penalización en el rendimiento del sistema frente a los que lo hacen con *http*. También se observa que a medida que se aumenta el tamaño del fichero enviado se incrementa la varianza del tiempo de servicio. Es un resultado predecible ya que cuanto mayor sea el fichero, su envío se verá mayormente afectado por el estado de la red y por la carga del servidor de ficheros.

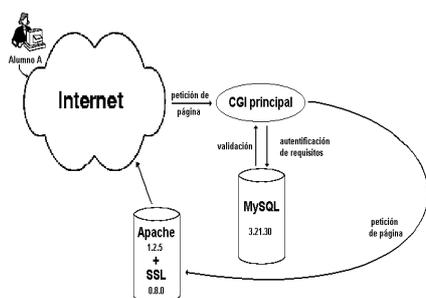


Fig 9. Evolución de la petición en el sistema de supervisión de cursos.

7. Generador de curso

7.1. Introducción

Esta aplicación es la encargada de automatizar el proceso de introducción de los contenidos que componen el curso en la base de datos de forma que sea más sencilla y utilizable por personal no experimentado. Con este módulo se intenta evitar el arduo trabajo de insertar los nombres, la ruta, los links a los que hace referencia, etc. de los distintos ficheros que conforman un curso. Está basado en un

manejo transparente de la base de datos y de los ficheros.

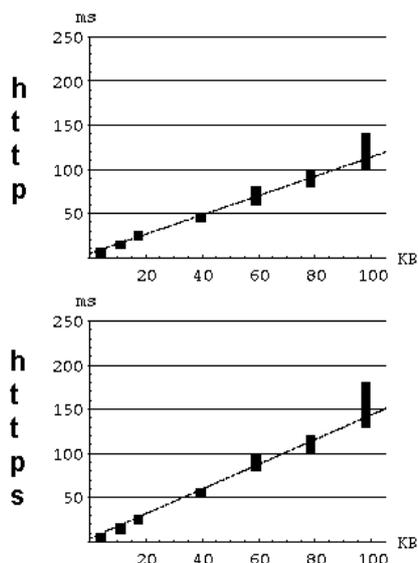


Fig 10. Comparativa de tiempos de espera de servicio del servidor http y https.

7.2. Arquitectura del generador

La aplicación debe interactuar con el servidor de la base de datos y con el servidor de Web.

El generador de curso debe estar en la misma máquina en la que está instalado el servidor de Web, en cambio la base de datos puede acceder a ella remotamente.

El generador de cursos interactúa con la base de datos de forma que envía peticiones para insertar nuevas entradas en alguna de sus tablas, borrar entradas obsoletas, modificar alguno de los campos de una entrada específica o actualizar alguna entrada realizada anteriormente.

7.3. Funcionamiento básico

El generador de cursos pretende permitir al administrador de curso diferentes funcionalidades como son: crear un nuevo curso, insertar una página a un curso existente, actualizar una página de un curso específico, eliminar una página de un curso dado, modificar o imponer requisitos a ciertas páginas del curso y por último visualizar las estadísticas referente a los cursos o alumnos del sistema.

El método de creación de cursos es el más importante de la aplicación global porque es la que introduce todos los ficheros que conforman un curso en la base de datos. En el proceso de creación de cursos se realiza una conversión de nombres de ficheros a identificadores numéricos y tratamiento de ficheros para cambiar los links a ficheros del curso. La conversión del nombre de fichero a identificador numérico se realiza para optimizar la búsquedas de la referencia de ficheros, ya que resulta más rápido buscar la referencia del fichero a través de un número que a través de un string, y para llevar un control más estricto de los contenidos del curso.

El sistema da de alta automáticamente el curso para que pueda ser accedido por los alumnos matriculados en él.

8. Conclusiones

Se ha presentado una herramienta de generación de cursos con la que la labor de publicación de contenidos en la red resulta sencilla y transparente para la persona que desee proveer información a través de Internet. Con esto se consigue que la persona encargada de introducir los cursos no tenga que tener conocimientos profundos del sistema. La introducción del curso es por lo tanto una tarea sencilla y se realiza de una forma visual e interactiva.

El sistema presentado es una herramienta multiplataforma debido a que todos los componentes que lo conforman son utilizables en todos los sistemas operativos. Además, proporciona un acceso supervisado al curso. De esta forma, el alumno no podrá acceder a los ficheros que no deba o no esté autorizado.

Cada alumno tiene en su mano herramientas conocidas, correo electrónico o comunicación online, para exponer sus dudas al profesor del curso que está siguiendo.

El profesor, por otra parte, podrá monitorizar la evolución de los alumno y obtener información sobre cuáles son las páginas más accedidas con lo que obtendrá conclusiones del éxito de los contenidos publicados en la red.

Referencias

- [1] P. J. E. Dekker, D. I. Beaver. "Report on ECDS: An Interactive Course on the Internet". University of Amsterdam.
- [2] B. Stiller, T. Walter. "Télépoly – An ATM-based Teleteaching Scenario". ETH Zürich, Institut für Technische Informatik und Kommunikation-netze.
- [3] P. Kutschera. "Combining Database technology with the World Wide Web for tele-teaching environments". University of Stuttgart.
- [4] http://selva.dit.upm.es/~isabel/user_man/
- [5] <http://www.mbone.com/>
- [6] Detron HB, Monty Program KB. "MySQL Reference Manual". 1998.
- [7] http://mysql.turbolift.com/mysql/3.21_C_API/
- [8] ftp://ftp.tcx.se/pub/Contrib/Old_versions/
- [9] RFC: Hypertext Transfer Protocol -- HTTP/1.0. <ftp://ftp.rediris.es/docs/rfc/19xx/1945>.
- [10] <http://cookiecentral.com/faq/>
- [11] E. E. Kim. "CGI Developer's Guide". Editorial Sams net. 1996.
- [12] D. Wagner, B. Schneier. "Analysis of the SSL 3.0 Protocol". 1996.
- [13] L. Lemay. "Teach yourself Java in 21 day". Editorial Sams net. 1996.