Bases de Datos

Acceso a bases de datos distribuidas mediante el uso de agentes móviles

César F. Acebal, Juan M. Cueva Lovelle
Universidad de Oviedo

<uov01490@correo.uniovi.es>
<cueva@pinon.ccu.uniovi.es>

Resumen: *en este proyecto pretendemos ejemplificar el uso de la reciente tecnología de agentes móviles para el acceso a bases de datos distribuidas a través de Internet. Para llevar a cabo este objetivo hemos tomado como dominio de aplicación el de los Sistemas de Información Geográfica (SIG), y se ha desarrollado una aplicación que recupera información de bases de datos relacionales que soportan la especificación OpenGIS [1].*

Para la implementación se ha utilizado el sistema de agentes móviles Aglets [2], basado en Java, que ha sido desarrollado por IBM Japón. La aplicación se ha programado enteramente en Java y el acceso a las bases de datos remotas se consigue mediante la tecnología JDBC que proporciona el propio lenguaje.

Palabras clave: *Agentes móviles,* ***aglets****, Java, JDBC, SQL, Sistemas de Información Geográfica, SIG, OpenGIS, objetos distribuidos.*

1. Introducción

Con el advenimiento de la era Internet y la globalización económica cada vez son más las empresas que experimentan la necesidad de compartir recursos geográficamente muy distantes unos de otros. De estos recursos, la información almacenada en bases de datos empresariales ocupa un lugar esencial [3]. La red Internet ofrece la infraestructura adecuada para conectar estos recursos a través de una amalgama de máquinas, sistemas operativos y redes de ordenadores de diferentes tipos.

En este contexto, la saturación del ancho de banda de la red se convierte en el problema clave a solucionar y aquí es donde aparece la tecnología de agentes móviles, al permitir interactuar localmente con esas bases de datos remotas.

Este artículo ilustra un posible uso de la tecnología para el acceso a bases de datos geoespaciales, según propone el *OpenGIS Consortium*, pero en la práctica permite conectarse y recuperar información de cualquier base de datos relacional, gracias a las facilidades ofrecidas por la tecnología *JDBC* de *Java* y el diseño abierto que se ha seguido en el desarrollo del proyecto.

2. Agentes móviles

Pero, ¿en qué consiste la tecnología de agentes móviles? ¿Es simplemente la nueva palabra de moda de la que pronto empezará a hablar todo el mundo --sin saber bien a qué se está haciendo referencia, como tantas veces ha sucedido en este mundo de la informática-- o se trata realmente de un nuevo y útil modelo de procesamiento distribuido?

A menudo se habla de «agente software» para referirse a más cosas que simplemente los agentes móviles. En este sentido, puede hablarse también de agentes inteligentes y representantes.

En el primer caso, los agentes son piezas de software a las que se ha dotado con algún grado de inteligencia artificial. En el segundo, pueden actuar como representantes para la realización de determinadas tareas sin nuestra presencia. Dependiendo de las instrucciones que les hayamos dado, podrían incluso tomar decisiones por cuenta propia en nuestro nombre, como realizar compras, reservar billetes, etcétera.

Estos dos términos pueden combinarse entre sí y con el de móviles, de manera que un agente puede ser inteligente pero no móvil, o ser ambas cosas a la vez, o ser también representante... y así sucesivamente.

Ahora bien, ¿qué entendemos realmente por agente móvil? De una manera sencilla podemos definir agente móvil como [4]: un programa capaz de detener su ejecución, viajar a través de una red (manteniendo intactos tanto el código como los datos) y reanudar la ejecución en un ordenador remoto.

2.1. ¿Por qué usar agentes móviles?

La tecnología de agentes móviles soluciona (o pretende solucionar) diversos problemas en diversos frentes. Por un lado, proporciona una solución al derroche de ancho de banda que se produce en la red en una arquitectura cliente/servidor. Este ancho de banda en una aplicación distribuida es un bien escaso y, por tanto, valioso. Una transacción o consulta realizada entre un cliente y el servidor puede requerir bastantes viajes por la red para completarse, cada uno de los cuales provoca un cierto tráfico de datos y consume ancho de banda. En un sistema en el que tengamos muchos clientes o mucho volumen de transacciones (o ambas cosas a la vez) posiblemente se sobrepase el ancho de banda disponible, lo que se traducirá en una disminución del rendimiento de la aplicación completa. Aplicando a este problema la tecnología de agentes móviles podríamos crear un agente que, dada la consulta o transacción a realizar, se trasladase desde el cliente al servidor, completase en él la operación y regresase con los resultados de la misma, necesitando de esta manera sólo dos viajes por la red al eliminar todo el tráfico intermedio de datos y resultados. En la **figura 1** podemos ver gráficamente este ahorro de ancho de banda.

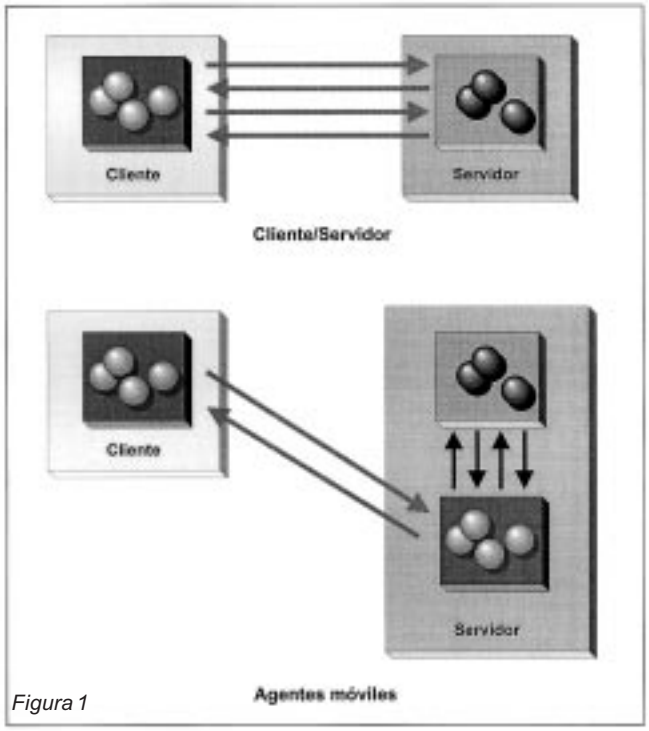


Figura 1

Mientras que los sistemas distribuidos tradicionales proporcionan transparencia de localización y datos móviles para permitir la comunicación entre objetos distribuidos, los sistemas basados en agentes móviles difieren de aquéllos en que proporcionan interacción local entre los objetos y tanto los datos como el código son móviles, es decir: es la computación misma la que se traslada hasta el sistema remoto a través de la red, mientras que en los modelos anteriores de objetos distribuidos lo que realmente se transmite son las llamadas a métodos remotos.

Otra cuestión es que en el diseño de una arquitectura tradicional cliente/servidor, el arquitecto de sistemas especifica los papeles del cliente y servidor de modo muy preciso en tiempo de diseño. El arquitecto toma estas decisiones sobre dónde residirá una funcionalidad concreta basándose en las restricciones del ancho de banda, tráfico en la red, volumen de transacciones, número de clientes y servidores, y muchas otras cuestiones por el estilo. Si estas estimaciones fueran incorrectas, o aun siendo válidas, el arquitecto tomase decisiones erróneas, el rendimiento de la aplicación se vería seriamente afectado.

Desgraciadamente, una vez que el sistema ha sido construido suele ser difícil, si no imposible, cambiar el diseño para solucionar estos problemas. Sin embargo, las arquitecturas basadas en agentes móviles son potencialmente mucho menos sensibles a este problema. En tiempo de diseño son necesarias menos decisiones y, sobre todo, el sistema es mucho más fácil de modificar después de construido.

Las arquitecturas de agentes también resuelven los problemas creados por conexiones de red intermitentes o poco fiables. En la mayoría de las aplicaciones de red de hoy día es necesario que la conexión esté activa todo el tiempo que dura la transacción o consulta. Si esta conexión deja de funcionar, el cliente a menudo vuelve a comenzar la transacción o consulta desde el principio, si es que es capaz de reiniciarla. Por el contrario, la tecnología de agentes móviles permite a un cliente despachar un agente hasta su destino cuando existe la conexión de red y luego desconectarse. El agente realizará la consulta o transacción por sí mismo y devolverá los resultados al cliente cuando restablezca la conexión.

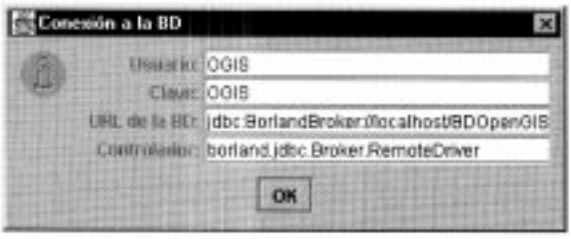


Figura 2. Configuración de la conexión a la base de datos

2.2. *Aglets*: un sistema de agentes móviles

El proyecto que se presenta en este artículo ha utilizado el sistema de agentes móviles *Aglets*, basado en *Java*, que ha sido desarrollado por IBM Japón [2]. Dicho sistema de agentes consiste en una *API* con una serie de clases e interfaces *Java* que proporcionan la movilidad y comunicación entre agentes. Asimismo, ofrece un contexto en el que los *aglets* pueden ejecutarse de forma segura. Por otro lado, al estar desarrollado en *Java* ofrece la necesaria portabilidad entre plataformas. De hecho, *Java* se está convirtiendo en el estándar de facto como lenguaje para la implementación de los distintos sistemas de agentes móviles [5].

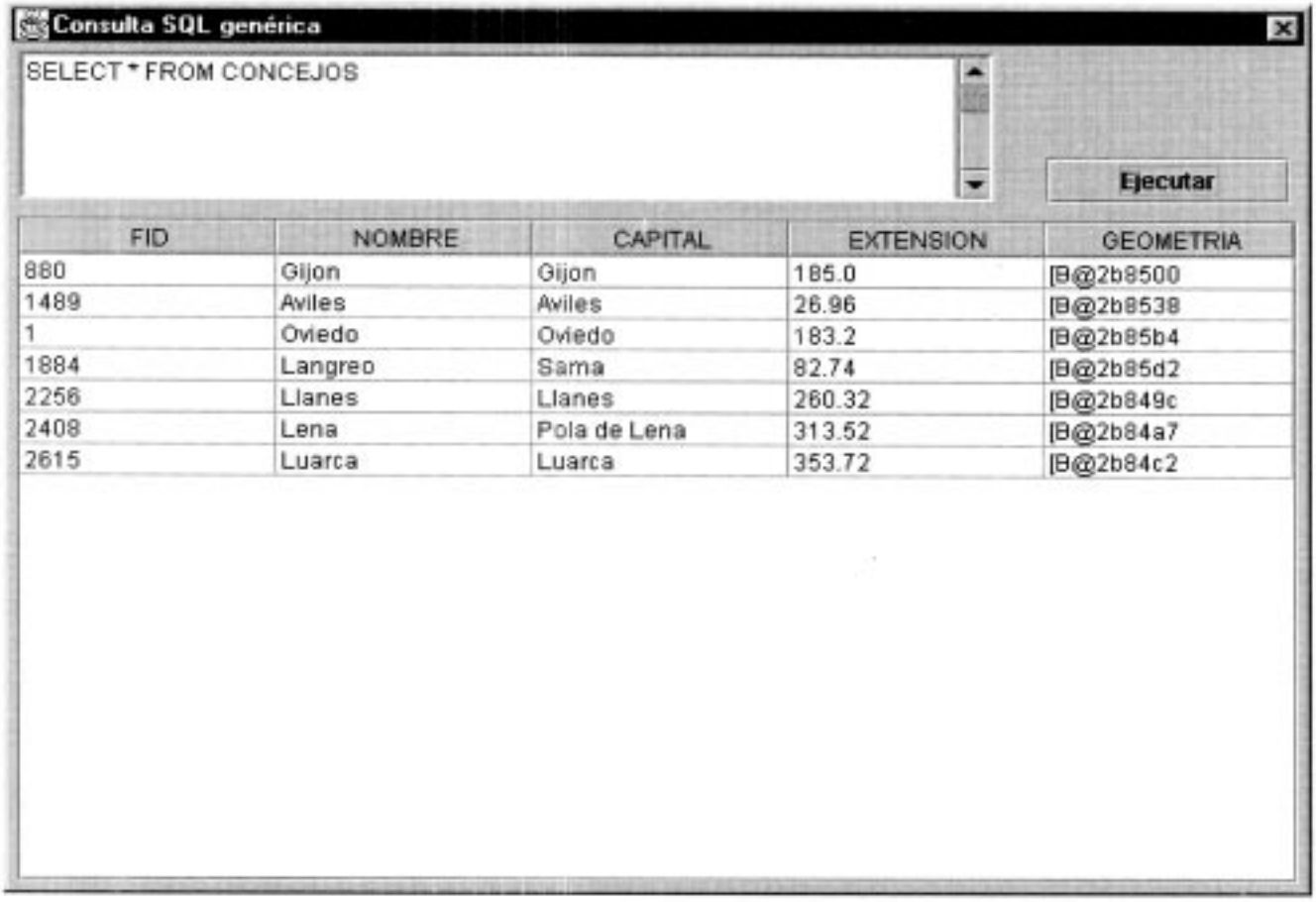
3. Una aplicación de los agentes móviles: recopilación de información de lugares dispersos

Nuestro proyecto, como ya se ha dicho, ejemplifica el uso de la tecnología de agentes móviles para la consulta y recuperación de información de bases de datos remotas. Aunque para el desarrollo de la aplicación de ejemplo se ha escogido como dominio de aplicación el de los Sistemas de Información Geográfica, ofreciendo una serie de consultas estadísticas sobre concejos (ayuntamientos), lo cierto es que la aplicación permite conectarse a cualquier base de datos relacional y realizar consultas *SQL* de modo interactivo, mostrando los resultados de las mismas en formato tabular.

3.1. Configuración de la consulta

La **figura 2** representa la ventana desde la que el usuario puede especificar los parámetros necesarios para que el *aglet* se pueda conectar mediante *JDBC* a la base de datos remota. Esto permite sustituir las bases de datos de los servidores remotos sin más que modificar aquí los parámetros necesarios. Asimismo, otro concepto fundamental en los agentes móviles es el de «itinerario». Así, nuestro agente no tiene por qué limitarse a consultar una sola base de datos, sino que se le puede indicar que recorra tantos servidores remotos como sea preciso, realizando la consulta en todos ellos (también podríamos indicarle, por ejemplo, que consultase primero los metadatos disponibles para ejecutar en función de ellos consultas diferentes en cada uno).

3.2. Ejemplo de consulta *SQL* interactiva



FID	NOMBRE	CAPITAL	EXTENSION	GEOMETRIA
880	Gijon	Gijon	185.0	[B@2b8500
1489	Aviles	Aviles	26.96	[B@2b8538
1	Oviedo	Oviedo	183.2	[B@2b85b4
1884	Langreo	Sama	82.74	[B@2b85d2
2256	Llanes	Llanes	260.32	[B@2b849c
2408	Lena	Pola de Lena	313.52	[B@2b84a7
2615	Luarca	Luarca	353.72	[B@2b84c2

Figura 3. Ejemplo de consulta SQL interactiva

Una vez que se han seleccionado los servidores remotos a los que viajará el *aglet* y los parámetros de conexión a la base de datos, el usuario está en condiciones de ejecutar cualquiera de las consultas disponibles. Como ejemplo de la versatilidad del sistema, se muestra (**figura 3**) la ventana desde la que se puede introducir una sentencia *SQL* interactiva, junto con los resultados de la misma en la parte inferior.

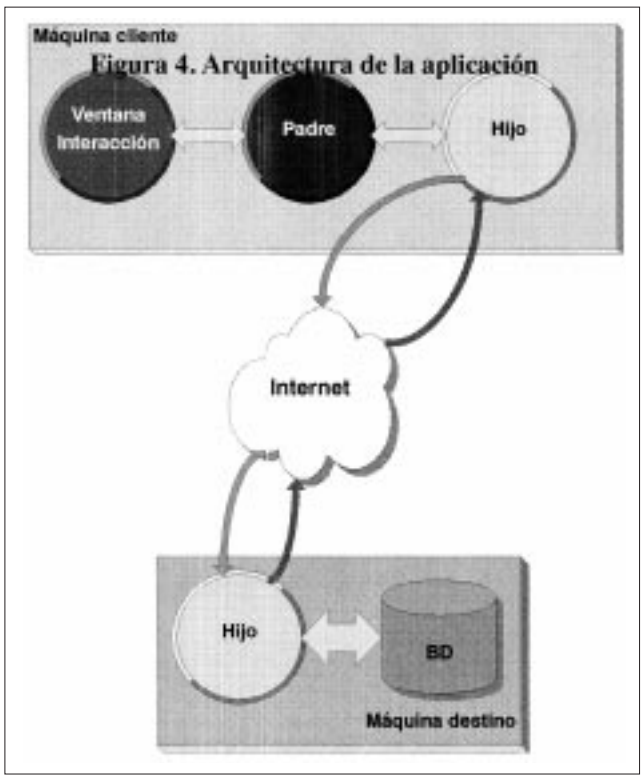


Figura 4

3.3. Arquitectura del sistema

Un «patrón de diseño» usual en las escasas aplicaciones que hacen uso de la tecnología de agentes móviles hoy día consiste en emplear un *aglet* estacionario en la propia máquina cliente, es decir, un *aglet* que no puede ser despachado a otros servidores y cuya misión es la de proporcionar una interfaz de comunicación entre la aplicación cliente y otros *aglets* creados por éste (a los que llamaremos por este motivo *aglets* hijo --y padre al estacionario). Estos *aglets* hijo son los que sí se despachan al *host* o *hosts* remotos especificados a realizar la labor para la que hayan sido programados. Esta arquitectura ha sido la que hemos empleado en nuestro proyecto, la cual se muestra gráficamente en la **figura 4**. Este diseño puede observarse también en el esbozo del diagrama de clases principales utilizadas en la ejecución de una consulta (**figura 5**).

4. Conclusiones

En este artículo hemos querido ofrecer una somera introducción a lo que son los agentes móviles, así como a uno de sus posibles y más claros usos: la consulta de bases de datos remotas con un impacto mínimo en la carga de la red, al eliminar todo el tráfico de resultados intermedios que se produce con las arquitecturas tradicionales de objetos distribuidos.

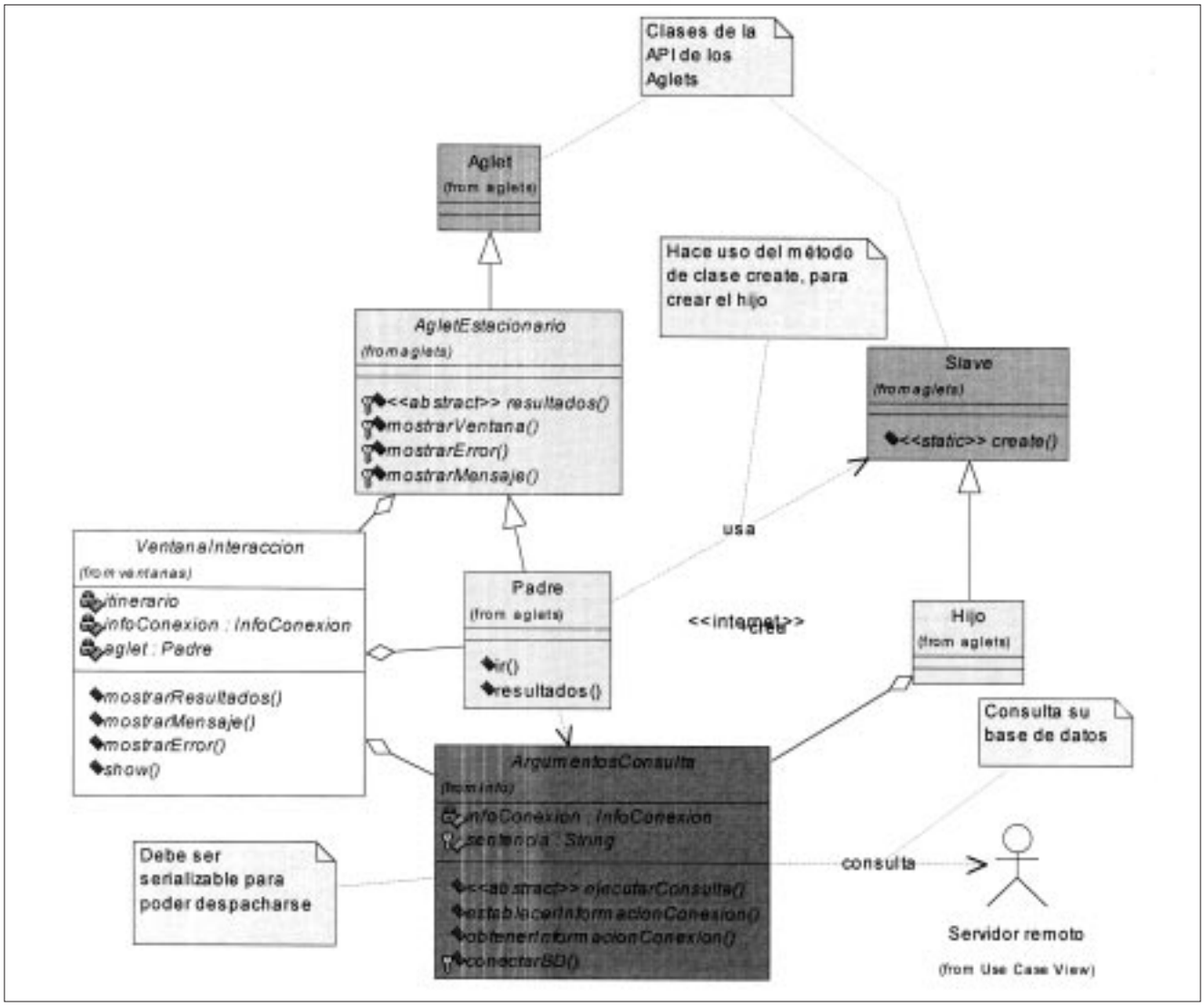


Figura 5. Esbozo del diagrama de clases principales que intervienen en una consulta

Para ello se presenta una aplicación real que hace uso de dicha tecnología [6]. En el desarrollo de esta aplicación creemos conseguir un patrón de diseño para futuras aplicaciones que requieran esta misma funcionalidad, esto es, la recuperación de información de bases de datos distribuidas. Este diseño ha sido facilitado gracias a la elección de *Java* como lenguaje de programación, que nos permite una fácil conexión a dichas bases de datos desde el propio lenguaje gracias a la tecnología *JDBC*.

Por último, aunque esta tecnología emergente no ha logrado aún consolidarse, creemos oportuna contemplarla como una opción más a tener en cuenta entre la ya abundante oferta existente de lo que se ha dado en llamar *middleware*. Con este objetivo hemos querido introducirla en este artículo al lector desconocedor de la misma, junto con una aplicación de ésta en un ámbito para el que nos parece adecuada.

Referencias

[1] *Open GIS Consortium, Inc.* (*OGC*); *www.opengis.org*
[2] *Aglets Software Development Kit* (*ASDK*) ; *www.trl.ibm.co.jp/aglets*
[3] Todd Papaioannou and John Edwards*; Mobile Agent Technology Enabling The Virtual Enterprise: A Pattern for Database Query*; *luckyspc.lboro.ac.uk/Docs/Papers/*
[4] Bill Venners; «Solve real problems with aglets, a type of mobile agent». *Javaworld, mayo 1997. www.javaworld.com*
[5] Jesús Arturo Pérez Díaz, Juan Manuel Cueva Lovelle y Benjamín López Pérez; «Different plataforms for mobile agents development. Is Java the dominant trend?», *International Workshop on Intelligent Agents on the Internet and Web, World Congress on Expert Systems*. Mexico City, Mexico, marzo 1998.
[6] César F. Acebal; *Agentes móviles aplicados a los Sistemas de Información Geográfica*. Proyecto Fin de Carrera, E.U. Ingeniería Técnica en Informática de Oviedo (Universidad de Oviedo), septiembre de 1998.

Nota

César F. Acebal es Ingeniero Técnico en Informática de Gestión; actualmente estudia 2º ciclo de Ingeniería en Informática en la Escuela Técnica Superior de Ingenieros Industriales e Ingenieros Informáticos de Gijón (Universidad de Oviedo).

Juan Manuel Cueva Lovelle es Catedrático de E. U. de Lenguajes y Sistemas Informáticos de la Universidad de Oviedo y Director de la Escuela Universitaria de Ingeniería Técnica en Informática de Oviedo (Universidad de Oviedo).