

DerEditor4GL: Software para la docencia en el diseño de Bases de Datos

Piedad Garrido Picazo, Manuel Coll Villalta, Francisco J. Martínez Domínguez

Departamento de Informática e Ingeniería de Sistemas (DIIS)
Escuela Universitaria Politécnica de Teruel, Universidad de Zaragoza
E-mail: {piedad@unizar.es, mhvcoll@gmail.com, F.Martinez@unizar.es}

Resumen

El presente trabajo presenta el desarrollo de una herramienta CASE (*Computer-Aided Software Engineering*) llamada DerEditor4GL [11], que genera lenguaje lógico conceptual y lenguaje SQL (*Structured Query Language*). Esta herramienta está pensada para su uso en la enseñanza de la asignatura Bases de Datos II, impartida a los alumnos de tercer curso de Ingeniería Técnica en Informática, especialidad Gestión.

Los objetivos de esta aplicación gráfica son por un lado, facilitar la comprensión de las distintas fases de diseño por las que pasa un sistema de información antes de ser procesado en un computador, por otro, permitir la visualización del esquema conceptual y lógico, y finalmente, la generación del código asociado a cada uno de los objetos de representación, en tiempo real y de forma transparente al usuario.

Esta herramienta CASE va a ser utilizada en las prácticas de la asignatura Bases de Datos II dentro de la titulación de Ingeniería Técnica en Informática de Gestión de la Escuela Universitaria Politécnica de Teruel.

1. Introducción

En el actual plan de estudios para la obtención del título de Ingeniero Técnico en Informática de Gestión en la Universidad de Zaragoza, el estudio de la disciplina de Bases de Datos se divide en dos asignaturas de carácter troncal y obligatorio, respectivamente: Bases de Datos I y Bases de Datos II [9, 10] ubicadas en quinto y sexto semestre de la titulación, respectivamente.

En la asignatura Bases de Datos I se definen conceptos tales como: qué es un Sistema de Información, sus componentes y funciones, qué se entiende por Sistema Gestor de Bases de Datos, así como qué es una Base de Datos. Posteriormente, se presenta una panorámica del modelo relacional de datos, se habla de la organización física de las bases de datos, y se trata en profundidad el SQL (*Structured Query Language*) [4] haciendo especial hincapié en cómo se ha ido adaptando a la evolución del modelo relacional de datos.

La asignatura de Bases de Datos II es una continuación de Bases de Datos I, cuyo principal objetivo es presentar una metodología de diseño de bases de datos relacionales basada en el modelo Entidad-Relación.

La docencia de ambas asignaturas se organiza en (4+2), 4 créditos de teoría y problemas y 2 créditos de trabajo en el laboratorio. Este trabajo consiste en la realización, de modo coordinado con las clases de teoría y problemas, de una serie de prácticas que refuerzan la comprensión de los conceptos estudiados y permiten su correcta aplicación.

En la presente ponencia se presenta una herramienta CASE (*Computer-Aided Software Engineering*) que asesorará al alumno a lo largo de la segunda y la tercera práctica de la asignatura Bases de Datos II, que versan sobre el análisis y diseño del esquema conceptual y lógico, respectivamente.

Previamente, el alumno ya habrá trabajado de forma teórica el aprendizaje de los distintos módulos y objetos que participan en el diseño.

El contenido del presente trabajo está organizado de la siguiente manera: en el segundo apartado se presenta en detalle la herramienta que ayuda al diseño de bases de datos.

En el tercer apartado se presenta la arquitectura software de DerEditor4GL y comentan los resultados y ficheros que genera.

En el apartado número 4, se explica el trabajo a realizar por los alumnos matriculados en la asignatura.

Finalmente, se enumeran una serie de posibles ampliaciones y mejoras para las futuras versiones de la herramienta.

2. Descripción de la herramienta

DerEditor4GL permite a los alumnos implementar de forma sencilla y atractiva, el modelo relacional teórico de cualquier base de datos.

De esta forma, los alumnos pueden hacerlo sin tener que entrar a comprender los detalles específicos y propios de cada uno de los editores gráficos que proporcionan los diferentes administradores corporativos de los sistemas gestores de bases de datos relacionales existentes

en el mercado, lo que en terminología de esta disciplina se conoce como esquema físico.

Cuando se plantean las prácticas de toda asignatura perteneciente a esta disciplina, hay que tener en cuenta que el alumno tiene doble tarea a realizar; por un lado, aún no ha terminado de asimilar el modelo estudiado en el módulo teórico de la asignatura, cuando por otro lado, se encuentra con que parte de los objetos de información que ha ido aprendiendo en clase, no se implementan de igual forma si se trabaja con diferentes gestores como MySQL, PostgreSQL, Oracle, SQLServer, etc.

DerEditor4GL permite al alumno realizar el diseño del esquema conceptual, ciñéndose a las pautas presentadas en las horas teóricas de la asignatura, y así obtener de forma sencilla e instantánea el pseudocódigo correspondiente al esquema lógico, para simultáneamente obtener su homólogo en SQL.

Características	DBDesigner	DeZign	xCase	DEREditor4GL
Plataformas soportadas	Windows/Linux	Windows	Windows	Windows/Linux Unix/MacOS/PowerOS
Licencia de uso	GPL	Propietario/Trial	Propietario/Trial	GPL
Capacidad de la interfaz	Alta	Alta/Adaptable	Exhaustiva	Sencilla
Formato y aspecto del diagrama	No	No	Sí	Sí
Generación de código	SQL	Múltiples	SQL	Descriptivo / Pseudo / SQL
Inserción manual de código	Sí	No	No	No
Ingeniería inversa	Sí	Sí	Sí	No
Formatos de exportación	XML	HTML/Doc/Jpeg	HTML/Rtf	Doc/Jpeg
Motor integrado de consultas	Sí	No	No	No
Conocimientos necesarios	Muy altos	Altos	Altos	Muy bajos
Simbología de diagramas	No estándar	No estándar	No estándar	Estándar
Uso de teoría diagramas E-R	No	Sí	Sí	Sí
Nuevos tipos de datos, dominios	No	No	Sí	Sí
Gestión de relaciones n-arias	No	Sí	No	Sí
Modelos jerárquicos y débiles	No	Sí	No	Sí
Chequeo automático de diseño	No	No	No	Sí

Figura 1. Comparativa de diferentes herramientas de diseño de Bases de Datos

Además, para verificar el correcto funcionamiento del diseño, la herramienta desarrollada incorpora un manejador de errores, guiando al usuario en todo momento hacia la obtención de la solución correcta.

La principal contribución de esta herramienta es precisamente la posibilidad de, no sólo obtener en un único paso el esquema conceptual, el esquema lógico y el código SQL asociado, sino también realizar su implementación de una forma sencilla y rápida, ya que el tiempo disponible para la realización de la práctica es de dos sesiones de dos horas cada una.

DerEditor4GL se trata de software libre adhoc, desarrollado una vez revisados otros productos significativos que perseguían objetivos similares, habiendo realizado previamente una comparativa y viendo las posibles carencias y oportunidades de mejora que ofrecían.

Los resultados de dicha comparativa se muestran en la Figura 1. Cabe destacar que este análisis ha sido llevado a cabo teniendo en cuenta tanto herramientas propietarias como herramientas de software libre existentes en el mercado.

De la interpretación de los parámetros escogidos para llevar a cabo la comparativa, presentados en la tabla, se deduce que DerEditor4GL soporta todas las características básicas.

Sin embargo, no proporciona funcionalidad de ingeniería inversa, inserción manual de código y motor integrado de consultas. Características que no afectan al buen funcionamiento y expectativas de la herramienta, pero que se deben considerar como posibles ampliaciones de la aplicación en las futuras versiones.

Por otro lado, es la única herramienta que cumple los requisitos en todos los aspectos teóricos de diseño. Además, es una aplicación flexible, que puede ser utilizada por cualquier tipo de usuario, ya sea novel o experto.

Finalmente, destacar que permite la definición de nuevos tipos de datos o dominios, gestiona relaciones n-arias, implementa agregaciones, relaciones reflexivas, entidades débiles, generalizaciones/especializaciones y chequea de forma automática el diseño.

3. Arquitectura y funcionamiento

En la siguiente figura se muestra la organización de la aplicación en módulos, y su interconexión.

El corazón de la aplicación es el módulo central **DEREdito**. Es el constructor de la aplicación y el encargado de iniciar los módulos **Config**, **DERGrafo**, y los módulos de E/S: **Archivador**, **Impresor** y **Visual**.

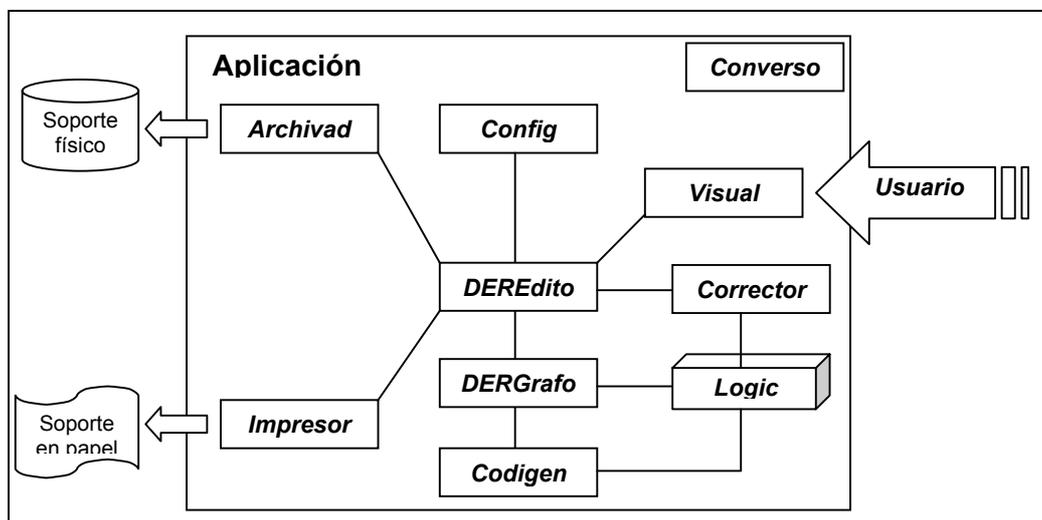


Figura 2. Arquitectura de la aplicación

Una vez los ha iniciado, carga la configuración y se mantiene en espera de recibir información del módulo **Visual**.

Por otro lado, se encuentran las librerías **Logic**, que permiten mantener un diagrama acorde a la teoría de los diagramas E-R, y el módulo **Corrector**, que es el encargado de realizar un análisis del diseño en busca de posibles errores que puedan generar código erróneo. Este análisis es realizado sobre cada elemento del diagrama, examinando sus datos mediante las librerías Logic. También analiza las asociaciones entre elementos, en busca de incompatibilidades.

4. Trabajo del alumno

De acuerdo con Kendall & Kendall [7], la ingeniería de sistemas asistida por ordenador es la aplicación de tecnología informática a las

actividades, las técnicas y las metodologías propias de desarrollo. Su objetivo es acelerar el proceso para el que han sido diseñadas.

En el caso de las herramientas CASE, se utilizan para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

Cuando se hace la planificación de la base de datos, la primera etapa del ciclo de vida de aplicaciones que acceden a un repositorio de información, se puede escoger una herramienta CASE que permita llevar a cabo el resto de tareas del modo más eficiente y efectivo posible.

El principal objetivo de esta herramienta es el de ofrecer al alumno la posibilidad de analizar de una forma muy sencilla y visual el correcto diseño y funcionamiento de las fases de todo diseño de base de datos que se precie.

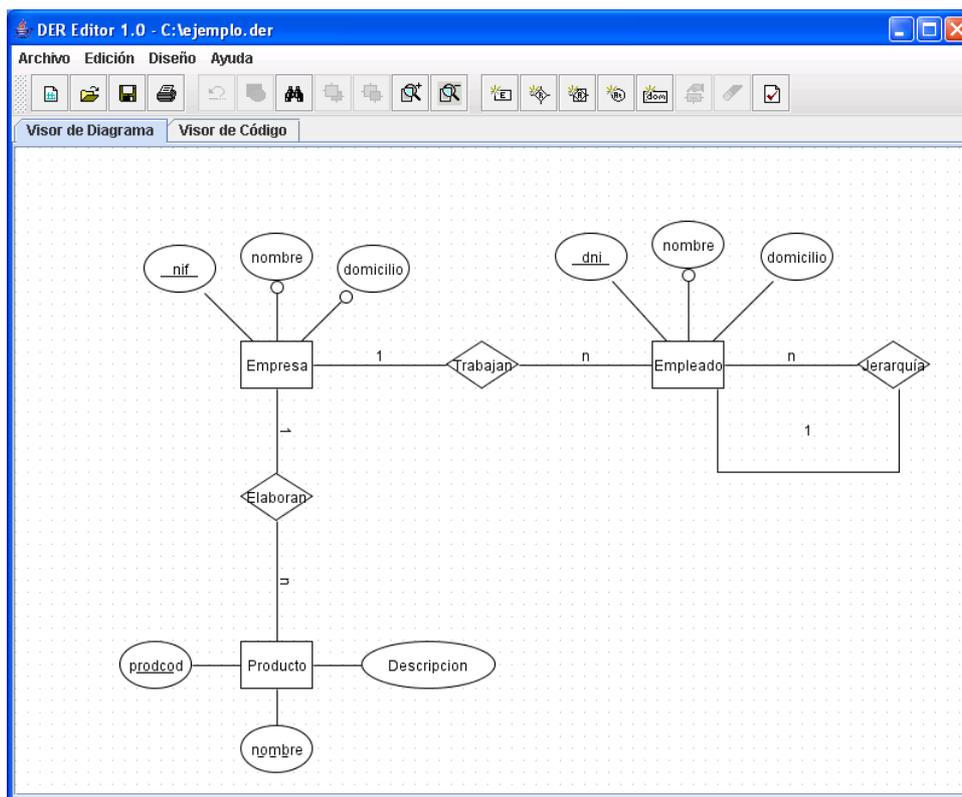


Figura 3. Ejemplo de Esquema Conceptual generado con DerEditor4GL

Se trata de una herramienta de software libre que ha sido implementada íntegramente en Java, por lo que es multiplataforma y se distribuye, bajo licencia GPL, actualmente en sourceforge.net [11] y próximamente en freshmeat.net.

Tal y como se ha demostrado en la comparativa previa, herramientas tales como el DBDesigner, DeSign, Xcase, etc., poseen dos claros inconvenientes: por un lado, no se ajustan a la teoría de diagramas entidad-relación, y por otro, la mayoría de estas herramientas generan código dependiente de los lenguajes de alto nivel predominantes en el mercado.

Si a esto le añadimos que para la utilización de los mismos se requiere tener unos elevados conocimientos teóricos de la materia, nos encontramos con que resulta complicada su utilización en un entorno educativo, sobretodo en sus primeras fases de aprendizaje.

El trabajo de los alumnos consiste en representar las entidades, atributos y relaciones, y sus peculiaridades, siempre ciñéndose a la información extraída en una fase de análisis previa del sistema de información real a procesar.

Todo objeto gráfico que necesitan, ya sea un rombo nominado, un arco, un rectángulo o una elipse, lo tienen disponible en una paleta de herramientas (ver figura 3), creada expresamente para esta aplicación, y cuyo contenido les va a ser muy familiar ya que se ajusta fielmente a la representación vista en clase.

En la primera sesión de prácticas, de dos horas de duración, los alumnos se dedicarán a analizar en detalle el sistema de información a implementar, con el resto de componentes de su grupo.

En la práctica 2, los alumnos deberán familiarizarse un poco con el entorno de trabajo e interactuar con él, para así obtener el esquema conceptual de su sistema de información.

En la práctica 3, deberán proceder a la interpretación del pseudocódigo que se irá generando de forma automática (ver algoritmo 1) mientras ellos van construyendo de forma gráfica, la estructura de su repositorio de información.

Este código se corresponderá con la notación vista en clase para la obtención del esquema lógico, lo que facilitará enormemente la comprensión y el repaso de los conceptos por parte de los alumnos.

```
-Inicio de pseudocodigo --
Dominio dom_cadena
Dominio dom_nif="00000000A" ( V
x(Dom x=dom_nif) -> (xdom_nif <=
"99999999Z" ^

xdom_nif >= "00000000A") )
Dominio dom_prodcod="A000" ( V
x(Dom x=dom_prodcod) ->
(xdom_prodcod <= "Z999" ^

xdom_prodcod >= "A000") )

/* Entidad Empresa: */
Empresa (nif:dom_nif,
         domicilio:dom_cadena,
         nombre:dom_cadena )
Clave Primaria { nif }
Valor No Nulo { domicilio }
Valor No Nulo { nombre }

/* Entidad Empleado: */
Empleado (dni:dom_nif,
          nombre:dom_cadena,
          domicilio:dom_cadena )
Clave Primaria { dni }
Valor No Nulo { nombre }

/* Relacion Trabajan: */
Trabajan (nif:dom_nif,
          dni:dom_nif )
Clave Primaria { dni }
Valor No Nulo { nif }
Clave Ajena { nif } hace
referencia a Empresa
Clave Ajena { dni } hace
referencia a Empleado

/* Relacion Jerarquia: */
Jerarquia (dni1:dom_nif,
           dni2:dom_nif )
Clave Primaria { dni1 }
Valor No Nulo { dni2 }
Clave Ajena { dni1 } hace
referencia a Empleado
Clave Ajena { dni2 } hace
referencia a Empleado

/* Entidad Producto: */
Producto (prodcod:dom_prodcod,
          nombre:dom_cadena,
          Descripcion:dom_cadena )
Clave Primaria { prodcod }
Unico { nombre }
```

```

/* Relacion Elaboran: */
Elaboran      (nif:dom_nif,
prodcod:dom_prodcod )
    Clave Primaria { prodcod }
    Valor No Nulo { nif }
    Clave Ajena { nif } hace
referencia a Empresa
    Clave Ajena { prodcod } hace
referencia a Producto

-- Fin de pseudocodigo --

```

Algoritmo 1. Pseudocódigo del ejemplo de la Figura 3

La interpretación del código es esencial, ya que antes de proceder a la elección de un gestor físico y exportar el código SQL generado al analizador de consultas, el esquema debe pasar lo que en base de datos se conoce como teoría de la normalización.

Este proceso no se realiza automáticamente, por razones obvias de complejidad y de diversidad de soluciones, quedando en manos del diseñador.

Los resultados que se generan por la aplicación, pueden volcarse en ficheros textuales o gráficos, dependiendo de la fase de diseño de la que se quiera obtener la información.

En el caso de llevarse a cabo una acción incorrecta, el sistema posee un sistema manejador de errores a nivel de cada módulo, y un módulo denominado **Corrector**, que es el encargado de realizar un análisis del diseño en busca de posibles errores que puedan derivar en código erróneo.

Todo ello, resulta muy útil para el alumno, ya que le permite profundizar de forma sencilla en el estudio y comprensión de las pautas a seguir para la obtención del código, y poder comprobar de forma visual cualquier situación teórica.

La evaluación de la práctica se realizará a través de una entrevista con los distintos grupos de alumnos, con preguntas relativas a los resultados obtenidos con DerEditor4GL.

La idea es ponerlo en práctica este curso académico para la realización de las prácticas de la asignatura, y poder así hacer una comparativa con los resultados previos obtenidos, cuando el alumno se tenía que enfrentar directamente al administrador corporativo de cualquier gestor de bases de datos comercial o libre disponible en el mercado.

Si la experiencia resulta positiva, también se espera incorporarlo como herramienta para la

evaluación de la asignatura, añadiendo algunos aspectos de seguridad.

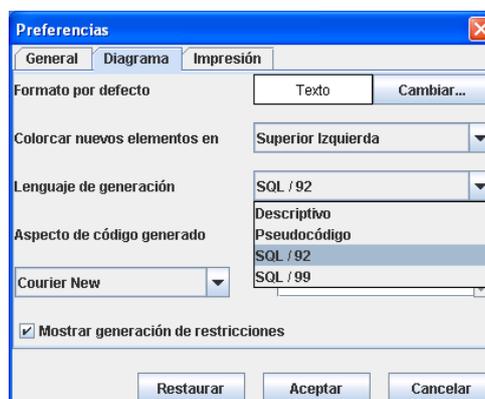


Figura 4. Cuadro de diálogo Preferencias

5. Análisis de ampliaciones y mejoras

Como se ha comentado anteriormente, para las futuras versiones de la herramienta, se sopesarán posibles ampliaciones de la herramienta en aspectos tales como:

- **Ingeniería inversa.** De esta forma, la aplicación podrá crear un diagrama Entidad-Relación, a partir de cualquier código escrito en alguno de los lenguajes soportados por el software DerEditor4GL.
- **Inserción manual de código.** Para que los alumnos puedan modificar el código directamente, y que dichos cambios queden reflejados en el diseño generado por la aplicación.
- **Motor integrado de consultas.** De esta forma los alumnos podrán introducir y probar diferentes consultas, en la base de datos creada.
- **Soporte de dominios universales.** Se puede modificar la aplicación para que incorpore una serie de tipos de datos universales y válidos para cualquier SGBD basado en SQL, evitando así la tarea de crear los dominios necesarios para cada diagrama, que puede acarrear un buen esfuerzo por parte del usuario.
- **Autocolocación de elementos.** Facilitando así el uso de la herramienta, al seleccionar

ésta de forma automática la mejor ubicación de los diferentes elemento y permitir una correcta visualización.

- **Aumento de los formatos de exportación.** Para ampliar las posibilidades de uso de la aplicación.

Por otro lado, dado que su implementación ha sido llevada a cabo a lo largo de siete meses, han aparecido diversas actualizaciones de algunas de las tecnologías aplicadas: por ejemplo Java 1.5 en lugar de 1.4.2, o JGraph 5.3.1 en lugar de 5.0.

Por lo que será necesario estudiar si alguna de las mejoras aparecidas puede ser utilizada en las nuevas versiones.

6. Conclusiones

DerEditor4GL es una aplicación capaz de facilitar el diseño y la gestión de diagramas, así como la generación de código, de manera limpia, efectiva, y que resuelve varios problemas detectados durante el uso de herramientas en las prácticas de este tipo de asignaturas por parte de los alumnos: su utilización no requiere grandes conocimientos previos, su manejo es muy intuitivo y la simbología utilizada se ajusta con la teoría impartida en clase.

Por otro lado, soporta modelos jerárquicos y de relaciones múltiples, aspectos que casi ninguna otra aplicación del sector soporta, entre otras prestaciones ventajosas.

Al generar código SQL (ver figura 4), ciñéndose estrictamente al estándar, es compatible con casi todos los gestores de bases de datos actuales.

Además, ya que se distribuye con licencia GPL (General Public License), se espera que sea una herramienta útil y bien aceptada por la comunidad universitaria. Animamos a sus futuros usuarios a contribuir en la obtención de versiones mejoradas, más robustas y seguras de la aplicación.

Referencias

- [1] Celma, Matilde, Casamayor, J. C., Mota, Laura. Bases de Datos Relacionales. Prentice Hall, 2003
- [2] Date, C. J. Introducción a los sistemas de Bases de Datos. Prentice Hall, 2001
- [3] Elmasri, Ramez A., Navathe, Shamkant B. Fundamentos de Sistemas de Bases de Datos. Addison Wesley, 2002
- [4] Groff, James R., Weinberg, Paul N. SQL: Manual de Referencia. Mc-Graw Hill, 2003
- [5] Hansen, Gary W., Hansen, James V. Diseño y administración de Bases de Datos. Prentice Hall, 2000
- [6] Harrington, Jan L. Relational database design. Morgan Kaufmann Publishers, 2002
- [7] Kendall, Kenneth, Kendall, Julie. Análisis y Diseño De Sistemas. Prentice Hall, 1997
- [8] Ullman, Jeffrey D., Widom, Jennifer. Introducción a los sistemas de Bases de Datos. Pearson, 1999
- [9] <http://eupt2.unizar.es/bd1>. Apuntes de la asignatura Base de Datos I
- [10] <http://eupt2.unizar.es/bd2>. Apuntes de la asignatura Base de Datos II
- [11] <http://dereditor.sourceforge.net>. Página web de DerEditor.
- [12] <http://www.fabforce.net/dbdesigner4>. Página web de DBDesigner 4.0.
- [13] <http://www.datanamic.com>. Página web de DeZign for Databases 3.
- [14] <http://Java.sun.com>. Página web de Java de Sun Microsystems.
- [15] <http://www.jgraph.com>. Página web de JGraph.
- [16] <http://www.xcase.com>. Página web de xCase Database Design.