

**IX Jornadas de Enseñanza Universitaria de la Informática
Jenui 2003**

Thomson Paraninfo, S.A., Madrid, 2003.

ISBN: 84-283-2845-5

Materia: Informática 681.3
Enseñanza superior 378

Formato: 240x170

Páginas 650

© Los autores. 2003

Diseño de portada: Nuria Hurtado Rodríguez

Maquetación: Inmaculada Medina Buló y Francisco Palomo Lozano

Primera edición: julio de 2003

I.S.B.N.: 84-283-2845-5

Depósito legal: M. 27.532-2003

Imprime: Closas-Orcoyen, S. L.

Polígono Igarza. Paracuellos de Jarama (Madrid)

CONTENIDOS

Ponencias

Tema estratégico: Enseñanza y uso de métodos formales en los estudios universitarios de informática 3

Métodos formales en programación: ¿desmitificar para motivar? 5
Inés Jacob
Universidad de Deusto

Verificación de Autómatas y Gramáticas 13
Jairo Rocha
Universitat de les Illes Balears

Tema estratégico: Educar/formar profesionales informáticos 19

Situaciones y Problemas: cómo preparar a los alumnos para lo que se les avecina 21
José Miró
Universitat de les Illes Balears

El futuro de la formación de los profesionales informáticos 29
Javier Oliver
Universidad de Deusto

Repercusiones del futuro espacio europeo de educación superior sobre las titulaciones universitarias de Informática en España 37
Fermín Sánchez, María-Ribera Sancho
Universitat Politècnica de Catalunya

Enseñando Inteligencia Emocional a Ingenieros en Informática 45
Esperanza Marcos, José María Cavero Barca
Universidad Rey Juan Carlos

Retos en la formación de profesionales de las tecnologías de la información: perfiles y aplicación en la formación universitaria	53
Luis Fernández Sanz	
<i>Universidad Europea de Madrid</i>	
Propuesta docente sobre Sistemas de Información Geográfica	61
Michael Gould, José Poveda	
<i>Universitat Jaume I</i>	
Arquitectura de ordenadores	69
Prácticas Experimentales de Memorias Cache	71
Julio Sahuquillo, José Flich, Jorge Real	
<i>Universidad Politécnica de Valencia</i>	
Prácticas de Diseño de Sistemas de Memoria	79
José Flich, Jorge Real, Julio Sahuquillo	
<i>Universidad Politécnica de Valencia</i>	
Nueva aplicación didáctica para electrónica digital	87
Javier García Zubía	
<i>Universidad de Deusto</i>	
Optimización de una implementación JPEG teniendo en cuenta la arquitectura actual de los procesadores	95
J.A. Padilla	
<i>Logic Factory</i>	
M. Anguita, F.J. Fernández, A.F. Díaz, A. Cañas, A. Prieto	
<i>Universidad de Granada</i>	
De la pizarra al circuito: una metodología para mejorar el aprendizaje en las prácticas de diseño lógico	103
José Luis Poza, Juan Carlos Cano, Juan Luis Posadas	
<i>Universidad Politécnica de Valencia</i>	
Motivando a los estudiantes en prácticas relacionadas con Estructura de los Computadores	111
F.J. Fernández, A. Cañas, A.F. Díaz, M. Anguita, H. Pomares, A. Prieto	
<i>Universidad de Granada</i>	

Automatización de prácticas en entornos masificados	119
Antonio García Dopico, Santiago Rodríguez de la Fuente, Francisco Javier Rosales García <i>Universidad Politécnica de Madrid</i>	
Atención a la diversidad (género, discapacidad, proyección social)	127
Aspecto de Género y Enseñanza de la Informática en Alemania	129
Esther Ruiz Ben <i>Albert-Ludwigs-Universität Freiburg</i>	
Calidad y evaluación de la docencia	135
Estudio del rendimiento académico de una asignatura con relación a asignaturas de cursos anteriores.....	137
Luisa Zúnica, Pedro Blesa, Rosa Alcover, Jorge Más, José M. Valiente <i>Universidad Politécnica de Valencia</i>	
E-valueate: un modelo de autoevaluación para la mejora de la enseñanza y el aprendizaje	143
M ^a . Ángeles Díaz, Ana Belén Martínez, Miguel Riesco, Carlos Iglesias <i>Universidad de Oviedo</i>	
Directrices éticas y legislación informática	149
Gestión académica y protección de datos	151
Xavier Canaleta, David Vernet <i>Universidad Ramón Llull</i>	

Evaluación del alumnado	159
Algoritmo para la evaluación de exámenes tipo test en sistemas e-learning avanzados	161
R. Barchino, J. M. Gutiérrez, J. Macías, S. Otón <i>Universidad de Alcalá</i>	
Sistema para la (auto) evaluación de los alumnos a través de la Web	167
Sergio Luján-Mora, Iván Mingot Latorre <i>Universidad de Alicante</i>	
SAM: Sistema de Autoevaluación Multimedia	175
J.M. Antelm, R. Mollá, R. Vivó, V. Vidal, A. Robles, M.L. Gil <i>Universidad Politécnica de Valencia</i>	
Evaluación continuada a un coste razonable	183
Miguel Valero-García, Luis M. Díaz de Cerio <i>Universitat Politècnica de Catalunya</i>	
Una apuesta por la motivación al alumnado en las asignaturas de programación: el sistema de evaluación continuada	191
Marisa Durán, Andrés Caro, Pablo G. Rodríguez <i>Universidad de Extremadura</i>	
Informática en otras carreras	197
Propuesta para la enseñanza de Informática en titulaciones de Ingeniería Química	199
Ana Belén Moreno Díaz, Juan José Pantrigo, Rosalía Peña <i>Universidad Rey Juan Carlos</i>	
Experiencia del ISPJAE en la formación Informática de los Ingenieros Industriales.....	207
Mavis Lis Stuart Cárdenas, Diana Aguilera Reina, Miguel Ángel Díaz Martínez, Yadary Ortega González <i>Instituto Superior Politécnico José Antonio Echeverría</i>	

Informática para profesionales de la Geología: docencia, aprendizaje y práctica	215
María Vaquero Domínguez, Roberto Therón Sánchez <i>Universidad de Salamanca</i>	
El comercio electrónico en los estudios empresariales.....	223
Pedro L. Pérez Serrano, Lourdes Moreno Liso <i>Universidad de Extremadura</i>	
La informática en los estudios de Gestión y Administración Pública	231
Francisco Araque <i>Universidad de Granada</i> Juan José Gaitán <i>IP Learning e-ducativa</i> Vlasta Hlavickova <i>Universidad de Económicas de Praga</i> Ernesto Zianni <i>Universidad Nacional del Litoral</i>	
Informática teórica	239
La teoría de autómatas y lenguajes formales, en la práctica.....	241
José A. Troyano, Víctor J. Díaz, Fernando Enríquez de S., Javier Barroso <i>Universidad de Sevilla</i>	
Una alternativa docente a la Máquina de Turing.....	249
Rafael Morales, Gonzalo Ramos, Francisco J. Vico, Francisco Triguero <i>Universidad de Málaga</i>	

Ingeniería del software	255
Incorporando Extreme Programming como Metodología de Desarrollo en un Laboratorio de Sistemas de Información	257
Francisco Letelier, José Hilario Canós <i>Universidad Politécnica de Valencia</i>	
Inclusión de patrones de diseño en un plan de estudios de Ingeniería Técnica en Informática de Gestión	265
Carlos López Nozal, Raul Marticorena Sánchez, Judith Antolín Sendino, Ignacio Cruzado Nuño <i>Universidad de Burgos</i>	
Aplicación práctica de un proceso basado en UML	273
Jesús García Molina, Marcos Menárguez Tortosa, Joaquín Nicolás Ros <i>Universidad de Murcia</i>	
Elaboración de Documentos de Requisitos en Asignaturas de Ingeniería del Software	281
Francisco J. García Peñalvo, M ^a . N. Moreno García <i>Universidad de Salamanca</i> Amador Durán Toro <i>Universidad de Sevilla</i>	
Inteligencia artificial	289
Un enfoque metodológico para la docencia en Ingeniería del Conocimiento	291
Amparo Alonso Betanzos, Bertha Guijarro Berdiñas <i>Universidad de A Coruña</i> Adolfo Lozano Tello <i>Universidad de Extremadura</i>	
Conceptos básicos en la enseñanza de la Inteligencia Artificial: datos, información y conocimiento	299
Margaret Miró-Julià <i>Universitat de les Illes Balears</i>	

Métodos pedagógicos innovadores	307
Debate y foro en el aula como metodología docente: estudio comparativo de su aplicación en la asignatura Sistemas de Transmisión de Datos.....	309
José L. Poza, Alberto Bonastre, José Oliver <i>Universidad Politécnica de Valencia</i>	
La mensajería instantánea al servicio de la docencia	317
Vicente Galiano Ibarra, Katja Gilly de la Sierra, Alejandro Pomares Padilla <i>Universidad Miguel Hernández</i>	
Adaptación de la técnica de planificación del trabajo personal aplicada a la educación.....	323
Antonio de Amescua, Juan José Cuadrado, Luis García, María Isabel Sánchez <i>Universidad Carlos III de Madrid</i>	
El proceso de aprendizaje: herramienta para el desarrollo de competencias profesionales en primero de informática.....	331
Juan José Escribano Otero, Estrella Gómez Fernández, María Teresa Villalba de Benito, Manuel Ortega Ortiz de Apodaca <i>Universidad Europea</i>	
Estudio de un Sistema de Aula Virtual como Apoyo a la Docencia Presencial.....	339
María del Pilar Romay Rodríguez <i>Universidad Europea de Madrid</i> Carlos E. Cuesta Quintero <i>Universidad de Valladolid</i>	
Los concursos de programación como herramienta didáctica.....	349
Agustín Cernuda del Río, Daniel Gayo Avello <i>Universidad de Oviedo</i>	

Cómo motivar al alumnado entrelazando las asignaturas Programación Avanzada y Tecnología de Computadores	357
Pedro Pablo Garrido Abenza, Héctor Francisco Migallón Gomis <i>Universidad Miguel Hernández</i>	
Una revisión de métodos pedagógicos innovadores para la enseñanza de la programación	363
Mercedes Gómez Albarrán <i>Universidad Complutense de Madrid</i>	
Docencia de la Programación Orientada a Eventos.....	371
Carlos Rioja del Río <i>Universidad de Cádiz</i>	
JDESK: Simulador de Eventos Discreto Basado en Web	377
Inmaculada García García, Ramón Mollá Vayá <i>Universidad Politécnica de Valencia</i>	
Terminales sin disco basados en GNU/Linux para docencia	385
David Úbeda, Katja Gilly, Salvador Alcaraz, Guillermo Martínez <i>Universidad Miguel Hernández</i>	
Multimedia e informática gráfica	393
El proyecto MEIIGA	395
Pedro M. Latorre, Francisco Serón <i>Universidad de Zaragoza</i>	
Organización curricular y planes de estudio.....	403
Reflexiones y experiencias sobre la enseñanza de POO como único paradigma	405
Daniel Gayo Avello, Agustín Cernuda del Río, Juan Manuel Cueva Lovelle, Marián Díaz Fondón, M ^a . Pilar Almudena García Fuente, José Manuel Redondo López <i>Universidad de Oviedo</i>	

Las asignaturas de programación en las universidades españolas	413
Julia González, Alberto Gómez	
<i>Universidad de Extremadura</i>	
La Universidad Española: Contenidos sobre Programación en los primeros cursos de las Titulaciones en Informática	421
Pedro J. Clemente, Pedro L. Pérez	
<i>Universidad de Extremadura</i>	
Los estudios universitarios en la sociedad de la información y el conocimiento: una propuesta de verificación de cobertura de contenidos curriculares “ <i>ad-hoc</i> ” mínimos y metodología docente asociada	431
Ferrán Virgós Bel, Antoni Pérez Poch	
<i>Universitat Politècnica de Catalunya</i>	
Del propósito de la materia de compiladores en la formación del ingeniero informático	439
JosuKa Díaz Labrador, José M ^a . Sáenz Ruiz de Velasco	
<i>Universidad de Deusto</i>	
Programación, algoritmos y estructuras de datos	447
Representaciones gráficas y Mundos Virtuales infinitos en las Prácticas de <i>Programación Lógica y Funcional</i>	449
Jose Emilio Labra Gayo	
<i>Universidad de Oviedo</i>	
Notación formalizada para la representación de árboles de seguimiento de algoritmos en Prolog	457
Nieves Pavón Pulido, Omar Sánchez Pérez	
<i>Universidad de Huelva</i>	
Entorno web de desarrollo para el aprendizaje de paradigmas de programación	465
Juan Ramón Pérez Pérez, M ^a . del Puerto Paule Ruiz,	
Martín González Rodríguez, Ramón González Suárez	
<i>Universidad de Oviedo</i>	

Un enfoque para la enseñanza de la depuración de errores en las asignaturas de programación	473
Sergio Luján-Mora	
<i>Universidad de Alicante</i>	
Mejora de la comprensión de las estructuras de datos.....	481
Raquel Lacuesta, Karmelo Urzelai	
<i>Universidad de Zaragoza</i>	
Análisis híbrido: una propuesta práctica	489
Francisco Palomo Lozano, Inmaculada Medina Bulo	
<i>Universidad de Cádiz</i>	
Robótica e informática industrial	497
Docencia de la Planificación y Desarrollo de un Proyecto de Informática Industrial	499
Lorenzo Moreno, Evelio González, Jonay Toledo,	
Leopoldo Acosta, Alberto Hamilton, J. Albino Méndez,	
Sergio Hernández, Marta Sigut, Nicolás Marichal,	
Santiago Torres, Jesús F. Montañez	
<i>Universidad de La Laguna</i>	
Propuesta para la Integración de Prácticas de Laboratorio en Intensificaciones de la titulación de ITIS	507
Vicent Lorente, Silvia Terrasa, Salvador Petit,	
Alfons Crespo	
<i>Universidad Politécnica de Valencia</i>	
Sistemas distribuidos y paralelos	513
Domótica y Edificios Inteligentes en la Universidad de Alicante	515
Jorge Azorín, Andrés Fuster, Francisco Maciá,	
Francisco J. Ferrández	
<i>Universidad de Alicante</i>	

Complejidad Algorítmica: de la Teoría a la Práctica.....	523
I. Dorta, C. León, C. Rodríguez, G. Rodríguez, A. Rojas	
<i>Universidad de La Laguna</i>	
Sistemas operativos	531
La programación concurrente y el interbloqueo en la asignatura de Sistemas Operativos	533
Miguel Riesco Albizu, Marián Díaz Fondón	
<i>Universidad de Oviedo</i>	
Simulador didáctico de gestión de memoria con interfaz de autoaprendizaje basado en WWW.....	539
Félix Buendía, Julio Sahuquillo, Juan Carlos Cano	
<i>Universidad Politécnica de Valencia</i>	
Tecnologías de la información en la gestión empresarial	547
Gestión de Clientes en el marco de los Portales Corporativos. Ensayo de enseñanza interdepartamental en la EUEEZ. Integración de las visiones empresarial y tecnológica.....	549
Javier Gutiérrez, María Jesús Lapeña, Pilar Urquizu	
<i>Universidad de Zaragoza</i>	
Telemática	557
Experiencia en la aplicación de un entorno visual como apoyo a la docencia de laboratorios presenciales.....	559
Javier Macías, José Javier Martínez, José María Gutiérrez, Roberto Barchino	
<i>Universidad de Alcalá</i>	
Desarrollo de actividades en grupos coordinados sobre el modelado y simulación del proceso de transmisión de datos.....	567
José Oliver, Alberto Bonastre, José L. Poza	
<i>Universidad Politécnica de Valencia</i>	

Demostraciones

Desarrollo de un simulador de programación del microprocesador Intel 8085.....	577
Manuel Rodríguez Álvarez, Ángel Manuel Gómez García, Pedro Mesas García, José Ignacio Ruiz Núñez, Alberto Prieto <i>Universidad de Granada</i>	
Jlógica, un entorno de diseño de sistemas digitales.....	581
José A. Álvarez, V.J. Ruiz, J.F. Sanjuan, Javier Roca P., Carlos A. Bermejo <i>Universidad de Almería</i>	
Estructura de computadores, simuladores e Internet	585
Javier García Zubía, José María Sáenz Ruiz de Velasco <i>Universidad de Deusto</i>	
CGRAPHIC: una herramienta gráfica para la enseñanza de los fundamentos de la programación (usando C)	589
Antonio J. Fernández, Jesús Millán Sánchez <i>Universidad de Málaga</i>	
Una Aplicación Interactiva para Visualizar las Hipótesis Generadas por Algoritmos de Aprendizaje Computacional	593
Santiago David Villalba Bartolomé, Juan José Rodríguez Díez <i>Universidad de Burgos</i>	
La herramienta ArtEM: aritmética entera y modular	597
Alfonso Gutiérrez, Violeta Migallón, José Penadés <i>Universidad Miguel Hernández</i> Héctor Migallón <i>Universidad de Alicante</i>	
Uso de las referencias bibliográficas en la Ingeniería Informática.....	601
Francisco J. García Peñalvo, Roberto Therón Sánchez, Ana B. Gil González <i>Universidad de Salamanca</i>	

SldDraw: Un trazador de árboles SLD	605
Francisco Gutiérrez	
<i>Universidad de Málaga</i>	
M ^a . del Carmen de Castro	
<i>Universidad de Cádiz</i>	
Sigraf: SIMulador de GRAFos	609
Judith Antolín Sendino, María Ruiz Ruiz,	
Carlos Pardo Aguilar, Juan J. Rodríguez Díez	
<i>Universidad de Burgos</i>	
Autoevaluación a través de Internet por medio de test.....	613
Pedro A. Castillo, Alberto Prieto, Antonio Cañas,	
Beatriz Prieto	
<i>Universidad de Granada</i>	
Una componente “e-Learning” de Aprendizaje Colaborativo para el Proyecto IDEFIX.....	617
T. Hernán Sagástegui Ch., José E. Labra G.,	
Juan M. Cueva L., José M. Morales G., María E. Alva O.,	
Eduardo Valdés, Cecilia García	
<i>Universidad de Oviedo</i>	

Terminales sin disco basados en GNU/Linux para docencia

David Úbeda, Katja Gilly, Salvador Alcaraz, Guillermo Martínez

Dpto. de Física y Arquitectura de Computadores

Universidad Miguel Hernández

03202 Elche (Alicante)

{ubeda,katya,salcaraz,g.martinez}@umh.es

Resumen

El avance hardware y software de los últimos tiempos ha sido espectacular. El nuevo software docente que se utiliza en nuestros laboratorios, obliga a una actualización continua a nivel hardware de los computadores, incluso llegando al extremo de tener que sustituir equipos por otros más potentes.

Por otra parte, los nuevos planes de estudio, con una carga docente práctica muy elevada, han provocado un uso intensivo de los laboratorios de informática, lo que origina una degradación progresiva del rendimiento de las máquinas, y obliga al personal técnico a realizar trabajos de mantenimiento, con costes elevados.

Tradicionalmente, hemos utilizado software bajo licencia de uso, que aunque con las ventajas de uso docente, también incrementan los costes de instalación y mantenimiento de las aulas de informática.

En este proyecto, planteamos una solución alternativa a los tradicionales laboratorios de informática basados en una configuración de máquinas monousuario. Nuestra propuesta nos permite obtener unas ventajas en ahorro económico, tanto en costes de adquisición y mantenimiento hardware y software, como costes de mantenimiento de personal técnico, así como una mejora del uso, gestión y administración del laboratorio. La solución consiste en la utilización de terminales sin disco duro bajo sistema operativo GNU/Linux [1]. Paralelamente, proporcionamos un aprendizaje al alumno de dicho sistema operativo y de las herramientas de trabajo que lo complementan, y el cuál se está imponiendo en los entornos tanto universitarios como empresariales, afianzándose en las diferentes áreas de Informática, Telecomunicaciones e Ingeniería en general.

1. Introducción

Uno de los objetivos históricos en las grandes multinacionales del sector tecnológico ha sido intentar comercializar equipos informáticos sin disco duro, con muy poca memoria y con un procesador de baja potencia, administrados desde un servidor, dado que es muy casual que un usuario tipo, sin unos conocimientos extensos en informática, consiga aprovechar el máximo rendimiento que le proporciona su máquina.

Estas empresas, incluso invirtieron un capital importante apostando por tal proyecto y algunas de ellas obtuvieron grandes pérdidas económicas, sin citar la gran pérdida de esfuerzo humano. Una de las primeras a comercializar dicho producto fue IBM, que tuvo que abandonar la idea rápidamente al ver como las ventas de este producto no aumentaban tan rápido como ellos esperaban.

Las principales causas para que tal proyecto no se llevara a cabo satisfactoriamente, fueron principalmente:

- La bajada de precios de los ordenadores personales (PC), apareciendo máquinas clónicas mucho más económicas que las que fabricaban gigantes como IBM.
- Internet por aquella época no se encontraba tan extendido como lo está en la actualidad.
- Un servidor potente, capaz de soportar algunas decenas de clientes terminales, era inalcanzable en cuanto a coste se refiere.
- Los ordenadores clónicos siguieron su descenso de precio hasta equipararse a los terminales sin disco, por lo que los compradores se decantaban por los primeros.

Aprovechando el fuerte crecimiento de Internet durante los años 1996 y 1997, varias fueron las empresas que retomaron de nuevo el proyecto de los terminales, creyendo que era la oportunidad esperada, aunque algunas de ellas decidieron apartarse de dicha empresa a tiempo y no perdieron un capital importante. La primera en lanzarse al abismo fue Oracle[2], que incluso fabricó algunas máquinas de prueba con un sistema operativo creado por ellos mismos.

Una vez dentro del ámbito universitario, la pregunta que nos planteamos es: *¿qué hace que apostemos desde nuestro área por tal proyecto?*

Nosotros, desde nuestra posición, buscábamos utilizar material docente de uso libre para impartir nuestras asignaturas, ya que de esta manera, nuestros alumnos serían capaces de poderse llevar el software a sus hogares y poder instalarlo sin ningún tipo de problema ni restricción alguna. Además de por este motivo, apostamos por el uso de GNU/Linux y por el software libre en general, ya que pensamos que abre infinitas puertas e ideas a nuestra imaginación, al menos muchas más que otro tipo sistema operativo cerrado propietario.

Una vez que disponíamos de un entorno estable y que nos garantizase un éxito en la impartición de nuestra docencia, decidimos adecuar tales necesidades a un ahorro económico por parte de nuestra área, y a un punto de partida totalmente distinto de lo que habíamos visto hasta ahora.

Tal apuesta no nos ha salido nada mal, ya que poco a poco hemos conseguido ir avanzando en proyectos con un nivel de escalabilidad bastante aceptable, gracias al uso de terminales.

Se decidió llevar a cabo dicho proyecto con asignaturas que encajaran perfectamente dentro de la filosofía que pretendíamos proponer, y así optamos por impartir todas las clases prácticas en un aula, que diseñamos previamente, dotada de terminales sin disco y utilizando un servidor central. Es tal la versatilidad que nos proporciona, que somos capaces de impartir la mayoría de docencia práctica de nuestras asignaturas del Área de conocimiento, entre las que cabe destacar las siguientes:

- *Telemática*: en la que usamos simuladores de redes totalmente libres y de uso gratuito, como pueda serlo CNET [3].
- *Complementos de telemática*: en la cual nos basamos en dos aplicaciones: SPIN[4]

(Simple Promela Interpreter) y XSPIN. La primera de ellas en modo texto y la segunda en modo gráfico (X-Window).

- *Fundamentos de Sistemas Operativos*: donde utilizamos software docente para análisis de llamadas al sistema en Linux y con conexión remota a servidor, o incluso local.
- *Programación en C y C++*: venimos usando entornos IDE basados en también en Linux para un entorno X-Window, además de los compiladores y linkadores necesarios.
- *Programación Concurrente*
- *Programación Multihilo*
- *Programación basada en Sockets*
- *Programación Java*[5]: Al igual que en C y C++ venimos utilizando entornos IDE de varias empresas, como Forte de Sun Microsystems, además de JDK[6] y otras herramientas.

Seguidamente vamos a tratar con cierta profundidad la problemática actual del mantenimiento de aulas de informática, y como la migración al sistema operativo GNU/Linux, nos ha ayudados a resolver tales problemas, hasta llegar al uso de terminales sin disco basados en dicho sistema.

1.1. El concepto de equipo obsoleto

El primer problema de un aula de informática surge cuando necesitamos instalar un software docente más o menos actual que va a consumir unos recursos bastante importantes en la máquina y que normalmente ésta no es capaz de rendirlos. Este motivo, generalmente, hace que nos planteemos un cambio de equipos por otros más modernos, con un microprocesador más potente, con más memoria RAM y con un disco duro más rápido y de mayor capacidad.

En la actualidad, en la mayoría de universidades, seguimos utilizando, mayormente, software docente basado en Microsoft Windows, a veces por comodidad, porque conocemos exhaustivamente ese software y no deseamos probar algún otro similar basado en otro sistema operativo, o porque no poseemos conocimientos en otros sistemas operativos, o es probable que no conozcamos la existencia de software que realice las mismas funciones que la versión de Microsoft Windows, y por tanto usamos el software basado

en este Sistema Operativo que consumirá bastantes recursos de nuestra máquina e incluso hará que nos planteemos un cambio de la misma.

Estos son algunos de los principales motivos por los que creemos que nuestros equipos antiguos se han quedado obsoletos.

Existen varias soluciones a este problema, entre las que cabe destacar dos, que suelen ser las más usadas por los responsables de aulas de informática de la propia universidad, o por los Técnicos de Laboratorio:

- La primera de ellas, a priori es la más sencilla, pero también es la más cara, y es comprar equipos nuevos que sustituyan a los antiguos.
- La segunda de ellas es proponer la migración de software a otro sistema operativo que nos plantee más ventajas, incluso sin renovar los equipos. Este sistema operativo podría ser GNU/Linux, que actualmente tiene un crecimiento bastante importante y que posee ciertas ventajas frente a cualquier otro sistema operativo propietario.

Llegados a este punto, podemos pensar que adoptando la segunda solución -la migración a GNU/Linux- se soluciona en parte la problemática de que los equipos puedan quedarse obsoletos. Esto en parte es cierto, pero no del todo, ya que aunque generalmente las herramientas basadas en GNU/Linux suelen consumir menos recursos que las basadas en Microsoft Windows, puede no ser suficiente y nuestros equipos antiguos aun así se quedaran obsoletos. Al menos, hasta aquí, nos quedaremos con que podemos disminuir considerablemente el consumo de recursos de nuestras máquinas, pero que probablemente no sea una solución factible a posteriori.

1.2. Coste elevado de Licencias

Otro de los problemas con el que nos encontramos en un aula, es el elevado coste de las licencias de software docente, teniendo que recurrir incluso a licencias de tipo Campus [7] para instalar dicho software en varias aulas de informática, con el gran desembolso que supone tales licencias por parte de la universidad.

En GNU/Linux, en cambio, nos encontramos con que la mayor parte del software es libre, y con

libre nos estamos refiriendo, como bien dice la comunidad GNU, a:

- La libertad de correr el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar como funciona el programa, y adaptarlo a sus necesidades (libertad 1). El acceso al código fuente es una precondition para esto.
- La libertad de distribuir copias de manera que se puede ayudar al vecino (libertad 2).
- La libertad de mejorar el programa, y liberar las mejoras al publico de tal manera que toda la comunidad se beneficia. (libertad 3). El acceso al código fuente es una precondition para esto.

En definitiva, no debemos pagar o pedir ningún tipo de permiso.

1.3. Restauración de equipos

Como sabemos, periódicamente, el disco duro de los ordenadores de las aulas de informática se restaura con una imagen que previamente se creó y replicó en todas y cada una de las máquinas. Este problema podemos solucionarlo mediante el uso de servidores, en los cuales se instala el software necesario una sola vez, y no es necesario instalarlo en los clientes, con la consiguiente comodidad y ahorro de tiempo.

Es latente el ahorro que supone no tener que pagar licencias de programas que se dedican a la replicación de imágenes de discos duros o particiones, e incluso, alargamos notablemente la vida de nuestros discos al no tener que restaurarlos periódicamente.

Imaginense pues, si además de que se consigue reutilizar los equipos antiguos, que nos ahorrásemos el dinero en licencias y que no necesitéramos disco duro, porque vamos a trabajar por red y con la memoria RAM.

2. Terminales Sin Disco basados en GNU/Linux

2.1. ¿Que es un terminal sin disco?

A grandes rasgos, podríamos definir un terminal sin disco como un conjunto de hardware que carga su sistema operativo por red y trabaja contra un servidor, o incluso se le podría instalar un disco duro para que además de trabajar contra un servidor, pudiese lanzar aplicaciones locales para evitar que todas se ejecuten en el servidor.

2.2. Ventajas de un terminal sin disco

Voy a pasar a resumir brevemente las numerosas ventajas que posee la utilización de terminales sin disco bajo GNU/Linux:

- Como ya hemos dicho, una de las principales ventajas es que estamos utilizando continuamente software libre, bajo licencia GPL [8], por lo que el coste es bajo e incluso nulo. Este tipo de software libre, posee un desarrollo abierto, y podemos disponer del código fuente del mismo en cualquier momento.
- El proyecto esta pensado para multipuesto, con todas las ventajas que ello conlleva. Imaginemos por un momento varias aulas de este tipo en nuestra universidad funcionando todas bajo un único servidor.
- No poseen disco duro, con la consecuente ventaja que éste no se puede romper al no existir, evitando algunos desastres que todo administrador ha sufrido en algun momento de su carrera, como grandes pérdidas de datos.
- La realización de copias de seguridad es muy sencilla ahora, puesto que basta con guardar en otra máquina o en algún soporte los datos contenidos en el home de los usuarios.
- Los terminales son de fácil reemplazo, ya que cualquier máquina nos serviría para utilizarla de terminal. Podríamos reutilizar pues, máquinas que la universidad se quiera deshacer de ellas.
- Tenemos la posibilidad de utilizar todo el software disponible en GNU/Linux, tanto de

oficina, como de diseño, etc. Algunos ejemplos son OpenOffice, Netscape...

- Podemos aprovechar la velocidad actual de las redes de área local, con velocidades de 100 Mbps o incluso de 1 Gbps con las conocidas Gigabit Ethernet, por lo que conseguimos además de velocidad, acabar con los cuellos de botella que se podrían producir en nuestra red.
- Tendremos también un arranque del terminal inmediato, con el consiguiente ahorro de tiempo.
- Optimizaremos al máximo nuestros recursos, ya que no necesitamos disco duro, y la cantidad de memoria RAM es muy baja. Hablaremos pues de un ahorro económico importante.
- Posibilita la gestión, administración y obtención de datos estadísticos del funcionamiento del aula.
- El diseño del aula es totalmente flexible, pudiendo añadir o eliminar máquinas fácilmente.
- También nos encontramos con que son muy fáciles de mantener, y poseen gran escalabilidad.
- Mediante el ahorro en máquinas para las aulas de informática, se consigue, al mismo tiempo, un incremento en la calidad de infraestructuras de los laboratorios, puesto que conseguimos máquinas más potentes invirtiendo en la ampliación o el cambio del servidor por otro con características superiores.
- Los alumnos mantienen las ventajas que obtenían mediante la utilización de otros sistemas operativos también en el uso de los terminales, y además, añaden otras como la robustez y la fiabilidad del sistema operativo GNU/Linux y la posibilidad de trabajar en las prácticas desde casa conectándose remotamente vía Telnet, SSH o X-Window remotas.

2.3. Inconvenientes de un terminal sin disco

El principal inconveniente que se nos plantea a la hora de llevar a cabo el diseño de un aula con terminales bajo Linux, es que necesitamos personal técnico con conocimientos avanzados de administración bajo éste sistema operativo.

Debemos tener en cuenta que todo el software se instalará en un servidor que deberá tener algún tipo de mantenimiento, como lo hubiese necesitado un aula repleta de computadores con Microsoft Windows, pero ahora, en lugar de tener que instalar el software en 40 ó 50 máquinas, sólo será necesario instalarlo en una: *nuestro servidor*.

Otro inconveniente bastante importante es el de la seguridad. Como veremos a lo largo de este artículo, los terminales sin disco utilizan el servicio NFS (Network File System) para la transferencia de archivos por red entre el servidor de terminales y los clientes. Dicho servicio tiene algunas carencias en materia de seguridad, por lo que formaremos una LAN donde ubicaremos todos nuestros terminales y así aislaremos de Internet todos los servicios que puedan ser más o menos vulnerables frente a posibles ataques desde el exterior. Por tanto, y si nuestro servidor tiene acceso a Internet, es sencillo deducir que poseerá dos interfaces de red: uno para nuestra red local y el otro para salir a la red de redes.

En nuestro diseño del aula, hemos optado por la creación de un firewall por software, aprovechando las características de ruteo que nos proporciona el kernel de GNU/Linux. Así hemos solucionado los posibles fallos de seguridad.

2.4. Especificaciones Técnicas

Hardware en el lado del servidor

El tipo de hardware y la calidad del mismo va a ser un punto importante en este lado, ya que pensemos que vamos a tener unos accesos bastante grandes a nuestro disco duro y un tráfico de red muy alto, incluso con un número elevado de máquinas cliente podríamos formar un cuello de botella en nuestra red local.

Los componentes más importantes, por tanto, serán el disco duro y la memoria RAM.

Es recomendable que el disco duro a instalar en el servidor sea SCSI, por la elevada velocidad que nos proporcionan este tipo de discos, lo que ayudará a que las aplicaciones se ejecuten en el cliente en un tiempo óptimo.

De la memoria RAM también dependerá la velocidad de ejecución de las aplicaciones en los clientes, y es tan importante como el disco duro. Por tanto es recomendable un tamaño elevado de dicha memoria.

También podríamos incluso incorporar una tarjeta Gigabit Ethernet para solucionar consecuentemente los posibles cuellos de botella que se podrían formar en nuestra red local. Este punto no es imprescindible, ni siquiera necesario,

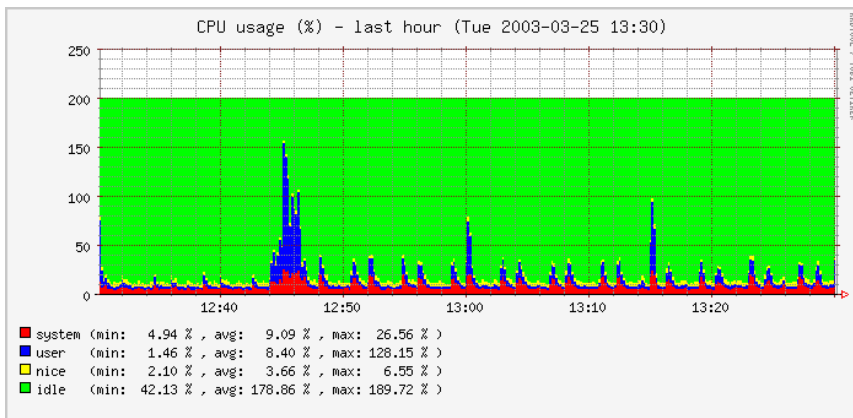


Figura 1. Uso de CPU

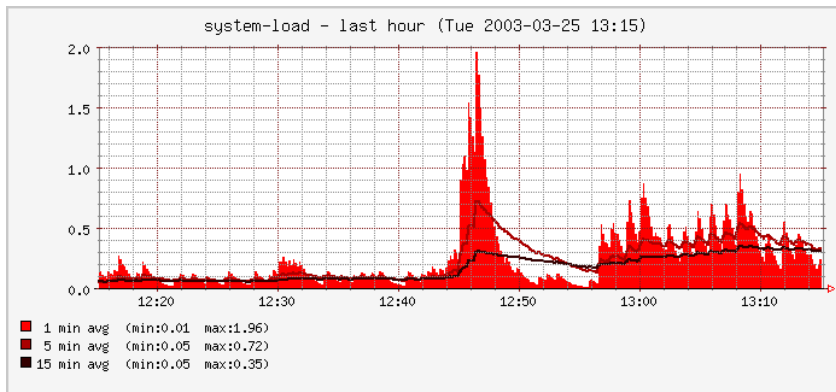


Figura 2. Carga del sistema

pero probablemente con un número muy elevado de clientes –en torno a 120- quizá se pueda producir un retardo en las conexiones debido a un pequeño cuello de botella en nuestra red.

Hardware en el lado del cliente

Respecto a los clientes, lo que se pretende es reutilizar las piezas o los ordenadores completos, así que no hacen falta unos componentes sobredimensionados, o con un coste elevado. Ni que decir tiene, que no es necesario que porten disco duro, ya que esto es la clave de este proyecto junto con la reutilización y ahorro.

Vamos a citar algunos ejemplos de configuraciones de clientes:

- Se han hecho pruebas con equipos i486 con 16MB de RAM, obteniendo resultados muy satisfactorios.
- Se ha trabajado con 100 estaciones de trabajo Pentium 166 con 32MB de RAM, destinadas a hoteles.

Servicios que se utilizan

Este proyecto se basa en una serie de servicios puntuales para su correcto funcionamiento.

Se realizan una serie de pasos desde que se pulsa el botón de encendido de un terminal hasta que obtenemos una pantalla para hacer login. En relación a los servicios, caben destacar que:

- Es necesario hacer una petición DHCP o BOOTP desde un cliente al servidor para obtener una dirección IP a través de la MAC. Esta petición se realizará en BROADCAST. Por tanto, un servidor DHCP será necesario desde el lado de nuestro servidor, ya que este le proporciona a los clientes, además de la IP, la máscara de red local, el path del kernel para bajar y el path del sistema de archivos a montar.
- Nuestros clientes descargan el kernel del sistema operativo a través de TFTP (Trivial File Transfer Protocol) así que será necesario también, dotar a nuestro servidor con servicios de TFTP.
- Para montar nuestro sistema de archivos en la máquina cliente, será necesario utilizar un servidor NFS (Network File System), por tanto será otro de los servicios a tener en cuenta.
- Si nos vamos a basar en terminales X-Window, también será necesario instalar un servidor de X-Window remotas, a través de conexiones XDMCP.

Tipos de terminales

Cabe destacar que se las estaciones de trabajo se pueden configurar de varias formas, atendiendo a nuestras necesidades.

Imaginemos, por ejemplo, que dedicaremos nuestros terminales exclusivamente a la programación en C, y que por lo tanto no será necesario desplegar un sistema X-Window. Tenemos pues una solución sin tener que usar X-Window que consiste en realizar conexiones en modo Telnet a nuestro servidor.

Ahora imaginemos que estamos utilizando los terminales para correr aplicaciones de diseño en CAD. Obviamente necesitaremos una infraestructura X-Window desplegada en nuestro cliente para tal fin.

Estas dos configuraciones serán las que podremos realizar con nuestros clientes, además de una tercera que será de depuración ante posibles fallos en la configuración de nuestro servidor o nuestros clientes.

Rendimiento del servidor

Durante nuestra extensa experiencia con terminales sin disco bajo GNU/Linux, hemos

procurado siempre ofrecer garantías educativas, pero también hemos apostado por soluciones fiables, robustas y eficientes a nivel de laboratorios.

Para conseguir este tipo de soluciones, previamente realizamos un exhaustivo estudio de nuestras ideas con el fin de lograr el máximo rendimiento y confianza, antes de ponerlas en marcha adaptándolas a nuestra docencia. Prueba de ello son las Figuras 1,2 y 3 que adjuntamos en este artículo, en las que reflejamos parte de un estudio del servidor a nivel de uso de CPU, de memoria y carga del sistema. En estas pruebas, concretamente, se ha probado el rendimiento del servidor durante el arranque de 20 terminales sin disco y la consiguiente validación de otros tantos usuarios en cada terminal.

En la Figura 1, podemos observar un pico de alrededor del 150% de uso de los dos procesadores a 800 MHz que tiene el servidor. Este pico esta provocado por la validación de los 20 usuarios instantáneamente en el sistema, lo que conlleva un uso muy grande de ambos procesadores, aunque podemos observar que nos resta un 50% libre. Vemos, también, que estos porcentajes se reducen a medida que los usuarios se validan y empiezan a trabajar con el sistema, obteniendo picos de uso de la CPU de alrededor

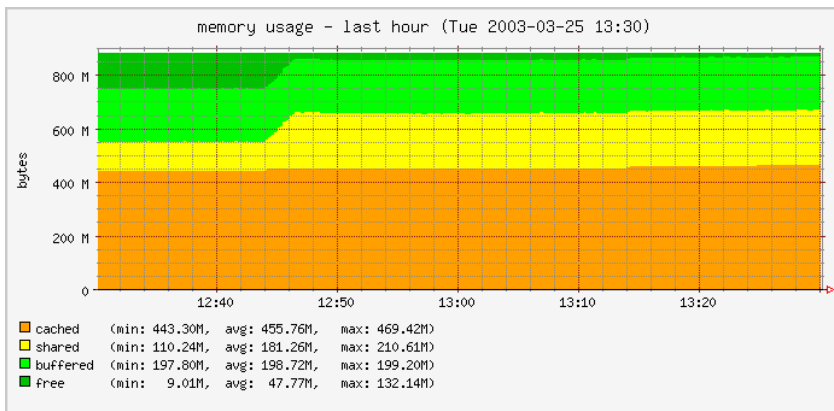


Figura 3. Consumo de memoria RAM

del 40% máximo.

En la Figura 2, podemos percatarnos que la carga del sistema, durante el proceso de arranque + validación de usuarios, ha sido muy pequeña, y que los procesos que ha arrancado –alrededor de 170 procesos con entorno gráfico basado en KDE– apenas tienen peso en nuestro servidor, aun siendo un entorno muy cargado, respecto a otros, en GNU/Linux.

En la Figura 3, podemos observar el uso de memoria RAM, que ha sufrido un incremento en el consumo de solo 100 Mbytes sobre el que ya tenía el servidor. La memoria de SWAP o de intercambio no ha sido utilizada por el sistema, lo que nos da un margen enorme de consumo.

3. Conclusión

Desde nuestra área de Tecnología de Computadores de la Universidad Miguel Hernández, apostamos firmemente por el proyecto que se muestra en este artículo, el cual nos proporciona la versatilidad que deseamos para impartir nuestra docencia práctica con unos resultados satisfactorios y además, que nuestros alumnos la reciban con la mayor calidad posible.

Representa también un desahogo laboral para nuestros técnicos y responsables de aulas, que una vez introducidos en el sistema operativo GNU/Linux, les resultará más sencillo y con un menor gasto de tiempo las configuraciones de las aulas de informática, pudiendo dejar configuradas varias de ellas con un mismo servidor.

Referencias y Bibliografía

- [1] GNU/Linux Debian
<http://www.debian.org>
- [2] Oracle
<http://www.oracle.com>
- [3] CNET Network Simulator
<http://www.cs.uwa.edu.au/cnet/>
- [4] SPIN
<http://netlib.bell-labs.com/netlib/spin/whatispin.html>
- [5] Java
<http://java.sun.com/>
- [6] JDK (Java Development Kit)
<http://java.sun.com/>
- [7] Licencias Campus Microsoft
<http://www.microsoft.com/education/?ID=Terms>
- [8] Licencia GPL
<http://www.gnu.org/copyleft/gpl.html>
- [9] Netscape
<http://www.netscape.com>
- [10] GNU
<http://www.gnu.org>
- [11] Linux Terminal Server Project
<http://www.ltsp.org>
- [12] K-12Linux Project
<http://www.k12ltsp.org>
- [13] X-Terminal at City of Largo
<http://dot.kde.org/995949998/>
- [14] Diskless Nodes HOW-TO for Linux
<http://www.tldp.org/HOWTO/Diskless-HOWTO.html>
- [15] Important notes on security issues for X-terminals
<http://etherboot.sourceforge.net/doc/html/security.html>
- [16] Swieskowski, Patrick. Making an X Terminal from a PC. Published in Issue 68 of Linux Gazette, July 2001
- [17] Kaszeta, Rich. Linux X Terminals A new use for old and outdated PCs, April 1998 Linux Gazette
- [18] Linux NFS-HOWTO:
<http://www.tldp.org/HOWTO/NFS-HOWTO/index.html>
- [19] XDM and X Terminal mini-HOWTO:
<http://www.tldp.org/HOWTO/mini/XDM-Xterm/index.html>
- [20] Debian packages page:
<http://www.debian.org/distrib/packages>
- [21] Girard, Kim. Ellison resurrects network computer. CNET News.com November 16, 1999
<http://news.com.com/2100-1001-233137.html?legacy=cnet>