



Universidad
Carlos III de Madrid
www.uc3m.es

RESUMEN-TRABAJO FIN DE GRADO

ACCESO WEB CONTROLADO AL EMULADOR DEL MAEMO

Autor: César Zurita Díaz

Titulación: Grado en Ingeniería Telemática

Tutor: Pablo Basanta Val

Co-tutora: Marisol García Valls

Fecha: 15 / 05 / 2013

Índice de Contenidos

1. Introducción.....	3
2. Estado del arte.....	3
2.1. Maemo SDK.....	3
2.2. VirtualBox.....	3
2.3. Apache Tomcat.....	4
2.4. Servlets.....	4
2.5. JSP.....	4
2.6. CSS.....	5
2.7. JavaScript.....	5
2.8. LDAP.....	5
2.9. FileUpload.....	5
3. Diseño de la solución.....	6
3.1. Cliente web.....	6
3.1.1. Modo autenticación.....	6
3.1.2. Modo consola.....	7
3.2. Servidor web.....	7
4. Validación y pruebas de rendimiento.....	7
5. Conclusiones.....	10

1. Introducción.

El proyecto consiste en desarrollar un conjunto de mecanismos para hacer accesible vía web el emulador del Maemo, con el fin educativo de crear una aplicación para la asignatura Arquitectura de Sistemas, impartida en los distintos grados de Ingeniería en Telecomunicaciones: Grado en Ingeniería Telemática, Grado en Ingeniería de Sistemas de Comunicaciones y Grado en Ingeniería de Sistemas Audiovisuales.

La idea principal consiste en cambiar el paradigma de ejecución en local de una imagen virtual de gran dimensión por una arquitectura multinivel (*multi-tier*), donde se ofrece una interfaz ligera en los clientes (interfaz web) que realiza las peticiones en remoto a los servicios que disponen de la lógica que se desea ejecutar.

Es decir, en la ejecución local la realización de la práctica requeriría que cada alumno tuviera una imagen propia instalada en su PC del laboratorio. Dicha imagen ocupa alrededor de 3GB, por lo que cada alumno debería descargarla previamente a la realización de la práctica. Con esta motivación, se propuso la idea de realizar una aplicación que pudiera acceder vía web a la máquina virtual.

Para este objetivo se analiza en profundidad la herramienta diseñada realizando pruebas de rendimiento para conocer los límites de dicho sistema de en un entorno real.

2. Estado del arte.

Para poder utilizar la aplicación del Maemo, es necesario tener un conocimiento más profundo acerca de las distintas tecnologías que la forman.

2.1. Maemo SDK.

Es el entorno de desarrollo para Maemo, el cual se puede instalar en ordenadores Linux y también se puede descargar como máquina virtual.

El SDK crea un entorno Maemo en el ordenador Linux, denominado ScratchBox, el cual permite realizar la compilación cruzada para el dispositivo.

Maemo SDK+ proporciona un entorno de desarrollo cruzado para dispositivos basados en Maemo de Nokia.

2.2. VirtualBox.

VirtualBox es un software de virtualización desarrollado por Oracle Corporation como parte de su familia de productos de virtualización.

Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como “sistemas invitados”, dentro de otro sistema operativo “anfitrión”, cada uno con su propio ambiente virtual. Por ejemplo, se podrían instalar diferentes distribuciones de GNU/Linux en VirtualBox instalado en Windows XP o viceversa.

Entre los sistemas operativos soportados (en modo anfitrión) se encuentran GNU/Linux, Mac OS X, OS/2 Warp, Windows, y Solaris/OpenSolaris, y dentro de éstos es posible virtualizar los sistemas operativos FreeBSD, GNU/Linux, OpenBSD, OS/2 Warp, Windows, Solaris, MS-DOS y muchos otros.

2.3. Apache Tomcat.

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets. Tomcat implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Sun Microsystems.

Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor Web Apache.

Tomcat puede funcionar como servidor Web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

La totalidad del proyecto está incluido en este servidor web, por lo que son necesarios conocimientos sobre configuración, análisis y gestión de Tomcat.

2.4. Servlets.

Los servlets son componentes de una aplicación web que se ejecutan en el servidor. El servidor web los carga para manejar peticiones del cliente y están escritos en código Java.

En el proyecto son necesarios servlets para: la portada de la aplicación, la autenticación del alumno en la base de datos de la universidad, el procesamiento de comandos en el lado del servidor, la subida de archivos a la aplicación y para el cierre de sesión.

2.5. JSP.

Los JSPs son una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

En la aplicación es necesario un JSP que se mueva del modo autenticación al modo consola dependiendo de si el alumno está autenticado.

2.6. CSS.

Las hojas de estilo en cascada hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL.

En la aplicación su uso más importante es dar formato al terminal web para emular a la consola de Linux. La correcta interacción entre las distintas entidades (servlets, JSP y CSS) hace posible el funcionamiento de la aplicación.

2.7. JavaScript.

JavaScript es un lenguaje de programación interpretado utilizado principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador. Existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS).

En la aplicación es utilizada mediante funciones dentro del JSP, para permitir comprobaciones en el lado del cliente y evitar el gasto de recursos innecesarios.

2.8. LDAP.

LDAP (Lightweight Directory Access Protocol) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también se considera una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

En el proyecto se utiliza el directorio de la Universidad Carlos III para que los alumnos puedan autenticarse en la aplicación.

2.9. FileUpload.

FileUpload es una librería que permite subir archivos de forma robusta a los servlets y aplicaciones web.

En la aplicación es utilizada para que el usuario suba sus propios ficheros en formato C o ZIP (no soporta otros).

3. Diseño de la solución.

El diseño de la aplicación web está dividido en dos grandes bloques: cliente web y el servidor web.

3.1. Cliente web.

El cliente web es la parte desde la que el usuario interactúa con la aplicación e incluye campos que contienen la información necesaria para la configuración de la posterior validación. Está compuesto por una página JSP, que incluye la presentación con código HTML, una hoja de estilo CSS que asocia los distintos componentes de manera ordenada y un servlet que gestiona las llamadas a la página y los datos que se envían desde ella.

En esta aplicación existen dos vistas desde el punto de vista del usuario. Es necesario autenticarse correctamente para poder acceder desde el modo autenticación al modo consola.

3.1.1. Modo autenticación.

El cual consta únicamente del bloque autenticación, donde el usuario deberá rellenar los campos para autenticarse. Si la autenticación se realiza correctamente se accederá al modo consola.



maem  nline

AUTENTICACION

NIA	Password
<input type="text"/>	<input type="password"/>

Validar

Bienvenido a la App de Maemo Online.

****Introduce tu NIA y pass de Campus Global para entrar.****

3.1.2. Modo consola.

Existen varios bloques dentro del modo consola: la consola de comandos, el directorio actual de trabajo (*pwd*), el bloque de subida de archivos y el bloque de usuario y salida (*logout*).



3.2. Servidor web.

El servidor web es el encargado de gestionar los comandos enviados por el cliente. Su función es, básicamente, desde un punto de vista menos técnico, recibir un comando, procesarlo, copiar la respuesta que generaría el terminal de bash y enviársela al cliente para que la imprima por pantalla.

Consta de un servlet de recepción de los comandos, el cual debe detectar si el comando introducido ha sido *cd* para cambiar el directorio actual de trabajo y enviarle al cliente el nuevo *pwd*.

4. Validación y pruebas de rendimiento.

La práctica sobre la cual se ha desarrollado este proyecto es "El emulador de Maemo" de la asignatura Arquitectura de Sistemas y una vez autenticados en la aplicación los alumnos deben realizarla.

El tiempo de validación de la aplicación depende de los comandos utilizados. Es necesario comprobar los tiempos de los distintos comandos, para estudiar la viabilidad de la aplicación en esta práctica.

En las siguientes tablas se pueden observar los distintos comandos de la práctica, el tiempo real, de usuario y de sistema, comprobados introduciendo el comando "time" antes de cada uno de ellos:

Comando	Tiempo real	Tiempo de usuario	Tiempo de sistema
<code>mkdir project</code>	0.052s	0.004s	0.012s
<code>sb2 gcc helloworld.c -o helloworld</code>	3.373s	0.596s	1.468s
<code>ls</code>	0.012s	0.004s	0.012s
<code>file helloworld</code>	0.075s	0.004s	0.012s
<code>sb2 -e ./helloworld</code>	1.747s	0.388s	1.052s
<code>./helloworld</code>	0.129s	0.004s	0.004s
<code>sb2 ./helloworld</code>	1.753s	0.532s	1.096s
<code>sb2 -e ls /usr/bin/helloworld</code>	1.715s	0.452s	1.040s
<code>sb2 -eR cp ./helloworld /usr/bin/helloworld</code>	1.893s	0.396s	1.240s
<code>sb2 -e /usr/bin/helloworld</code>	1.615s	0.488s	0.988s

Comando	Tiempo real	Tiempo de usuario	Tiempo de sistema
<code>sb2 -e ls /usr/bin/helloworld</code>	2.015s	0.412s	1.108s
<code>/usr/bin/helloworld</code>	0.005s	0.008s	0.000s
<code>sb2 -eR rm /usr/bin/helloworld</code>	1.739s	0.416s	1.136s
<code>mkdir usr && mkdir usr/bin</code>	0.013s	0.004s	0.012s
<code>cp helloworld usr/bin/</code>	0.027s	0.004s	0.012s
<code>tar cvfz helloworld-1.0.0.tar.gz usr/</code>	0.065s	0.004s	0.028s
<code>fakeroot alien helloworld-1.0.0.tar.gz</code>	6.017s	1.400s	2.660s
<code>sb2 -eR dpkg -i ./helloworld_1.0.0-2_all.deb</code>	10.299s	1.756s	2.336s
<code>sb2 -e /usr/bin/helloworld</code>	1.696s	0.460s	1.044s
<code>sb2 -eR apt-get remove helloworld</code>	>4minutos (Comando no viable vía web)		

Al introducir el último comando vía web el tiempo de respuesta en la aplicación supera los 4 minutos, por lo que se desaconseja totalmente su utilización.

5. Conclusiones.

El objetivo del proyecto era la creación de una aplicación web para la asignatura de Arquitectura de Sistemas. Para ello se trabajó con el fin de conseguir una maqueta que cumpliera las necesidades básicas principales de la práctica “El emulador de Maemo”.

En este aspecto, se ha conseguido la maqueta de una aplicación funcional y real con muchas posibilidades en el campo de la docencia.

El desarrollo del trabajo ha sido similar al planteamiento inicial del proyecto aunque en ocasiones se han tenido que estudiar otras soluciones ante problemas imprevistos, tales como, descartar la viabilidad de un terminal web, descartar el uso de la base de datos MySQL y los problemas del comando `cd` en Bash.

Finalmente y teniendo en cuenta los resultados de las pruebas realizadas, se ha llegado a la conclusión de que la aplicación es viable y que se adapta completamente al entorno docente para la cual ha sido ideada.

Bien es cierto, que existen una serie de ampliaciones y posibilidades que no están en la versión final de la aplicación pero que, en un futuro, sería muy interesante poder estudiar incorporarlas:

- **Dar a la aplicación la función multiusuario.**
- **Posibilidad de subir la aplicación a la nube.**
- **Exportar el cliente fuera de la máquina virtual.**
- **Incorporar tecnología AJAX en el lado del cliente (*client-side*).**

Estas son las cuatro principales mejoras por las que sería interesante continuar un estudio acerca de la posibilidad de incorporarlas.