

# Comparativa de herramientas para la enseñanza de lenguajes relacionales

Javier J. Gutiérrez, María J. Escalona, Darío Villadiego, Manuel Mejías

Dpto. de Lenguajes y sistemas Informáticos

Universidad de Sevilla

Avd. Reina Mercedes s/n

41012 Sevilla

e-mail: {javierj, escalona, risoto}@lsi.us.es

## Resumen

El estudio del modelo relacional aplicado a las bases de datos es una parte fundamental en los estudios de toda Ingeniería Informática, tanto técnica como superior. Los lenguajes de consulta relacionales forman una parte importante de este estudio. Nuestra experiencia nos ha mostrado que, para los alumnos, es difícil conocer cuándo las consultas expresadas en un papel en términos de estos lenguajes son correctas y responden satisfactoriamente a los enunciados planteados.

En este trabajo se describe una herramienta de apoyo que permite realizar consultas expresadas en lenguajes relacionales sobre cualquier base de datos. Esta herramienta se ha desarrollado siguiendo el modelo del software libre. Además este trabajo expone los resultados de un estudio comparativo entre esta herramienta y otras herramientas que se puede descargar libremente de Internet.

## 1. Motivación

Actualmente el modelo relacional para organizar y gestionar sistemas de bases de datos sigue estando de plena vigencia. Por ello, cobran una importancia vital a la hora de la enseñanza de sistemas de bases de datos los fundamentos de este modelo. Estos fundamentos incluyen sus lenguajes relacionales, ya que son la base teórica sobre la que se han desarrollado todas las herramientas de bases de datos que los alumnos van a manipular en sus profesiones [13]. La importancia del aprendizaje del modelo relacional ha sido expuesta en varios trabajos como [7], [9] y [13].

Es muy importante, en este proceso de aprendizaje de los lenguajes relaciones, disponer de herramientas que permitan ejecutar expresiones

y mostrar sus resultados. Con herramientas de este tipo es posible motivar a los alumnos a que investiguen por ellos mismos las posibilidades de los lenguajes relacionales, facilitar el entendimiento de los operadores relacionales, desarrollar nuevas expresiones y comparar las ventajas de utilizar diversas expresiones relacionales. En resumen, mejorar las técnicas de autoaprendizaje de la materia [9].

Este trabajo se divide en dos partes. En la primera se presenta una herramienta libre de código abierto llamada RelationalQuery o RQuery [12]. El objetivo principal de esta herramienta es ofrecer a los alumnos la posibilidad de poder ejecutar consultas en los lenguajes relacionales clásicos. En la segunda parte se realiza un análisis comparativo entre herramientas para ejecutar consultas en lenguajes relacionales.

La sección 2 ofrece una breve introducción al modelo relacional y a los lenguajes relacionales. La sección 3 detalla las características de la herramienta RQuery. La sección 4 muestra el análisis comparativo. Por último, la sección 5 exponen las conclusiones y los futuros trabajos.

## 2. Introducción al modelo relacional y a los lenguajes relacionales

El modelo de datos relacional fue presentado por Tedd Codd en 1.970 [3]. Este modelo está basado en el concepto de relación matemática y el conjunto de manipulaciones que es posible llevar a cabo sobre estas relaciones. La base de las manipulaciones permitidas sobre el modelo la ofrecen los lenguajes relacionales. Originalmente Codd definió dos lenguajes relacionales: álgebra relacional y cálculo relacional como la base de los modelos relacionales.

Lenguaje	Consulta
Álgebra relacional	<code>project nombre (empleado)</code>
Cálculo relacional orientado a tuplas	<code>X : empleado / X.nombre</code>
Cálculo relacional orientado a predicados	<code>empleado(x: nombre)</code>
SQL	<code>SELECT nombre FROM empleado;</code>

Tabla 1. Ejemplo de consulta en los distintos lenguajes relacionales y SQL.

El álgebra relacional es un lenguaje procedural o de alto nivel, mientras que el cálculo relacional es un lenguaje no procedural. Sin embargo, ambos lenguajes son equivalentes. Para cada expresión del álgebra, se puede encontrar una expresión equivalente en el cálculo, y viceversa.

El cálculo relacional tradicionalmente se ha dividido en dos lenguajes: el cálculo relacional orientado a tuplas y el cálculo relacional orientado a predicados, según el tipo de variables que se manejan. El cálculo relacional orientado a tuplas (CRT) emplea variables-tupla, las cuales pueden tomar el valor de cualquier tupla de la relación. En el cálculo relacional de dominios (CRD) se utilizan variables-dominio, que toman valores de los dominios asociados a los campos de las relaciones.

En la tabla 1, se muestra una consulta para obtener los nombres de los empleados, tanto en álgebra relacional, como en los cálculos relacionales y SQL.

Tanto el álgebra relacional como los cálculos relacionales son lenguajes formales, muy matemáticos y poco amigables. Sin embargo deben estudiarse porque sirven para ilustrar las operaciones básicas que todo lenguaje de manejo de datos debe ofrecer. Además, han sido la base para otros lenguajes relacionales de manejo de datos de más alto nivel [7], [13].

En la siguiente sección se describe brevemente nuestra herramienta creada para facilitar el aprendizaje y práctica con estos lenguajes.

### 3. Descripción de la herramienta RelationalQuery

#### 3.1. ¿Por qué una nueva herramienta?

Para contestar correctamente a esta pregunta sería necesario estudiar las herramientas existentes en

la actualidad. Este estudio se ha pospuesto para la sección 4 con el fin de incluir en él a esta herramienta. Si se puede adelantar que, como se verá con mayor detalle en la siguiente sección, no se ha encontrado ninguna herramienta que cumpla todas las siguientes características. Estas características se describirán con más detalle en la comparativa y conclusiones.

- Herramienta libre con código fuente disponible libremente por Internet.
- Multiplataforma.
- En español.

Siguiendo las recomendaciones de [5], esta herramienta se plantea para el marco concreto de facilitar el aprendizaje en carreras universitarias.

#### 3.2. Descripción de la arquitectura

RQuery se ha desarrollado siguiendo el modelo del software libre [11]. Una de las ideas principales de este modelo es reutilizar tantos componentes como sea posible. En la figura 1 se muestran los distintos componentes libres. Esta reutilización acorta el tiempo de desarrollo y aumenta la calidad del mismo al emplear código ya probado y depurado.

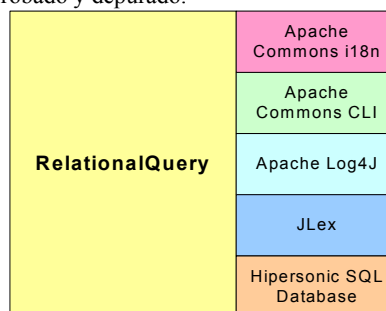


Figura 1. Componentes libres utilizados en RelationalQuery.

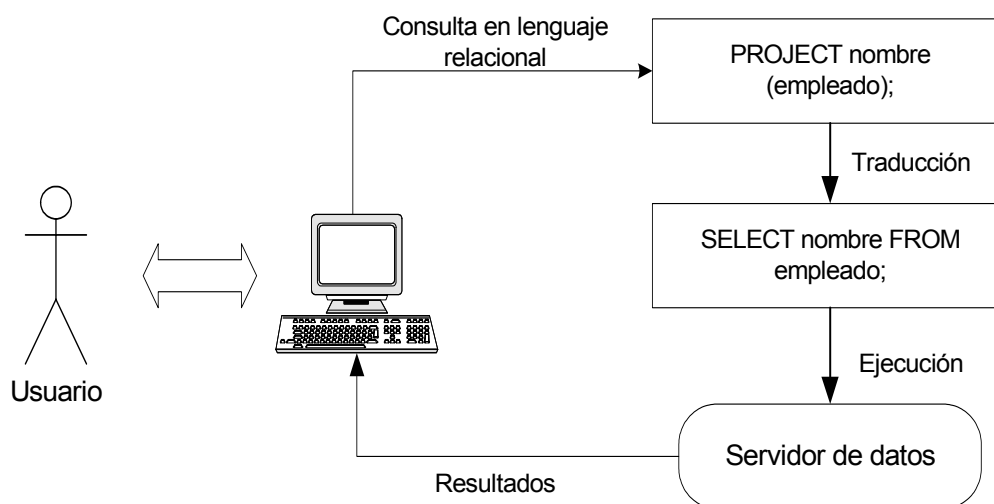


Figura 2. Proceso de traducción y ejecución de una consulta.

Las referencias a los componentes utilizados se pueden encontrar en [2] y [8].

Para el desarrollo de esta herramienta se ha optado por delegar todas las operaciones de consulta sobre un servidor de bases de datos relacionales. La aplicación realiza una traducción de expresiones en lenguaje relacional a consultas SQL. Estas son ejecutadas en el servidor de bases de datos. Como se puede ver en la figura 1, la base de datos seleccionada ha sido Hipersonic SQL [8] por estar escrita íntegramente en Java, soportar SQL-92 y disponer de una versión con licencia libre.

Las ventajas de esta aproximación respecto de codificar directamente las consultas sobre los datos, se enumeran a continuación.

- Es posible utilizar cualquier base de datos que disponga de un conector o driver JDBC
- Es posible utilizar cualquier herramienta de administración de bases de datos.
- Es posible aprovechar toda la información almacenada en bases de datos ya existentes.
- Ayuda a los alumnos a prácticas tareas de gestión de bases de datos en SQL.
- Ofrece un conjunto de dominios muy ricos (todos los tipos de datos soportados por SQL-92).

- Permite reutilizar componentes ya creados, como herramientas de bases de datos, o herramientas para análisis léxico y semántico.

El proceso de traducción y ejecución de una consulta se muestra en la figura 2.

A la hora de implementar los distintos lenguajes relacionales se ha optado por una arquitectura modular donde cada uno de los lenguajes tiene su propio conjunto de módulos, y comparte otros módulos comunes con el resto de la aplicación. Los módulos propios de cada lenguaje son el analizador léxico, sintáctico y traductor a sentencias SQL, ya que cada lenguaje tiene unas reglas de traducción distintas. Las interfaces gráficas, factorías, y módulos de ejecución de consultas y acceso a bases de datos son compartidos por todos los lenguajes.

En la figura 3 se muestran los componentes comunes a todos los lenguajes y los componentes dependientes de cada uno de los lenguajes relacionales soportados.

En las dos siguientes secciones se describe con más detalle las características más importantes de RQuery.

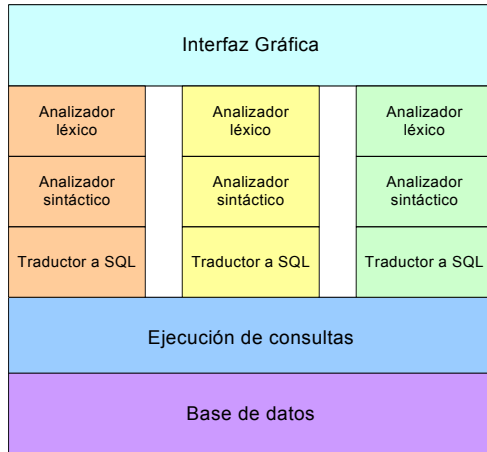


Figura 3. Estructura modular de la aplicación.

### 3.3. Traducción sistemática de expresiones de lenguajes relacionales a SQL

Como se ha visto en la sección anterior, el corazón de RQuery es el traductor de consultas de lenguajes relacionales a SQL. No ha sido el objetivo de este trabajo crear ningún método formal, simplemente crear algo práctico que se pudiera implementar y de utilidad para la comunidad educativa.

Cada lenguaje ha sido estudiado de manera independiente y se ha desarrollado un mecanismo sistemático y automatizable para obtener un equivalente en SQL de cualquier consulta

expresable en cualquiera de los tres lenguajes relacionales.

Actualmente está completamente implementada la traducción desde álgebra relacional a SQL con la excepción del operador división. Respecto a los lenguajes del cálculo relacional, ya se ha desarrollado los mecanismos de traducción, pero aún no han sido implementados.

### 3.4. Interfaces de usuario.

RelationalQuery ofrece varias interfaces de usuario en función de la preferencia de cada usuario o de la potencia de la máquina.

La más básica es una interfaz de línea de comandos que permite ejecutar en modo de proceso por lotes un archivo de texto con consultas. Una captura de esta interfaz se muestra en la figura 4.

RelationalQuery también incluye dos interfaces gráficas escritas en Java/SWING. La primera es una interfaz extremadamente sencilla y adecuada para equipos menos potentes. Esta interfaz divide la ventana en dos áreas, en el área superior se ejecutan las consultas mientras que el área inferior se muestran los resultados. Una captura de ejemplo se muestra en la figura 5.

Tanto en esta, como en la interfaz anterior, es necesario indicar el lenguaje a utilizar desde la línea de comandos, por lo que no es posible cambiar de lenguaje sin salir de la aplicación y volver a ejecutarla.

```

c:\PLI\FreeQuery>rquerydemo
-df: ../demo/database.sql
-qf: ../demo/queries.ar
-type: RA

Not Persistent : ../demo/database.sql
Init: ../demo/database.sql
ParserBuilder: ../demo/database.sql
End Init: ../demo/database.sql

Query: restrict [ not < name LIKE 'Tool 1' > ] < tools > ;
Result:
! COD_I ! NAME !
-----
! T2    ! Tool 2 !
! T3    ! Tool 3 !
! T4    ! Tool 4 !
! T5    ! Tool 5 !

```

Figura 4. Interfaz de línea de comandos de RQuery.

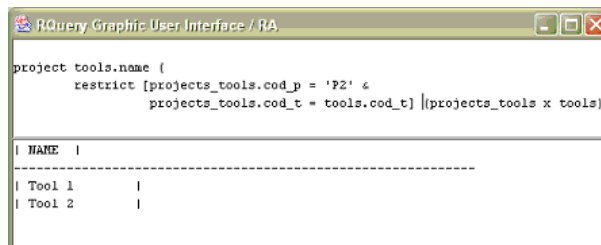


Figura 5. Interfaz gráfica básica.

La tercera interfaz gráfica es una versión evolucionada de la anterior y ofrece más opciones, como la posibilidad de almacenar y recuperar consultas o la opción de cambiar de lenguaje sin necesidad de abandonar la aplicación. Una captura de ejemplo se muestra en la figura 6.

El componente de interfaces gráficas es uno de los aspectos más activos dentro del proyecto RelationalQuery. En futuras interfaces está previsto incorporar más opciones que faciliten la edición y depuración de consultas, así como una interfaz multiventana (MDI) para poder tener varias consultas activas al mismo tiempo.

#### 4. Comparativa de herramientas de lenguajes relacionales

En este apartado se analizan y comparan brevemente tres herramientas que permiten la ejecución de consultas en álgebra relacional. Las tres herramientas seleccionadas son LEAP [10], RelationalQuery [12] y WinRDBI [14]. Se han elegido estas herramientas porque cumplen dos condiciones: la primera que es cualquiera pueda

descargarlas de Internet y la segunda que estén lo suficientemente desarrolladas para poder ser utilizadas.

No se han podido incluir en esta comparativa herramientas españolas como [7] por no estar disponibles para descarga.

##### 4.1. Breve presentación de las herramientas

RQuery [12] ya ha sido comentada con detalle en los apartados anteriores, por lo que esta sección va a describir brevemente las otras dos herramientas.

WinRDBI [14] consta de una interfaz de usuario escrita en Java, la cual ejecuta las consultas mediante una librería Prolog implementada con librerías dinámicas DLL, lo cual no permite su utilización en una plataforma distinta de Windows. Existe también una versión para Linux disponible para descarga. Esta herramienta soporta SQL, álgebra relacional y los lenguajes del cálculo relacional. En la figura 7 se muestra una captura de esta herramienta.

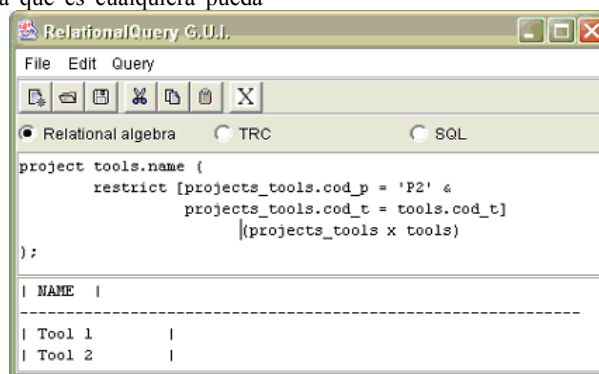


Figura 6. Interfaz gráfica avanzada.

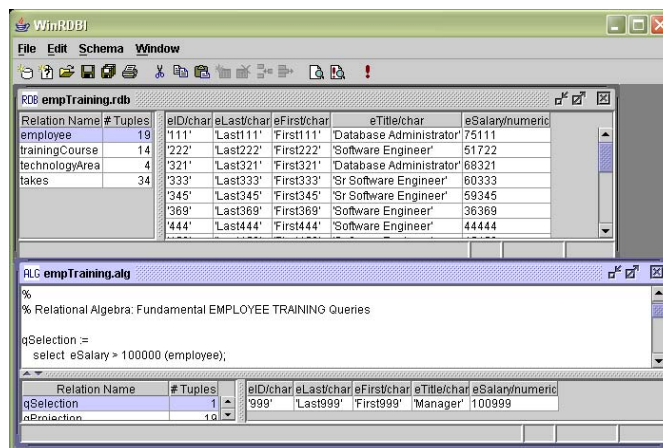


Figura 7. Herramienta WinRDBI en acción.

LEAP [10] se ha desarrollado en C, por lo que debe ser compilado en cualquier plataforma donde se quiera utilizar. Actualmente solo se distribuye una versión compilada para entornos Windows. Esta herramienta sólo soporta álgebra relacional. Una captura de esta herramienta se muestra en la figura 8.

En la tabla 4 se recogen las versiones de las herramientas analizadas en esta comparativa. En el apartado 4.2 se detallan las características analizadas de cada herramienta así como los resultados de la comparativa.

Herramienta	Versión
LEAP	1.2.5
WinRDBI	3.10
RelationalQuery	0.010

Tabla 2. Versiones de las herramientas empleadas en la comparativa.

#### 4.2. Comparativa de herramientas

Se han establecido 14 características para ser analizadas en cada una de las herramientas. Estas características están centradas en describir la utilidad pedagógica de la herramienta y las facilidades de uso que ofrecen a los alumnos. Se ha establecido una clasificación donde, para cada categoría, se indica la herramienta que más la satisface con un primer puesto, la que queda en un nivel intermedio y la que más pobremente la satisface, o no la satisface en absoluto, con un tercer puesto. En la tabla 4 se muestran las 14 características analizadas para cada una de las herramientas en esta comparativa.

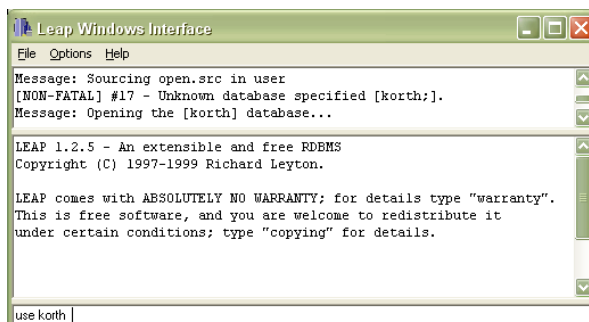


Figura 8. Interfaz gráfica de la herramienta

LEAP.Característica	1° puesto	2° puesto	3° puesto
Calidad de la documentación	RelationalQuery / LEAP	WinRDBI	X
Cantidad de ejemplos	WinRDBI	LEAP	RelationalQuery
Cantidad de interfaces de usuario disponibles	RelationalQuery	LEAP	WinRDBI
Código fuente disponible	RelationalQuery / LEAP	WinRDBI	X
Facilidad de descarga	RelationalQuery / LEAP	WinRDBI	X
Facilidad de administración	WinRDBI	RelationalQuery	LEAP
Idiomas de la interfaz de usuario	RelationalQuery	LEAP	WinRDBI
Lenguajes de consultas	WinRDBI	RelationalQuery	LEAP
Licencia más libre	RelationalQuery	LEAP	WinRDBI
Multiplataforma	RelationalQuery	LEAP	WinRDBI
Opciones de configuración	LEAP	RelationalQuery	WinRDBI
Operadores de álgebra relacional	LEAP	RelationalQuery	WinRDBI
Sencillez de uso	WinRDBI	RelationalQuery	LEAP
Servidor centralizado	RelationalQuery	LEAP	WinRDBI

Tabla 3. Categorías analizadas.

Las herramientas RelationalQuery y LEAP han obtenido el primer puesto de calidad de la documentación ya que ambas cuentan con un manual extenso y actualizado. La documentación de WinRDBI, en cambio, es pobre y obsoleta. Sin embargo, aunque el número de ejemplos que se incluyen en LEAP es el mayor de todas las herramientas, WinRDBI se alza con el primer puesto ya que se distribuye con un completo ejemplo implementado en los cuatro lenguajes que soporta

LEAP, al igual que RelationalQuery, ofrece también una interfaz gráfica y otra de línea de comandos. Sin embargo su interfaz gráfica es muy pobre y no ofrece distintas alternativas para equipos más o menos potentes.

Tanto RelationalQuery como LEAP pueden descargarse libremente de sus respectivos sitios web incluido su código fuente. Para descargar WinRDBI, en cambio, es necesario registrarse en un formulario y esperar a recibir el enlace adecuado a través del correo electrónico.

Tanto LEAP como WinRDBI solo están disponibles en inglés. Sin embargo al estar disponible el código fuente de LEAP sería posible traducirla al español. Al estar los mensajes de la aplicación incrustados en el código sería necesario recompilar la herramienta. RelationalQuery es la herramienta mejor preparada para soportar nuevos idiomas ya que todos los mensajes del sistema se

almacenan en archivos XML externos fácilmente modificables.

En el número de lenguajes de consultas disponibles, actualmente WinRDBI es un claro ganador. RelationalQuery actualmente sólo soporta álgebra relacional y SQL, mientras que LEAP sólo soporta álgebra relacional.

La licencia que otorga más libertad es la licencia de RelationalQuery [1], ya que tanto herramienta como sus componentes pueden reutilizarse en proyectos libres o propietarios. La licencia de LEAP [6] sólo permite reutilizarla en programas libres con la misma licencia o compatible, mientras que WinRDBI se distribuye bajo una licencia propietaria que no permite su redistribución, lo cual es un serio inconveniente en entornos educativos.

Al estar desarrollada en Java, RelationalQuery puede ejecutarse en cualquier plataforma que disponga de una máquina virtual Java sin ninguna modificación. LEAP puede ejecutarse en cualquier plataforma que disponga de compilador de lenguaje C, pero es necesario compilar el código para cada plataforma específica. WinRDBI solo distribuye versiones para sistemas Windows y Linux.

LEAP es la herramienta que más operadores del álgebra relacional implementa, ya que incluye todos los operadores definidos. RelationalQuery incluye todos los operadores del álgebra relacional

menos la división, y WinRDBI no incluye división ni intersección.

Una característica importante en una herramienta pedagógica es permitir que el conjunto de datos esté situado en un único ordenador, y diversas copias de la aplicación puedan acceder a estos datos. Con esto se consigue que todos los alumnos tengan un conjunto consistente de datos de prueba, facilitar la instalación y minimizar el mantenimiento. La herramienta que mejor implementa esta característica es RQuery, ya que la tecnología JDBC aísla tanto de la marca del servidor de bases de datos como de su ubicación. LEAP también proporciona esta característica aunque debe realizarse una configuración especial de la herramienta, mientras que WinRDBI no la soporta. La clasificación final de las herramientas, según el número de puestos de cada tipo obtenidos, se muestra en la tabla 5.

Herramienta	1º puesto	2º puesto	3º puesto
RelationalQuery	8	5	1
LEAP	5	6	3
WinRDBI	4	3	7

Tabla 4. Clasificación final de las herramientas.

En la siguiente sección se exponen las conclusiones que se pueden extraer de esta comparativa.

## 5. Conclusiones

Como planteamos en la sección 3.1, esta comparativa justifica la creación de una nueva herramienta. Sin embargo, a pesar de los buenos resultados obtenidos en la comparativa, RelationalQuery no es aún la herramienta definitiva. Su éxito en la comparativa se debe a que ha podido partir de la base de los errores de las anteriores herramientas. Tampoco ninguna de las otras dos herramientas analizadas es la herramienta definitiva. A pesar de todas sus carencias, WinRDBI es actualmente la única herramienta que ofrece soporte para todos los lenguajes relacionales y la internaz de usuario más sencilla. RQuery espera ofrecer todos los lenguajes en un corto periodo de tiempo. LEAP ha sido diseñado y construido solo para álgebra relacional, por lo que es muy improbable que incorpore otros lenguajes en un futuro. Uno de factores principales en los buenos resultados de RQuery ha sido la filosofía de código abierto [11] que ha permitido reutilizar muchos componentes

ya existentes para ahorrar tiempo y esfuerzo en la construcción.

Creemos muy interesante hacer un llamamiento a otras universidades, departamentos e instituciones que disponen, o están desarrollando, herramientas similares para que las liberen a las comunidad, aunque mantengan el código propietario. Además la liberación del código ha servido de base para otra herramienta libre para extraer volcados de tablas de bases de datos y que puede ser utilizada para generar archivos SQL, o archivos en formato WinRDBI [4]. Esta herramienta ha sido muy útil para la comparativa de la sección 4.

## Referencias

- [1] Apache Software Foundation License. [www.apache.org/licenses/](http://www.apache.org/licenses/).
- [2] Apache Project. [www.apache.org/](http://www.apache.org/).
- [3] Codd, E. F. 1970. A Relational Model of Data for Large Shared Databanks. Communications of the ACM, June 1970.
- [4] DumpTable. <http://javahispano.net/dumptable/>
- [5] Gewerc, A. 2002. Diseño de entornos de aprendizaje. [www.quadernsdigitals.net/](http://www.quadernsdigitals.net/)
- [6] GNU General Public Licence. [www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html)
- [7] Hernández, C., et-al. 2002. Una Herramienta para el Aprendizaje del Álgebra Relacional. VIII Jornadas de Enseñanza Universitaria de la Informática. Cáceres.
- [8] Hipersonic SQLDB. <http://hsqldb.sourceforge.net>
- [9] Jordá, P. A., et-al. 2001. Mejoras en el aprendizaje de la informática en otras escuelas universitarias. VII Jornadas de Enseñanza Universitaria de Informática.
- [10] LEAP. <http://leap.sourceforge.net/>
- [11] Raymenod, Eric S. The Cathedral and the Bazaar. <http://es.tldp.org/catedral-bazar/>.
- [12] RelationalQuery. [relationalquery.dev.java.net](http://relationalquery.dev.java.net)
- [13] Sosa, A.R., et-al. 2002. Enseñanza de la parte estática del modelo relacional de bases de datos basada en las Nuevas Tecnologías de la Información y las Comunicaciones. [www.ucf.edu.cu/publicaciones/anuario2002/N TIC/articulo22.pdf](http://www.ucf.edu.cu/publicaciones/anuario2002/N TIC/articulo22.pdf)
- [14] WinRDBI. [www.eas.asu.edu/~winrd](http://www.eas.asu.edu/~winrd)