

## PROGRAMA PRÁCTICO DE UNA ASIGNATURA DE INGENIERÍA DEL SOFTWARE DISTRIBUIDO

Asier Perallos<sup>1</sup>, Rebeca Cortazar<sup>1</sup>

<sup>1</sup>*Departamento de Ingeniería del Software ESIDE (Universidad de Deusto)*  
e-mail: perallos@eside.deusto.es, cortazar@eside.deusto.es

**RESUMEN:** En esta contribución describimos de manera detallada el programa práctico de la asignatura *Ingeniería del Software III*, del plan de estudios implantado en 1996 en ESIDE (Universidad de Deusto). Esta asignatura es troncal y se impartirá por primera vez a los alumnos de 5º durante el primer cuatrimestre del curso 2000-2001. Mediante su docencia, nuestra intención es acercar a los alumnos los conceptos involucrados en el desarrollo de software distribuido, así como afianzar algunos de ellos mediante las clases de laboratorio que describimos.

### 1.- INTRODUCCIÓN

Durante el próximo curso 2000-2001, se impartirá por vez primera la asignatura *Ingeniería del Software III*, cuyo programa teórico ya presentamos en JENUI'99. En aquel artículo [1], justificamos los contenidos de la asignatura, que van orientados hacia el desarrollo de aplicaciones distribuidas, tanto desde un punto de vista técnico como desde el punto de vista de los aspectos metodológicos. Por otra parte, el plan de prácticas quedó someramente planteado, puesto que era un tema que en aquel momento no había sido elaborado. Lógicamente, durante este curso académico hemos abordado la labor de preparar los contenidos de las sesiones de laboratorio cuyo objetivo, contenido, organización y metodología se detallan a continuación.

### 2.- PROGRAMA PRÁCTICO

El objetivo fundamental de las sesiones de laboratorio es facilitar al alumno la comprensión de una parte de los contenidos teóricos estudiados [2]. Las prácticas permitirán que el alumno compruebe por su cuenta el funcionamiento real de los conceptos explicados en clase y mostrarán una visión práctica de éstos. Al mismo tiempo, le darán la oportunidad de familiarizarse con tecnologías novedosas y por tanto, muy necesitadas de profesionales en el mercado, logrando que los alumnos terminen la carrera debidamente preparados para satisfacer la demanda del mercado laboral.

Las sesiones de laboratorio se basarán en el desarrollo de aplicaciones distribuidas centrándose en tres áreas: la comunicación entre objetos remotos, la tecnología de componentes y el acceso remoto a bases de datos. Concretamente, las tecnologías usadas para la realización de las prácticas serán RMI, CORBA y JDBC [3], todo ello sobre la plataforma Java.

La selección de estas tecnologías se fundamenta en los siguientes motivos:

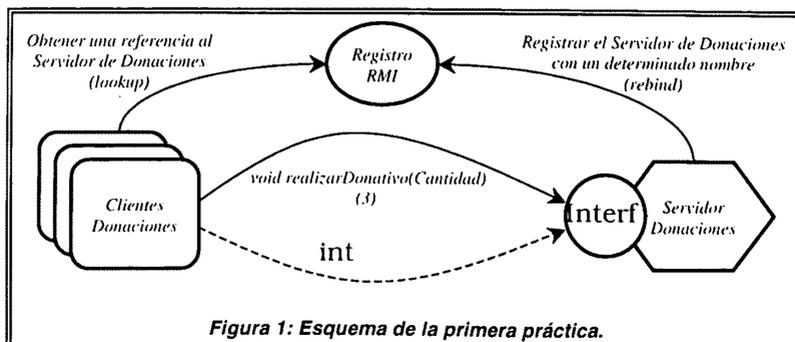
- Son tecnologías actuales, no excesivamente complejas de utilizar y que se complementan muy bien con los conceptos teóricos estudiados, permitiendo experimentar con ellos y contrastarlos.
- El uso de estas tecnologías no requiere de grandes recursos software ni hardware. El software necesario está basado en la plataforma Java y se encuentra disponible de forma gratuita, por lo tanto se posibilita que las prácticas puedan ser realizadas fuera de la universidad.
- Los conocimientos necesarios para su utilización se amoldan perfectamente al perfil de nuestros alumnos. Éstos conocen todos los conceptos de análisis y diseño orientado a objetos al haber cursado la asignatura troncal de *Ingeniería del Software II*, poseen conocimientos de programación orientada a objetos al haber cursado la asignatura de *Tecnología de la Programación* y conocen la plataforma Java al haber realizado múltiples prácticas basadas en ella a lo largo de la carrera.
- Por último, estas tecnologías, junto con otras del temario teórico, son lo suficientemente extensas y novedosas como para animar a los alumnos a que realicen proyectos de fin de carrera basados en ellas.

Los créditos prácticos de la asignatura serán cubiertos mediante la realización de tres prácticas que se desarrollarán siguiendo la siguiente pauta: en primer lugar, se analizará el funcionamiento de la tecnología en estudio mediante un ejemplo trivial (como el clásico 'hola mundo') y a continuación se investigará un ejemplo más complejo. Sobre este último se plantearán modificaciones que los estudiantes deberán implementar y entregar de manera obligatoria. A continuación, se expone el contenido de cada una de las prácticas.

### a) La primera práctica

Tendrá como objetivo la invocación de métodos de un objeto remoto a través de RMI. En primer lugar se expondrán los fundamentos necesarios para poder utilizar RMI y se comprobará su funcionamiento con un ejemplo trivial como el típico 'hola mundo'.

Seguidamente, se presentará una sencilla aplicación distribuida consistente en un Servidor de Donaciones al que se conectan varios clientes a través de RMI para realizar Donativos. Utilizaremos este software base como ejemplo de la tecnología RMI y como práctica plantearemos una modificación: se debe añadir una nueva funcionalidad al servidor que permita a los clientes consultar la cantidad total donada que ha sido recogida hasta ese momento. Esta modificación requerirá que los alumnos inserten un nuevo método en la interfaz remota del servidor y que lo invoquen desde los clientes a través de RMI (Figura 1).

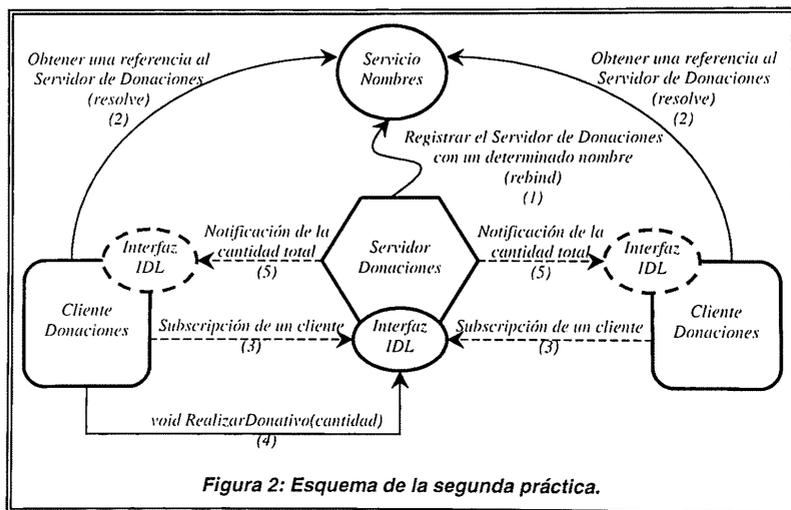


Los objetivos a cubrir en esta práctica son que los alumnos se familiaricen con un mecanismo sencillo de comunicación entre objetos remotos como es RMI y que conozcan todos los elementos que hacen posible esta comunicación como son entre otros, el registro RMI y las interfaces remotas.

### b) La segunda práctica

La segunda práctica estará basada en la tecnología de componentes centrándose en el modelo de componentes CORBA [4]. Partirá de la misma aplicación de Donativos que la usada en la práctica anterior pero implementada mediante componentes CORBA en lugar de RMI. También utilizaremos esta aplicación como ejemplo para explicar la tecnología CORBA y como práctica plantearemos una modificación sobre ella. Dicha modificación consistirá en la implementación de un patrón *Observer* [5] que permita a los clientes ser notificados en todo momento de la cantidad total donada. Así, cada vez que un cliente realice un donativo, el resto de clientes verá actualizada automáticamente esa cantidad.

Para implementar este patrón se deberá establecer un Sistema de Suscripción entre los clientes y el servidor, para lo cual será necesario que los clientes sean convertidos también en componentes CORBA (Figura 2).

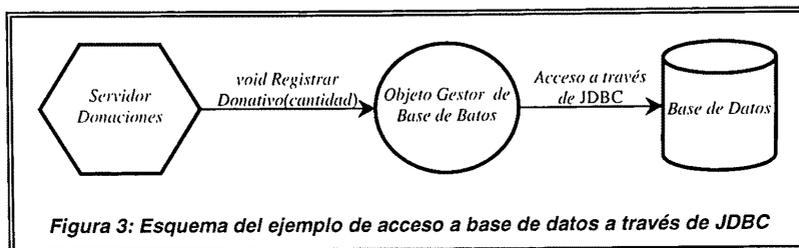


Los objetivos a cubrir en esta práctica son que los alumnos se familiaricen con la tecnología de componentes y más en concreto con el modelo de componentes CORBA y que sean capaces de reconocer dentro de una aplicación los elementos de la arquitectura CORBA estudiados en las clases teóricas.

### c) La tercera práctica

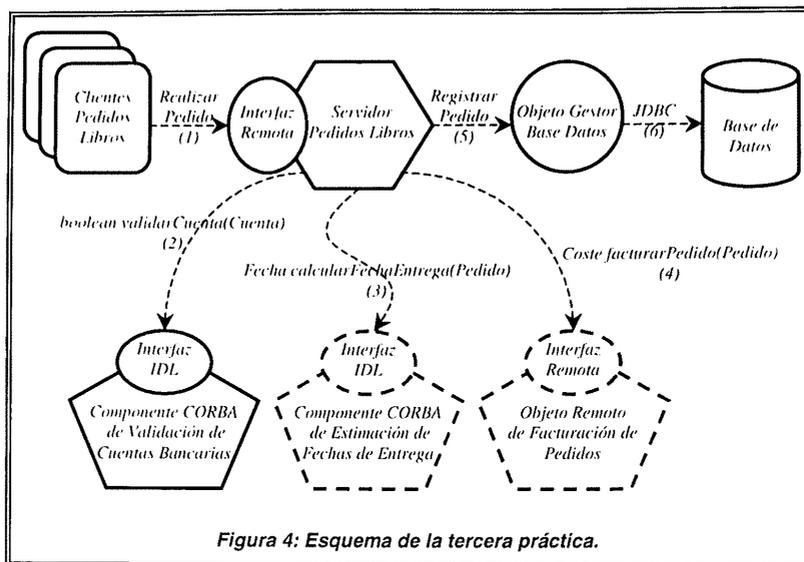
Ésta tendrá dos partes diferenciadas: una primera dedicada a JDBC y una segunda parte para el planteamiento de una práctica de dimensiones mayores, integradora de las tecnologías probadas.

Para explicar el acceso a bases de datos remotas a través de JDBC, se añadirá a la aplicación de Donativos un objeto que se encargue del acceso a la base de datos de manera que nuestros alumnos dispongan de esta información como ejemplo. El nuevo objeto se encargará de, a través de JDBC, ir almacenando en la base de datos todas las donaciones que van realizando los clientes así como la fecha y la hora en que se realiza cada donación (Figura 3).



No plantearemos ninguna ampliación sobre este ejemplo ya que consideramos que van a utilizar los accesos a bases de datos con suficiente detalle en las ampliaciones que se propongan en la siguiente etapa.

En esta segunda parte se integrarán todas las tecnologías usadas en las prácticas anteriores: RMI, CORBA y JDBC. Su finalidad será desarrollar una aplicación distribuida de pedidos de libros (tipo Amazon). Para ello, se proporcionará a los alumnos el núcleo de la aplicación ya implementado y su labor será realizar una serie de ampliaciones sobre él. El funcionamiento de la aplicación será el siguiente: los clientes se conectarán al servidor remoto a través de RMI para realizar sus pedidos. Este Servidor de Pedidos será quien contenga toda la funcionalidad necesaria para realizar un pedido. Para ello, hará uso de una serie de componentes externos que se encargarán de realizar funciones tales como realizar la Facturación del pedido, la Estimación de su fecha de entrega o la Verificación de la validez del nº de cuenta bancaria del comprador. Algunos de estos componentes serán componentes CORBA y otros implementarán interfaces remotas RMI. Además, el servidor almacenará toda la información relativa a los compradores, los libros y los pedidos en una base de datos remota a la que accederá a través de JDBC. Sobre el núcleo de la aplicación los alumnos deberán realizar una serie de ampliaciones (Figura 4).



Atendiendo a sus características, las ampliaciones se pueden clasificar en tres tipos diferentes:

- El primer tipo de ampliación consistirá en utilizar los servicios proporcionados por algunos de los componentes externos con los que se debe comunicar la aplicación. En este caso, los componentes a invocar se proporcionarán ya implementados a los alumnos.
- El segundo tipo consistirá en que los alumnos implementen algunos componentes externos, tanto componentes CORBA como componentes que implementen interfaces remotas y los invoquen.
- El tercer tipo consistirá en añadir alguna funcionalidad nueva a la aplicación como, por ejemplo, la adición de nuevos compradores. Para que esto sea posible, los alumnos deberán realizar desde el cliente invocaciones RMI al servidor y deberán implementar mediante JDBC los accesos que el servidor realice a la base de datos.

Los objetivos a cubrir en esta práctica son que los alumnos sepan integrar en una única aplicación todas las tecnologías estudiadas y que comprendan la arquitectura de una aplicación distribuida de cierta complejidad así como su implementación.

### 3.- NECESIDADES DE SOFTWARE

La herramienta de desarrollo que se usará en las prácticas será *Java 2 SDK*. Esta herramienta incluye todo lo necesario para poder realizarlas: contiene el paquete *java.rmi* que permite realizar invocaciones remotas de métodos a través de RMI, implementa un ORB CORBA, posee un compilador de IDL a Java y proporciona el paquete *java.sql* para el acceso a bases de datos a través de JDBC.

#### 4.- ORGANIZACIÓN DE LAS CLASES PRÁCTICAS

Las clases de prácticas se impartirán en aulas del centro de cálculo donde se dispone de un cañón de proyecciones que permitirá mostrar el código de los ejemplos y su ejecución desde el ordenador del profesor, una pizarra que servirá de apoyo en las explicaciones y ordenadores donde los alumnos podrán ir realizando tanto los ejemplos guiados (ejemplos que realiza toda la clase, paso a paso, al mismo tiempo que el profesor), como las modificaciones propuestas.

Las clases de prácticas se impartirán durante las semanas 6ª, 8ª y 10ª del cuatrimestre (meses de noviembre y diciembre) en grupos de tres sesiones cada semana; por tanto, un total de 9 sesiones de una hora de duración. Evidentemente, esto implica una coordinación entre las clases teóricas y las prácticas para garantizar que los contenidos ya han sido previamente impartidos. (Esto nos ha llevado a modificar la planificación de temas, ya que por ejemplo, originalmente se estudiaría CORBA en el mes de diciembre, trastocando así las prácticas. Hemos solucionado el problema dividiendo ese tema en dos partes: CORBA Básico (con contenido suficiente para las prácticas) y CORBA Avanzado, donde se explica el resto).

Para la realización de las prácticas los alumnos formarán grupos de 2 personas. Cada grupo deberá entregar el código fuente de las tres prácticas, siendo éste requisito indispensable para superar la asignatura. Las entregas de las dos primeras prácticas están planificadas para las semanas 7ª y 9ª del cuatrimestre. La entrega de la tercera la hemos pospuesto hasta el mes de enero por ser la que más trabajo requiere. Además, consideramos importante esta entrega paulatina de los resultados de las prácticas porque la experiencia nos ha demostrado que funciona mejor que unir todas las entregas en una y proporcionar una plazo amplio.

#### 5.- CONCLUSIONES

Hemos descrito el programa práctico que esperamos impartir en el curso 2000-2001 en la asignatura de *Ingeniería del Software III*. Creemos que se adecua para lograr el objetivo fundamental que se persigue con esta asignatura, tanto en su vertiente teórica como práctica: enseñar los conocimientos necesarios para el desarrollo de software distribuido.

#### 6.- BIBLIOGRAFÍA

- [1] R. Cortazar, A. Barredo, J.L. Del Val, "Propuesta de Contenidos para una Asignatura de Ingeniería del Software Distribuido". V Jornadas sobre la Enseñanza Universitaria en Informática (JENUI'99), La Almunia de Dª Godina (Zaragoza), 25 y 26 de Octubre, pg:141-146.
- [2] R. Orfali, D. Harkey, J. Edwards, "The essential Client/Server Survival Guide". John Wiley & Sons, 3ª Ed, 1999.
- [3] G. Seshadri, "Enterprise Java Computing: Applications and Architecture". SIGS Books, 1999.
- [4] Object Management Group, "The common object request broker: architecture and specification". Object Management Group, Revisión 2.3.1. Octubre 1999.
- [5] E. Gamma, R. Helm, R. Johnson, J. Vlissides, "Design Patterns, Elements of Reusable Object-Oriented Software". Addison-Wesley, 1995.