Cómo proponer prácticas avanzadas para sistemas distribuidos y que el alumno no muera en el intento

Diego Marcos Jorquera, José Vicente Berná Martínez, Virgilio Gilart Iglesias,
Héctor Ramos Morillo, Francisco Maciá Pérez
Dpto. de Tecnología Informática y Computación
Universidad de Alicante
03080 Alicante
email: {dmarcos, jvberna, vgilart, hramos, pmacia}@dtic.ua.es

Resumen

La realización de prácticas complejas en el ámbito de los sistemas operativos distribuidos, que profundicen tanto en los aspectos inherentes a las tecnologías básicas como en los más avanzados y con mayor nivel de abstracción, se ve seriamente afectada por factores como la falta de tiempo para su desarrollo, el elevado número de alumnos matriculados y, sobre todo, por la gran diversidad de tecnologías, modelos y herramientas implicados. En la mayor parte de los casos, este tipo de prácticas se centran, bien en las tecnologías básicas (sockets, RMI, RPC), bien en aquéllas de carácter más abstracto (aplicaciones y servicios Web). Si en el primer caso los alumnos no tienen tiempo para comprender y asimilar los conceptos más avanzados, necesarios para abordar aplicaciones sofisticadas, en el segundo caso, carecerán de los conocimientos básicos que les permitirán aprovechar de forma eficaz y eficiente los recursos computacionales sobre los que se ejecutan las aplicaciones. En este artículo se propone un enfoque que permite plantear prácticas de amplio calado, que contemplen todos los aspectos descritos, junto con el trabajo en grupo y dentro de un marco establecido por estándares abiertos. Como caso concreto se describe la práctica propuesta para la asignatura obligatoria de quinto curso, Sistemas Operativos en Red, impartida por el Departamento de Tecnología Informática y Computación de la Universidad de Alicante.

1. Introducción

Los sistemas distribuidos en general e Internet en particular están cambiando la forma de concebir, realizar e implantar las aplicaciones informáticas. En todo este proceso concurren multitud de tecnologías, modelos, herramientas, estándares, perfiles profesionales y modelos de negocio que hacen muy complicado plantear asignaturas que cubran adecuadamente todos estos factores.

En estas condiciones resulta muy difícil conocer todas las tecnologías TIC involucradas, diferenciarlas y discriminar cuándo, cómo y dónde emplear cada una de ellas.

Su estudio requiere prácticas complejas y muy elaboradas. Sin embargo, este tipo de prácticas, para que puedan ajustarse a los tiempos disponibles, no suelen plantear casos muy realistas.

Un caso realista debería considerar la organización del trabajo por grupos, la adopción de estándares, la elección de las herramientas y modelos más adecuados y la programación distribuida.

En muchas ocasiones, para poder contemplar estos requerimientos, las prácticas propuestas suelen resultar muy guiadas, con lo que el alumno no adquiere una verdadera conciencia del proyecto global.

Nuestra propuesta es plantear una aplicación que deba ser desarrollada utilizando diferentes metodologías y con diferente grado de abstracción. Para los niveles de abstracción mayores se establecerán grupos de prácticas que elaboren de forma totalmente independiente módulos funcionales de la aplicación y que, finalmente, puedan ser integrados dentro de un marco de referencia establecido por estándares abiertos, evitando especificaciones ad hoc, al menos en los niveles de servicios.

2. Estado actual

Durante las últimas décadas las tecnologías de comunicación han sufrido muchos y diversos cambios, apreciándose una clara evolución por la

cual la programación de aplicaciones distribuidas utiliza tecnologías cada vez más cercanas a las capas superiores del modelo OSI [8].

De esta manera, en la actualidad, el abanico de tecnologías que se utiliza es muy amplio y abarca desde los niveles de abstracción más bajos, donde se trabaja de forma muy cercana al funcionamiento físico de la red, hasta niveles de abstracción muy altos, donde no se tiene un conocimiento real de los procedimientos de comunicación que actúan por debajo.

En los niveles más bajos encontramos como principal tecnología de red los *sockets* [4], un interfaz de comunicación altamente extendido y que suele presentarse muy integrado con el sistema operativo, aportando la base para tecnologías de nivel superior. En el entorno docente, las prácticas realizadas con *sockets* están muy generalizadas ya que establecen los fundamentos de la comunicación y permiten el entendimiento de sus procesos más básicos.

En un nivel de abstracción superior a los *sockets* encontramos RPC (*Remote Procedure Call*) desarrollado por Sun Microsystems [5], y que nos permite realizar llamadas a procedimientos independientemente de su localización, buscando con ello la transparencia de acceso para el programador [1].

A más alto nivel encontramos tecnologías orientadas a objetos, que abstraen aun más los procesos de comunicación. Entre éstas cabe destacar RMI (*Remote Method Invocation*), utilizado en Java, DCOM (*Distributed Component Object Model*)dentro de las plataformas Windows de Microsoft y CORBA (*Common Object Request Broker Architecture*) [6] que pretende conseguir un alto nivel de transparencia en la invocación remota.

Por último, en el nivel de abstracción más alto, dentro de las tecnologías orientadas al uso de un *middleware*, encontramos modelos de componentes software distribuidos, apoyados en tecnologías *middleware* como J2EE de Sun o .NET *framework* de Microsoft; y servicios Web con protocolos como SOAP y UDDI definidos sobre HTTP y basados en XML.

Todos estos mecanismos y paradigmas de comunicación tienen una alta aceptación en el desarrollo de aplicaciones distribuidas y en la actualidad se pueden considerar tecnologías válidas y maduras.

A estas tecnologías, se le suman una amplia variedad de *patrones de implementación* y *frameworks* referentes a la creación de aplicaciones distribuidas que hacen uso de las tecnologías comentadas, dando soporte a las diferentes arquitecturas, como por ejemplo, la arquitectura cliente-servidor de 2, 3 o n-niveles.

Habitualmente, el enfoque adoptado para imbuir los conocimientos teóricos sobre la programación distribuida pasa por proponer prácticas fundamentadas en tecnologías con un bajo nivel de abstracción [9], [12]. En concreto, el entorno de trabajo usado suele ser el de librerías de *sockets* bajo el sistema operativo Linux [7] mediante C estándar o C++. Si bien esto permite al alumno adquirir los conocimientos necesarios para la comprensión de la tecnología básica, les impide adquirir una visión más abstracta, imprescindible para la resolución de problemas asociados a los sistemas distribuidos.

Otro enfoque, cada vez más presente, consiste en orientar las prácticas a tecnologías de mayor nivel de abstracción, relegando las tecnologías básicas a un segundo plano [10], [11]. Esto ofrece al alumno un acercamiento más realista al desarrollo de las aplicaciones distribuidas que se diseñan actualmente; sin embargo, no les permite llegar a asimilar adecuadamente los fundamentos básicos de la tecnología.

3. Nuestro enfoque

Para que el alumno conozca las principales tecnologías y paradigmas de la computación distribuida se propone como práctica para la asignatura Sistemas Operativos en Red (SOR) un caso realista y profundo, que será implementado mediante el uso de diferentes metodologías de comunicación. En concreto se propone el uso de tres tecnologías: de nivel de abstracción bajo (sockets), medio (RMI) y alto (*Webservices*).

Para que el alumno comprenda que cada nivel de abstracción es adecuado para la resolución de un problema determinado, y que en algunos casos una solución basada en *sockets* es más apropiada que otra basada en *Webservices* y viceversa, la práctica se abordará desde dos puntos de vista diferentes (ver figura 2): en el primer enfoque, se propone resolver la aplicación mediante una arquitectura cliente-servidor de 2-niveles y será implementada con *sockets* y RMI; el segundo enfoque es más ambicioso, proponiendo una arquitectura de 3-niveles, mediante el uso de *Webservices*.

Ambos enfoques coinciden en el punto de partida. La propuesta de aplicación se basa en un protocolo de aplicación definido para la práctica y se proporciona a los alumnos a modo de RFC (Request For Comments).

La implementación mediante sockets y RMI se realizará de forma individual y parcialmente guiada, proporcionando al alumno el esqueleto básico de la aplicación. De esta manera el alumno puede centrarse en los aspectos técnicos relacionados con las tecnologías implicadas, dejando más tiempo para la realización de las prácticas con modelos más abstractos (*Webservices*) en los que se potenciará mucho más la toma de decisiones de diseño.

Para la implementación de la práctica con *Webservices* se establecerán grupos de alumnos, distribuyendo el conjunto global de la aplicación de manera que cada grupo será responsable de la realización de un módulo de la aplicación.

De esta manera se potenciará el trabajo cooperativo, organizado por grupos, y se contemplan los problemas derivados de la interacción, desarrollo en paralelo e integración, siempre en un marco establecido por estándares abiertos.

Con la separación de la práctica en dos bloques diferenciados se llega a un compromiso entre las metodologías de prácticas guiadas y las que fomentan la toma de decisiones por parte del alumno.

En todos los casos se aportará al alumno unos documentos de especificación donde se describirá detalladamente los protocolos de aplicación que se implementarán (RFC's). Con ello se logrará que el alumno:

- Tome conciencia de que, según se incrementa el nivel de abstracción de la tecnología, disminuye el nivel de detalle de la especificación. Por ejemplo para la práctica de sockets de determinará hasta el tipo y orden de los datos a transmitir, mientras que para la de Webservices simplemente se describirán los servicios ofertados.
- Se familiarice con el desarrollo de aplicaciones complejas, donde ceñirse a la especificación planteada es vital para un correcto trabajo en paralelo. De hecho el alumno será consciente de que un módulo desarrollado por él podrá ser sustituido por el de otro compañero, sin que esto afecte al funcionamiento global de la aplicación.
- Consolide sus conocimientos sobre las técnicas relacionadas con la integración y la reutilización del software.

Si bien en la asignatura existen otros conceptos teóricos que no son abordados de forma explícita en este planteamiento, con la realización de estas prácticas se pretende establecer un hilo conductor sobre el que contemplar el resto de conocimientos asociados a la asignatura. Por ejemplo en las prácticas de *sockets* se aplicarán técnicas de resolución DNS o en las prácticas de RMI se propondrán técnicas de seguridad.

4. Descripción de la práctica

Los objetivos generales de la asignatura Sistemas Operativos en Red consisten en proporcionar conocimiento concreto al tiempo que una visión global e integradora sobre los protocolos, configuraciones y estrategias que dan soporte a las aplicaciones que deben ejecutarse sobre sistemas distribuidos y redes de computadores.

Se trata de una asignatura anual de quinto curso, obligatoria de la titulación Ingeniería en Informática y que consta de nueve créditos repartidos por igual entre teoría y práctica.

La práctica planteada para la asignatura consiste en la implementación de una *agencia de viajes*. Según esta propuesta un cliente dispondrá de una aplicación de usuario mediante la que podrá realizar la organización y reserva de un viaje en el que se incluye: venta de billetes de avión, reserva de habitaciones de hotel, sistema de pago con tarjetas de crédito y servicio meteorológico. La funcionalidad de estos cuatro servicios no tendrá por que ser real, ya que no es éste el objetivo de la práctica. De hecho, estos módulos podrían ser simples generadores de respuestas aleatorias o prefijadas.

La aplicación de usuario requerirá al hipotético cliente los siguientes datos:

- Nombre del cliente.
- Ciudad origen del viaje.
- Ciudad destino del viaje.
- Número de personas.
- Categoría del hotel (número de estrellas).
- Fecha inicial del viaje.
- Fecha final del viaje.
- Número de tarjeta de crédito.
- Caducidad de la tarjeta de crédito.
- Número de teléfono móvil.

La aplicación de usuario realizará una primera validación de los datos, informando al usuario de los posibles errores cometidos.

La lógica de la aplicación será la siguiente:

- En primer lugar se valida la tarjeta de crédito. Si no es válida se informará al usuario de ello y se cancelará el proceso.
- Se consultará las condiciones meteorológicas previstas en el destino entre las fechas del viaje. Si se prevé mal tiempo se informará al usuario de ello y se cancelará el proceso.
- Se obtendrá un listado de todos los hoteles disponibles para la ciudad de destino y categoría indicada. Si no se obtiene ninguno se informará al usuario de ello y se cancelará el proceso.
- Por cada uno de los hoteles se consultará su disponibilidad para el número de personas indicado y entre las fechas del viaje. El primer hotel que cumpla las condiciones será designado como alojamiento del viaje, no realizándose aun la reserva. Si ninguno de los hoteles cumple las condiciones se informará al usuario de ello y se cancelará el proceso.
- Se obtendrá un listado de todos los vuelos con origen y destino en las ciudades indicadas. Si no se obtiene ninguno se informará al usuario de ello y se cancelará el proceso.
- Por cada uno de los vuelos obtenidos se consultará su disponibilidad para el número de personas indicado y entre las fechas del viaje. Igualmente, el primer vuelo que cumpla las condiciones será designado como transporte del viaje, no realizándose aun la reserva del mismo. Si ninguno de los vuelos cumple las condiciones se informará al usuario de ello y se cancelará el proceso.
- Se suman los importes del hotel y vuelo y se informa al usuario sobre la cantidad total, esperando su confirmación.
- Si el usuario confirma afirmativamente se consultará si el importe total puede ser cargado a la tarjeta de crédito. Si no es posible se informará al usuario de ello y se cancelará el proceso.
- Se procederá a reservar el hotel para las fechas y número de personas indicadas. Si no es posible se informará al usuario de ello y se cancelará el proceso.
- Se procederá a reservar el viaje para las fechas y número de personas indicadas. Si no es posible se informará al usuario de ello y se cancelará el proceso.
- Se procederá a realizar el cargo del importe total en la tarjeta de crédito. Si no es posible por falta de crédito o error se informará al usuario de ello y se cancelará el proceso. Notar que los procesos de anulación de las reservas de hotel y vuelo en caso de error no serán implementadas.
- Por último se activará el servicio de información meteorológica para el cliente. Se indicará para ello el número de teléfono móvil, la localidad y el periodo de tiempo.

La aplicación hará uso de un conjunto de módulos que implementan los cuatro servicios contemplados.

El servicio de reserva de hoteles dispondrá de las siguientes funcionalidades:

- Obtención de listado de hoteles. Se le indicará la localidad de los hoteles a listar así como la categoría de los mismos. El resultado será una lista con los identificadores de los hoteles encontrados.
- Consulta de disponibilidad de un hotel. Se indicará el identificador del hotel, el periodo de tiempo y el número de personas para los que se desea consultar la disponibilidad. El resultado será *afirmativo* si hay habitaciones disponibles o *negativo* en caso contrario.
- Reserva de habitaciones. Se indicará el identificador del hotel, el periodo de tiempo y el número de personas para los que se desea reservar. El resultado será *afirmativo* si se ha podido realizar la reserva o *negativo* en caso contrario.

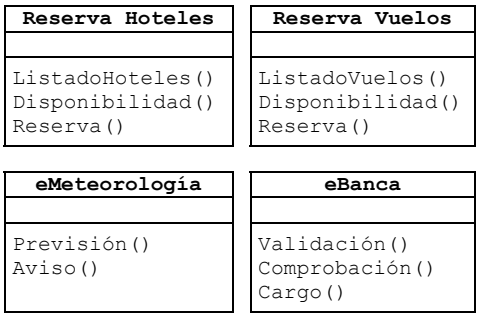

Figura 1. Servicios de la aplicación

El servicio de reserva de vuelos dispondrá de las siguientes funcionalidades:

- Obtención de listado de vuelos. Se le indicará las localidades origen y destino del vuelo. El

resultado será una lista con los identificadores de los vuelos encontrados.
- Consulta de disponibilidad de un vuelo. Se indicará el identificador del vuelo, el periodo de tiempo y el número de personas para los que se desea consultar la disponibilidad. El resultado será *afirmativo* si hay plazas disponibles o *negativo* en caso contrario.
- Reserva de vuelo. Se indicará el identificador del vuelo, el periodo de tiempo y el número de personas para los que se desea reservar. El resultado será *afirmativo* si se ha podido realizar la reserva o *negativo* en caso contrario.

El servicio meteorológico (eMeteorología) dispondrá de las siguientes funcionalidades:
- Consulta de tiempo. Se indicará un periodo de tiempo a evaluar. El resultado será *afirmativo* si se prevé buen tiempo o *negativo* en caso contrario.
- Activación del servicio de aviso meteorológico. Se indicará el número de teléfono móvil, localidad y el periodo de tiempo.

El servicio de pago con tarjeta (eBanca) de crédito dispondrá de las siguientes funcionalidades:
- Validación de tarjeta. Se le indicará el número de la tarjeta y la fecha de caducidad. El resultado será afirmativo si la tarjeta es válida o negativo en caso contrario.
- Comprobación de cargo. Se le indicará el número de la tarjeta y la cantidad a comprobar. El resultado será afirmativo si se puede cargar dicha cantidad a la tarjeta o negativo en caso contrario. El cargo no se efectuará, sólo se realiza una comprobación de su viabilidad.
- Cargo. Se le indicará el número de la tarjeta y la cantidad a cargar. El resultado será afirmativo si se ha cargado dicha cantidad a la tarjeta o negativo en caso contrario.

Cada uno de estos servicios tendrá que implementar su propia validación de argumentos para cada una de las funcionalidades, evitando problemas de seguridad o malfuncionamiento en su uso.

A continuación se describen los dos bloques en los que se estructura la práctica. En el primer bloque se abordarán las implementaciones en los niveles bajo y medio y en el segundo bloque los relacionados con el nivel más alto de abstracción.

4.1. Bloque 1: Nivel bajo y medio

En este bloque la arquitectura de la aplicación a implementar estará estructurada a dos niveles: uno donde se situaría la aplicación de usuario y otro donde residirían la agencia de viajes con los diferentes servicios.

La aplicación de usuario será la responsable de consultar los datos necesarios de los servicios ofertados, implementando la lógica de la aplicación.

La aplicación servidora implementará los diferentes servicios que se ofertan, aportando un interfaz al cliente.

Los interfaces de comunicación estarán normalizados mediante el protocolo que se describe en el pertinente documento de especificación, uno para la práctica en *sockets* y otro para la de RMI. Para cada práctica también se aportará un esqueleto base sobre el que desarrollar la aplicación.

En la especificación para la implementación de la práctica en *sockets* se detallarán los siguientes aspectos:
- Puertos TCP/IP utilizados.
- Listado de comandos aceptados por cada uno de los servicios.
- Formato interno de cada uno de los comandos indicando: orden, tipo y descripción de cada campo que conforma el comando y posibles respuestas al mismo.
- Descripción de la estructura de aplicación aportada como base para la realización de la práctica.

En la especificación para la implementación en RMI se indicará:
- Puertos TCP/IP utilizados.
- Definición de las diferentes clases a crear, una para cada servicio.
- Definición de los métodos de cada clase, detallando los argumentos de los mismos. Cada uno de los métodos implementará una funcionalidad del servicio.
- Descripción de la estructura de aplicación aportada como base para la realización de la práctica.

4.2. Bloque 2: Nivel alto

En este bloque, donde el desarrollo se realiza mediante *Webservices*, se replantea la arquitectura

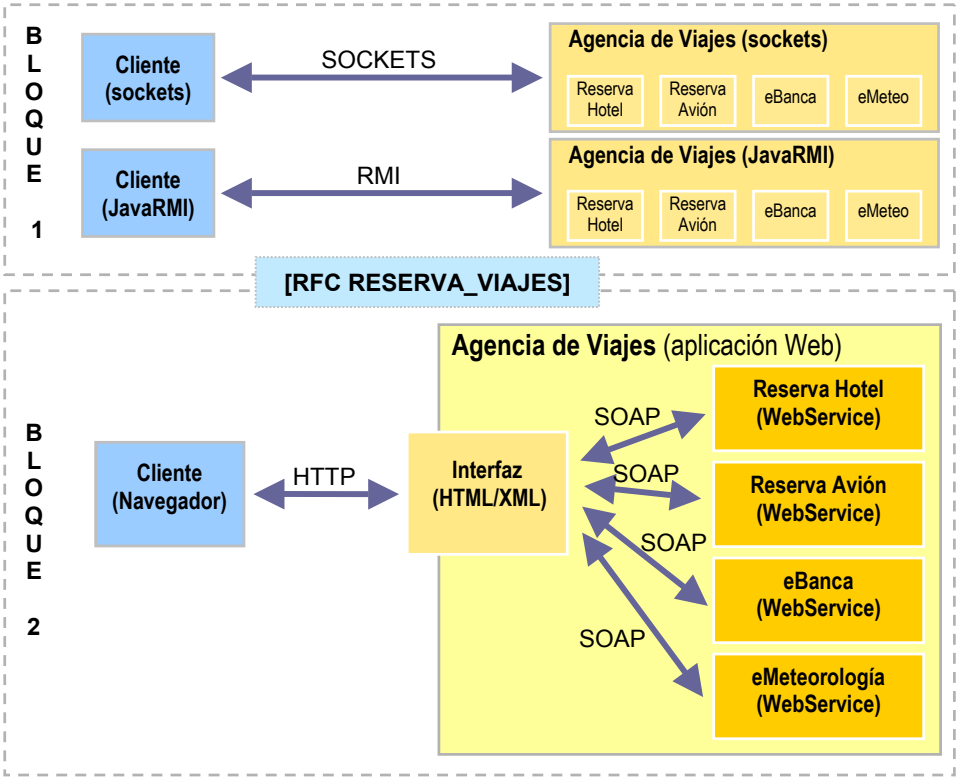

Figura 2. Bloques funcionales de la aplicación, organizada por niveles de abstracción junto con la relación entre los mismos basada en protocolos y estándares abiertos adecuados a cada nivel.

de la aplicación de manera que el alumno observe como esta tecnología se adapta mejor a arquitecturas más complejas. Por ello se distribuyen los servicios para obtener una aplicación más desacoplada, con la que ganar en flexibilidad, reutilización, escalabilidad y desarrollo en paralelo.

Así pues la aplicación estará estructurada a tres niveles:

- Una interfaz de usuario, que en este caso es un simple navegador.
- Un servidor, donde se implementa la lógica de la aplicación y el interfaz con el cliente.
- Un conjunto de servicios, implementados como webservices.

En este caso el documento de especificación será mucho menos detallado, y se limitará a normalizar los nombres y los métodos a implementar. De esta manera todos los alumnos crearán los *webservices* con la misma definición ayudando a la posterior integración de todos los componentes desarrollados.

4.3. Planificación

Para el caso de la asignatura de Sistemas Operativos en Red se dispone de un total de quince sesiones de clases prácticas de dos horas de duración cada una.

Como se puede observar en la Tabla 1 se ha distribuido el tiempo de manera que el número de sesiones es menor cuando las prácticas son guiadas que cuando no lo son.

Bloque	Práctica	Sesiones
1	Introducción	1
	Sockets	3
	RMI	3
2	Establecimiento de grupos	1
	Webservices	3
	Interfaz	2
	Integración	1
	Repaso/Recuperación	1

Tabla 1. Planificación de las prácticas.

5. Conclusiones

La práctica propuesta permite la experimentación tanto con tecnologías básicas (*sockets*, RMI) como de niveles de abstracción mayores (aplicaciones y servicios Web).

La metodología empleada para su realización fomenta el trabajo en grupo y la interacción, basándola en definiciones normalizadas mediante estándares abiertos.

Al abordar diferentes metodologías, tecnologías y herramientas relacionadas con los sistemas distribuidos, y presentarlas de forma progresiva, ordenadas crecientemente por nivel de abstracción, el alumno aprende a distinguir qué nivel es el más adecuado a cada problema y en cada momento.

En la actualidad estamos trabajando para diseñar un completo entorno de prácticas que permita al alumno centrarse más en la realización de las mismas.

En este sentido estamos trabajando en la confección de un *Live-CD* que contenga todas las herramientas e infraestructuras software necesarias que permitan al alumno realizar las prácticas mediante un entorno completo, portable e independiente en lo posible del hardware, facilitando la migración a otros laboratorios y el trabajo en casa.

Éste es el primer año en el que se ha cursado esta asignatura ya que pertenece al nuevo plan de estudios. Se han ofertado dos turnos de prácticas para un total de cincuenta alumnos, los cuales se han distribuido en grupos de cuatro personas. Hasta el momento la experiencia está resultando muy positiva, lográndose los objetivos establecidos y percibiendo un alto grado de interés y motivación por parte de los estudiantes.

Referencias

[1] Basanta Gutiérrez, B., Tajes Martínez, L. *Tecnologías para el desarrollo de Sistemas Distribuidos: Java versus Corba*. X jornadas de paralelismo. La Manga del Mar Menor, Murcia, septiembre 1999.

[2] Coulouris, G., Dollimore, J. y Kindberg, T. *Sistemas distribuidos. Conceptos y diseño*. Ed. Addison Wesley, 3ª edición, 2001.

[3] Liu, M.L. *Computación Distribuida. Fundamentos y Aplicaciones.* Addison Wesley, 2004.

[4] Mary Campione y Kathy Walrath. *The Java Language Tutorial: Object-Oriented Programming for the Internet.* Paperback.

[5] Sun Microsystems. http://www.sun.com.

[6] Robert Orfali & Dan Harkey. *Client/Server Programming with Java and Corba. 2nd. Ed*. John Siley & sons, 1998

[7] Tackett & Gunter. *Utilizando Linux 2ªed*. Prentice Hall.

[8] Tanenbaum, A. S. *Distributed operating systems*. Prentice Hall Internacional, 1995.

[9] Universidad Carlos III Madrid. http://www.uc3m.es

[10] Universidad Politécnica de Madrid. http://www.upm.es

[11] Universidad de la Rioja. http://www.unirioja.es

[12] Universidad de Zaragoza. Escuela Universitaria Politécnica de Teruel. http://www.unizar.es/centros/eupt