

**SOFTCAR, SOFTWARE EDUCATIVO PARA EL CÁLCULO RELACIONAL COMO APOYO A LA ASIGNATURA DE BASE DE DATOS**AUTORES: Angel E. Figueredo León<sup>1</sup>Agustín A. Ortiz Díaz<sup>2</sup>Elena Martínez Pérez<sup>3</sup>Laura Palomino Mariño<sup>4</sup>DIRECCIÓN PARA CORRESPONDENCIA: [afigueredol@udg.co.cu](mailto:afigueredol@udg.co.cu)

Fecha de recepción: 16 - 06 - 2018

Fecha de aceptación: 30 - 07 - 2018

## RESUMEN

El presente trabajo propone el estudio y aprendizaje del lenguaje de consultas estructuradas del cálculo relacional de bases de datos. Dicha investigación se basa en el problema existente en la carrera de Ingeniería Informática de la Universidad de Granma en cuanto a la inexistencia de un software educativo que facilite a los estudiantes comprender dicho tema de la asignatura Sistemas de Bases de Datos. Es por tal motivo que el objetivo de este trabajo es desarrollar una herramienta que implemente las funcionalidades necesarias para garantizar la recepción y asimilación del conocimiento de manera más fácil para el estudiante. Por previo estudio se tomó la decisión de desarrollar la herramienta con software libre, utilizando para ello Eclipse SDK como IDE y Java como lenguaje de programación. Para el modelado se utiliza la herramienta case Rational Rose, y todo el proceso de desarrollo está basado en la metodología RUP. Lo anteriormente planteado dio como resultado una aplicación de escritorio que responde a la problemática actual.

**PALABRAS CLAVE:** Sistemas de bases de datos; analizador sintáctico; cálculo relacional.

**SOFTCAR, EDUCATIONAL SOFTWARE FOR RELATIONAL CALCULATION AS A SUPPORT FOR THE DATABASE SUBJECT**

## ABSTRACT

---

<sup>1</sup> Máster en Ciencias. Profesor de la Facultad de Ciencias Informáticas, Naturales y Exactas. Universidad de Granma. Cuba.

<sup>2</sup> Doctor en Ciencias. Profesor de la Facultad de Ciencias Informáticas, Naturales y Exactas. Universidad de Granma. Cuba. E-mail: [aotrizd@udg.co.cu](mailto:aotrizd@udg.co.cu)

<sup>3</sup> Ingeniera. Profesora de la Facultad de Ciencias Informáticas, Naturales y Exactas. Universidad de Granma. Cuba. E-mail: [emartinezp@udg.co.cu](mailto:emartinezp@udg.co.cu)

<sup>4</sup> Máster en Ciencias. Profesora de la Facultad de Ciencias Informáticas, Naturales y Exactas. Universidad de Granma. Cuba. E-mail: [zoe1580@gmail.com](mailto:zoe1580@gmail.com)

The current paper proposes the study and learning of structured queries language for the relational algebra in data bases. Such research is based on the existing problem in the Information Engineering major at the University of Granma concerning the lack of an educational software to facilitate the students the understanding of such a matter for the Database Systems subject. It is for this reason that the objective of this work is to develop a tool that implements the necessary functions to guarantee the students the knowledge reception and assimilation in an easier and friendlier way. According to a previous study a decision was made to develop such a tool with free software, using Eclipse SDK as IDE and Java as the programming language. For modelling, the Rational Rose case tool is used, and the whole development process is based on the RUP methodology. The above stated resulted in a desktop application that responds to the current problem.

**KEYWORDS:** Database systems; syntactic analyzer; relational calculus.

## INTRODUCCIÓN

Desde sus inicios el hombre ha buscado la forma de mejorar su calidad de vida y su forma de trabajo, para ello ha buscado métodos adecuados tales como la tecnología que ha desarrollado a través de la ciencia. Esto ha permitido llegar a grandes inventos científicos desde la calculadora hasta la computadora, este gran avance ha llevado a la humanidad a tener un enorme desarrollo social, pues la computadora se ha convertido en pocos años en parte integrante de la vida cotidiana y ha permitido que se exploren de manera más amplia y exactas disciplinas cuyos avances no serían posible de ningún otro modo.

El mundo de hoy, está inmerso en una nueva revolución tecnológica basada en la informática, que encuentra su principal impulso en el acceso y en la capacidad de procesamiento de información sobre todos los temas y sectores de la actividad humana.

El sistema educativo actual se encuentra sumido en una época de grandes cambios. La necesidad de innovación tecnológica y los cambios que se realizan en la docencia propician a su vez un importante avance en las estructuras docentes. El uso de las nuevas tecnologías aplicadas a la docencia y la rápida implantación de herramientas software permiten dinamizar y compartir contenidos de modo que amplían el espectro de trabajo en el aula y trasladan el entorno docente a otros lugares y otra división de tiempos distintos a los de la docencia tradicional.

Las instituciones educativas en nuestro país están particularmente integradas a las computadoras como una herramienta interactiva para el aprendizaje. Los programas de Educación Asistida por Computadora (CAE), pueden solicitar retroalimentación del usuario y responder de manera apropiada. En forma similar, programas interactivos de aprendizaje pueden enseñar, hacer pruebas de comprensión y repaso basado en lo aprendido por el estudiante.

Las universidades cubanas han estado totalmente inmersas en un proceso de informatización, con el objetivo de lograr una mayor eficiencia tanto en la gestión académica como administrativa, ya que es de vital importancia para el correcto funcionamiento de una universidad, y en general de cualquier centro de educación.

La Universidad de Granma es uno de los centros de educación superior del país que se encuentra enfrascada en la automatización del proceso docente educativo, tanto de pregrado como de postgrado.

Por tanto, es de especial interés desarrollar un software educativo que proporcione un mecanismo mediante el cual los estudiantes pueden explorar las diferentes posibilidades del lenguaje de consultas estructuradas, de manera que vieran el resultado de la consulta que habían propuesto en ese mismo momento y pudieran realizar las oportunas modificaciones, ya que actualmente la carrera de Ingeniería Informática no cuenta con un software que permita a los estudiantes comprender y comprobar resultados obtenidos en un ejercicio de cálculo relacional, sólo se cuenta con el criterio del profesor encargado de impartir la asignatura, por lo que se dificulta la forma en que estos estudiantes pueden adquirir a mayor profundidad los conocimientos necesarios y suficientes con relación al tema.

## DESARROLLO

### 1. Utilización de los Software Educativos.

Esta nueva etapa del desarrollo social trae consigo la introducción de las tecnologías de la información y la comunicación, soportes de la información y canales de comunicación relacionados con el almacenamiento, procesamiento y transmisión digitalizados de la información. Se viven día a día las consecuencias de todos estos avances tecnológicos y el proceso de enseñanza-aprendizaje no escapa a ello. Precisamente, la utilización de la computadora como medio de enseñanza, ha popularizado la utilización de programas para ordenadores, creados con la finalidad específica de ser utilizados como medio didáctico, para facilitar los procesos de enseñanza y de aprendizaje. Son llamados softwares educativos, programas educativos o programas didácticos (Couturejuzón, 2006).

Los Software Educativos (SE), se definen de forma genérica como aplicaciones o programas computacionales que facilitan el proceso de enseñanza-aprendizaje. Algunos autores los conceptualizan como cualquier programa computacional, cuyas características estructurales y funcionales sirven de apoyo al proceso de enseñar, aprender y administrar, o el que está destinado a la enseñanza y el autoaprendizaje y además permite el desarrollo de ciertas habilidades cognitivas (Vidal y otros., 2010).

Con la utilización de un programa didáctico para la transmisión de los conocimientos, está implícita la aplicación del principio del carácter audiovisual de la enseñanza, puesto que los adelantos tecnológicos aseguran poder utilizar

prácticamente todos los elementos necesarios para la comprensión del contenido mediante la representación dinámica de todos los objetos, esquemas, figuras, gráficos y tablas necesarios, así como del sonido para propiciar la relación de lo concreto con lo abstracto. La adecuación y vigencia de estos principios en el uso de las nuevas tecnologías de la educación evidencian su utilidad y aplicabilidad en cualquier forma de enseñanza (Labarre y Valdivia, 2006).

## 2. Características del SOFTCAR.

### 2.1 Modelo del dominio

Los objetos del dominio representan los elementos que existen o los eventos que suceden en el entorno en el que trabajará el sistema. Por esa razón, se propone un modelo del dominio con el objetivo de definir los tipos más importantes de objetos que representan las cosas que existen o eventos que suceden en el entorno en el que trabaja el sistema.

El diagrama de la figura 1, visualiza y relaciona los principales conceptos del dominio, el cual va a contribuir posteriormente a identificar algunas de las clases que se utilizarán en el sistema a desarrollar.

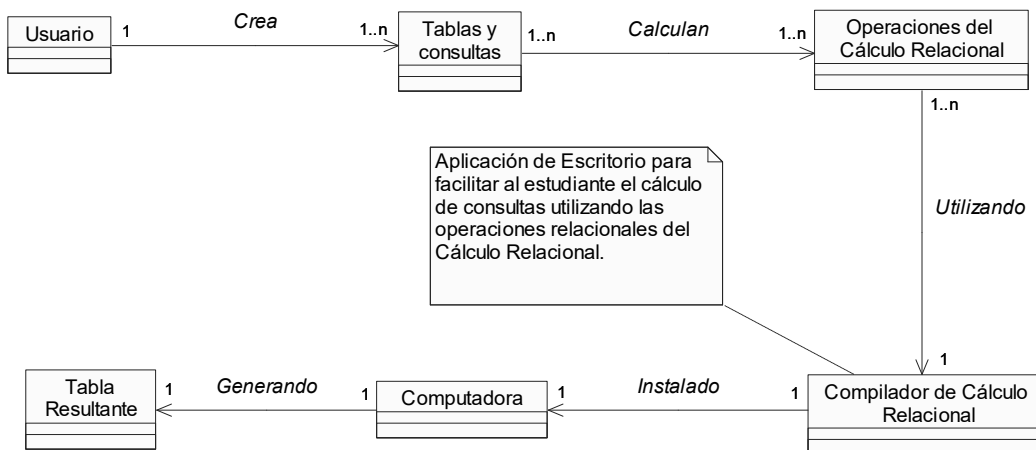


Figura 1. Diagrama de clases del dominio.

### 2.2 Requerimientos funcionales

Los requisitos funcionales son las acciones que deberá ser capaz de realizar el sistema para satisfacer las necesidades del usuario. Definen los límites de la aplicación y por lo general, se describen mejor a través del modelo de caso de uso y los casos de uso como tal. De acuerdo con los objetivos planteados, el sistema debe de ser capaz de:

R1- Crear tablas.

R2- Introducir datos en las tablas.

R3- Guardar tablas.

R4- Eliminar tablas.

R5- Modificar tablas.

R6- Mostrar tablas.

R7- Introducir consulta.

R8- Procesar los datos.

R9- Guardar el resultado final.

R10- Mostrar ayuda.

### 2.3 Técnicas de compilación utilizadas

En la comunicación hombre-máquina existe una dificultad real; las computadoras operan sobre bits y registros, y los hombres se entienden por medio de idiomas. Los lenguajes de programación son el vehículo de comunicación entre el hombre y la computadora. Los compiladores o traductores son programas encargados de transformar el programa fuente en un programa objeto equivalente. Este programa objeto debe de estar expresado, en última instancia, en un lenguaje que la máquina entienda directamente (Mora, 1983).

Implementar un intérprete es un proceso difícil, ya que requiere de un gran esfuerzo durante su desarrollo debido a que es preciso diseñar e implementar algoritmos y estructuras de datos muy complejos (Toro y otros, 1997)

En este trabajo se utilizó una gramática de tipo LL(1), la cual permiten construir de forma automática un analizador determinista descendente con tan sólo examinar en cada momento el símbolo actual de la cadena de entrada (símbolo de preanálisis) para saber qué producción aplicar (Aho. y otros, 1990).

Además se utilizó un analizador sintáctico descendente predictivo no-recursive sin llamadas recursivas a procedimientos, que utiliza una pila auxiliar de símbolos terminales y no-terminales. Para determinar qué producción debe aplicarse en cada momento, se busca en una tabla de análisis sintáctico (parsingtable) en función del símbolo de preanálisis que se está observando en ese momento y del símbolo en la cima de la pila se decide la acción a realizar (Vivancos, 2000). En la figura 2 se muestra un analizador sintáctico descendente dirigido por tabla

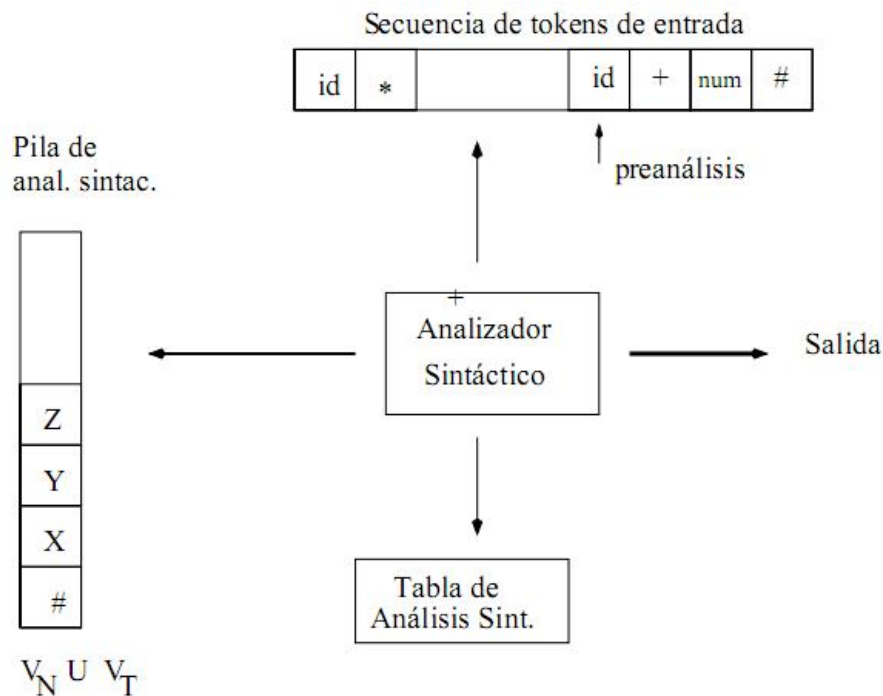


Figura 2. Esquema de un analizador sintáctico descendente dirigido por tabla.

El análisis sintáctico dirigido por tabla es más eficiente en cuanto a tiempo de ejecución y espacio, puesto que usa una pila para almacenar los símbolos a reconocer en vez de llamadas a procedimientos recursivos (Herrera, 1982). A continuación se expone el algoritmo de análisis sintáctico predictivo dirigido por tabla.

Algoritmo de análisis sintáctico predictivo dirigido por tabla.

Entrada: una cadena  $w$ , una tabla de analizador sintáctico y una gramática  $G$ .

Salida: Si  $w \in L(G)$ , la derivación más a la izquierda de la cadena  $w$ , sino una indicación de error

Método:

Como configuración inicial se tiene en el fondo de la pila el símbolo #, el axioma  $S$  en la cima y la cadena  $w\#$  en el buffer de entrada.

Iniciar preanálisis con el primer símbolo de  $w\#$ .

*repetir*

Sea  $X$  el símbolo de la cima de la pila

si  $X$  es un terminal o # entonces

si  $X == \text{preanálisis}$  entonces

extraer  $X$  de la pila y obtener nuevo componente léxico

sino error();

```

sino
    si M[X,preanalysis] = X → Y1 Y2 ... Yn entonces
        extraer X de la pila
        meter Yn ... Y2Y1, con Y1 en la cima de la pila
    sino error();
hasta_que X == # (* la pila está vacía y hemos procesado toda la entrada *)
    Cadena reconocida con éxito

```

#### 2.4 Tecnologías utilizadas

Para el desarrollo de la aplicación informática se hizo uso de la metodología del Proceso Unificado de Desarrollo Software (RUP), ya que permite definir un sistema de software; para detallar los artefactos en el sistema; para documentar y construir. Durante todos los pasos del ciclo de vida del software y facilitando el mejoramiento de éste, se hizo uso como herramienta CASE, el Rational Rose Enterprise Edition ya que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. JAVA fue lenguaje de programación escogido para implementar el software, porque es un lenguaje multiplataforma, con el cual se pueden desarrollar programas que se ejecuten sin problemas en sistemas operativos como Windows, Linux, Mac y Unix.

Una vez definido el lenguaje de programación, se procedió a definir el entorno de desarrollo, en este caso se hizo uso del Eclipse SDK 3.2, ya que es una poderosa y completa plataforma de programación, desarrollo y compilación de elementos tan diversos como sitios Web, programas en C++ o aplicaciones Java. Incluye todas las herramientas y bibliotecas estándar de Java necesarias para crear applets y aplicaciones.

#### 3. SOFTCAR como software educativo para el cálculo relacional.

La interfaz de usuario, es la categoría de diseño que crea un medio de comunicación entre el hombre y la máquina. Además, el diseño identifica los objetos y acciones de la interfaz y crea entonces un formato de pantalla que formará la base del prototipo de interfaz de usuario.

El diseño de la interfaz es un punto fundamental a tener en cuenta a la hora de presentar la aplicación, porque se considera que es la cara mostrada al usuario y debe ser lo más amigable y comprensible posible.

En el diseño de las pantallas se tuvieron en cuenta varios aspectos: organización de los elementos en la pantalla, dónde se coloca la información y cómo se estructura. Las ventanas de la interfaz deben ser diseñadas de forma uniforme garantizando:

- El equilibrio en la organización de la información, por ejemplo, todas las ventanas que muestran información siempre la mostrarán en el mismo orden.



- La optimización de la cantidad de elementos en la pantalla, ayudando al fácil manejo y mejor comprensión de la información mostrada en pantalla.
- La unidad, donde cada elemento de la pantalla se debe diseñar siguiendo un patrón de tamaño, colores y formas.

A continuación se muestra las opciones del sistema que permite realizar todas las operaciones relacionadas con las tablas de una base de datos.



Figura 3: "Operaciones con las tablas".

El sistema analiza sintácticamente y semánticamente una consulta, la ejecuta y devuelve los resultados, los mismos deben ser claros y precisos para que el usuario pueda comprender la forma en que el sistema los muestra, dando la posibilidad de guardarlos. A continuación en la figura 4 se presenta una vista del sistema mostrando los resultados obtenidos luego de calcular una consulta.



Figura 4: "Ejemplo de cómo se muestran los resultados dentro del sistema."



El tratamiento de los errores es uno de los elementos fundamentales para el buen funcionamiento de un sistema, pues garantiza la armonía y facilidad de uso de la aplicación. Es importante partir del hecho que los usuarios son altamente propensos a cometer errores, por grandes o pequeños que los errores sean. El sistema debe prevenir y ayudar a corregir los errores mediante el uso de mensajes altamente descriptivos sobre el origen de los mismos y la posibilidad de su corrección. Los errores pueden ser generados por los usuarios en el momento de entrar los datos o generados por el sistema frente a algunas acciones y de manera inesperada. Estas tareas de detección de errores se realizan con el objetivo de garantizar al usuario el máximo de calidad y fiabilidad. Mediante la figura 5 se muestra un ejemplo de tratamiento de errores.



Figura 5: "Ejemplo del tratamiento de las excepciones dentro del sistema."

### 3. Discusión.

El éxito de un software se puede definir como calidad del sistema, de los datos y del servicio. La aplicación propuesta en esta investigación impacta de manera positiva en la preparación de los estudiantes de la carrera de Ingeniería en Informática en la asignatura de Bases de Datos evidenciados en los resultados obtenidos por los estudiantes. Con su implantación, no es necesario mejorar el equipamiento del que se dispone pues se desarrolló en función de los requerimientos técnicos existentes. Las tecnologías que requiere para su funcionamiento están basadas en software libre por lo que puede ser modificado en caso de cualquier error técnico o necesidad de ponerles nuevas prestaciones.

Para evaluar su calidad se empleó el criterio de expertos mediante el método Delphi. Éste consiste en la utilización sistemática del juicio intuitivo de un

grupo de expertos para obtener un consenso de opiniones informadas. Es considerado como uno de los métodos subjetivos de pronosticación más confiables (Blanco, López y Mengual, 2010; García y Suárez, 2013).

Los expertos que evaluaron la calidad de la aplicación fueron 20. Respondieron una encuesta que se confeccionó a partir de los siguientes indicadores generales de evaluación:

1. Resulta una interfaz amigable y fácil de operar.
2. Garantiza la disponibilidad de la información actual e histórica.
3. Facilita el análisis de la información actualizada.
4. Garantiza la seguridad de la información.
5. Constituye una herramienta útil para la preparación de los estudiantes en la asignatura de Bases de Datos.
6. Eleva la calidad de la preparación de los estudiantes.
7. Garantiza la obtención correcta de la información.

Después de realizado el procesamiento de la encuesta los resultados arrojaron que la aplicación cumple con todos los indicadores antes mencionados.

Con el propósito de valorar la efectividad de la propuesta, se seleccionó un grupo de expertos sobre la base de la labor profesional desarrollada por estos. Se utilizó la metodología de preferencia debido a la imposibilidad de réplica. Las categorías son de la siguiente forma: excelente 5, bien 4, regular 3, deficiente 2, mal 1 y descartado 0.

Tabla 1. Resultados de Opinión de Experto

Nº	Indicadores	EXPERTOS					
		A	B	C	D	F	Media
1	Capacidad de motivación al estudiante	5	5	5	5	5	5
2	Adecuación a los destinatarios	5	4	5	4	4	4,4
3	Fomento del autoaprendizaje	4	4	4	4	4	4
4	Estímulo a la creatividad del estudiante	5	5	4	4	4	4,4
5	Operatividad del medio soporte del sistema de ejercicios	5	5	4	5	5	5
Media		4,8	4,6	4,4	4,4	4,4	

La media general de los expertos fue 4,52.

## CONCLUSIONES

En la presente investigación se desarrolló un sistema informático que automatiza el proceso del cálculo relacional, facilitándole a la carrera de Ingeniería en Informática de la universidad de Granma una herramienta eficiente como apoyo al proceso docente-educativo en la asignatura de Bases de Datos.

#### BIBLIOGRAFÍA

- Aho., A., Sethi., V., & J, U. (1990). *Compiladores: principios, técnicas y herramientas*.
- Blanco, J.E., López, A. y Mengual, S. (2010). Validación mediante método Delphi de un cuestionario para conocer las experiencias e interés hacia las actividades acuáticas con especial atención al Windsurf. *ÁGORA*, 12(1), 75-96.
- Couturejuzón, D. L. (2006). Cumplimiento de los principios didácticos en la utilización de un software educativo para la educación superior *Revista Cubana Educación Médica Superior*, 17(1), 53-57.
- García, M. y Suárez, M. (2013). El método Delphi para la consulta a expertos en la investigación científica. *Revista Cubana de Salud Pública*, 39(2), 253-267.
- Herrera, E. (1982). *Técnicas de Compilación*.
- Labarre, G., & Valdivia, G. (2006). *Principios de la enseñanza de Software Libre*. La Habana: Editorial Pueblo y Educación.
- Mora, K. (1983). *Lenguajes de Programación y Técnicas de Compilación*.
- Toro, M., Corchuelo, R., & Troyano, J. A. (1997). *Apuntes de Compiladores*.
- Vidal, M., Gómez, F., & Ruiz, A. M. (2010). *Software educativos Scielo*, 24(1), 97-110.
- Vivancos, E. (2000). *Compiladores I: Una introducción a la fase de análisis*.

