Universidad de las Ciencias Informáticas
Facultad 9
Departamento de Programación

Sistema para la Integración del Proceso de Normalización de Bases de Datos Relacionales con Gestores de Bases de Datos (SINORGES)

Tesis presentada en opción al Título de Máster en Informática Aplicada

Autor: Lic. Yanet Espinal Martín

Universidad de las Ciencias Informáticas

e-mail: yanete@uci.cu

Tutor: Msc. Manuel Enrique Puebla Martínez

Universidad de las Ciencias Informáticas

e-mail: mpuebla@uci.cu

Ciudad de La Habana, noviembre del 2009

"Año del 50 Aniversario del Triunfo de la Revolución"

Cuba

Dedicatoria

A mi familia más cercana: mis padres, mis abuelos, mi hermana, mis sobrinos, mi primo y en especial mi esposo, por soportar todos mis caprichos en momentos estresantes.

Agradecimientos

Toda obra humana por humilde que sea necesita el concurso de varios, jamás el hombre alcanzó la meta por sí solo, siempre fue de alguna manera asistido en el empeño, y si algo lo enaltece es la gratitud.

Al terminar un trabajo de esta índole son muchas las personas que han colaborado en la realización del mismo, a todos ellos muchas gracias y en especial a:

- *Mi tutor y esposo: Manuel Enrique Puebla Martínez, por su ayuda incondicional, apoyo y tolerancia en este periodo.*
- *Mis estudiantes queridos: Sandor Escobar Ruiz y Sisley Sosa Vázquez, por su dedicación, interés y responsabilidad mostrada en el transcurso del desarrollo del proyecto, convirtiéndose en protagonistas también de este resultado.*
- *Todos los profesores del claustro de la maestría.*
- *A mis amigas y amigos que me quieren y aprecian. En especial a Isachi Abreu Gil, Julio Alberto Leyva, Rafael Rodríguez Puente y Baby.*
- *Todas aquellas personas que de una manera u otra contribuyeron a que hoy pudiera hacer realidad este sueño.*

DECLARACIÓN JURADA DE AUTORÍA Y AGRADECIMIENTOS

Yo Yanet Espinal Martín, con carné de identidad 82012731033, declaro que soy el autor principal del resultado que expongo en el presente trabajo titulado "*Sistema para la Integración del Proceso de Normalización de Bases de Datos Relacionales con Gestores de Bases de Datos (SINORGES)*", para optar por el título de Máster en Informática Aplicada.

El trabajo fue desarrollado durante el período comprendido entre el 2008-2009 en colaboración con mi tutor MSc. Manuel Enrique Puebla Martínez y dos estudiantes de 3er año de la carrera Ciencias Informáticas: Sandor Ruiz Escobar y Sisley Sosa Vázquez que forman parte del trabajo y soy tutora de su formación investigativa-profesional, quienes me reconocen la autoría principal del resultado expuesto en esta investigación.

Finalmente declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de La Habana a los ___ días del mes de _____ del año _____.

<Firma del maestrante en tinta azul>

Resumen

Dado el crecimiento vertiginoso de los volúmenes de información en la sociedad actual y por consiguiente, el aumento de la información manipulada por los Sistemas Informáticos (SI), se hace necesario optimizar el tratamiento de los grandes volúmenes de información que genera nuestra sociedad, así como, integrar dicha optimización con los Sistemas Gestores de Bases de Datos (SGBD) existentes; facilitando el desarrollo de los SI.

Reconocida la necesidad explicada anteriormente, se propuso el desarrollo de un SI que integrara el proceso de normalización de Bases de Datos Relacionales (BDR) con los SGBD; con el objetivo de elevar la calidad de los modelos de datos y favorecer la interrelación de la etapa de diseño con la etapa de implementación de las diferentes soluciones informáticas que se realizan en la Universidad de Ciencias Informáticas (UCI). Además, se incorporó un módulo con objetivos docentes que permite la utilización del proceso de normalización de BDR en el proceso de enseñanza, beneficiando la enseñanza de esta temática a través de la visualización de algoritmos básicos del proceso.

El principal aporte de la investigación es la concepción y obtención de un SI capaz de automatizar e integrar el proceso de normalización de BDR y su aplicación en la enseñanza, con SGBD existentes.

Palabras Claves: Diseño de Bases de Datos; Normalización de Bases de Datos Relacionales; Sistema Informático.

Abstract

Given the explosive growth in the volumes of information in today's society and therefore increasing the information manipulated by computer systems (CS), it is necessary to optimize the processing of large volumes of information generated by our society and integrate this optimization with the Database Management System (DBMS) in place by facilitating the development of CS.

Recognized the need explained above, we propose the development of an CS that integrates the process of normalization of Relational Database (RDB) with the DBMS, with the aim of improving the quality of data models and encourage the interrelationship of the stage design to the stage of implementation of different solutions that are conducted at the Universidad de las Ciencias Informáticas (UCI). Also included is a module with teaching objectives, which allows use of the standardization process of RDB in the teaching process, benefiting the teaching of this subject through the display of basic algorithms of the process.

The main contribution of this research is the design and procurement of an CS capable of automating and integrating the process of normalization of RDB and their application in teaching, with existing DBMS.

Keywords: Database Design; Normalization of Relational Databases; Computer System.

Índice

Índice de Ilustraciones

Índice de Tablas

Introducción

Introducción

Desde los albores de la década de los sesenta, ante el fenómeno del crecimiento exponencial de la información, que constituye sin lugar a dudas una parte fundamental en la vida de las personas y en las sociedades contemporáneas; la necesidad de su tratamiento marcó evidentemente la multiplicación de la demanda de soluciones concretas para agilizar el proceso documental, llamado desde aquel entonces con el término "la revolución de la tecnología de la información". Tanto así, que Manuel Castells tituló a su ambicioso estudio sobre las transformaciones en las sociedades contemporáneas justamente, *La era de la información,* como una manera de nombrar "*el mundo surgido en las postrimerías del siglo XX…*" (Cabrera, Agosto del 2004).

La revolución de la tecnología de la información marcada como la "convergencia de tecnologías de la microelectrónica, la informática (computadoras y software), las telecomunicaciones, la optoelectrónica y la ingeniería genética" (Cabrera, Agosto del 2004), ha provocado algunos efectos que no se dejarán de mencionar por su gran repercusión sobre nuestra sociedad actual.
Según Cabrera:

- ✓ Se ha producido más información en los últimos treinta años que en los 500 anteriores.
- ✓ El volumen total de información científico-técnica se duplica cada cinco años.
- ✓ El 75% de la información disponible hoy día se ha generado en los últimos veinte años.
- ✓ La Información existente se duplica hoy cada 5 años; para 2010 se duplicará cada 72 días. (Saavedra, Octubre 2003).

El crecimiento considerable del volumen de información que se manipula en la sociedad actual, junto a la explosión tecnológica y otros factores que se derivan, ha propiciado un elevado desarrollo y utilización de los sistemas de gestión de la información (SGI) por el hombre; con el objetivo de aumentar la productividad y reducción de los costes, además de ofrecer una mayor disponibilidad de la información. Evidentemente, se hizo necesario el estudio y la utilización de técnicas o mecanismos desde la propia confección de los SGI de múltiples propósitos, para garantizar una adecuada y eficiente manipulación y obtención de la misma.

Actualmente, este incremento de la información también está influyendo de manera proporcional en el aumento del volumen de información que gestionan los SGI; lo que provoca una mayor dificultad en la

manipulación de la información y hace completamente engorroso la aplicación, de forma manual, de los mecanismos existentes para lograr un diseño correcto de la información que se almacena; aspecto de vital importancia que determina la eficiencia de los SGI.

DISEÑO TEÓRICO DE LA INVESTIGACIÓN

SITUACIÓN PROBLEMÁTICA

En el proceso y construcción de toda aplicación informática, el diseño de las bases de datos ocupa un lugar importante dentro de su desarrollo; a tal punto que esta puede verse como un proceso relativamente independiente dentro de la confección de un sistema, compuesto por una serie de etapas a realizar.

En el diseño de una base de datos relacional pueden encontrarse anomalías de inserción, eliminación, actualización de los datos, así como restricciones artificiales en la estructura de los mismos y dependencia entre ellos, implicando grandes problemas en la gestión y obtención de su información.

En la actualidad existen varias herramientas de software a nivel mundial que comprueban y dan solución a estas anomalías en la etapa del diseño de forma automática para modelos de BDR. Las herramientas encontradas fueron estudiadas y de ellas se analizaron las ventajas y desventajas que brindan para su uso. Este estudio arrojó como resultado que en estas herramientas están presentes uno o varios inconvenientes que van desde la interfaz visual, el idioma, los mecanismos para obtener los resultados, la integración con las próximas etapas del diseño que intervienen en el proceso de desarrollo de la aplicación, la falta de profundización en el proceso de normalización de BDR, hasta su ambiente de desarrollo propietario. Además, ninguna proporciona mecanismos que garanticen la integración de dicho resultado con los SGBD.

PROBLEMA

Del análisis de esta situación se desprende el siguiente problema a resolver:

La inexistencia de la integración del proceso de normalización de bases de datos relacionales con los sistemas gestores de bases de datos, provoca pérdida en la calidad de los modelos de datos y dificulta la interrelación de la etapa de diseño con la etapa de implementación de dichos modelos.

Se plantea como **objeto de estudio**: *Diseño de bases de datos relacionales*, enmarcado dicho estudio en el siguiente **campo de acción**: *Procesos de normalización de bases de datos relacionales y la integración con los SGBD.*

Como **objetivo general** se plantea:

Desarrollar un Sistema Informático que integre el proceso de normalización de bases de datos relacionales con los sistemas gestores de bases de datos; con el objetivo de elevar la calidad de los modelos de datos y favorecer la interrelación de la etapa de diseño con la etapa de implementación de las diferentes soluciones informáticas que se realizan en la UCI.

MARCO CONCEPTUAL

Para dejar claro el significado que para la presente investigación tienen todas las categorías y términos fundamentales que se emplean en el planteamiento del problema, los objetivos y la hipótesis se define el concepto de:

Calidad en los modelos de datos: *Modelos de bases de datos* con una estructura estable y lógica de manera que no sufran (o presenten escasas) anomalías de almacenamiento como: redundancia de datos, anomalías de actualización, anomalías de inserción, anomalías de borrado; modelos que puedan modificarse fácilmente para adquirir nuevos requerimientos, que permitan la recuperación sencilla de los datos en respuesta de solicitudes de los usuarios.

Se plantea la siguiente idea a defender:

El desarrollo de un Sistema Informático capaz de normalizar bases de datos relacionales e integrarse con los sistemas gestores de bases de datos, permitirá elevar la calidad en los modelos de datos y favorecerá la interrelación de la etapa de diseño con la etapa de implementación.

Objetivos específicos:

1. Evaluar la teoría y empleo de la normalización de BDR dentro del diseño de las bases de datos, así como el nivel de aplicaciones de la misma.

2. Valorar los productos disponibles que se utilizan para la normalización y el diseño de BDR, identificando sus ventajas y desventajas para el cumplimiento de los objetivos trazados en la investigación.
3. Identificar vía de integración con los SGBD.
4. Identificar estrategia que permita la utilización del proceso de normalización de BDR en el proceso de enseñanza y asimilación de sus algoritmos de forma independiente, por parte de los estudiantes de la UCI.
5. Implementar el sistema informático propuesto.

DISEÑO METODOLÓGICO DE LA INVESTIGACIÓN

Población o Universo: Desarrolladores o Estudiantes de la UCI.

Entre los **Métodos Científicos** utilizados en esta investigación se destacan los siguientes:

1. **Métodos teóricos :**
   - ✓ El ***método analítico-sintético***: Al descomponer el problema de investigación en el proceso de normalización de BDR y la generación de Código de Consulta Estructurado (SQL)[1] para los SGBD, con el objetivo de profundizar en el estudio de cada uno de ellos de forma independiente, y luego sintetizarlos en la solución de la propuesta.
   - ✓ El ***método hipotético-deductivo***: Para la elaboración de la idea a defender en la investigación y para proponer nuevas líneas de trabajo a partir de los resultados parciales.
   - ✓ El ***método sistémico***: La inclusión del proceso de normalización de BDR integrado con los SGBD en un único sistema para la corrección del modelo de datos como un todo, considerando la estrecha vinculación entre estos procesos que intervienen en el diseño de BDR.
   - ✓ Los ***métodos históricos-lógicos***: Para el estudio crítico de los trabajos previos que abordan la problemática planteada, para utilizarlos como puntos de referencia y de comparación con los resultados alcanzados.

1 Lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones.

2. **Métodos empíricos:**
   - ✓ ***Encuesta***: A especialistas de la producción y de la docencia en la UCI, con el objetivo de corroborar la situación problemática planteada en la presente investigación. (Ver encuesta en el **Anexo II**).
   - ✓ ***Entrevista:*** A especialistas de la producción y de la docencia de elevado nivel científico o metodológico, con el objetivo de corroborar la idea a defender planteada en la presente investigación.

El **Aporte** de la presente investigación se expresa en la obtención de un Sistema Informático capaz de automatizar e integrar el proceso de normalización de bases de datos relacionales y su aplicación en la enseñanza, con SGBD existentes.

A continuación se abordarán algunos criterios que justifican la investigación:

1. **Conveniencia y utilidad metodológica:** La herramienta será de gran utilidad para ***los estudiantes*** de la UCI o de carreras afines al perfil informático. Les servirá para apropiarse de nuevos conocimiento en la teoría del proceso de normalización de BDR, a través de la visualización de algoritmos básicos dentro del proceso; permitiéndole observar en cada paso de la ejecución los resultados que se van obteniendo hasta llegar a la solución.
   A ***los diseñadores*** les facilitará la construcción de modelos de BDR normalizados de forma automática, brindándole su integración directa con los SGBD como *PostgreSQL*, *Microsoft SQL Server y MySQL*; obteniendo mayor nivel de confiabilidad pues se utilizan algoritmos demostrados y probados matemáticamente por expertos en la materia que garantizan un alto grado de seguridad en la obtención de la solución.
   Actualmente la mayoría de los diseñadores de Bases de Datos (BD) realizan el diseño de la BD e integración con los SGBD de forma manual o independiente, permitiendo la inserción del error humano.

2. **Relevancia social:** El sistema contribuirá a que los futuros estudiantes de segundo año de la UCI, emerjan con un mayor nivel de conocimientos de la asignatura de "Sistemas de Bases de Datos" en la teoría de normalización de BDR, lo cual repercutirá de forma positiva en el desarrollo de soluciones informáticas realizadas por los estudiantes de la UCI y con ello brindará un mayor beneficio para nuestra sociedad.

La **Significación Práctica** de los resultados obtenidos en esta investigación se concentran, en las bondades que brinda la herramienta para mejorar el diseño de BDR en cualquier solución informática y trabajo colaborativo. En sentido general el sistema "SINORGES" y sus funcionalidades pueden resultar de gran utilidad dentro de la etapa de diseño de una BDR para su validación y corrección del modelo de datos, brindándole una mayor facilidad al diseñador.

El documento de tesis que se presenta, está compuesto por Introducción, Tres Capítulos, Conclusiones, Recomendaciones, Referencias Bibliográficas y Dos Anexos. El Primer Capítulo de "Fundamentos Teóricos", se expone una evaluación del estado del arte referente al surgimiento y los diferentes modelos de bases de datos que han existido hasta la actualidad, así como, definiciones y características de los mismos, haciendo énfasis fundamentalmente en las BDR, su impacto en el mercado, para entrar en el estudio de su etapa de diseño, principalmente en el proceso de normalización. Además, se describen y analizan algunos de los software existentes para la normalización y el diseño de BDR. El Segundo Capítulo "Justificación de la propuesta. Análisis y Diseño", se expone los diferentes elementos que fundamentan el desarrollo de la propuesta, se realiza una comparación de la propuesta con las diferentes herramientas existentes, se describe el flujo del proceso de normalización de BDR, la vía de integración con los diferentes SGBD, así como, la descripción de las diferentes funcionalidades del sistema. Se presentan los resultados obtenidos de la aplicación de la Metodología "Proceso Unificado de Rational" (RUP) por la cual se orientó el proceso de desarrollo de estas herramientas. En Tercer Capítulo "Implementación del Sistema. Validación y Prueba", se exponen las consideraciones para la selección del lenguaje en el que se desarrolló la herramienta, los artefactos generados por RUP en la etapa de implementación, los resultados arrojados tras la aplicación de encuestas a especialistas de computación en la UCI para la validación de la situación problemática planteada en la presente investigación, junto con las pruebas realizadas al sistema y el análisis de la viabilidad de la propuesta.

Desarrollo

Capítulo 1. Fundamentos Teóricos

En este capítulo se expone el estado del arte de los sistemas gestores de bases de datos, así como, definiciones, características, sus tipos y la repercusión en el mercado mundial. Se plasma un estudio detallado sobre la etapa de diseño de las bases de datos relacionales, específicamente la utilización de la técnica de normalización de bases de datos relacionales y algunos aspectos a tener en cuenta para su aplicación. Se realiza un análisis de algunos software existentes a nivel mundial con este mismo fin y otros desarrollados particularmente para el diseño de BDR. Se finaliza con las conclusiones generales del capítulo.

1.1 LA EVOLUCIÓN DE LOS SISTEMAS DE BASES DE DATOS

Los sistemas de información existen desde las primeras civilizaciones. El concepto más esencial de sistema de información no ha variado desde los registros romanos, por poner un ejemplo. Los datos se recopilaban, se estructuraban, se centralizaban y se almacenaban convenientemente. El objetivo inmediato de este proceso era poder recuperar los datos u otros datos derivados de ellos en cualquier momento, sin necesidad de volverlos a recopilar, paso que solía ser el más costoso o incluso irrepetible. El objetivo ulterior de un sistema de información, no obstante, era proporcionar a los usuarios información fidedigna sobre el dominio que representaban, con el objetivo de tomar decisiones y realizar acciones más pertinentes que las que se realizarían sin dicha información (Orallo, mayo 2002). Surgen así los primeros sistemas de bases de datos llamados "*sistemas de ficheros",* con el objetivo de informatizar el manejo de los archivadores manuales, proporcionándole un acceso más eficiente a los datos. Los *sistemas de ficheros* almacenaban datos durante un largo periodo de tiempo y permitían la existencia de grandes cantidades de estos. Sin embargo, los sistemas de ficheros no garantizaban, generalmente que los datos no se perdieran ante fallos bastante triviales, además presentaban una serie de inconvenientes como (Marqués, 10 de febrero del 2001):

- ✓ Separación y aislamiento de los datos.
- ✓ Duplicación de datos.
- ✓ Dependencia de datos.
- ✓ Formatos de ficheros incompatibles.
- ✓ Consultas fijas y proliferación de programas de aplicación.

Todas atribuidas por dos factores fundamentales:

- ✓ La definición de los datos se encuentra codificada dentro de los programas de aplicación, en lugar de estar almacenada aparte y de forma independiente.
- ✓ No hay control sobre el acceso y la manipulación de los datos más allá de lo impuesto por los programas de aplicación.

Para trabajar de un modo más efectivo, surgieron las *bases de datos* y los *sistemas de gestión de bases de datos*.

1.2 CONCEPTO DE BASE DE DATOS (BD)

Muchas son las definiciones que pueden encontrarse en la literatura acerca de BD, aunque la más referenciada es la del Dr. Christopher J. Date**.** Este define a las **BD** como: *un conjunto de* ***datos persistentes*** *que es utilizado por los sistemas de aplicación de alguna empresa dada.* (Date, 2001)

Datos persistentes: Es una costumbre referirse a los datos de la BD como "persistente", (¡aunque en realidad éstos podrían no persistir por mucho tiempo!). Por persistentes queremos decir, de manera intuitiva, que el tipo de datos de la BD difiere de otros datos más efímeros, como los datos de entrada, los datos de salida, las instrucciones de control, los bloques de control de software, los resultados intermedios y de manera general, cualquier dato que sea de naturaleza transitoria. En forma más precisa, se dice que los datos de la BD "persisten" en primer lugar una vez aceptados por el SGBD para entrar en la BD, en lo sucesivo sólo pueden ser removidos de la base de datos por alguna solicitud explícita al SGBD, no como un mero efecto lateral de por ejemplo: algún programa que termina su ejecución. (Date, 2001).

Otras definiciones tomadas son las siguientes:

- ✓ Rosa María, en su libro "Diseño de Bases de Datos" enuncia que es: *un conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que una BD puede considerarse una colección de datos variables en el tiempo*. (María Mato Gracía, Octubre 1999).
- ✓ Peter Rob y Carlos Coronel, en su libro "Sistemas de bases de datos: diseño, implementación y administración" plantean que es: una estructura de computadora integrada, compartida que aloja un conjunto de: (Peter Rob, 2003)

- Datos para el usuario final, es decir, hechos en bruto interesantes para este.
- *Metadatos*[1] o datos sobre datos, mediantes los cuales se integran los datos.

Hoy en día, sin embargo, solemos asociar las bases de datos con los ordenadores, y su gestión no suele ser manual, sino altamente automatizada. Más concretamente, la tecnología actual insta a la delegación de la gestión de base de datos a tipos de aplicaciones de software específicas denominadas *SGBD* o, simplemente, *sistemas de bases de datos*. Por esta razón, hablar de la tecnología de *bases de datos* es prácticamente lo mismo que hablar de la tecnología de los *sistemas de gestión de bases de datos* (Orallo, mayo 2002).

1.3 CONCEPTO DE SISTEMAS GESTORES O DE GESTIÓN DE BASES DE DATOS (SGBD)

Según Rosa María, un ***Sistemas de gestión de bases de datos*** es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez. (María Mato Gracía, Octubre 1999).

Según C. J. Date, un ***Sistemas de gestión de bases de datos*** es una aplicación que permite a los usuarios definir, crear y mantener la *base de datos*, y proporciona acceso controlado a la misma (Date, 2001).

1.3.1 FUNCIONES BÁSICAS DE LOS SGBD

Un SGBD realiza varias funciones importantes que garantizan la integridad y la consistencia de los datos de una base de datos. La mayoría de estas funciones son transparentes para los usuarios finales, y casi todas pueden realizarse sólo mediante un SGBD. Estas funciones incluyen (Peter Rob, 2003):

- ✓ La administración de un diccionario de datos.
- ✓ La administración de almacenamientos de datos.
- ✓ Transformación y representación de los datos.
- ✓ Administración de la seguridad, control de acceso a usuarios múltiples.

1 Describen las características de los datos y las relaciones que vinculan a aquellos que están incluidos en la base de datos.

- ✓ Administración de respaldos y recuperación.
- ✓ Administración de la integridad de los datos, lenguajes de acceso a bases de datos e interfaces de programación de aplicaciones e interfaces de comunicación con base de datos.

En la *Fig. 1* se ilustra el concepto que el SGBD mantiene entre la BD y los usuarios. Incluso, el SGBD actúa como intermediario entre los usuarios y la BD, transformando las solicitudes del usuario en código complejo necesario para atender dichas solicitudes.

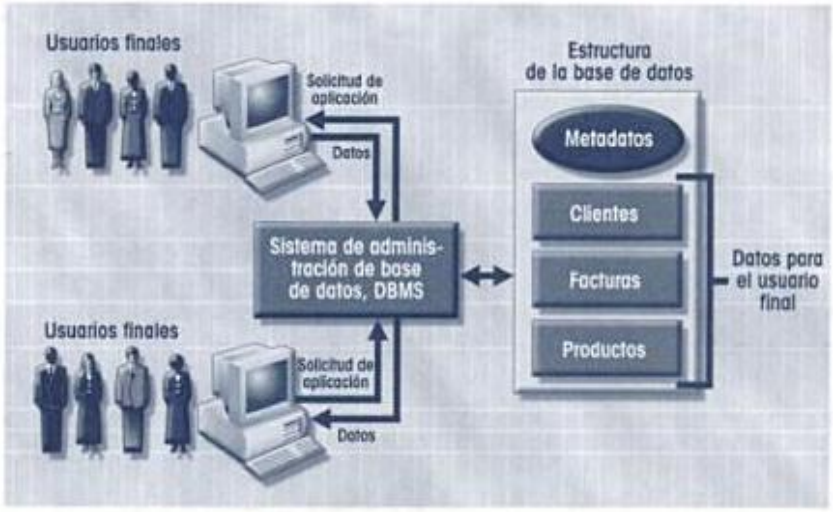


Fig. 1 Interacción del SGBD con el usuario final y la BD**.** (Peter Rob, 2003)

De la mayoría de las bibliografías consultadas por el autor de la investigación, coincidiendo con varios artículos encontrados en la web, entre los SGBD más utilizado a nivel mundial actualmente podemos encontrar: *Oracle, MySQL, PostgreSQL, Microsoft SQL Server*; existen otros gestores menos reconocidos como: *SQLite, Microsoft Access, DBase, Advantage Database, Fox Pro, Paradox, Open Access, NexusDB, Sybase IQ, WindowBase, IBM Informix*, etc.

1.4 PRIMEROS SISTEMAS DE BASES DE DATOS

Los primeros verdaderos sistemas de bases de datos, evolucionados de los sistemas de ficheros, obligaban a que el usuario visualizara los datos de manera muy parecida a como se almacenaban. Los primeros sistemas de ficheros habían logrado pasar del código máquina a un lenguaje ensamblador con ciertas instrucciones de acceso a disco, tales como la línea AS de International Business Machines

(IBM). No es de extrañar que con este nivel de abstracción la manera de recuperar los datos estuviera estrechamente ligada al lenguaje de programación utilizado. Un avance importante lo constituyó el comité formado en la ***CO****nference on* ***DA****ta* ***SY****stems and* ***L****anguages*, (CODASYL), en 1960 estableciendo el ***CO****mmon* ***B****usiness-****O****riented* ***L****anguage* (COBOL) como un lenguaje estándar para interrelacionar con datos almacenados en ficheros. Aunque hoy en día puede parecer un lenguaje "muy físico", en aquella época representó lo que se vinieron a llamar los lenguajes de programación de tercera generación. Pero pronto los discos magnéticos empezaron a sustituir a las cintas magnéticas, lo que supuso una re-concepción del almacenamiento, al pasarse del *acceso secuencial al acceso aleatorio* (este paso es el que se conoce como el paso de los sistemas de bases de datos de primera a segunda generación).

Durante la década del sesenta empezaron a aparecer distintos *modelos de datos o modelos de base de datos* para describir la estructura de la información en una base de datos, con el objetivo de conseguir una independencia un poco mayor entre las aplicaciones y la organización física de los datos. (Orallo, mayo 2002).

1.5 MODELOS DE BASE DE DATOS

Las ideas teóricas de bases de datos están representadas en los modelos de bases de datos. Entre los conceptos buscados, los más completos a consideración del autor sobre el ***modelo de bases de datos*** son:

- ✓ El emitido por el Dr. Ian Graham, en el libro "Métodos Orientados a Objetos", el cual pronuncia que *es un formalismo matemático que consta de una notación para describir los datos y las estructuras de datos (información), y de un conjunto de operaciones válidas que se pueden utilizar para manipular estos datos, o al menos, los símbolos que los representan* (Graham I. , 1996).
- ✓ El emitido por los Drs. Mario Piattini y Miguel A. en el libro "Fundamentos y Modelos de Bases de Datos" como:

  "... instrumento que se aplica a una parcela del mundo real (universo del discurso) para obtener una estructura de datos a la que denominamos esquema. Esta distinción entre el modelo (instrumento) y el esquema (resultado de aplicar el instrumento) es importante... Es importante también distinguir entre mundo real y universo del discurso, este último es la visión que del mundo real tiene el diseñador... podemos definir un modelo de datos como un

conjunto de conceptos, reglas y convenciones que permiten describir los datos del universo del discurso." (Miguel, Piattini 1997)

En el proceso de abstracción que conduce a la creación de una base de datos desempeña una función prioritaria el *modelo de datos*. El *modelo de datos*, como abstracción del universo de discurso, es el enfoque utilizado para la representación de las entidades y sus características dentro de la base de datos, y puede ser dividido en tres grandes tipos (Batini, 2004):

- ✓ ***Modelos lógicos basados en registros:*** Modelos utilizados para describir los datos en los modelos conceptual y físico. A diferencia de los modelos lógicos basados en objetos, se usan para especificar la estructura lógica global de la BD y para proporcionar una descripción a nivel más alto de la implementación.
  - *Modelo jerárquico.*
  - *Modelo red o reticular.*
  - ***Modelo relacional.***
- ✓ ***Modelos lógicos basados en objetos:*** Modelos utilizados para describir datos en el nivel conceptual y el externo. Se caracterizan porque proporcionan capacidad de estructuración bastante flexible y permiten especificar restricciones de datos.
  - *Entidad-Relación:* Se basa en una percepción del mundo compuesta por objetos, llamados entidades, y relaciones entre ellos. Las entidades se diferencian unas de otras a través de atributos.
  - *Semántico:* Modelos dedicados específicamente a representar la realidad sobre la cual versa la base de datos.
  - *Orientado a Objetos:* Se basa en objetos, los cuales contienen valores y métodos, entendidos como órdenes que actúan sobre los valores, en niveles de anidamiento. Los objetos se agrupan en clases, relacionándose mediante el envío de mensajes.

  Dentro de los modelos lógicos basados en objetos, el más extendido es el *modelo entidad-relación,* seguido por el *modelo orientado a objetos.*
- ✓ ***Modelos físicos de datos:*** Describen como se almacenan los datos en la computadora, representando informaciones tales como: las estructuras de los registros, el ordenamiento de los registros y las rutas de acceso. No abundan tantos modelos físicos como lógicos, los más usuales son: *modelo unificador y la memoria de marco.*

La descripción detallada del surgimiento, características particulares, ventajas y desventajas del *modelo jerárquico* podemos encontrarlo en (Peter Rob 2003), (Carlos Batini, et al., 1992) y del *modelo de red* en (Marqués 10 de febrero del 2001). Se analizarán en este capítulo: *el modelo relacional, el modelo entidad-relación y el modelo orientado a objetos* por sus repercusiones en el mercado desde su surgimiento hasta la actualidad.

1.5.1 MODELO RELACIONAL. BASES DE DATOS RELACIONALES (BDR)

En 1970 el Dr. Codd, de los laboratorios de investigación de IBM, escribió un artículo presentando el *modelo relacional*. En este artículo, presentaba también los inconvenientes de los sistemas previos, el jerárquico y el de red. Entonces, se comenzaron a desarrollar muchos sistemas relacionales, apareciendo los primeros a finales de los setenta y principios de los ochenta. Uno de los primeros es System R de IBM, que se desarrolló para probar la funcionalidad del modelo relacional, proporcionando una implementación de sus estructuras de datos y sus operaciones. Esto incitó dos grandes desarrollos (Marqués, 10 de febrero del 2001):

- ✓ El desarrollo de un lenguaje SQL, que se ha convertido en el lenguaje estándar de los sistemas relacionales.
- ✓ La producción de varios SGBD relacionales durante los años ochenta, como DB2 y SLQ/DS de IBM, ORACLE de ORACLE Corporation. Otros sistemas relacionales de microordenadores son: Paradox y dBase IV de Borland, Access de Microsoft, FoxPro y RBase de Microrim.

Los SGBD relacionales constituyen la **"*segunda generación de los SGBD*"**. La popularidad de este modelo ha ido creciendo lenta pero firmemente, de manera que el término relacional ha llegado a ser común entre los profesionales informáticos. El modelo relacional de datos es un modelo simple, potente y formal para representar la realidad. También ofrece una base firme para enfocar y analizar formalmente muchos problemas relacionados con la gestión de BD, como el diseño de la BD, la redundancia, la distribución, etcétera. El formalismo y una base matemática son las piedras angulares en el desarrollo de la teoría de las bases de datos relacionales (Batini, 2004).

La estructura fundamental del modelo relacional es la "***relación"***, es decir una tabla bidimensional constituida por *filas (tuplas)*[1] y *columnas (atributos)*. Las relaciones representan las entidades que se consideran interesantes en la base de datos. Cada instancia de la entidad encontrará sitio en una tupla

1 Las **tuplas** en una relación son un conjunto en el sentido matemático del término, es decir una colección no ordenada de elementos diferentes.

de la relación, mientras que los atributos de la relación representan las propiedades de la entidad (Quiroz, 2003).

Por ejemplo, si en la base de datos se tienen que representar personas, podrá definirse una relación llamada *"Personas"*, cuyos atributos describen las características de las personas. Cada tupla de la relación *"Personas"* representará una persona concreta, tendríamos la relación representada de esta manera:

Personas (*RFC, nombre, apellido, sexo, estadoCivil, fechaNacimiento***)**

Como características favorables significativas de este modelo podemos citar:

- ✓ *Independencia estructural:* Se pueden realizar cambios en la estructura de la base de datos sin que se vea afectado la capacidad de los SGBD para acceder a los datos, no siendo así para el caso de los modelos jerárquicos y de red.
- ✓ *Simplicidad conceptual mejorada:* El modelo de BDR es más simple a nivel conceptual que los modelos anteriores, su atención se concentra en el panorama lógico de la base de datos e ignora las características de almacenamiento de los datos físicos.
- ✓ *Diseño, ejecución, administración y uso más fácil de las bases de datos:* El modelo relacional logra al mismo tiempo independencia de los datos e independencia estructural, lo que hace más fácil su diseño y administrar su contenido.
- ✓ *Capacidad de consultas ad hoc:* Una de las razones por las que el modelo relacional tiene una posición dominante en el mercado es su poderosa y flexible capacidad de consulta. En la mayoría del software de la BDR, el lenguaje SQL, es un lenguaje de cuarta generación, le permite al usuario especificar qué debe hacer sin tener que decir cómo se debe hacer. El SGBD relacional utiliza SQL para trasformar la consulta del usuario en el código tecnológico requerido para recuperar los datos solicitados. Por consiguiente, la BD relacional requiere menos programación que otras BD o ambientes.
- ✓ *Un poderoso sistema de administración de base de datos:* Un buen SGBD relacional es una pieza de software mucho más compleja que el SGBD utilizado en las BD jerárquicas y de red. Su complejidad se debe a que realiza muchas más (y más complejas) tareas tanto para sus diseñadores como para sus usuarios. Por ende, un buen SGBD relacional disminuye la complejidad física de un sistema tanto para el diseñador como para el usuario final de la base de datos.

Gracias a que el SGBD relacional realiza las tareas del hardware ocultas, no hay necesidad de enfocarse en los aspectos físicos de la base de datos. Por el contrario, los esfuerzos deberán concentrarse en la parte lógica de la base de datos relacional y ***en su diseño***.

El modelo relacional también presenta algunos fallos, siendo uno de ellos su limitada capacidad para modelar los datos (SGBD, 2004). Entre sus desventajas podemos citar:

- ✓ *Gastos indirectos sustanciales para el software y el hardware del sistema:* El mismo SGBD relacional que esconde la mayoría de las complejidades del sistema, también es la causa de que se requieran gastos sustanciales para el sistema operativo y para el hardware. Simplemente se requiere una computadora más poderosa para realizar todas las tareas asignadas por el SGBD relacional. Por lo que, el sistema de bases de datos relacional tiende a ser más lento que otros sistemas. Sin embargo, gracias a la rápida evolución de la capacidad del hardware y a las constantes mejoras de los sistemas operativos, la etiqueta de *"lento"* ha comenzado a desvanecerse.
- ✓ *El diseño y la ejecución deficientes son más propicias:* En cierto sentido, la facilidad de uso del ambiente relacional también se puede convertir en un inconveniente. El software relacional, particularmente al nivel de las microcomputadoras, es tan fácil de usar que las personas relativamente inexpertas generan con facilidad reportes y consultas útiles sin pensar mucho en la necesidad de diseñar una base de datos apropiada. La falta de un diseño apropiado tiende a hacer más lento el sistema y a producir anomalías en los datos encontrados en los sistemas de archivos.

1.5.2 MODELO ENTIDAD-RELACIÓN (E-R). BASES DE DATOS RELACIONAL

En 1976, el Dr. Peter Chen presentó el modelo entidad-relación, que es la técnica más utilizada en el diseño de bases de datos, con el objetivo de erradicar las deficiencias del *modelo relacional* para representar los datos de una manera más real (Orallo, mayo 2002).

Debido a la simplicidad de este modelo, su expresividad y la relativamente sencilla transformación de este modelo a un esquema lógico relacional se popularizó rápidamente como herramienta para representar el modelo conceptual de un sistema de información, creando la clásica separación en las etapas del desarrollo de un sistema de información: *diseño conceptual, diseño lógico, diseño físico e implantación.*

A pesar de la claridad y facilidad que aportaba el modelo E-R para modelar problemas reales, el modelo E-R era exclusivamente estático, expresaba las entidades de la realidad y sus relaciones. El diseño conceptual basado en los Diagramas E-R (ERD) adolecía, por tanto, de dinámica. Para paliar esta carencia, inicialmente se solía combinar el E-R con los populares Diagramas de Flujo de Datos (DFD) (Baizán, 1987).

En 1979, Codd intentó subsanar algunas de las deficiencias de su modelo relacional con una versión extendida denominada RM/T en 1979 y posteriormente RM/V2 en 1990 (Marqués 2001). Alrededor de la mitad de los ochenta, algunas aplicaciones exigían mayor expresividad en los datos con los que trabajaban. Como respuesta a la creciente complejidad de dichas aplicaciones surgió el nuevo modelo de datos: *orientado a objetos*. Esta evolución representa la ***"tercera generación de los SGBD".***

1.5.3 MODELO ORIENTADO A OBJETO. BASES DE DATOS ORIENTADAS A OBJETOS (BDOO)

La idea de una base de datos orientada a objetos se articuló por primera vez en 1984 con el sistema prototipo *GemStone*. Uno de los sistemas más famosos de los finales de los ochenta y principios de los noventa fue el sistema *ObjectStore*. Al principio de los noventa, los primeros Sistemas de Gestión de Bases de Datos Orientados a Objetos (SGBDOO, o simplemente, SGBDO) empezaron a aparecer en el mercado, a partir de compañías como *Objectivity* (Stonebraker, 1999).

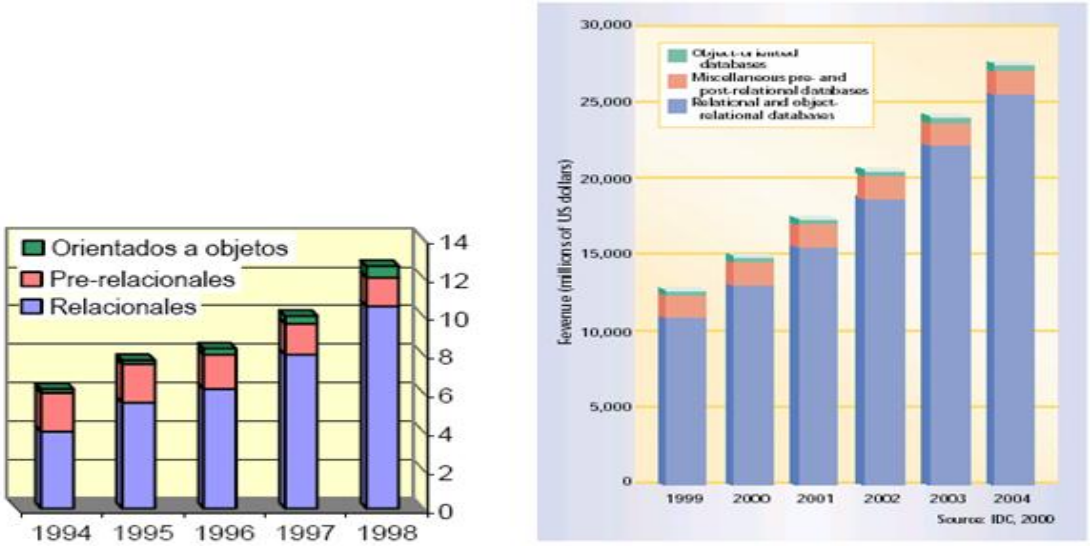
En el modelo orientado a objeto la información sobre una entidad se almacena como un objeto persistente y no como una fila en una tabla, aspecto que en principio, lo hace más eficiente en términos de requerimientos de espacio y asegura que los usuarios puedan manipular los datos sólo de las maneras en las que el programador haya especificado. También es más eficiente en el uso de espacio de disco requerido para las consultas, pues en vez de almacenar la consulta, simplemente se construye una serie de índices (punteros) a los objetos seleccionados; añadiéndole las ventajas derivadas del modelo orientado a objetos, ya explotadas en sus lenguajes de programación, la mayor expresividad y su adecuación para almacenar muchos tipos de datos diferentes.

Se pudiera pensar, por lo antes mencionado desde aquel entonces, que hoy en día las bases de datos orientadas a objetos superarían en la práctica a las bases de datos relacionales. Veamos a continuación

algunos estudios, realizados por el autor de la presente investigación, sobre el comportamiento en el mercado mundial de los SGBD.

1.6 EL FUTURO DEL MODELO RELACIONAL. IMPACTO EN EL MERCADO DE LAS BASES DE DATOS RELACIONALES

Ya desde principios de 1990 se hablaba del "***fin del modelo relacional***" y su sustitución por las "*bases de datos orientadas a objetos"*. Pero el caso es que en el año 2001 de 8 mil 884 millones de dólares pagados por licencias de bases de datos, 7 mil 107 correspondieron al modelo relacional. Esto supone un 80,4% de los *nuevos* sistemas, con lo que no parece haber ninguna señal clara de que la situación vaya a cambiar en el futuro (Graham C. , 6 May 2002). Más significativo es el hecho de que en el año 2000 las ventas para bases de datos relacionales tuvieron un incremento del 15% en tanto, para bases de datos orientadas a objetos y de otro tipo tuvieron un incremento negativo (IDG, 23 de mayo del 2001) (SMITT, 2002). O sea, después de más de 15 años, el mercado de las bases de datos orientadas a objetos no supone más de un 5 % del mercado de las relacionales, como se puede ver en las siguientes gráficas que muestra la *Fig. 2* desde 1994 hasta el 2004:



(a) (b)

Fig. 2 Ingresos de las Ventas Mundiales de SGBD
(a) Ingresos entre 1994 y 1998 (miles de millones de dólares). (Miguel, Piattini,1997)
(b) Ingresos entre (1999 y 2000) y predicciones (2001-2004). (Leavitt, August 2000)

Existen varias razones para explicar este hecho:

- ✓ En primer lugar, las bases de datos de objetos acarrean consigo algunas de las propiedades no deseables de los modelos pre-relacionales. El programador tiene que tener mucha información sobre la estructura de los datos. Si se conocen las propiedades de los objetos, la consulta es rápida y simple. Pero la realidad es que en muchos casos se desconocen las identidades de los objetos. Lo que preocupa o interesa es almacenar los atributos de los objetos y relacionar los valores de estos atributos, aspecto en el que el modelo relacional es más sencillo. (Orallo, mayo 2002)
- ✓ En segundo lugar, el hecho de que las organizaciones sean capaces de alterar los métodos de bajo nivel utilizados en los SGBDO, hace que sea más difícil para terceros el hacer productos añadidos. Mientras que las bases de datos relacionales se pueden beneficiar del software realizado por otras firmas, los usuarios de SGBDO tienen que producir el software en casa para adaptarse a sus propias particularidades, parte de ellas incorporadas al *comportamiento* de los objetos. (Orallo, mayo 2002)
- ✓ En tercer lugar, para las BD orientadas a objetos aún no ha sido propuesto un buen modelo de lenguaje de consulta, inducido por varias razones (Carlos Batini, et al., 2004):
    - Primeramente, la estructura del modelo de datos que es más compleja y la ausencia de una definición formal, son obstáculos predominantes para definir un lenguaje que sea simple y poderoso. Se han sugeridos lenguajes que se derivan de los modelos semánticos y funcionales. Hasta hoy en día no ha emergido ninguno y los estudios demuestran que la tarea de diseñar un lenguaje de consultas simples para modelos de datos tan complejo no es nada fácil.
    - Como segundo problema, es que los lenguajes de consultas y el encapsulamiento son en cierto sentido conceptos contradictorios. El principio del encapsulamiento estipula que la estructura de los datos (implementación) debe esconderse de los usuarios, mientras que las consultas deben en lo esencial ser capaces de ver los atributos de un objeto.
- ✓ En cuarto lugar, las organizaciones tienden a ser conservadoras en relación con las bases de datos que utilizan. Muchas organizaciones aunque utilizan lenguajes orientados a objetos para las aplicaciones, desconfían de los lenguajes orientados a objetos en general por no considerarlos suficientemente estables para trabajar con información crucial para la organización. Mito o realidad, el hecho es que no acaban de decidirse por embarcarse en un

SGBDO y siguen aferrados al SQL para realizar sus informes y al SQL embebido para interrelacionar las aplicaciones con el SGBD, manteniendo una separación que consideran imprescindible.

La adición de nuevas características al modelo relacional es asunto de intenso debate así como la modernización y adecuación del lenguaje SQL a las exigencias siempre cambiantes de un entorno de gran competencia. Sin duda, el modelo ha superado la prueba del tiempo. Los proveedores han añadido características de "objetos" a sus productos, lo cual permite a los usuarios definir sus propios tipos de datos. En resumen, el modelo relacional de bases de datos es un estándar de la industria consolidado, una tecnología confiable y eficiente que estará entre nosotros aún por muchos años antes de que sea desplazada por una nueva y mejor (Quiroz, 2003).

1.7 DISEÑO DE BASES DE DATOS RELACIONALES

Las dos últimas décadas se ha caracterizado por un fuerte crecimiento en el número e importancia de las aplicaciones de bases de datos. Las BD son componentes esenciales de los sistemas de información, usadas en todos los ordenadores. El diseño de BDR se ha convertido en una actividad popular, desarrollada no sólo por profesionales sino también por no especialistas.

A finales de la década de 1960, cuando las bases de datos entraron por primera vez en el mercado de software, los diseñadores de bases de datos actuaban como artesanos, con herramientas muy primitivas: *diagramas de bloques y estructuras de registros* eran los formatos comunes para las especificaciones, y el diseño de BD se confundía frecuentemente con la implantación de las BD. Esta situación ahora ha cambiado: los métodos y modelos de diseño de BD han evolucionado paralelamente con el progreso de la tecnología en los sistemas de BD. Se ha entrado en la era de los sistemas relacionales de BD, que ofrecen poderosos lenguajes de consultas, herramientas para el desarrollo de aplicaciones e interfaces amables con los usuarios, La tecnología de BD cuenta ya con un marco teórico, que incluye la teoría relacional de datos, procesamiento y optimización de consultas, control de concurrencia, gestión de transacciones y recuperación, etc. (Batini, 2004)

Desafortunadamente, las metodologías de diseño de BD no son muy usadas; la mayoría de los diseñadores confían muy poco en ellas para llevar a cabo el diseño, lo cual se considera con frecuencia, una de las principales causas de fracaso en el desarrollo de los sistemas de información. A menudo, subestiman el tiempo o los recursos necesarios para un proyecto de BD, lo que proporciona que estas

sean inadecuadas o ineficientes en relación con la demandas de la aplicación, la información que gestiona sea limitada y con el riesgo de no poder extender el diseño sin tener que reestructurar este, al surgir nueva información relacionada con lo modelado y el mantenimiento es difícil.

Muchos de estos problemas se deben a la falta de claridad que permita entender la naturaleza exacta de los datos. En muchos casos, los datos se describen desde el comienzo del proyecto en términos de las estructuras finales de almacenamiento; no se da peso a un entendimiento de las relaciones que existen entre ellos, que sean a su vez independientes de los detalles de manipulación y visualización.

1.7.1 SURGIMIENTO DE LA TÉCNICA DE NORMALIZACIÓN DE BDR

La normalización de BDR fue introducida por el *Dr. Edgar F. Codd*[1] a principios de los 70, poco después de crear el modelo relacional (Baizán, 1987), es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional, es una etapa posterior a la correspondencia entre el esquema conceptual y el esquema lógico. La normalización pretendía añadir calidad a los diseños de BDR evitando la aparición de datos redundantes y los problemas de actualización que conllevan. Normalizar *esquemas de relación*[2] de una BDR supone, en primer lugar, comprobar una serie de normas, llamadas "*formas normales"*, basadas en dos conceptos fundamentales: "*Dependencias Funcionales y Claves"*, y en segundo lugar, descomponer los esquemas que no las cumplen para obtener esquemas normalizados.

Otra definición de ***normalización***, que merece la pena resaltar fue pronunciada por ***el grupo de Bases de Datos Avanzadas (LABDA), de la Universidad Carlos III de Madrid***, los cuales expresan: *que es la descomposición sin pérdida de información ni de semántica de la relación universal (o de una colección de relaciones equivalentes a la misma) en una colección de relaciones en el que las anomalías de actualización (inserción, borrado y modificación) no existan o sean mínimas*.

1 Brillante matemático y científico de la computación, creador de modelo de datos relacional y toda su teoría, trabajo toda su vida en la IBM, donde realizaba sus labores de investigación en el área de la computación, considerado uno de los más prestigiosos en el campo de la informática.

2 Estructura abstracta que define a una relación a través de un nombre, un conjunto de atributos y un conjunto de dependencias funcionales.

1.7.1.1 DEPENDENCIAS FUNCIONALES (DF)

Existen varias definiciones de DF propuesta por diferentes autores prestigiosos de la literatura, todas muy relacionada con la teoría de conjunto. Algunas de estas definiciones la podemos encontrar en: (Date, 2001) como: Sea ***R*** un esquema de relación, y sean ***X, Y*** $\subseteq$ ***R***. Se dice que ***X*** *determina funcionalmente a* ***Y,*** o que, ***Y*** *depende funcionalmente de* ***X*** y lo denotamos con ***X*** $\rightarrow$ ***Y***, si y sólo si para cualquier instancia ***r*** no existen dos tuplas que coincidan en ***X*** y no coincidan en ***Y***.

(Baizán, 1987) como: Una sentencia de la forma ***X*** $\rightarrow$ ***Y***, que se lee: "***X*** *implica* ***Y*** " o "***Y*** *depende de* ***X***", siendo ***X*** e ***Y*** subconjunto del conjunto de atributos del esquema de relación (***R***). Si (***X*** $\rightarrow$ ***Y***) es subconjunto del conjunto de dependencias funcionales de ***R***, quiere decir que para toda ocurrencia ***r*** de ***R***, el valor de ***X*** determina unívocamente el valor de ***Y***. (Si ***X*** $\rightarrow$ ***Y*** y también ***Y*** $\rightarrow$ ***X***, de ***X*** e ***Y*** se dice que son descriptores equivalentes).

Intuitivamente, dado un conjunto de dependencias funcionales, hay otras que se deducen como consecuencia de ellas. Al conjunto inicial de DF, completado con todas las dependencias que de el se deducen, se le denomina ***cierre*** y se representa por DF^+ (obviamente: DF es subconjunto de DF^+). El cálculo de DF^+ se fundamenta en los denominados "**Axiomas de Armstrong"** (Baizán, 1987)**.**

- ✓ *Reflexividad:* Si $\forall$***X***, entonces ***X*** $\rightarrow$ ***X***.
- ✓ *Aumentatividad:* Si ***X*** $\rightarrow$ ***Y*** entonces ***X'*** $\rightarrow$ ***Y*** ***X*** $\subseteq$ $\forall$ ***X'***
- ✓ *Proyectividad:* Si ***X*** $\rightarrow$ ***Y*** entonces ***X*** $\rightarrow$ ***Y'*** $\forall$ ***Y'*** $\subseteq$ ***Y***
- ✓ *Aditividad:* Si ***X*** $\rightarrow$ ***Y*** y ***U*** $\rightarrow$ ***W*** entonces ***X*** $\cup$***Y*** $\rightarrow$ ***U*** $\cup$ ***W***
- ✓ *Transitividad:* Si ***X*** $\rightarrow$ ***Y*** y ***Y*** $\rightarrow$ ***Z*** entonces ***X*** $\rightarrow$ ***Z***.

Es sumamente importante resaltar que ***no podemos deducir a partir de una instancia*** r ***que dependencias funcionales se cumplen en R***. Las dependencias funcionales representan restricciones de la realidad. Por consiguiente, ***la única manera de determinar las dependencias funcionales que se cumplen en un esquema R es analizando cuidadosamente las restricciones de la realidad que estamos representando***. O sea, las DF es un concepto que deriva del significado de los datos y no del comportamiento de la relación.

El autor considera necesario, a la hora de realizar el análisis y diseño de cualquier sistema informático para la gestión de información, primeramente ***conocer*** cuáles son las restricciones o relaciones que existen entre sus datos y como segundo paso ***representar*** de manera formal, con la notación que lo exige, cada una de estas restricciones o DF. Se ha detectado que en la mayoría de las documentaciones que describen el diseño de la BDR de aplicaciones informáticas, tanto en trabajos de diplomas presentados por estudiantes de nuestra universidad como en los artefactos relacionados con esta etapa del modelo del sistema, no se especifican las restricciones o relaciones existentes entre los datos; lo que trae consigo que resulte difícil la comprobación de la validez del modelo que se representa, atendiendo a los diferentes niveles de normalizaciones que existen.

1.7.1.2 DEPENDENCIAS FUNCIONALES COMPLETA O TOTAL

Definición: Se dice que el atributo **Y** de **R** *depende funcionalmente por completo o totalmente* del atributo **X** de **R**, si depende funcionalmente de **X** y no depende funcionalmente de ningún subconjunto de **X** (Ullman, 1988).

Coincidiendo con esta definición y expresado en otras palabras, podemos decir que: *no existe un subconjunto **Z** de atributos componentes de **X** de los cuales **Y** dependa funcionalmente.*

1.7.1.3 DEPENDENCIAS FUNCIONALES TRANSITIVAS

Definición: Entre dos descriptores X e Y tiene lugar una dependencia transitiva, cuando se cumplen las siguientes condiciones (Miguel,A et al. 1999):

- ✓ $X \cap Y = \emptyset$
- ✓ $\exists Z : X \cap Z = \emptyset \; Y \cap Z = \emptyset$
- ✓ $\exists Z : X \rightarrow Z \; , \; Z \nrightarrow X, \; Z \rightarrow Y$

1.7.1.4 CLAVES O LLAVES

Definición: Sea **R** un esquema de relación, **F** un conjunto de dependencias funcionales y $\mathbf{X} \subseteq \mathbf{R}$. Se dice que **X** es clave de **R** si y sólo si **X** es un conjunto minimal de atributos que determina funcionalmente todos los atributos en **R**, es decir: (Miguel, A et al. 1999).

(a) $X \rightarrow R \in F^+$

(b) No existe $Y \subseteq X$, tal que $Y \rightarrow R \in F^+$ (X es minimal)

Puede suceder que exista más de una clave para un esquema ***R***. En casos como estos, suele designarse a una de ellas como ***clave primaria***. El término ***clave candidata*** se usará para aquellas claves que no han sido designadas como primarias.

Cuando un conjunto de atributos ***X*** satisface la condición (a) pero no la condición de minimalidad, se dirá que ***X*** es una ***superclave***.

Se llamará ***atributos primos o principales*** a todos aquellos que participan en alguna clave, y ***atributos no primos o no principales*** a aquellos que no participan en ninguna clave.

1.7.2 FORMAS NORMALES

Las relaciones se pueden clasificar por tipos de anomalías de modificación a las cuales son vulnerables. En la década de 1970, los teóricos relacionales como el Dr. Codd, Dr.Boyce, Dr. Ronald Fagin, entre otros, investigaron acerca de estos tipos de anomalías y cuando encontraban una anomalía, la clasificaban y pensaban en una manera de prevenirla. Cada vez que ocurría se mejoraban los criterios de diseño de las relaciones. Estas clases de relaciones, así como las técnicas para prevenir anomalías, se denominaban ***Formas Normales.*** En un trabajo posterior a su artículo de 1970, Codd y otros definieron la Primera Forma Normal (1FN), Segunda Forma Normal (2FN) y Tercera Forma Normal (3FN). Más tarde en el año 1974, debido a que la 3FN no contemplaba algunos casos particulares, el Dr. Boyce ayudó a Codd a redefinir la 3FN y fue lo que se llamó Forma Normal de Boyce-Codd (FNBC), todas esta basadas en *dependencias funcionales*. Posteriormente en 1977 y 1979 se definieron la Cuarta Forma Normal (4FN) y la Quinta Forma Normal (5FN), estas dos últimas creadas por el Dr. Ronald Fagin (Date, 2001). En la *Fig. 3.* se muestra cómo se pueden agrupar las relaciones en función de su estado de normalización, así como, la relación de cada una de ellas.

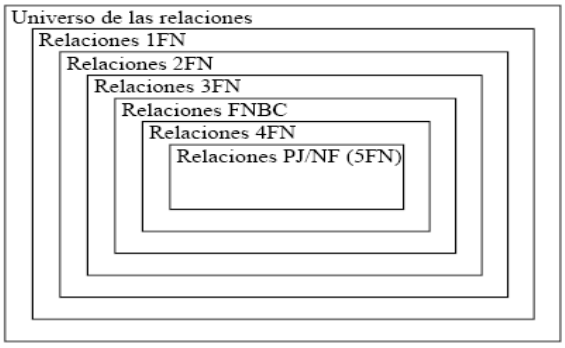


Fig. 3 Agrupamiento de relaciones en función de su estado de normalización. *(Date, 2001)*

El proceso de normalización de Codd (*eliminación de grupos repetitivos*, *eliminación de dependencias parciales*, *eliminación de dependencias transitivas*), reduce (en ocasiones elimina por completo) la incidencia de anomalías de inserción, eliminación y actualización (Baizán, 1987). Muchos investigadores consideran que una base de datos que cumpla con las restricciones de la 3FN y FNBC presenta un diseño adecuado. Por ello, el autor de esta investigación, solo ha centrado su estudio, y el desarrollo de la aplicación, hasta la FNBC.

1.7.2.1 PRIMERA FORMA NORMAL (1FN)

Definición: El término 1FN describe una relación en la cual (Kroenke, 2003):

- ✓ Todos los atributos claves están definidos.
- ✓ No existen grupos repetidos en la tabla. En otras palabras, cada intersección de fila/columna puede contener uno y sólo un valor, no un conjunto de valores.
- ✓ Todos los atributos son dependientes de la clave primaria.

1.7.2.2 SEGUNDA FORMA NORMAL (2FN)

Definición: El término 2FN describe una relación en la cual (Peter Rob, 2003):

- ✓ Está en 1FN.
- ✓ Todos sus *atributos no primos* tienen *dependencias funcionales total* respecto a cada una de las claves.

Aún podemos observar que es posible que una relación que cumpla con 2FN exhiba dependencias transitivas; es decir uno o más atributos pueden ser funcionalmente dependientes de *atributos no primos*.

1.7.2.3 TERCERA FORMA NORMAL (3FN)

Definición: El término 3FN describe una relación en la cual (Peter Rob, 2003):

- ✓ Está en 2FN.
- ✓ Ningún *atributo no primo* depende transitivamente de ninguna clave.

1.7.2.4 FORMA NORMAL DE BOYCE-CODD (FNBC)

Definición: Una relación está en FNBC, si todo determinante[1] en ella es una *clave candidata* (Kroenke, 2003).

Otra definición equivalente, la encontramos en (Baizán, 1987) como:

Definición: Una relación se encuentra en FNBC cuando para todas dependencias no trivial $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ (o sea, tal que $\boldsymbol{Y} \not\subseteq \boldsymbol{X}$, $\boldsymbol{Y} \neq \emptyset$), $\boldsymbol{X}$ es *clave o superclaves.*

Evidentemente, si una tabla contiene solamente una *clave candidata*, la 3FN y la de FNBC son equivalentes. Si se plantea esta proposición de otra manera, la FNBC puede ser violada sólo si la relación contiene más de una *clave candidata.*

La mayoría de los diseñadores consideran la FNBC como un caso especial de la 3FN. De hecho, si se utilizan las técnicas mostradas, la mayoría de las tablas se ajustan a los requerimientos de FNBC una vez que se llega a la 3FN. Así pues: ¿cómo puede estar una relación en 3FN y no estarlo en FNBC? Para responder esta pregunta se debe tener en cuenta que existen dependencias transitivas cuando un *atributo no primo* depende de otros atributos que tampoco son primos. En otras palabras, una relación está en 3FN si está en 2FN y no existen dependencias transitivas. Pero ¿qué pasa cuando un *atributo no primo* es el determinante de un *atributo primo*? Esta condición no viola la 3FN, sin embargo no satisface los requerimientos de FNBC, porque este requiere que todo determinante en la relación sea una *clave candidata.*

1.7.3 VENTAJAS DE LA TÉCNICA DE NORMALIZACIÓN DE BDR

Uno de los retos en el diseño de la base de datos relacionales es el de obtener una estructura estable y lógica tal que:

- ✓ El sistema de base de datos no sufra de anomalías de almacenamiento.
    1. **Redundancia de datos:** Repetición de datos en un sistema**.**
    2. **Anomalías de actualización:** Inconsistencias de los datos como resultado de redundancias y actualizaciones parciales.
    3. **Anomalías de inserción:** Imposibilidad de incluir datos en la base de datos debido a la ausencia de otros.

1 Es cualquier atributo cuyo valor determina otros valores de atributos.

4. **Anomalías de borrado:** Pérdida de información no intencionadas debido a que se han borrado otros datos.

- ✓ El modelo lógico pueda modificarse fácilmente para admitir nuevos requerimientos.

Una base de datos implantada sobre un modelo bien diseñado tiene mayor esperanza de vida, aunque aumente su tamaño o su ambiente sea dinámico, que una base de datos con un diseño pobre, y será lo suficientemente flexible para incorporar nuevos requerimientos o características adicionales.

La normalización de BDR se lleva a cabo por cuatro razones fundamentales:

- ✓ Estructurar los datos de forma que se puedan representar las relaciones pertinentes entre los datos.
- ✓ Permitir la recuperación sencilla de los datos en respuesta a las solicitudes de consultas y reportes.
- ✓ Simplificar el mantenimiento de los datos actualizándolos, insertándolos y borrándolos.
- ✓ Reducir la necesidad de reestructurar o reorganizar los datos cuando surjan nuevas aplicaciones.

En términos más sencillos la normalización trata de simplificar el diseño de una base de datos relacional, a través de la búsqueda de la mejor estructuración que pueda utilizarse con las relaciones involucradas en ella.

1.7.4 CONSIDERACIONES SOBRE ASPECTOS A TENER EN CUENTA EN EL USO DE LA TÉCNICA DE NORMALIZACIÓN DE BDR

Todo lo que se ha reflejado hasta ahora da muestras de las principales bondades y ventajas del proceso de normalización en la etapa de diseño para BDR, que son significativas, fundamentalmente en el desarrollo de aplicaciones informáticas con fines comerciales, así como también, en el campo de la educación, dentro del proceso de enseñanza de esta teoría que juega un papel primordial en la asignatura de BD de todas las carreras de corte Informática. Sin embargo, hay que tener en cuenta un conjunto de aspectos importantes que se deben llevar presentes para poder decidir la aplicación de esta técnica y el nivel de normalización para el diseño de los diferentes problemas, que puede atentar contra el rendimiento y uso efectivo del sistema que se esté desarrollando. Sobre las limitaciones de la utilización de esta teoría, no se encontraron en la bibliografía consultada criterios relevantes, que

entrara en contradicción con lo hasta aquí planteado. Independientemente de su importancia, se han publicado algunos trabajos que advierten sobre algunos aspectos a considerar a la hora de realizar el proceso, para un trabajo más correcto y efectivo con ellos.

Algunas de estas consideraciones son:

- ✓ ***Desnormalizar*:** La "*desnormalización"* consiste en renunciar a tener la BDR en una forma normal más alta, *permitiendo en muchos casos redundancia en la información almacenada*, en beneficio de la ***"eficiencia"*** con respecto al tiempo de respuesta de las consultas realizadas sobre dicha información. El proceso de normalización exige, muchas veces, situar en diferentes tablas datos relacionados entre sí, lo que puede provocar en dependencia de la frecuencia con que se consulten los datos relacionados una demora en el tiempo de respuesta, debido, que si volvemos a juntarlos se requiere ejecutar la operación de reunión (*join*). Si se decide mantener dichos datos juntos en la misma tabla, se ahorran tales operaciones. Esto supone documentar y gestionar correctamente la redundancia de datos que ello implica.

- ✓ El programador o diseñador debe tener presente que el diseño de las bases de datos relacionales es un proceso que requiere tiempo, se necesita realizar un estudio exhaustivo sobre toda la información que se desea modelar y las relaciones que existen entre ellas.

El colectivo de especialista en (et al, 2008) opinan sobre el artículo: *"Considerando la introducción de redundancia (Desnormalización)"* que:

...Si el costo de la obtención de información a partir de los datos relacionados fuera considerablemente más grande que el costo de su mantenimiento entonces surge la necesidad de desnormalizar.

El introducir elementos redundantes en la tabla de índice y estructuras puede resultar beneficioso, e incluso producir mejoras en el rendimiento de bases de datos. Donde para determinar los niveles de normalización a utilizar en la base de datos se deberán tener en cuenta varios factores como el tamaño de la aplicación, el número de accesos concurrentes sobre las tablas, la velocidad de las consultas, entre otros. Por este motivo, el uso de una base de datos normalizada no es siempre la mejor alternativa. ***Por lo general, si un número importante de las consultas precisan de la combinación de más de cinco o seis tablas, es aconsejable su uso.*** *Es importante tener en cuenta que sólo se debería implementar si presenta notables ventajas de rendimiento.*

Los Drs. Peter Rob y Carlos Coronel, en el libro: "*Sistemas de Bases de Datos: Diseño, Implementación y Administración"* opinan:

La pureza de la normalización a menudo es difícil de sostener en el ambiente moderno de bases de datos. Los conflictos entre eficiencia en el diseño, requerimientos de información y velocidad de procesamiento, a menudo se resuelven mediantes compromisos que incluyen la desnormalización. En algunas ocasiones, el uso de formas normalizadas de menos nivel ayuda a eliminar un problema estructural potencialmente serio, un ejemplo lo encontramos en los "Almacenes de Datos", donde se puede comprobar que las formas normales de menos nivel ocurren (e incluso se requieren) en estas bases de datos especializadas. Rutinariamente, el "Almacén de Datos" utiliza estructuras en 2FN en su complejo ambiente de datos de fuente y niveles múltiples. En resumen, aunque la normalización es muy importante, sobre todo en el llamado ambiente de bases de datos de producción, examinando hasta ahora, 2FN ya no se desatiende tanto como antes. (Peter Rob, 2003)

No se pretende minimizar el problema de trabajar con tablas que contienen *dependencias transitivas o parciales*, o ambas, en un ambiente de base de datos de producción. Independientemente de la posible creación de anomalías problemáticas de datos, las tablas no normalizadas en un ambiente de bases de datos de producción tiende a mostrar estos efectos (Peter Rob, 2003):

- ✓ Las ***actualizaciones de los datos son menos eficientes*** porque los programas que leen y actualizan de las tablas deben tratarse con tablas más grandes.
- ✓ La ***indexación es mucho más complicada***. Simplemente no es práctico construir todos los índices requerido para todos los atributos que pueden estar localizados en una sola tabla no normalizada.
- ✓ Las tablas no normalizadas no producen estrategias simples para crear visualizaciones.

En correspondencia con todo lo antes expresado por los Drs. Rob y Coronel y el colectivo de especialistas citados anteriormente, el autor de esta investigación resalta ***que la no normalización en el modelo de datos no siempre implica mayor velocidad en el acceso a la información***, como pudieran considerar muchos diseñadores, todo depende del número de juntas o (*join*) que deben procesar las consultas, donde es válido resaltar a consideración del autor que no son muy frecuentes las consultas que en su implementación se excedan de ochos juntas de relaciones, aunque no niega que en determinados modelos pudieran existir. Por lo tanto, se considera siempre necesario el proceso de normalización aunque no se llegue a niveles superiores de este como FNBC.

1.8 HERRAMIENTAS EXISTENTES PARA LA NORMALIZACIÓN DE BASES DE DATOS RELACIONAL

Existen en el mundo herramientas computacionales y no computacionales, para realizar el proceso de la normalización de BDR. Dentro de las computacionales, se encuentran los sistemas que automatizan este proceso. Estas herramientas brindan ventajas para el desarrollo de sistemas informáticos en su etapa de diseño, las cuales referenciamos a continuación:

- ✓ La comprobación automática del nivel de normalización de la estructura lógica de sus datos.
- ✓ Corrección de su diseño.
- ✓ Velocidad y fiabilidad de obtención del proceso.

Por otro lado, resultaría muy difícil poder obtener el máximo de las bondades que pueden brindar, sin tener comunicación directa con los diferentes SGBD.

En el campo de las herramientas no computacionales, se encuentra todo una teoría de lógica matemática que tiene por objetivo fundamental orientar en cómo realizar y aplicar en el diseño de BDR las técnicas de normalización; para que puedan ser utilizados de forma más efectiva para cada una de las BD.

Todas estas herramientas no computacionales que pueden ser utilizadas en el proceso de elaboración de un SI en su etapa de diseño, no garantizan que este proceso se haga de forma rápida, sin carecer de errores dados propiamente por la presencia humana. Como respuesta a ellos es que han surgido las herramientas computacionales que automatizan este proceso.

A continuación se ofrece las características más importantes de las herramientas encontradas.

1.8.1 JMathNorm

Es una herramienta interactiva de escritorio, con una interfaz gráfica descrita en Java. Permite obtener modelos normalizados hasta la FNBC, incluyendo también como parte del sistema la 2FN y 3FN. Arroja además, operaciones que forman parte del proceso de normalización como: el *cálculo del cierre de descriptores, cobertura minimal del conjunto de dependencias funcionales, eliminación de atributos extraños en la relación, eliminación de dependencias funcionales redundantes.* Su interface

desarrollada en Java hace uso de la componente *J/Link*[1], para lograr una integración directa con el *Mathematica*[2] (Karakaya, ICCS 2007). A consideración del autor de la presente investigación, este es uno de los grandes beneficios que brinda la herramienta; pues la hace ser más robusta, confiable y eficiente a la hora de obtener el resultado de los diferentes algoritmos implementados en la aplicación.

Otras características importantes de la herramienta ***JMathNorm,*** que se ilustran en la *Fig.4*, son:

- ✓ Permite adicionar atributos al modelo de manera interactiva por el usuario.
- ✓ Permite crear el conjunto de dependencias funcionales que representan las reglas del negocio, con la notación usada en la mayoría de las bibliografías existentes de BD.
- ✓ Permite mostrar el resultado final de cada uno de los procesos particulares, que no son más que resultados de los diferentes pasos que forman parte de la técnica de normalización, para cualquiera de los niveles que se encuentre analizando.

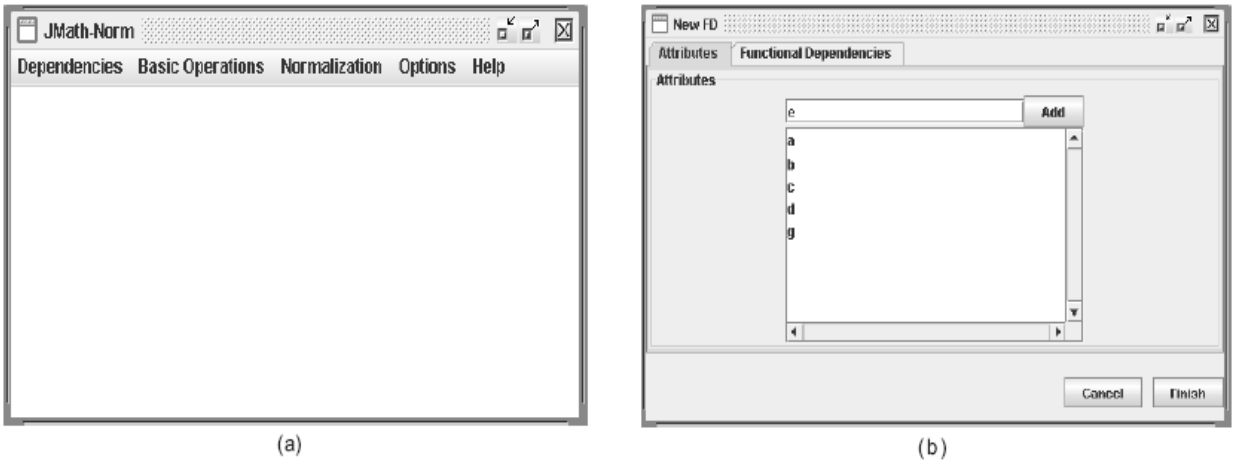


Fig. 4 Interfaces de algunas operaciones básicas del JMathNorm
(a) Menú principal. (b) Definición de atributos.

1 Conjunto de herramientas que integra Mathematica y Java. Permite llamar a Java desde Mathematica de una manera completamente transparente.

2 Programa utilizado en áreas científicas, de ingeniería, matemáticas y áreas computacionales, herramienta poderosa de cómputo que permite realizar cualquier tipo de cálculo algebraico o numérico y generar gráficos y análisis profundos.

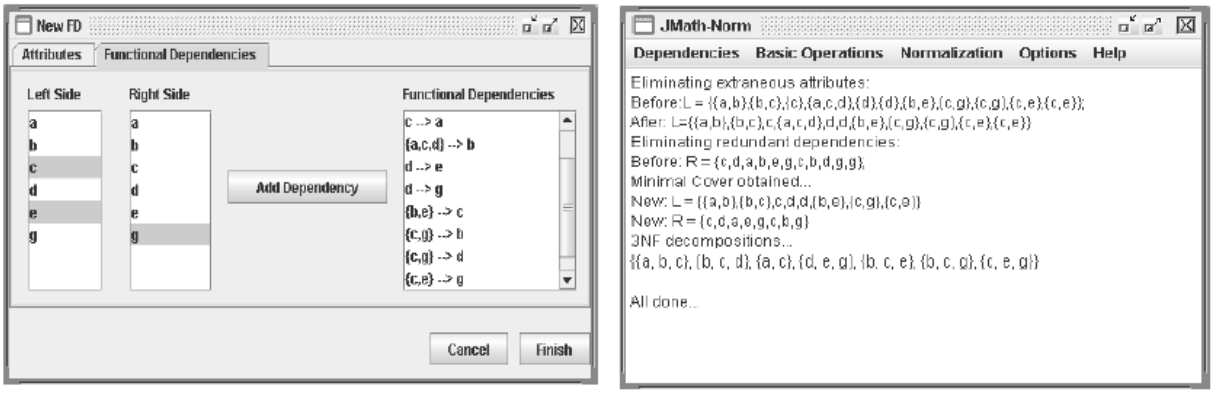


Fig. 4 Interfaces de algunas operaciones básicas del JMathNorm
(c) Creación del conjunto de DF. (d) Proceso de normalización (3FN) (Yazici, 2006)

A pesar de encontrarnos con un gran número de características significativas es importante mencionar que esta herramienta, en la actualidad, no cuenta con algunas funcionalidades que son necesarias para su mejor aprovechamiento, como son: la capacidad de integrarse con los SGBD; no muestra el proceso de obtención de las claves candidatas, ni lo tiene agregado como una prestación del sistema; no permite *cargar* un modelo de BD almacenado, lo que puede convertirse en un ejercicio tedioso para el estudiante por el tiempo que puede demorarse con respecto a la entrada de sus datos; así como tampoco permite *salvar* los modelos para que sean utilizados nuevamente y el alfabeto de la interfaz es en inglés.

Actualmente el sistema no se encuentra dispononible en la web para su explotación. Las imágenes mostradas anteriormente son el producto de una colaboración entre los autores de la herramienta JMathNorm y el autor de la presente investigación.

1.8.2 NORMIT

Sistema tutorial web basado en restricciones para la enseñanza, con una arquitectura centralizada, creado para resolver problemas de normalización de bases de datos, cuyo modelo de dominio es representado por un conjunto de 81 problemas de restricciones independientes. Maneja una Base de Conocimiento Basada en Restricciones (CBM), la cual se manipula a través del lenguaje de programación *Lisp*[1] y como servidor de aplicaciones web utiliza el *AllegroServe* (MITROVIC, 2002).

1 Es el segundo lenguaje de programación más antiguo (después de Fortran) de alto nivel. Lenguaje multi-paradigma, o sea, de tipo orientado a objeto y declarativo.

Sistema capaz de analizar y ofrecerle una retroalimentación a los estudiantes del tema, como objetivo de complementar y ejercitar los conocimientos adquiridos en el aula (MITROVIC, 2005). Para la resolución de los problemas se basa en la estrategia metodológica de la *auto-explicación*[1], exigiéndole por cada acción que realice por primera vez los estudiantes una explicación de las respuestas seleccionadas. Para las acciones posteriores del mismo tipo, se requiere una explicación sólo si se realiza la acción incorrectamente (MITROVIC, 2002).

Para la utilización del sistema el estudiante necesita autentificarse para iniciar sesión, una vez autenticado recibe una breve descripción del sistema y la normalización de BDR en general. El estudiante para seleccionar el problema a desarrollar el ***Normit*** muestra el listado de todos los problemas pre-definidos, de manera que pueda escoger uno que le resulte interesante (MITROVIC, 2005).

La normalización de BDR es una tarea de procedimiento: el estudiante transita por una serie de pasos para analizar la calidad de la base de datos de su problema. ***Normit*** le muestra de forma ordenada por ventanas los pasos que este debe completar para darle la solución completa al problema. A continuación se comentarán brevemente e ilustrarán en la *Fig.5* algunos de estos pasos.

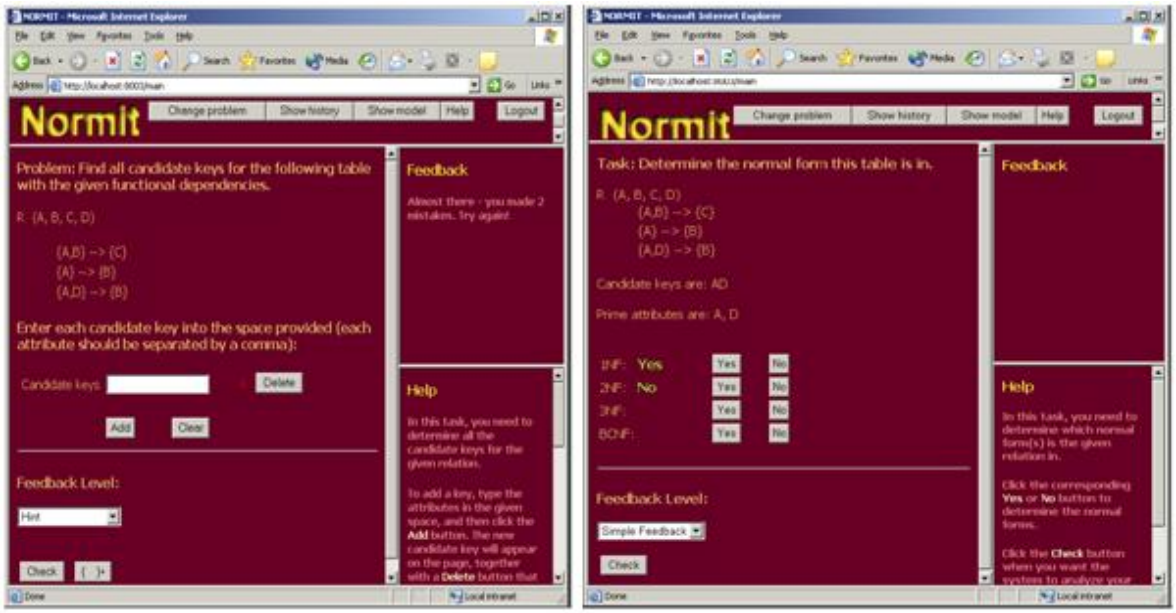
- ✓ *Determinar las claves candidatas:* El estudiante necesita analizar el problema seleccionado para determinar todas las posibles claves.
- ✓ *Determinar el cierre de un conjunto de elementos:* Si el estudiante no está seguro si un conjunto de atributos forma una clave candidata o no, puede calcular el cierre de ese conjunto en función del conjunto de dependencias funcionales dadas. *Fig. 5 (a)* última sección.
- ✓ *Determinar atributos principales:* La interfaz de esta tarea se muestra una vez que el estudiante determina con éxito todas las claves candidatas para el problema en cuestión.
- ✓ *Simplificar las dependencias funcionales*.
- ✓ *Determinar la forma normal del esquema en cuestión:* El estudiante tiene que especificar si la relación se encuentra 1FN, 2FN, 3FN, o en FNBC, como se muestra en la *Fig. 5 (b)*. Si el estudiante piensa que la relación no cumple una forma normal, el sistema requiere que el estudiante especifique las dependencias funcionales que violan esa forma normal.

---

1 Es un tipo de explicación desarrollada para uno mismo, con el fin de aclararse una circunstancia, proceso que incluye tanto recordar principios ya conocidos como sustituir principios antiguos por otros nuevos.

- ✓ *Si es necesario, descomponerlo hasta FNBC.*

Se evidencia en la descripción anterior, las considerables ventajas que brinda esta herramienta en el entorno educacional para la *ejercitación* de todo el proceso de normalización de BDR. Siendo las más significativas, según el criterio del autor de la presente investigación: la posibilidad de explicar las respuestas, ayudando al estudiante tanto a corregir sus ideas erróneas como a fomentar y profundizar la comprensión de la teoría mediante el desarrollo de sus propias explicaciones, a través de la estrategia "*auto-explicación*"; la posibilidad de brindarle pista de la solución y de proporcionarle a cada usuario su nivel de dominio en las diferentes materias que engloba esta técnica.



(a) (b)

Fig. 5 Interfaces de algunas funcionalidades del Normit
(a) Determinación de las claves candidatas. (b) Determinación de los niveles de normalización
(MITROVIC, 2002)

A pesar de estos elementos significativos, presenta algunos inconvenientes tales como: el sistema no proporciona ayuda sobre los conceptos básicos del dominio abarcado, o sea, sólo es provechoso para estudiantes que posean conocimientos del la normalización; no admite introducir nuevos problemas de estudio por los estudiantes, sólo le permite la ejercitación a través de los ya existentes en la base de conocimiento; el alfabeto de la interfaz es en inglés; no tiene la capacidad de integrarse con los SGBD.

1.8.3 WEB-BASED TOOL

Sistema web desarrollada con *applet de Java*[1] para mejorar la enseñanza y el aprendizaje del proceso de normalización de bases de datos; fácil de utilizar para los estudiantes, permite obtener modelos normalizados hasta la 3FN a partir de un conjunto de dependencias funcionales, asociadas a una relación, introducidas por el usuario sin exceder las 10 unidades.

Algunas de las prestaciones que brinda esta herramienta:

- ✓ Permite *adicionar* nuevas dependencias funcionales. Se puede observar en la *Fig. 6* que aparece seguidamente.
- ✓ Permite *normalizar* el modelo de datos compuesto por el conjunto de dependencias funcionales introducida por los estudiantes a 3FN, mostrándolo en una interfaz independiente
- ✓ Permite *mostrar paso a paso el resultado de la normalización* transitando por los diferentes niveles 1FN, 2FN hasta la 3FN. Se puede observar más detalladamente en la *Fig. 7* que aparece a continuación.
- ✓ Se encuentra en desarrollo la funcionalidad de *imprimir*. (A Web-Based Tool, 2006)

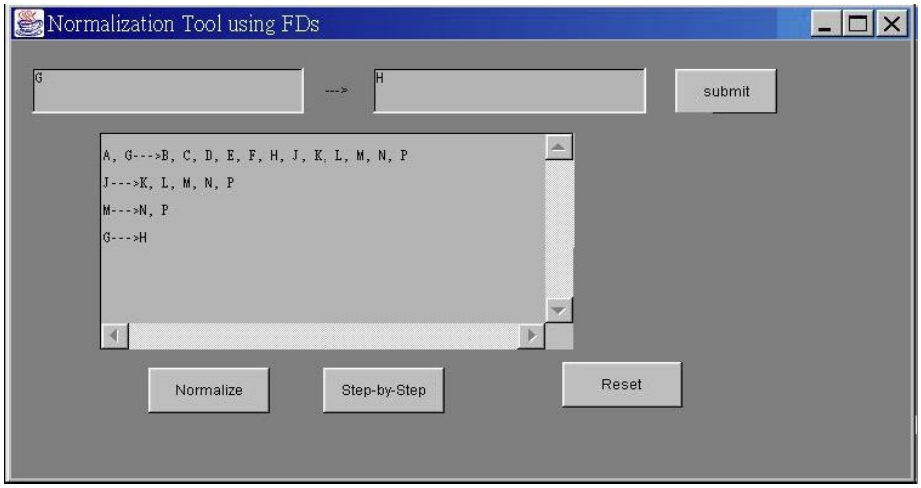


Fig. 6 Interfaz principal del sistema Web-Based Tool (A Web-Based Tool, 2006)

1 Es un código JAVA que carece de un método principal, se utiliza principalmente para el trabajo de páginas web, es un pequeño programa que es utilizado en una página HTML y representado por una pequeña pantalla gráfica dentro de esta.

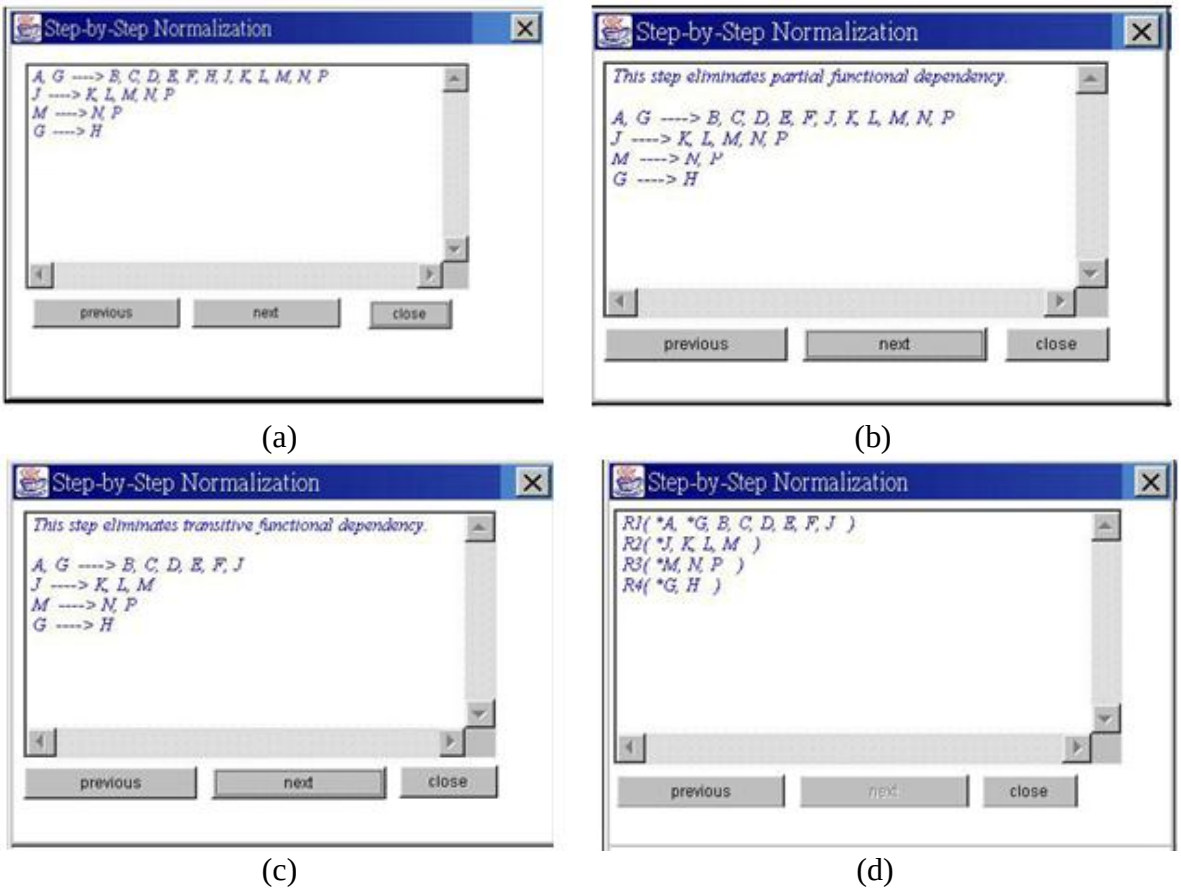


Fig. 7 Interfaces que representan las etapas el proceso de normalización (a) Conjunto de DF iniciales. (b) 2FN. Eliminación de DF parciales (c) 3FN. Eliminación de DF transitivas. (d) Modelo normalizado (3FN). (A Web-Based Tool, 2006)

El autor de la presente investigación considera que Web-Based Tool, es el software que menos prestaciones brinda (de los analizados) para facilitar el proceso de enseñanza de la normalización de BDR para los estudiantes. Como aspecto positivo vale la pena resaltar su interfaz de fácil entendimiento y la opinión favorable que presentan la mayoría de los estudiantes encuestados según Hsiang-Jui Kung en su artículo: *"A Web-Based Tool to enhance teaching/learning database normalization"*. Como elemento negativo podemos citar los siguietes: no refleja el procedimiento de la obtención de los resultados; no incluye dentro de sus funcionalidades la obtención de las claves candidatas, cierre de un descriptor y el cálculo de la cobertura minimal del modelo; normaliza sólo hasta la 3FN; restringe el modelo a diez unidades, como máxima cantidad de dependencias funcionales que se pueden introducir; no permite *cargar* un modelo de BD almacenado, lo que puede convertirse en un ejercicio tedioso para el estudiante por el tiempo que puede demorarse con respecto a la entrada

de sus datos; así como también, no permite *salvar* los modelos para que sean utilizados nuevamente, estas últimas características pueden influir de manera negativa en la utilización de la herramienta por los estudiantes; no está creada para identificar el nivel de normalización que poseen los modelos de datos introducidos por el usuario; restringe a crear modelo de datos de una sola relación; además no incluye la posibilidad de integrarse con los SGBD.

1.8.4 DBDesignTools

Es una librería en construcción (versión 1.0) escrita en el lenguaje de programación C++, con el fin de resolver algunos de los problemas de la estructura lógica del modelo de datos que se muestra en cualquier diseño de BDR. Esta librería implementa algunos algoritmos relacionado con la normalización de BDR como son : el cálculo del cierre de un conjunto de dependencias funcionales, la proyección de un conjunto de atributos con respecto a un conjunto de dependencias funcionales, equivalencia entre conjuntos de dependencias funcionales, cierre de un conjunto de atributos, cálculo de claves candidatas, cobertura minimal de un conjunto de dependencias funcionales y la comprobación de los diferentes niveles de normalización hasta FNBC (Martín, 2006).

El autor de la presente investigación considera, que a pesar de constar con una herramienta que ponga en manos de nuestra comunidad la implementación de un gran número de algoritmos importantísimos, que son usados casi en su totalidad como complemento para darle solución a los diferentes problemas del diseño lógico de las BDR, aún no permite transformar el modelo actual en modelos normalizados que cumplan con los diferentes niveles de normalización, dígase 2FN, 3FN y FNBC. Además, si se analiza profundamente la implementación de cada una de estas funcionalidades, se percibe que algunos de los algoritmos implementados en *DBDesignTools* no son los más eficientes, atendiendo a su complejidad temporal; aspecto que el autor de esta herramienta reconoce y pone en manos de la comunidad de desarrolladores para su optimización.

1.8.5 LDBN - Learning Database Normalization

Sistema web creado por Nikolay Georgiev en el 2008, con el fin de brindarle al estudiante una forma fácil y suficiente de comprobar sus conocimientos de la teoría de normalización de bases de datos relacionales, desarrollado por la parte del cliente en *JavaScript* usando la tecnología *Ajax* para lograr un interfaz del sistema más interactiva al usuario, y por el servidor con PHP y MySQL como SGBD, con una arquitectura descentralizada, donde del lado del cliente implementa todas las funcionalidades

del sistema y del lado del servidor sólo se utiliza para almacenar los datos creados y la gestión de los usuarios, reduciéndole carga al servidor, garantizándole que pueda manejar más usuarios a la vez. LDBN permite al estudiante a partir de un esquema relacional-universal (URF)[1], ya sea creado por él o cargado de la aplicación, realizar todo el proceso de normalización del esquema hasta FNBC, incluyendo 2FN y 3FN usando la teoría de la descomposición, regido por un conjunto de pasos que les son brindados como guía para la obtención de la solución. Así como también, brinda la posibilidad de obtener la solución correcta del URF en caso que sea solicitado por el usuario. (Georgiev, September 2008).

Las prestaciones principales que brinda esta herramienta:

- ✓ Permite *resolver* un *URF:* Es utilizado para generar una solución del URF y presentarla al usuario, en la *Fig. 8* se muestra un ejemplo de esta funcionalidad en el sistema. Para la obtención de la solución el estudiante debe seguir los siguientes pasos que se describen a continuación:
  - Determinar la cobertura minimal del conjunto de DF del esquema.
  - Descomponer el esquema relacional dado en 2FN, 3FN y FNBC.
  - Determinar una clave principal para cada nueva relación obtenida.
- ✓ Permite cargar URF: Presenta un listado de todos los URF almacenados en la BD para que el usuario pueda seleccionar.
- ✓ Permi*te crear* URF: Brinda interfaz para adicionar nuevas URF al sistema.
- ✓ Posibilita *administrar* usuarios: Para usuarios no registrados permite crear nuevas cuentas y a los usuarios registrados ingresar al sistema con su nombre de usuario y contraseña.
- ✓ Permite publicar comentarios: Les brinda a los usuarios registrados la posibilidad de comentar un trabajo, con el objetivo de intercambiar conocimientos o dudas con los demás usuarios.

LDBN es, de los software analizados anteriormente con enfoque docente, el que mejor interfaz de usuario y prestaciones brinda para representar todo el proceso de normalización de BDR, esta característica lo puede convertir en uno de los favoritos en esta esfera; como aspectos positivos a resaltar, desde el punto de vista docente, tenemos: la posibilidad de mostrar la evaluación con exactitud de los pasos que componen todo el proceso de normalización, les permite a los usuarios comunicarse entre sí, a través de comentarios o mensajes para consultar dudas o inquietudes que presenten al

1 Una relación con todos los atributos y el conjunto de dependencias funcionales de la DB.

enfrentarse a problemáticas diferentes, permite cargar un URF lo que asegura un uso muy fácil y rápido de la herramienta sin necesidad de realizar la entrada completamente por teclado.

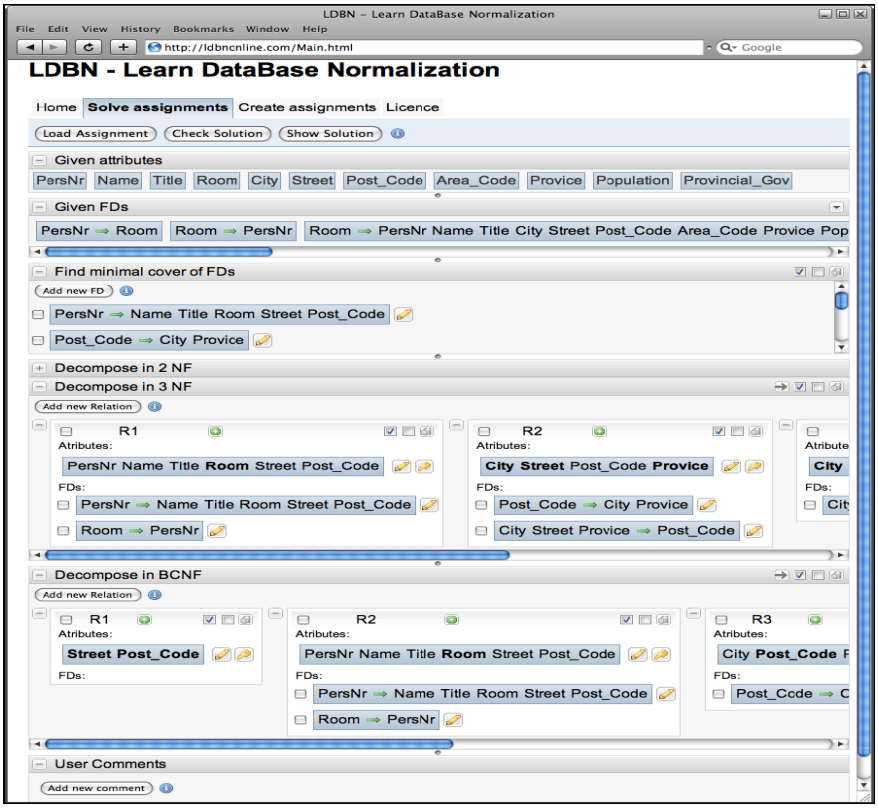


Fig. 8 Generando solución del URF con el LDBN. (Georgiev, September 2008)

A pesar de todas sus bondades, presenta al igual que el resto algunos inconvenientes, los más significativos están relacionados con la no representación del proceso de obtención de las claves candidatas, así como el cálculo de cierre de descriptores, los cuales constituyen dos pasos de vital importancia para comenzar a aplicar el proceso de normalización, y que le resultaría de mucho apoyo al estudiante para alcanzar la solución correcta, además no cuenta con la capacidad de integrarse con los SGBD, y el alfabeto del lenguaje se encuentra en inglés.

Existe otras herramientas para la normalización de bases de datos relacionales, como son: RENO, DBNormalization, MICRO y NOCAT, todas estas con similares características a las mencionadas anteriormente, con el objetivo de resaltar sus bondades y deficiencias más puntuales teniendo en cuenta: *su disponibilidad para su uso, funcionalidades básicas como: cálculo de cierre, recubrimiento*

o cobertura minimal, conjunto de claves, normalización hasta la FNBC, representación de los pasos dados para la obtención de la solución y un breve comentario de su tipo, mostraremos la *Tabla 1.1* que representa el estudio realizado en el ***"XV JENUI. Barcelona, julio de 2009"*** sobre todas estas herramientas.

Herramienta	Disponible Encontrada	Cierre, cobertura mínima, claves	2FN 3FN FNBC	Muestran proceso	Comentarios
RENO [16]	Sí	Sí	Sólo FNBC	No	Acompaña a libro
NORMIT [14]	Sí	Sí	Sí	Sí	Sin nuevos ejercicios
DBNormalization [21]	Sí	Sí	Sí	Sí	Guía el proceso
lbdn [10]	Sí	No cierre ni claves	Sí	Sí	Proyecto Google code
"Web-based" tool [12]	Sí	No	Hasta 3FN	No	Applet de Java
Micro [7]	No	Si	Si	-	Prototipo
NOCAT [17]	No	Sí	Sí	-	Prototipo
DBDesignTools [13]	No	Sí	Sí	-	Librería en C++
JMathNorm [23]	No	No claves	Si	No	En Mathematica

Tabla 1.1 Herramientas para la normalización de base de datos. (César Domínguez Pérez, et al. 8-10 julio 2009)

La mayoría de las herramientas, calculan los resultados automáticamente, o sea, su utilidad docente para lo cual están diseñadas podemos compararla a una calculadora, que muestra el resultado pero no el proceso ni los pasos intermedios para conseguirlo. (César Domínguez Pérez,et al. 8-10 julio 2009). Además podemos resaltar que casi todas están pensadas sólo para fines docentes y no como instrumento necesario para desarrollar sistemas informáticos que presenten una alta calidad en sus modelos de datos, aspecto de vital importancia en el marco productivo de la UCI, para el desarrollo de nuevas soluciones informáticas con fines comerciales.

Después del análisis realizado de las diferentes herramientas mencionadas, concluimos que en ellas persisten un conjunto de inconvenientes y limitantes que dificultan su utilización en el proceso productivo, tampoco brindan la posibilidad de visualizar los algoritmos utilizados en el proceso de normalización.

1.9 HERRAMIENTAS EXISTENTES PARA MODELAR BASES DE DATOS RELACIONAL

Existen un gran número de herramientas CASE (Computer Aided Software Engineering) para modelar BDR que comienzan desde el modelo E-R o modelo conceptual mediante la creación de ERD y permiten la integración directa con los diferentes SGBD vigentes actualmente en el mercado

mundial, entre estas herramientas podemos mencionar: *DBDesigner, SQLDesigner, Embarcadero.ER/Studio, Erwin, EasyCASE, CaseStudio, Data Arquitect, ERCreator, XCase, SmartDraw, Microsoft Visio 2003, ERECase.*

En este epígrafe el autor de la investigación ofrecerá la descripción de las herramientas CASE más populares en el diseño de BDR, enfatizando en las herramientas libres, sin dejar de mencionar algunas de carácter propietario muy reconocidas a nivel mundial por las variedad de prestaciones que brindan.

1.9.1 DBDesigner

Es un programa de modelado de bases de datos relacionales desarrollado por *FabFORCE.net,* gratuito con *licencia GPL,* actualmente se encuentra en la cuarta versión de su desarrollo. Entre todas sus características, podemos destacar (Blog de Web a Medida. Potenciado por Joomla, 2008):

- ✓ Modelados realizados en XML.
- ✓ Interfaz de usuario amigable.
- ✓ Permite hacer ingeniería inversa desde bases de datos como *MySql, Oracle, Microsoft SQL Server* y cualquier base de datos ODBC.
- ✓ Modo de diseño y de consulta. Permite realizar por una parte el modelado de la base de datos y por otra realizar consultas sobre las tablas y construir consultas en lenguaje SQL para *php, Kylix* y otros lenguajes de programación.
- ✓ Permite generar ficheros *.sql* con las sentencias necesarias para crear el modelo plasmado en la parte gráfica.
- ✓ Sincronización automática con bases de datos *MySql*. Esta función es una de las más interesantes porque permite actualizar las tablas y relaciones existentes en una base de datos *MySql* que haya instalada en un ordenador local o remoto, sin necesidad de ejecutar ningún scripts.
- ✓ Colocación automática de las foreign key.
- ✓ Modelo avanzado de impresión.

En el 75% de la bibliografía analizada durante la búsqueda de información, los usuarios o diseñadores lo catalogan como un software potente y fácil de utilizar para diseñar modelos de bases de datos para el SGBD *MySQL*.

1.9.2 Embarcadero ER / Studio

Herramienta líder reconocida en el mercado mundial para el modelado de datos, desarrollada por la empresa Embarcadero Technologies, *software propietario* para el modelado de datos, fácil de usar y multinivel para el diseño y construcción de bases de datos a nivel físico y lógico, soporta un gran número de SGBD como: *Oracle, Microsoft SQL Server, MySQL, PostgreSQL, Microsoft Access*, entre otros. Actualmente se encuentra en la octava versión de su desarrollo, a continuación se muestra en la *Tabla 1.2* sus principales características a destacar (Embarcadero Technologies, 2009).

Características de producto	Descripción
Ambiente de diseño y modelado altamente productivo	
Gráficos y diseño avanzados	Automáticamente crea diagramas altamente navegables y legibles
Capacidades de diseño multinivel	Permite muchos diseños físicos desde una arquitectura central lógica.
Transformaciones automatizadas y personalizadas	Facilita la derivación de un diseño físico desde uno lógico y checa la compatibilidad con una plataforma de base de datos objetivo.
Múltiples formatos de presentación	Publica modelos y reportes en una variedad de formatos incluyendo HTML, RTF, esquemas XML.
Soporte completo al ciclo de vida de la base de datos	
Ingeniería hacia adelante	Genera código fuente para algunos diseños de base de datos.
Ingeniería inversa	Construye modelos gráficos desde una base de datos o esquema existente.
Gestión de modelado empresarial	
Integración de metadatos	Importa y exporta metadatos desde una variedad de fuentes incluyendo plataformas de BI, UML y soluciones de modelado de datos y esquemas XML.
Generación de Esquemas XML	Asegura que los proyectos XML tales como aquellos utilizando Arquitectura Orientada a Servicio (SOA) están basados en los mismos estándares y metadatos que los modelos al modelarlos en ER/Studio y generar XSD
Diseño de bases de datos de calidad	
Validación del modelo completo	Automatiza las revisiones de modelos, y refuerza estándares con más de 50 chequeos para validar modelos lógicos y físicos para definiciones de objetos faltantes, dominios no utilizados, índices únicos e idénticos y relaciones circulares.
Migración automática de llaves foráneas	Mantiene claves externas para asegurar la integridad referencial en los diseños
Diseño y seguridad	
Clasificación de datos	Categoriza y etiqueta datos y objetos de acuerdo al nivel de seguridad y privacidad que deben ser aplicados a la información
Gestión de permisos	Permite modelado de usuarios, roles y permisos en niveles lógico y físico

Tabla 1.2 Algunas características del producto "Embarcadero ER / Studio"
(Embarcadero Technologies, 2009)

1.9.3 ERwin

Herramienta para el diseño de base de datos relacionales. Brinda productividad en diseño, generación, y mantenimiento de aplicaciones desde el modelo lógico de los requerimientos de la información hasta el modelo físico. Herramienta propietaria que se encuentra en la séptima versión de su desarrollo, como principales características podemos destacar:

- ✓ Genera automáticamente las tablas, *stored procedure* y *triggers* para los principales tipos de base de datos.
- ✓ Hace fácil el diseño de una base de datos.
- ✓ Genera automáticamente vistas, índices, reglas de integridad referencial (llaves primarias, llaves foráneas), valores por defecto y restricciones de campos y dominios.
- ✓ Soporta un conjunto de bases de datos compatibles como: *Oracle, Microsoft SQL Server, Sybase, DB2, Informix, Microsoft Access, Paradox, SQLBase, Sybase, etc.*
- ✓ El mismo modelo puede ser usado para generar múltiples bases de datos, o convertir una aplicación de una plataforma de base de datos a otra.
- ✓ Permite hacer ingeniería inversa desde las bases de datos compatibles.

Su limitante fundamental es que no es multiplataforma, o sea, se limita sólo a un sistema operativo: *"Windows"*.

1.9.4 SQLDesigner

Es una herramienta web, de código abierto *(Open Source)* para realizar el modelo de bases de datos relacionales de forma rápida y sencilla. La herramienta es capaz de (MANCOMUN. Centro de Referencia a Servicios de Software Libre, 2009):

- ✓ Realizar diseños pueden ser guardados, exportados e importados desde XML.
- ✓ Permitir la generación automática de scripts SQL para la creación de base de datos en SGBD como: *MySQL, SQLite, Microsoft SQL Server, Oracle, PostgreSQL*.
- ✓ Importar esquema procedente de una base de datos existente que sean compatibles con los SGBD antes mencionados.

A consideración del autor de la investigación, de las herramientas analizadas es una de las que menores prestaciones les brinda a los usuarios.

Dado un análisis profundo de las herramientas citadas anteriormente realizado en (Carlos García González, 2007) donde evidencian un conjunto de desventajas presentes en estas, llevó a estos autores cubanos al desarrollo de la herramienta ERECase con el objetivo de brindar una herramienta que incluyera aspectos novedosos del modelo conceptual y diera solución a las principales deficiencias encontradas en las herramientas CASE.

1.9.5 ERECase

Herramienta para el diseño y validación estructurar de esquemas conceptuales, realizada en la Universidad Central "Marta Abreu" de las Villas, Cuba. Esta herramienta aún no ha salido al mercado, está siendo probada y utilizada actualmente en la docencia de pregrado en las carreras de Ciencias de la Computación e Ingeniería Informática y en la maestría de Ciencia de la Computación en la provincia Villa Clara.

La herramienta CASE es capaz de (Carlos García González, 2007):

- ✓ Manipular las funciones generales del diseño como: inserción y eliminación de diagramas, mover, redimensionar, copiar, cortar y pegar.
- ✓ Permitir una mayor expresividad en el modelado conceptual. Entre estas construcciones se incluyen las entidades débiles y fuertes, interrelaciones de asociación recursiva, binaria y ternaria, jerarquías de generalización/especialización, agregación, dependencia de existencia en asociaciones binarias.
- ✓ Validación estructural del esquema conceptual. Realizando chequeos como:
    - Exclusividad de nombres.
    - Chequeos de identificación.
    - No se permiten ciclos entre conjuntos de entidades débiles.
    - Chequeo de validez estructural de ciclos entre conjuntos de entidades que contengan interrelaciones recursivas, binarias y ternarias.
- ✓ Generación del esquema lógico, detección y corrección de inconsistencia en este.
- ✓ Generación del esquema físico para Microsoft SQL Server.

Es importante resaltar que el módulo de "Detección y Corrección de Inconsistencias" en el esquema lógico, detecta sólo un tipo de inconsistencia que puede manifestarse cuando entre uno o varios esquemas de relación las restricciones de integridad referencial forman ciclos, sin embargo, el autor de

la investigación considera que constituye la herramienta CASE más expresiva y rigurosa examinada en este epígrafe, orientada al diseño de BDR.

Al terminar con el estudio y análisis de todas las herramientas concebidas para el diseño de BDR, donde fueron descritas sus principales características y desventajas presentadas en ellas, se puede afirmar que a pesar de constar nuestra comunidad con un gran número importante de herramientas CASE, ricas en funcionalidades utilices que ayudan a modelar e integrar con los diferentes SGBD existentes, aún ninguna está concebida para detectar redundancias en los modelos de datos e anomalías de inserción, eliminación y actualización que pudieran estar presentes en el diseño del modelo de datos, atendiendo a su estructura lógica, así como, corregir o normalizar modelos de datos hasta FNBC.
Esto apunta a la necesidad de desarrollar un Sistema Informático para la integración del proceso de normalización de BDR con los SGBD más usados.

1.10 CONCLUSIONES DEL CAPÍTULO

Del estudio realizado sobre los SGBD, conceptos, características, y el análisis minucioso de las diferentes herramientas que existen actualmente en el mercado, con el propósito de modelar o diseñar modelos de bases de datos relacionales; así como también automatizar el proceso de normalización de BDR, concluimos:

- ✓ El estudio y la utilización de la técnica de normalización en la etapa de diseño de los SGBD relacionales es de vital importancia para el desarrollo de un adecuado modelo lógico de la información, minimizando la presencia de anomalías de almacenamiento y garantizando una elevada calidad, robustez y extensibilidad de los modelos de datos realizados en esta etapa para el desarrollo de los diferentes SGI.
- ✓ Aproximadamente el 90% de las herramientas encontradas que automatizan el proceso de normalización de BDR, presentan un enfoque académico pensadas para ser usada en la docencia y no en el proceso productivo.
- ✓ Las herramientas CASE analizadas en este capítulo, al igual que los SGBD, no atacan el proceso de normalización de BDR
- ✓ El 100% de las herramientas analizadas en el presente capítulo, que automatizan el proceso de normalización de BDR, no integran su resultado con los SGBD relacionales.

Capítulo 2. Justificación de la propuesta. Análisis y Diseño del Sistema

En el presente capítulo se exponen las principales características de la herramienta, basándose fundamentalmente en la concepción de la propuesta de solución ya planteada.
Se realiza una comparación con las soluciones homólogas ya documentadas en el capítulo uno, basándose en algunos parámetros seleccionados por el autor
Se describen los pasos a seguir en el proceso de normalización de BDR, junto con una descripción detallada de las funcionalidades del sistema. Además, se presentan los resultados obtenidos de la utilización de la Metodología de Desarrollo de Software RUP, que guió el proceso de construcción de esta herramienta.

2.1 FUNDAMENTACIÓN DE LA PROPUESTA DEL SISTEMA "SINORGES"

El análisis realizado a los software desarrollados para la normalización de BDR y para el diseño de BDR, documentado en el capítulo uno de la presente investigación, arrojó que en ellos están presentes una o varias limitaciones que dificultan su utilización generalizada, tanto en el entorno educacional como en el ámbito productivo, para elevar la calidad de los modelos de datos.
La utilización en el ámbito productivo constituye el campo de acción a abordar con la propuesta de solución que se fundamentará en el presente capítulo.
Las limitaciones a las que se hacen referencia en las aplicaciones desarrolladas para la normalización de BDR van desde aspectos menos significativos como diseño de la interfaz, idioma que utilizan, hasta aspectos más importantes como: no implementación de funcionalidades básicas del proceso y la visualización directa de los resultados, con comportamiento similar a una calculadora, no integración con los SGBD. La limitante fundamental que se aborda en las aplicaciones desarrolladas para el diseño de BDR es la no validación y corrección de la estructura lógica del modelo de datos, a través del proceso de normalización.
Estudios realizado por el autor, fundamentalmente hacia la mayoría de las universidades de nuestro país y en especial a lo interno de nuestra Universidad, permiten expresar que no se conocen, ni se utilizan herramientas que garantizan el proceso de normalización.

Si nos centramos en la misión de la UCI, citada en (Proyecto Estratégico (2008-2012)), constar con un producto propio desarrollado en nuestra comunidad que permita a través de su utilización la integración, de forma automática, de todo el proceso de normalización con los diferentes SGBD, sería de valioso aprovechamiento y utilidad para toda la familia de desarrolladores en el ámbito productivo.
Al presente, la UCI no cuenta con ningún sistema para la normalización de modelos de datos como los descritos en el capítulo uno, ni en el entorno docente como la mayoría de los sistemas mencionados, con el objetivo de elevar el aprendizaje de nuestros estudiantes sobre estos temas en la asignatura de "Sistemas de Bases de Datos", ni en el marco productivo con el fin de obtener modelos de bases de datos con un menor porciento de redundancias y anomalías en su información, durante el desarrollo de nuevas soluciones informáticas. Este proceso se desarrolla de forma manual por las personas que transitan por el rol de diseñadores de bases de datos en los diferentes proyectos productivos, y en muchos casos sin dedicarle el tiempo suficiente, desempeñándolo de manera intuitiva, o sea, sin un análisis previo y profundo de todas las restricciones del negocio (*dependencias funcionales*), lo cual conlleva a resultados incorrectos con respecto a la información que gestiona el sistema.

La detección de una inconsistencia en la estructura lógica de los datos o no flexibilidad del diseño para responder a las necesidades de los clientes, provocaría realizar una reestructuración en el modelo de BDR, que pudiera afectar parcial o totalmente la implementación de las diferentes funcionalidades del sistema y la estructuración lógica del modelo de datos almacenado, válido sólo en el caso de no detectarlo en la etapa de diseño, lo cual tiende a ser muy común, por no constar en nuestra comunidad con herramientas que comprueben y den solución a estas anomalías presentadas en esta etapa. Lo que influye de manera negativa en el tiempo de desarrollo del proyecto y hasta podría ocasionar, en algunos casos, resultados incorrectos en la obtención de la información gestionada por el usuario.

Por tales argumentos surge la necesidad de desarrollar un Sistema Informático propio y específico para automatizar el proceso de normalización de BDR, que sirva de apoyo para el diseño de BDR en el desarrollo de soluciones informáticas; así como también, elevar el aprendizaje de nuestros estudiantes, integrándose de manera automática con los diferentes SGBD.

Entre los requerimientos, a consideración del autor, que debería cumplir un sistema informático para lograr los objetivos propuesto de la investigación podemos enumerar:

- ✓ Debe tener una interfaz amigable y brindar la mayor flexibilidad posible para modificar el entorno de trabajo.
- ✓ El mecanismo que se emplee para la entrada de los datos del modelo debe ser el más fácil posible.
- ✓ Debe brindar como funcionalidades principales todos los algoritmos relacionados con la teoría del proceso de normalización de BDR, dígase: cálculo de la clave, cierre de descriptores, cierre del conjunto de dependencias funcionales, cobertura minimal, equivalencia entre dos conjuntos de dependencias funcionales, verificación del nivel de normalización y normalización de la BDR.
- ✓ Debe proporcionar alguna técnica de visualización para la mayoría de los algoritmos utilizados que complementan toda la teoría de normalización. Permitiéndole a los estudiantes o usuarios entender de manera más rápido el proceso.
- ✓ ***El modelo resultante debe poder integrarse con los SGBD. Lo que permitirá una mayor interoperabilidad de los modelos que se elaboren con la herramienta.***
- ✓ El modelo resultante debe poder *salvarse* en el computador, permitiendo de esta manera su recuperación u utilización más adelante.
- ✓ Debe brindar la posibilidad de *cargar* modelos de datos creados en los SGBD o herramientas de diseño de BDR. Lo que les facilitará en gran medida al usuario la entrada del modelo de datos a normalizar o validar y será otro mecanismo que favorecería la integración del sistema con otras aplicaciones.
- ✓ El idioma que se utilice debe estar en función de los usuarios finales.

Estos requerimientos dan solución a algunas de las limitaciones identificadas en las herramientas analizadas, pero no engloba todas las deficiencias citadas en el ámbito educativo de las herramientas que tratan el proceso de normalización de BDR, pues el objetivo fundamental de la propuesta no se centra en el marco educativo, hacia el aprendizaje de los estudiantes, sino se encuentra más orientado a la integración del proceso de normalización de BDR con algunos de los SGBD actuales y utilizados por nuestra universidad, para el desarrollo de aplicaciones informáticas.

Al analizar el comportamiento de la mayoría de las herramientas existente en el mercado mundial para automatizar el proceso de normalización, descrito en el capítulo anterior, con respecto a la propuesta, se percibió que la generalidad de estas tratan de guiar al usuario (estudiante), mediante una secuencia de pasos lógicos que van describiéndole el flujo de los procesos que intervienen en el negocio, para

lograr alcanzar los diferentes niveles de normalización hasta la FNBC, aspecto importante que el autor de esta investigación no deja de reconocer, pues le permite a los estudiantes la ejercitación de dicha teoría, y es significativo para elevar sus conocimientos. Sin embargo, es válido realizar este análisis desde otro punto de vista, inducido fundamentalmente por el entorno productivo en el que nos desarrollamos en la Universidad, y es que sólo se centran en el adiestramiento de este conocimiento y no explotan la fortaleza que representaría la teoría de la normalización integrada con los diferentes SGBD para el desarrollo de sistemas informáticos.

En la *Tabla 2.1*, se puede apreciar con mayor claridad los sistemas desarrollados para automatizar el proceso de normalización de BDR analizados con anterioridad, cómo se comportan con relación a la propuesta "SINORGES", evidenciando como ninguna de las herramientas existentes permite su integración con los SGBD.

Software/Criterios de Comparación	**Web-Based Tool**	**Normit**	**DBDesign Tool**	**LDBN**	**JMathNorm**	**SINORGES**
Interfaz amigable	No	Sí	-	Sí	No	Sí
Algoritmos Disponibles	No claves. No cierre de descriptores. No cierre de DF. No cobertura minimal. No equivalencia entre DF. Sólo normaliza hasta 3FN	No equivalencia entre dos conjuntos de DF	No normaliza, sólo comprueba los niveles de normalización	No claves. No cierre de descriptores. No equivalencia entre DF	No claves. No cierre de DF. No equivalencia entre dos conjuntos de DF	Todos
Representación del proceso/visualización de algoritmos	No	Sí	-	Sí	No	A través de un seudo-código
Exportar a SGBD	**No**	**No**	**-**	**No**	**No**	**Sí**
Salvar Modelo Relacional	No	No	-	Sí	No	Sí
Cargar Modelo Relacional (realizado desde la propia herramienta)	No	No	-	Sí	No	Sí
Importar Script	**No**	**No**	**-**	**No**	**No**	**Sí**
Idioma en la interfaz	Inglés	Inglés	-	Inglés	Inglés	Español

Tabla 2.1 Comportamiento de las herramientas existentes con relación a la propuesta.

2.2 PROCESO DE INTEGRACIÓN CON LOS SGBD RELACIONALES

La variante más utilizada que implementan la mayoría de las herramientas desarrollada para modelar bases de datos relacionales e integrarse con los diferentes sistemas gestores de bases de datos es la creación de "*scripts de base de datos*" en el lenguaje *SQL*, puesto que se puede asegurar que todas las sentencia escrita en *SQL* será interpretable por cualquier motor de datos o SGBD.

¿Qué son los Scripts de Bases de Datos?

Los scripts de base de datos son archivos adicionales que contienen instrucciones del lenguaje *SQL* utilizadas para crear una base de datos y sus objetos. Se pueden generar scripts a partir de los objetos de una base de datos existente y agregar dichos objetos a otra base de datos mediante la ejecución de los scripts en esa base de datos. De ese modo, se vuelve a crear la estructura completa de la base de datos y todos sus objetos.

El esquema para los objetos generados puede guardarse en un único archivo de scripts SQL o en varios archivos que contengan los esquemas de cada objeto. También se puede guardar el esquema generado para un solo objeto, o un grupo de objetos, en uno o varios archivos de scripts SQL. Entre otros ejemplos de archivos de scripts SQL que se pueden generar, podemos citar los siguientes:

- ✓ Una base de datos completa guardada en un solo archivo de scripts SQL.
- ✓ Un esquema de una, varias o todas las tablas de una base de datos guardado en uno o más archivos de scripts SQL.
- ✓ Un esquema de tablas e índices guardado en un archivo de scripts SQL, procedimientos almacenados guardados en otro archivo de scripts SQL, y reglas y valores predeterminados guardados en otro archivo de scripts SQL.

La propuesta de sistema "SINORGES" que se plantea en esta investigación está diseñada para garantizar la creación de scripts para los SGBD como: *PostgreSQL, Microsoft SQL Server y MySQL*. Además brinda la posibilidad de *cargar* modelos físicos de datos creados en los SGBD mencionados anteriormente y en la herramienta de diseño de BDR "Embarcadero ER / Studio", la cual además de constituir otra variante que favorece a la integración con los SGBD y el ER / Studio, les facilitará en gran medida al usuario la entrada del modelo de datos a normalizar o validar.

2.3 PROCESO DE NORMALIZACIÓN DE BDR

Para llevar a cabo el proceso de normalización de BDR, es necesario transitar por un conjunto de algoritmos previos que me permiten descomponer el modelo original en un conjunto de relaciones más sencillas y simples, en término de las operaciones que se ejecutan sobre ellas.

El proceso de normalización de BDR parte del *modelo relacional*, con el objetivo de representar el diseño lógico de la realidad que describe el modelo, es decir, conocer la semántica de los atributos de la relación reflejada a través del conjunto de *dependencias funcionales*; término principal que se utiliza para poder realizar dicho proceso, descrito con detalles en el capítulo uno.

El conjunto de dependencias funcionales que se establecen en cada una de las relaciones del modelo relacional, no constituyen las únicas dependencias funcionales que se cumplen y existen en el esquema relacional. ¿Cómo determinarlas?

Para darles respuesta a esta interrogante se pensó en el ***algoritmo para el cálculo del Cierre de un Conjunto de Dependencias Funcionales*** (***DF*$^+$*)** aplicando los *Axiomas de Armstrong*, que permiten derivar dependencias a partir de un conjunto inicial ***DF***. El inconveniente del método es su excesiva complejidad computacional (es del tipo *NP-completo*), aún cuando el número inicial de ***DF*** sea pequeño, el número total de ***DF***$^+$ sería muy grande. Por otro lado, no todas las dependencias en ***DF***$^+$ son útiles, por lo tanto, se debe buscar, en la manipulación de las dependencias funcionales que exige la teoría de normalización, métodos que no estén basados en el cálculo de ***DF***$^+$. El ***algoritmo para el cálculo del cierre de un descriptor*** es el recomendado en la mayoría de las bibliografías relacionada con el diseño de BD en la teoría relacional, propuesto en ***(Ullman, 1988)***, con una *complejidad temporal* para el caso peor (O-grande) de ***O(n*m)***, donde ***n*** es el número de atributos del esquema relacional y ***m*** es el número de ***DF*** de la relación. El mismo es utilizado por casi todos los algoritmos que intervienen en el cálculo de ***DF***$^+$ para obtener esquemas que cumplan con los diferentes niveles de normalización; constituyendo el eje central en la implementación de ***algoritmos para la determinación del conjunto de claves o llaves primarias,*** primer concepto importante dentro de los esquemas de relaciones en un modelo de datos.

Entre los libros consultados por el autor, muy pocos abordan de manera formal la representación de un *algoritmo para el cálculo de las claves primarias* de un esquema de relación, esta afirmación podemos encontrarla también en el artículo de *(César Domínguez Pérez,et al. 8-10 julio 2009)*, como se puede mostrar en la Tabla 2.2.

Libro	Total nº páginas	Normalización nº páginas	%	Cuestiones/ Ejercicios	Algoritmo claves
Elmasri y Navathe 2007 [8]	988	74	7,49%	72	No
Connolly y Begg 2007 [5]	1269	56	4,41%	17	No
Date 2004 [6]	983	74	7,53%	30	No
Silberschatz y otros 2007 [22]	522	40	7,66%	30	No
Rob y Coronel 2004 [19]	838	40	4,77%	30	No
Kroenke 2003 [11]	671	30	4,47%	25	No
Piattini y otros 2006 [16]	946	184	19,45%	12	Sí
Rivero y otros 2005 [18]	574	101	17,60%	22	Sí
Promedio	**848,9**	**74,9**	**9,2%**	**29,7**	

Tabla 2.2 Tratamiento bibliográfico sobre la normalización de BDR

(César Domínguez Pérez,et al. 8-10 julio 2009)

Una de las variantes más triviales que se pudiera pensar para desarrollar dicho algoritmo, sería calcular el cierre para cada conjunto de descriptores del esquema de relación, (representado en este caso por ***R***) y seleccionar aquellos que coincidan con el conjunto de atributos de ***R***. Esta variante a pesar de ser la más trivial, suponía hacer, $\sum_{i=1}^{n} C_n^i$ *(Combinaciones de cálculo de cierre de descriptores)*, en el caso peor el algoritmo estaría acotado por una función exponencial. Esta variante la utiliza la herramienta (***DBDesignTools (Librería en C++)***) para la determinación de las claves primarias. "SINORGES" usa un algoritmo más eficiente, con respecto a la complejidad temporal, para darle solución a esta funcionalidad y es precisamente el algoritmo descrito en (Baizán, 1987).

La ***complejidad temporal***, de los algoritmos comentados anteriormente, es proporcional al conjunto de atributos y dependencias funcionales que pertenecen al modelo. Por lo tanto, lograr reducir dichos conjuntos sin afectar ninguna de las restricciones del negocio que el modelo de datos describe, contribuye proporcionalmente a disminuir la complejidad temporal; no sólo de estos métodos, sino también de los algoritmos de normalización que los usan para su funcionamiento. Por tal motivo se hace necesario sintetizar dichos conjunto a través de ***algoritmos para la determinación de coberturas minimales*** de esquemas relacionales encontrado en (Herrera, 2006). El resultado de este algoritmo es el punto de partida de todos los algoritmos de normalización que fueron implementados en el sistema "SINORGES".

Para la obtención de esquemas normalizados en 3FN existen dos grandes grupos de algoritmos:

- ✓ ***Algoritmo de síntesis.***
- ✓ ***Algoritmo de descomposición.***

El *Algoritmo de Síntesis* basado en las definiciones de esquemas sintetizados propuesto por el Dr. P.Bernstein, constituye el primer intento satisfactorio para instrumentar el proceso de normalización planteado por el Dr.Codd en 1970. Su descripcion formal se encuentra en (Baizán, 1987).

El *Algoritmo de Descomposición* basado en la teoría de descomposión de relaciones, es la técnica más nombrada en las diferentes fuentes consultadas por el autor de la presente investigación, utilizada para la normalización de esquemas, no sólo para 3FN sino también para formas normales superiores como FNBC, 4FN y 5FN. Es el método que utilizan la mayoría de las soluciones existentes mencionados en el capítulo anterior. Su inclinación hacia este método está proporcionada por la capacidad de permitir modelos normalizados sin pérdida de información o ***descomposición sin pérdida de información*** *(Lossless Join) (sus siglas en inglés LLJ)* y modelos normalizados con ***preservación de dependencias funcionales,*** aspectos importantes que deben tenerse siempre muy presentes si se pretende obtener verdaderos modelos de datos con elevada calidad en su diseño. Para su demostración formal existen un conjunto de definiciones matemáticas basadas en la teoría relacional que se pueden encontrar en (Baizán, 1987) (Herrera, 2006). Es válido aclarar que al desarrollar el proceso de normalización por descomposición a FNBC no siempre se puede conseguir la ***preservación de dependencias funcionales*** que se encontraban inicialmente**,** no siendo así para el caso de 3FN.

La propuesta del sistema "SINORGES" que se presenta en esta investigación, está diseñada para brindar los dos grupos de algoritmos para realizar el proceso de normalización de esquemas relacionales hasta FNBC, y deja en manos de los diseñadores o especialistas de BD su elección.

2.4 SISTEMA "SINORGES"

2.4.1 CONCEPCIÓN DEL SISTEMA "SINORGES"

2.4.1.1 REPRESENTACIÓN DEL MODELO RELACIONAL

El sistema "SINORGES" es un software que automatiza el proceso de normalización de BDR iniciado en el modelo relacional y permite la interacción directa con los diferentes SGBD. Brinda mecanismos fáciles y cómodos de inserción y eliminación de *atributos, dependencias funcionales y relaciones*, así como también para la modificación de sus propiedades. El modelo relacional en el sistema se describe mediante un conjunto de *relaciones*, las cuales están representadas en forma de tablas, como se puede

apreciar en la *Fig. 9*, donde cada una de estas relaciones contiene sus *atributos propios* y su conjunto de *dependencias funcionales*, en correspondencia con la teoría de normalización. Puede ser observado desde el sistema de una mejor manera en la *Fig. 9 (a), (b), (c)*.

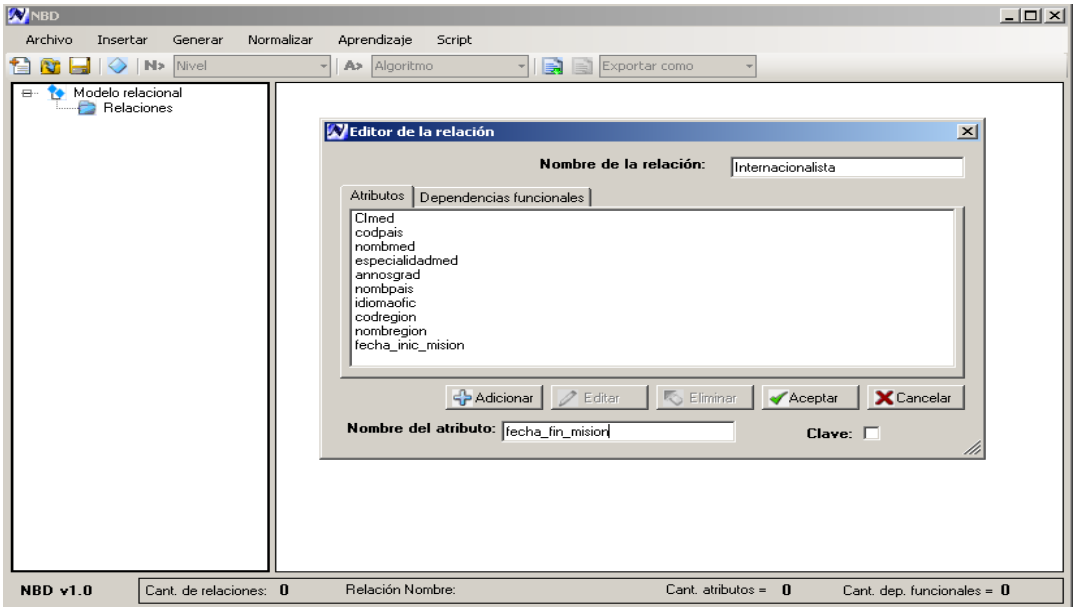


Fig. 9 Representación de la entrada de una nueva Relación con el Sistema "SINORGES"
(a) Representación de la entrada de Atributos pertenecientes a la nueva Relación

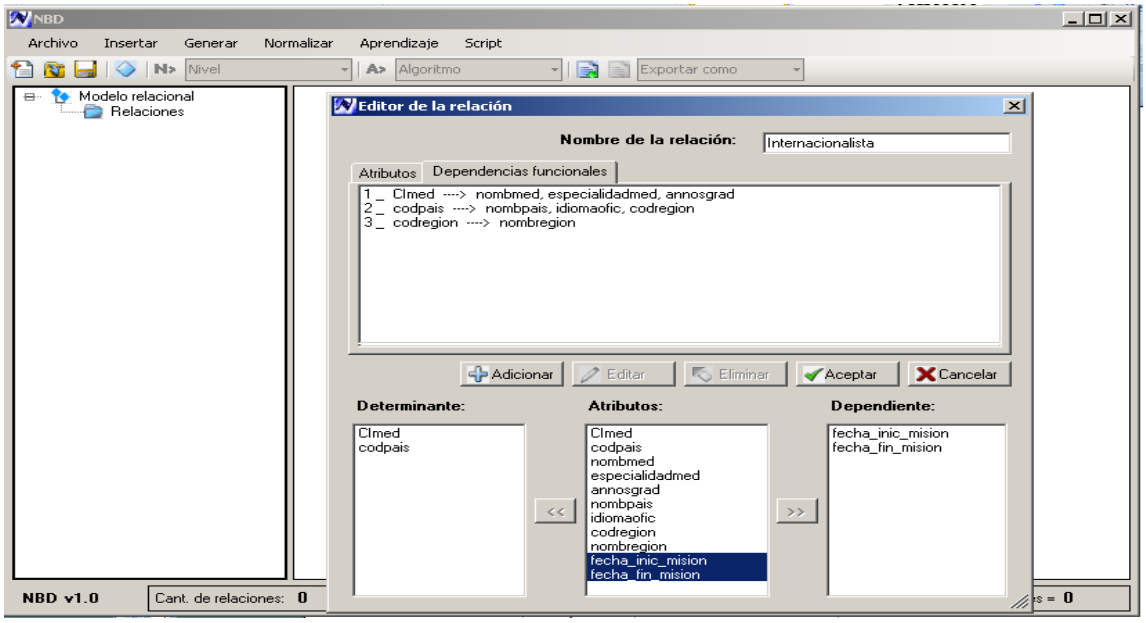


Fig. 9 Representación de la entrada de una Relación con el Sistema "SINORGES"
(b) Representación de la entrada del conjunto de DF pertenecientes a la nueva Relación

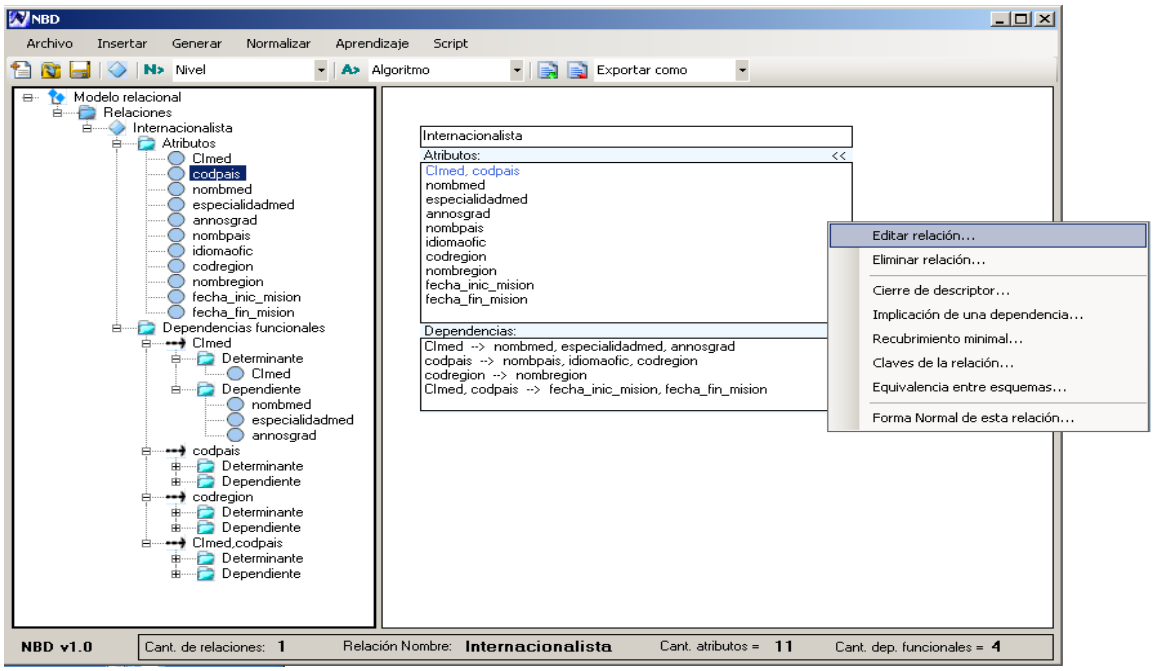


Fig. 9 Representación de la entrada de una Relación con el Sistema "SINORGES"
(c) Representación de la nueva Relación creada

En el modelo representado en la *Fig 9 (c)*, se muestra como es identificada la *clave primaria*, resaltada con respecto a los demás atributos de la relación en función del conjunto de *dependencia funcionales* que fueron introducidas por el usuario. Esta identificación puede ser producida de manera automática por el sistema, en caso de no haber especificado ninguna clave el usuario, o de forma manual; este último caso siempre es verificado por el sistema para evitar selecciones erróneas de estas. Brinda además, en la parte izquierda de la ventana principal una representación general (en forma de árbol) de todas las *relaciones* que componen el modelo, con la posibilidad de editar, eliminar y realizar cualquier algoritmo básico, que se verá con más detalles posteriormente para cada relación del modelo.

Otro mecanismo soportado por el sistema con el propósito de facilitar la entrada de los modelos de datos para el usuario, es la posibilidad de ***cargar modelos físicos de BD*** realizados en los diferentes SGBD como: PostgreSQL, Microsoft SQL Server y MySQL, o en la herramienta de diseño de BDR "Embarcadero ER / Studio" para los SGBD mencionados anteriormente. De manera que al ser elegida esta opción por el usuario el sistema cargará en la ventana principal y transformará el modelo físico de BD seleccionado, por su respectivo modelo lógico, conservando todas sus propiedades descritas en el modelo físico para realizar posteriormente la validación y corrección del mismo. Este aspecto

constituye un aporte importante desarrollado en el sistema que favorece su integración con herramientas muy utilizadas no sólo en la UCI, sino también a nivel mundial para el diseño y la manipulación de los datos.

2.4.1.2 VISUALIZACIÓN DE LOS ALGORITMOS PRINCIPALES DEL SISTEMA "SINORGES"

La visualización de software y especialmente la visualización de algoritmos, contribuye favorable y efectivamente en la comprensión de los mismos. Ayuda tanto en la tarea de interpretación del programa subyacente como en la de diseño, depuración, prueba y mantenimiento del software. Para ello se debe disponer de sistemas amigables con el usuario (facilidad de interacción hombre-máquina, en cuyo diseño se debe poner especial interés en la percepción humana). (Almeida, 2003)

La visualización de software presenta importantes beneficios educativos que estimulan y ayudan a los estudiantes en las distintas situaciones del aprendizaje, entre ellas se pueden mencionar:

- ✓ Facilitan el desarrollo de destrezas por la oportunidad de realizar prácticas adicionales. Los estudiantes tienen una nueva forma para experimentar los algoritmos. Además de resolver los ejercicios en papel y escribir los programas, ellos pueden percibir visualmente los algoritmos y estudiar sus características observando e interactuando con la animación. (Lawrence, et al. 2000).

- ✓ Asisten en el desarrollo de habilidades analíticas y promocionan las predicciones, pues los estudiantes deben coleccionar sus propios datos para el análisis del algoritmo y los subsecuentes diseños de algoritmos mejorados. Ofrecen distintas ventajas comparadas frente a la ayuda ofrecida por la enseñanza tradicional, tal como el libro de texto y el pizarrón.

- ✓ Ofrecen un buen soporte al docente y son de gran ayuda en la clase durante la explicación de la conducta dinámica de un algoritmo.

- ✓ Permiten la exploración, jugando interactivamente, de las peculiaridades de un software, mejorando la comprensión individual de los estudiantes. Permite, a los mismos, manipular el software y sus entradas, formular hipótesis de la conducta del algoritmo y luego estudiar las acciones resultantes, verificando o refutando sus ideas. Los capacita para poder desarrollar un nuevo algoritmo o variantes más eficientes de ellos, además de aprender el código. La interactividad agrega un nuevo nivel de efectividad al ambiente de aprendizaje, y es una

> herramienta apropiada en concepto de enseñanza, ya que fuerza a los aprendices a tomar parte de la lección, y no simplemente observar un movimiento. La interactividad ayuda a los estudiantes a adquirir una experiencia invalorable en la resolución de problemas (Merril, 2006).

Es importante el valor educativo de las técnicas de visualización de software. La animación de algoritmos y la visualización de programas ayudan a los estudiantes a comprender los conceptos de software y también a los docentes en su tarea de enseñar dichos conceptos. Distintas experiencias con respecto a la aplicación de la Visualización de Software en la enseñanza de la programación se han realizado en diversas universidades del mundo. Actualmente se emplean como recurso didáctico tanto en los cursos elementales como en los avanzados (Basik, et al. 2006).

Aunque todas las visualizaciones de algoritmos tienen el mismo propósito, el uso de las mismas puede variar considerablemente, nuestro propósito es fortalecer el conocimiento y el aprendizaje de los diferentes algoritmos que intervienen en el proceso de normalización de BDR de nuestros estudiantes UCI, y que pueda servir de apoyo a la asignatura de "Sistemas de Bases de Datos" y ser usada:

- ✓ Para acompañar una lectura y ayudar a comprender los conceptos claves que explica el profesor durante la clase.
- ✓ En un laboratorio formal donde los estudiantes interactúan con las computadoras.
- ✓ Para uso informal por los estudiantes fuera de la clase, en su tiempo libre, para ayudar a comprender más acerca de un determinado algoritmo.
- ✓ Para realizar un aprendizaje personalizado e individualizado.

El sistema "SINORGES" brinda la mayoría de los algoritmos utilizados dentro del proceso de normalización entre ellos tenemos: Cálculo de Cierre de Descriptores de un esquema de relaciones, Cálculo del Recubrimiento Minimal del conjunto de DF de un esquema, Equivalencia entre dos conjuntos de DF, Cálculo de las Claves Primarias de un esquema de relación. Estos algoritmos son visualizados a través de seudo-código de manera sencilla y fácil de interpretar por cualquier estudiante universitario de la especialidad de informática, donde a través de un indicador que va recorriendo paso a paso cada una de las instrucciones del algoritmo (manipulado por los usuarios, a través del botón siguiente de la ventana, como se puede apreciar en la Fig. 10), permite observar los cambios que se van produciendo en el *conjunto solución* para la obtención del resultado final del algoritmo,

proporcionándole además una descripción más detallada asociada a cada iteración de este indicador. (Ver detalles, en la Fig. 10 (a) y (b)).

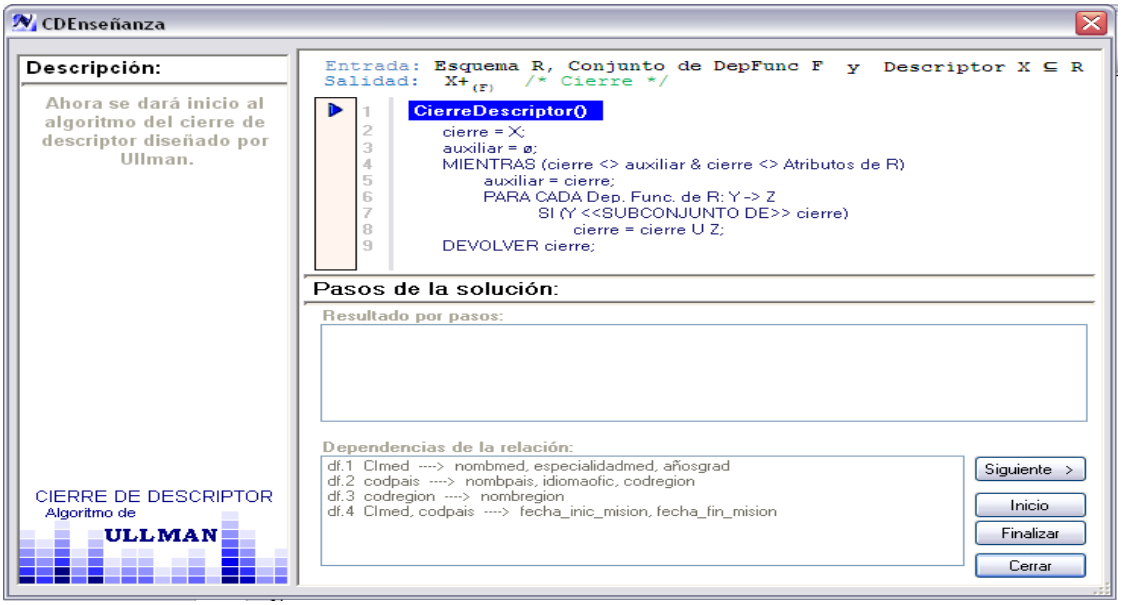


Fig. 10 Visualización del Cálculo del Cierre de un Descriptor a través del Sistema "SINORGES"
***(a) Inicialización del Proceso de Visualización para determinar el* (CImed, codregion)$^+$**

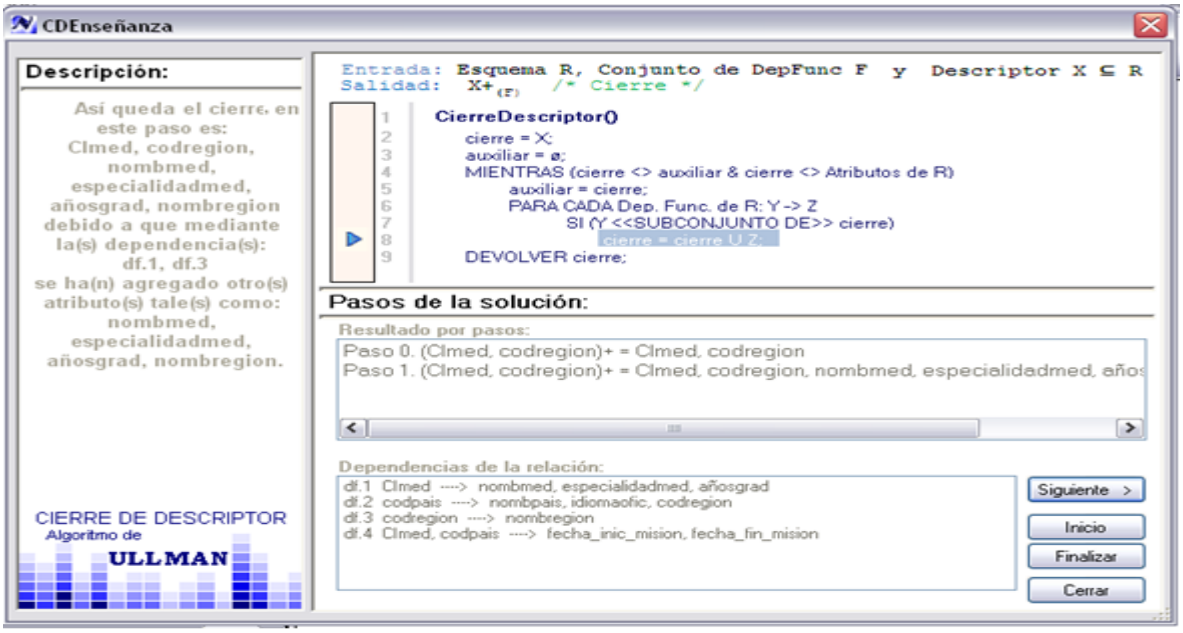


Fig. 10 Visualización del Cálculo del Cierre de un Descriptor a través del Sistema "SINORGES"
***(b) Proceso de Visualización para determinar el* (CImed, codregion)$^+$.**

2.4.1.3 REPRESENTACIÓN DE LOS ALGORITMOS DE NORMALIZACIÓN DEL SISTEMA "SINORGES"

El sistema "SINORGES" le permite comprobar al usuario en que nivel de normalización se encuentra el modelo introducido (Ver *Fig. 9(c)*); además de brindarle la posibilidad de transformar el modelo en niveles más alto de normalización, dígase 2FN, 3FN y FNBC, ofreciéndole para el caso de 3FN la *normalización por síntesis de Bernstein* y la *normalización por descomposición* con *preservación del conjunto de dependencias funcionales y sin pérdida de información*. Para todos los casos al realizar la selección por el usuario del nivel de normalización que desea para su modelo, el sistema le ofrecerá una vista previa donde le mostrará cómo quedará el modelo una vez realizado el proceso, representándole las interrelaciones que se establecen entre cada *esquema de relación* resultante, así como, las nuevas claves de los esquemas normalizados resaltadas, con el objetivo de lograr una mejor comprensión del proceso que se realizó para el usuario (Ver los detalles en la *Fig.11(a)*). Una vez conforme con la transformación propuesta por el sistema puede pulsar *"Aplicar"* y le mostrará el esquema relacional normalizado en la ventana principal del sistema, como se puede apreciar en la *Fig. 11(b)*, en caso de no conformidad puede retomar su modelo inicial al presionar *"Cancelar"*.

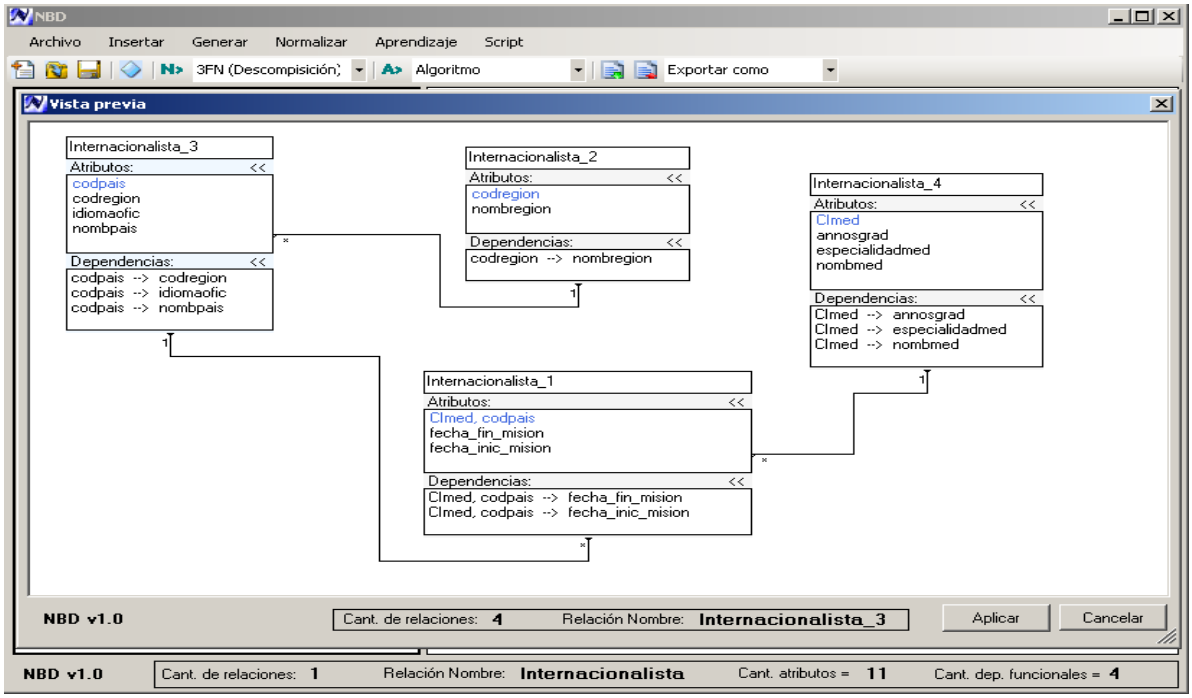


Fig. 11 Representación del proceso de Normalización a 3FN por el Sistema "SINORGES" (a). Vista previa de un esquema normalizado por descomposición en 3FN

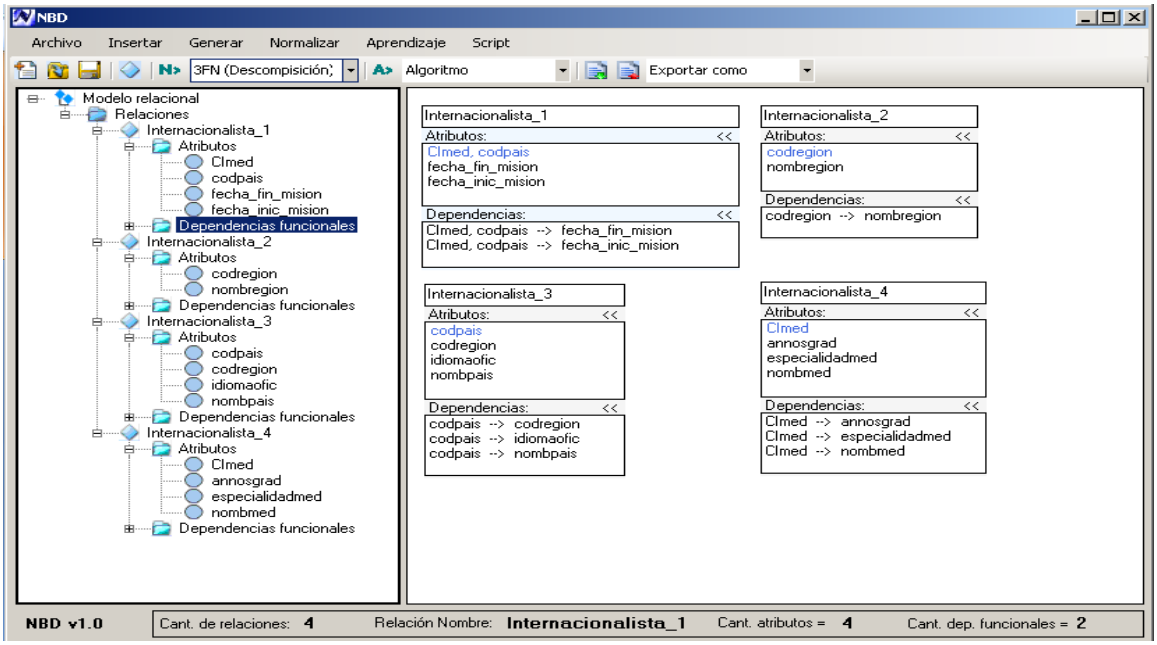


Fig. 11 Representación del proceso de Normalización a 3FN por el Sistema "SINORGES" (b). Representación del resultado final del esquema normalizado por descomposición en 3FN

2.4.1.4 REPRESENTACIÓN DE LA GENERACIÓN DE SCRIPTS EN EL SISTEMA "SINORGES"

Una vez normalizado el esquema relacional introducido por el usuario, el sistema es capaz de generar scripts en lenguaje SQL, que le permite ser exportado a los SGBD más utilizados por nuestra universidad, como: *PostgreSQL, Microsoft SQL Server y MySQL*, cumpliendo con las sintaxis propias del lenguaje para cada uno de ellos. Para realizar estas operaciones el sistema le proporciona una interfaz al usuario para que realice la entrada de los diferentes tipos de datos asociados a los atributos introducidos en la confección del esquema relacional, junto con otras características necesarias como tamaño, escala, precisión, tipos de valores (Ver detalles en la *Fig. 12*), además es capaz de verificar de forma automática, similar a como lo desarrollan los SGBD relacionales, la integridad de los datos almacenados, dígase *restricciones de dominios, reglas de integridad de entidades y regla de integridad referencial*. Luego, se procede a la generación automática de un fichero con extensión ***.sql***, con el objetivo de garantizar su integración con los SGBD relacionales antes mencionados, de manera que los SGBD puedan nutrirse del resultado final del sistema. Le facilita al usuario además, en caso de desearlo, visualizar este código en modo lectura, como se puede observar en la *Fig. 12(a)*.

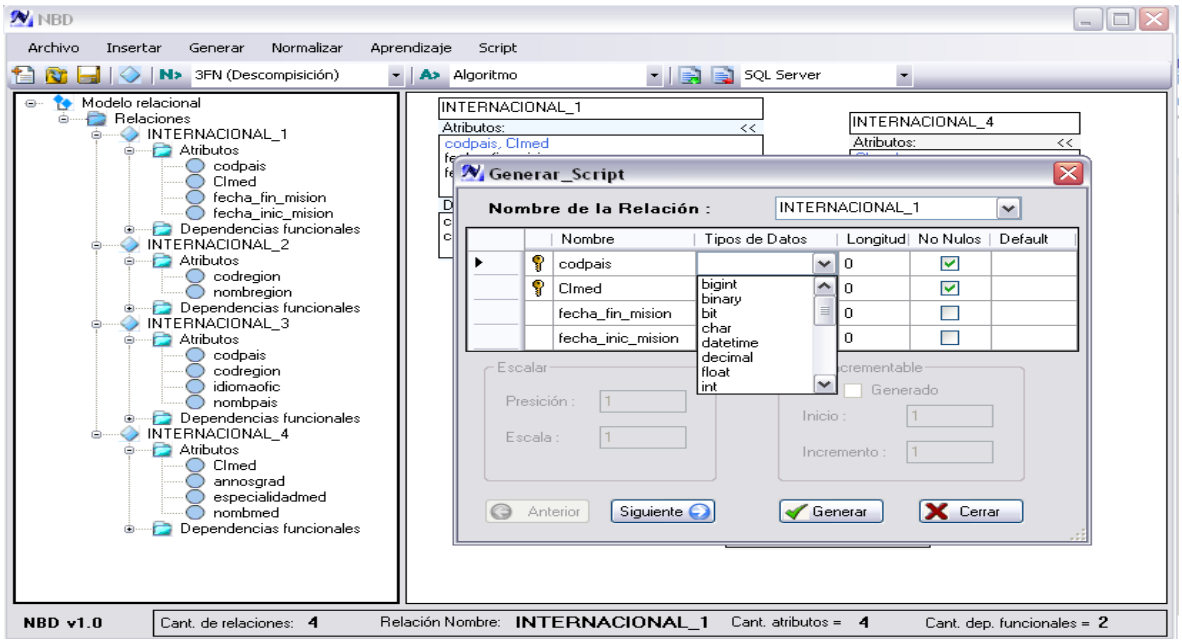


Fig. 12 Representación de la entrada de datos necesarios para la generación de scripts en el Sistema "SINORGES"

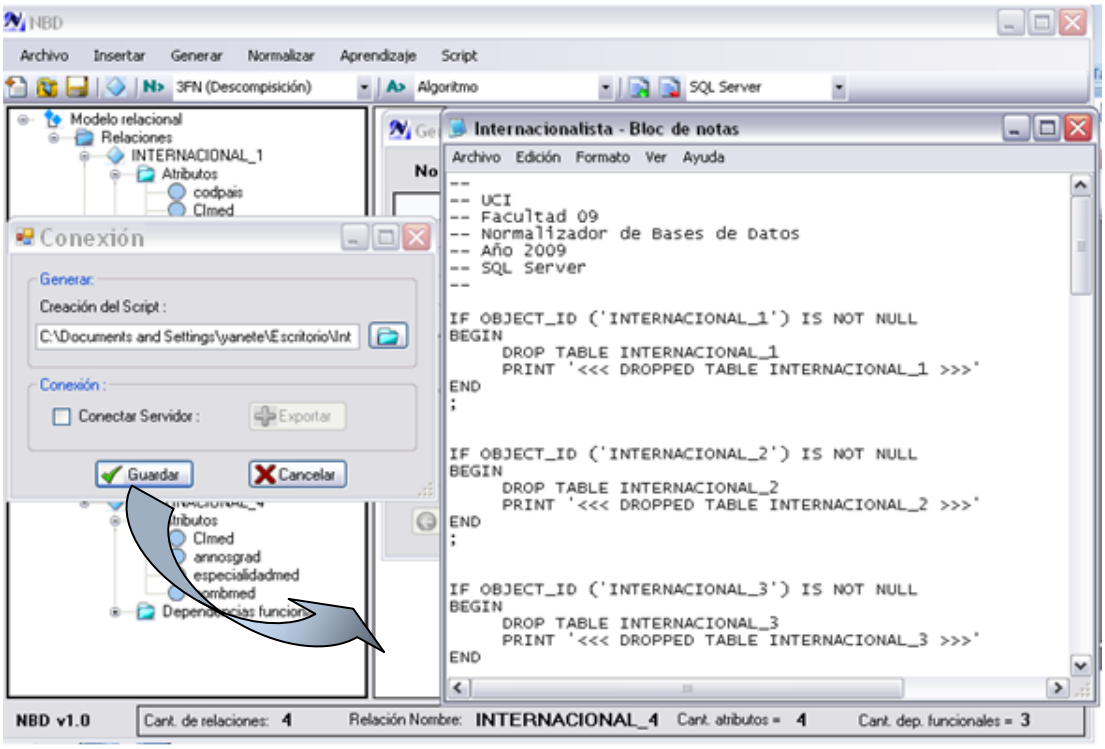


Fig. 12 Representación de la entrada de datos para la generación de scripts en el Sistema "SINORGES"

(a) Representación del script generado para "Microsoft SQL Server".

2.4.1.5 REPRESENTACIÓN DE LA GENERACIÓN DEL MODELO FÍSICO POR EL SISTEMA "SINORGES"

Una vez normalizado el esquema relacional introducido por el usuario, el sistema es capaz de generar el modelo físico de BD y exportarlo directamente hacia los SGBD *PostgreSQL, Microsoft SQL Server y MySQL*, o sea, crea la BD normalizada en el SGBD seleccionado por el usuario; para lograrlo le brinda una interfaz amigable, (Ver detalles en las *Fig. 13(a) y Fig. 13(b)*) donde en dependencia del SGBD seleccionado y la entrada de los datos necesarios para lograr la conexión al servidor de DB, les genera automáticamente la BD descrita por el modelo normalizado, lista para ser manipulada e implementada desde el SGBD.

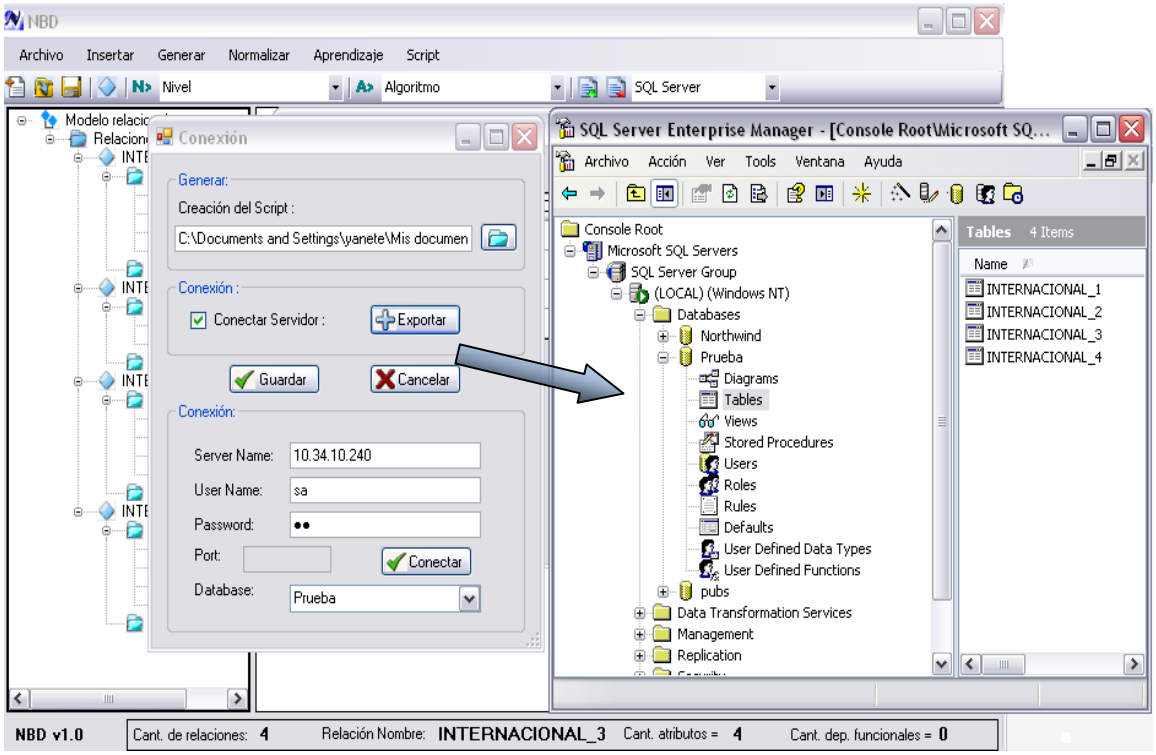


Fig. 13 Representación de la generación del modelo físico de BD en el Sistema "SINORGES"
(a) Representación del modelo físico "Prueba" exportado al SGBD "Microsoft SQL Server"

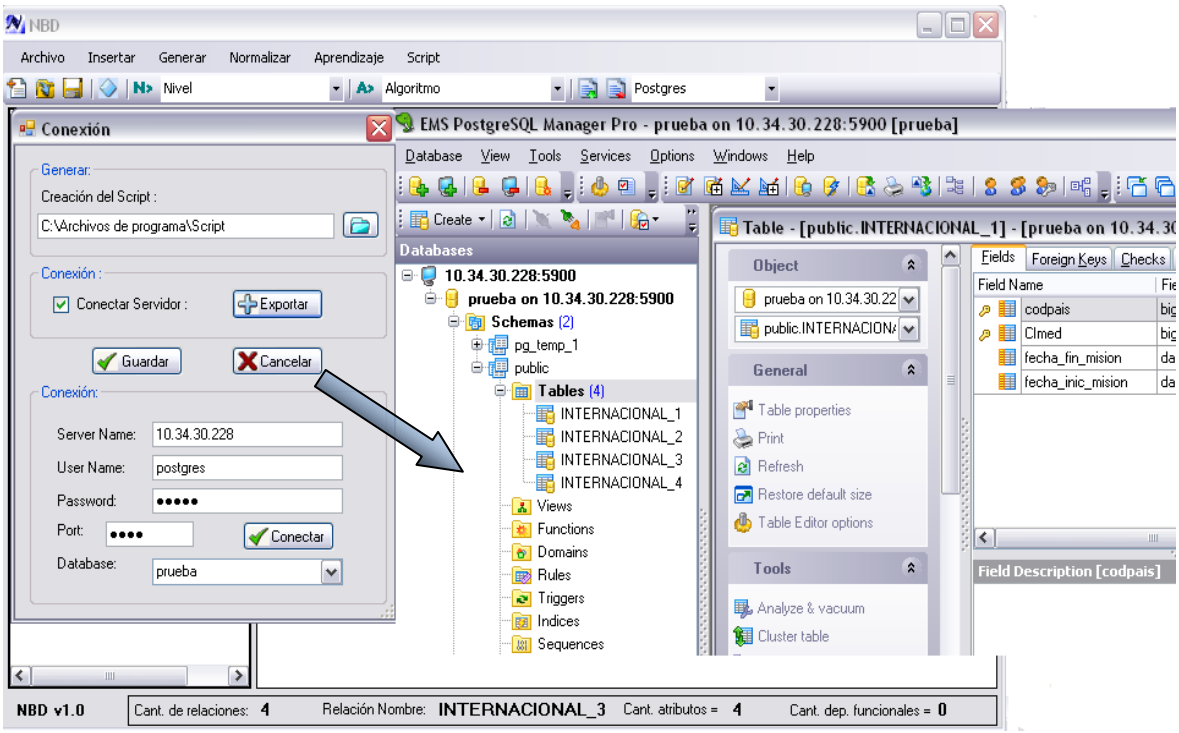


Fig. 13 Representación de la generación del modelo físico de BD en el Sistema "SINORGES" (b) Representación del modelo físico "Prueba" exportado al SGBD "PostgreSQL"

2.4.2 CONSTRUCCIÓN DEL SISTEMA "SINORGES"

Se describe a continuación el resultado del proceso de construcción del sistema "SINORGES", siguiendo lo que se propone en RUP, a través de algunos de los artefactos que se elaboraron.

2.4.2.1 MODELO DE DOMINIO

De acuerdo a la metodología seleccionada se pueden utilizar de forma opcional dos artefactos fundamentales para la comprensión del contexto en el que se insertará la herramienta a desarrollar, ellos son: el *Modelo de Negocio y el Modelo de Dominio* (Jacobson, I et al. 2000). Como primer paso al diseño del sistema se expresará el entendimiento ganado del negocio a través del *modelo de dominio*, que es descrito en la *Fig. 14* que se muestra a continuación:

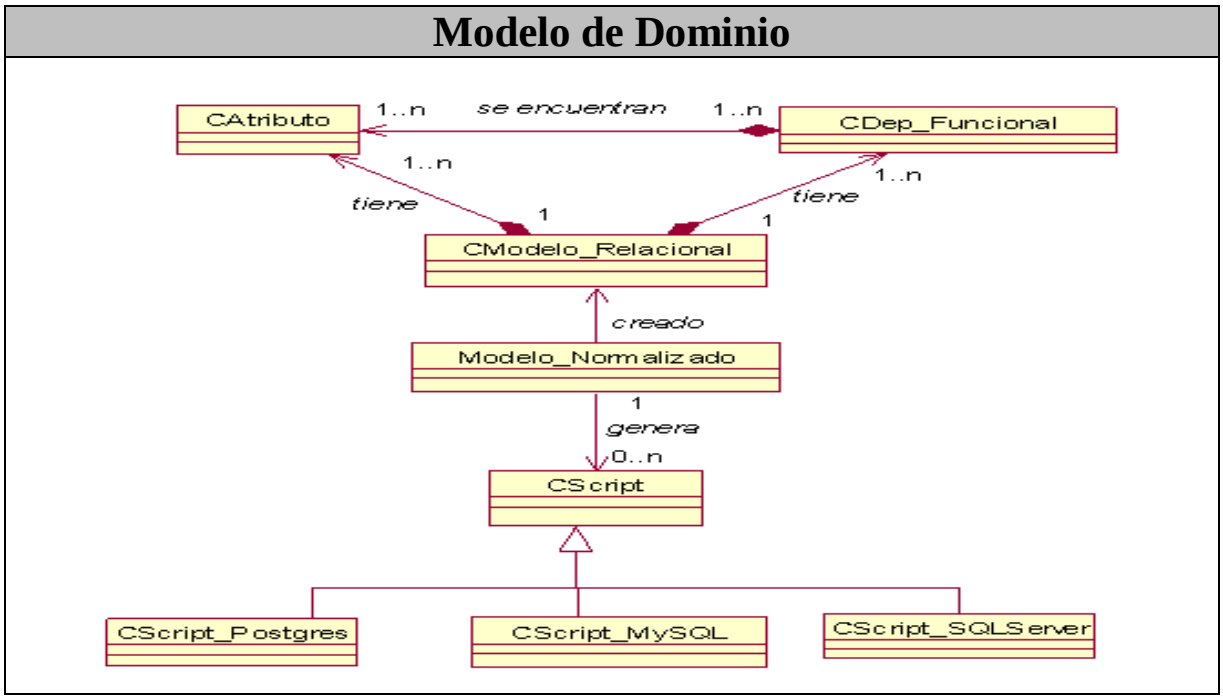


Fig. 14 Representación del Modelo de Dominio del Sistema "SINORGES"

En este modelo se han representado los conceptos fundamentalmente que están relacionados con el modelo relacional que es el punto de partida del sistema, algunos de ellos corresponden con los elementos que lo componen. Se muestra que un *Modelo Relacional* está compuesto por *Atributos* y *Dependencias Funcionales*, donde a partir de este modelo se obtiene el modelo relacional normalizado, y en dependencia de las necesidades del usuario el sistema puede generar scripts de BD en el lenguaje SQL para *PostgreSQL, Microsoft SQL Server y MySQL*.

2.4.2.2 ESPECIFICACIONES DE LOS REQUISITOS DEL SISTEMA "SINORGES"

Las funcionalidades del sistema y el Diagrama de Casos de Uso del sistema han sido modelado utilizando el listado de las característica que se muestra en el **Anexo I**, ambos artefactos propuestos en RUP para este propósito (Jacobson, I et al. 2000).

En el Diagrama de Casos de Uso del Sistema que se muestra en la *Fig.15*, se pueden apreciar los diferentes procesos que componen este sistema, las relaciones que existen entre cada uno de ellos y su interacción con el actor del sistema.

2.4.2.3 DIAGRAMA DE CLASES DEL SISTEMA "SINORGES"

El Diagrama de Clase es el diagrama principal de diseño y análisis para un sistema. Es una aproximación a un Caso de Uso guiado hacia el análisis orientado a objetos, el diagrama de clases se desarrolla a través de información obtenida en los Casos de Uso, Diagramas de Secuencia y Diagramas de Colaboración. Los objetos encontrados durante el análisis son modelados en términos de la clase a la que instancian, y las interacciones entre objetos son referenciados a relaciones entre las clases instanciadas (Jacobson, I et al. 2000). A continuación en la *Fig. 16* se representa el Diagrama de Clases de Análisis del sistema.

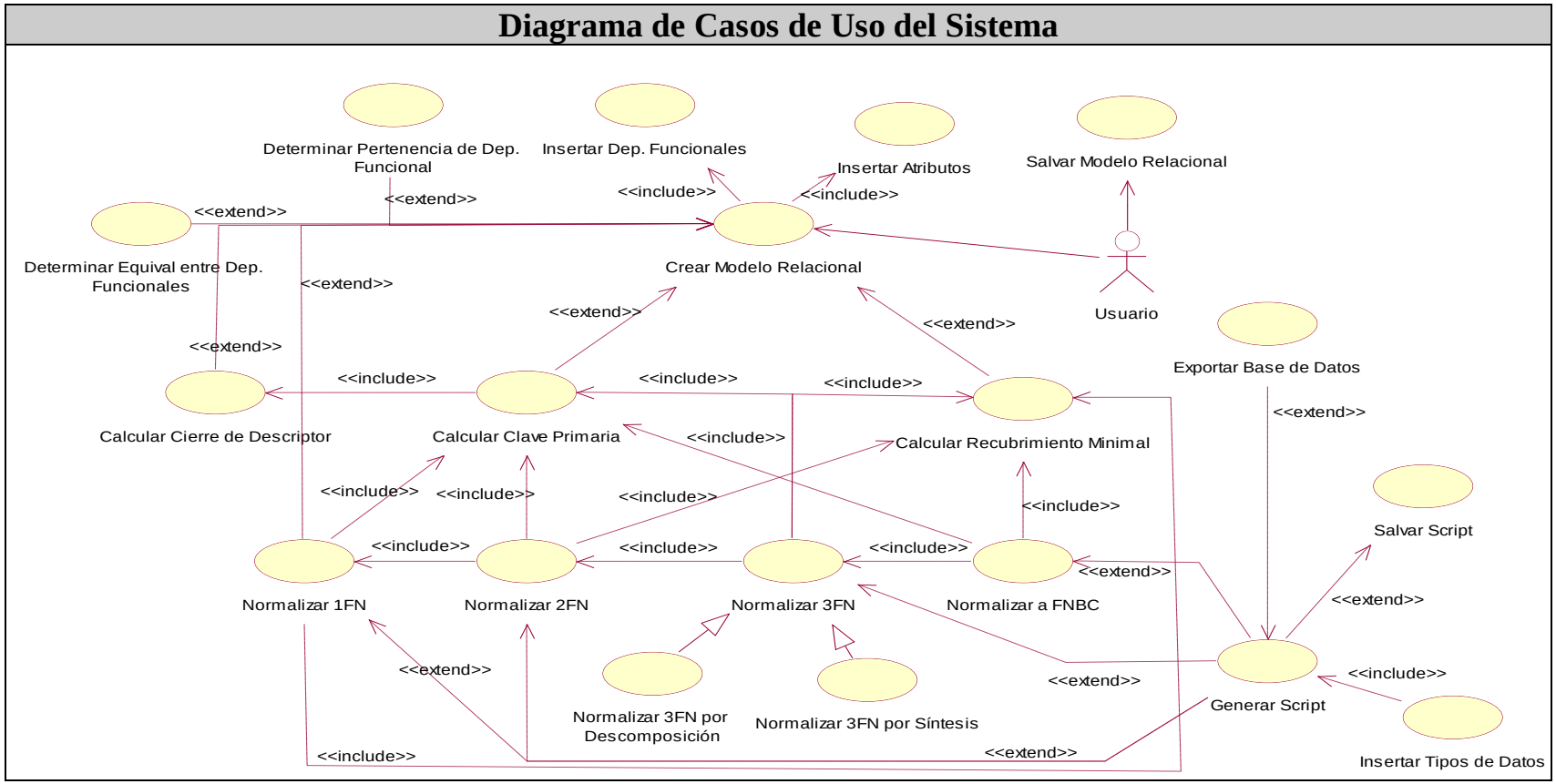


Fig. 15 Representación del Diagrama de Casos de Uso del Sistema "SINORGES"

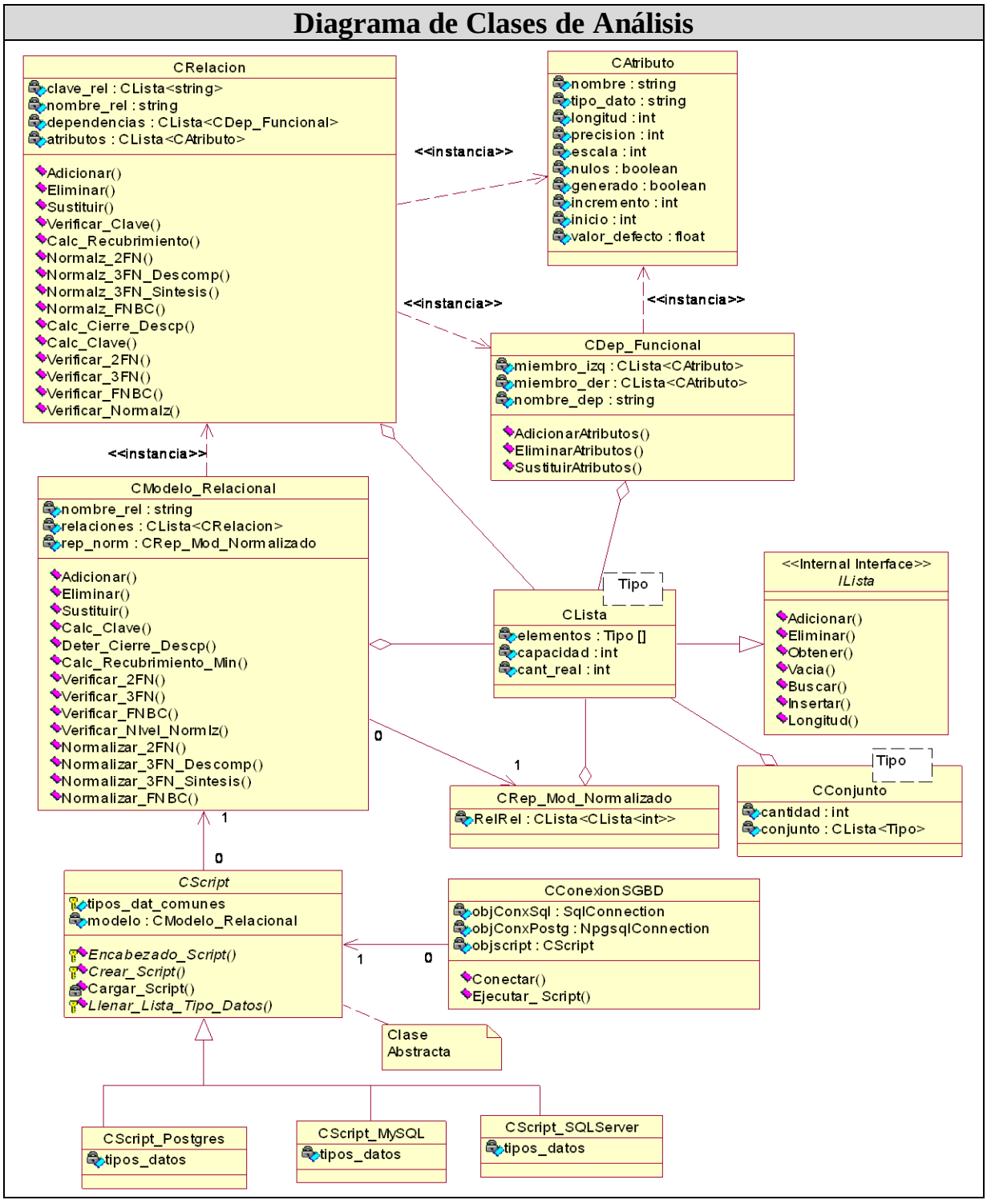


Fig. 16 Representación del Diagrama de Clases del Sistema "SINORGES"

2.5 CONCLUSIONES DEL CAPÍTULO

Al finalizar este capítulo, después de abordar la justificación de la fundamentación de la propuesta descrita por el autor, así como, haber realizado un análisis profundo de las características que presentan las herramientas existentes en el mercado y realizar una comparación de dichas herramientas con la propuesta, teniendo en cuenta los factores identificados por el autor de la investigación, podemos concluir que:

- ✓ La propuesta de solución permitirá la integración del proceso de normalización con los SGBD, aspecto que no se incluye en las herramientas que normalizan, haciéndola superior al compararla con las herramientas analizadas teniendo en cuenta la posibilidad de integración con otros sistemas (Interoperabilidad) y para el desarrollo de soluciones informáticas.

- ✓ La propuesta de solución permitirá la "*visualización de algoritmos*", técnica que no utilizan las herramientas que normalizan hoy en día.

- ✓ La utilización de la Metodología de RUP permitió un desarrollo exitoso de las herramientas propuestas.

Capítulo 3. Implementación del Sistema "SINORGES". Validación y Prueba

En el presente capítulo se abordan las consideraciones que se tuvieron en cuenta para la selección del lenguaje utilizado para el desarrollo de la propuesta del Sistema Informático, así como, una breve descripción de sus características principales y ventajas que ofrece el lenguaje. Se describe el modelo de componente del sistema propuesto, para brindar un mejor entendimiento de las relaciones entre las clases y componentes que se definieron e implementaron en el sistema. Además, se plasma un análisis de los resultados arrojados en la encuesta realizada a especialistas de la computación de la UCI, con el objetivo de corroborar la situación problemática planteada en esta investigación. Se ejecutaron algunas pruebas de aceptación a diferentes funcionalidades básicas para determinar fallas en la solución alcanzada, frente algunos casos de estudio desarrollados y por último se realizó un pequeño estudio de la viabilidad de la propuesta.

3.1 IMPLEMENTACIÓN DEL SISTEMA "SINORGES"

3.1.1 CONSIDERACIONES DE LA SELECCIÓN DEL LENGUAJE DE DESARROLLO

La implementación de la herramienta propuesta se llevó a cabo utilizando como lenguaje de programación ***C Sharp (C#)***, bajo el Entorno de Desarrollo Integrado (IDE) de software para su realización ***SharpDevelop 3.0*** y como plataforma de desarrollo ***Microsoft. NET Framework 3.5***. Se realizó un estudio de otros lenguajes de programación que se utilizan en la actualidad como: ***Java***, ***C++***, ***Python,*** su elección principalmente fue precedida por ofrecer una rápida, fácil, segura y robusta manera de desarrollar aplicaciones, además de contar con el vasto conocimiento que tiene creado el equipo de desarrolladores del sistema en dicho lenguaje.

SharpDevelop es un IDE gratuito para. NET en Windows, bajo la Licencia Pública General de GNU: Lesser General Public License (LGPL) 2.1 muy similar a Microsoft Visual Studio.

El **Framework de .Net** es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables. Las principales ventajas de este entorno son (CSharp-Online.NET):

- ✓ **Código administrado**: El Common Language Runtime (CLR) realiza un control automático del código para que este sea seguro, es decir, controla los recursos del sistema para que la aplicación se ejecute correctamente.
- ✓ **Interoperabilidad multilenguaje**: El código puede ser escrito en cualquier lenguaje compatible con .Net, porque siempre se compila en código intermedio Microsoft Intermediate Language (MSIL). Soporta múltiples lenguajes de programación y aunque cada lenguaje tiene sus características propias, es posible desarrollar aplicaciones con cualquiera de estos lenguajes. Existen más de 30 lenguajes adaptados a .Net, desde los más conocidos como C#, Visual Basic, C++, Delphi(Object Pascal) hasta otros lenguajes menos conocidos como Perl o Cobol.
- ✓ **Compilación just-in-time**: El compilador Just-In-Time (JIT) incluido en el Framework compila el código intermedio (MSIL) generando el código máquina propio de la plataforma. Se aumenta así el rendimiento de la aplicación al ser específico para cada plataforma.
- ✓ **Recolector de basura**: El CLR proporciona un sistema automático de administración de memoria denominado recolector de basura (garbage collector). El CLR detecta cuándo el programa deja de utilizar la memoria y la libera automáticamente. De esta forma el programador no tiene por que liberar la memoria de forma explícita, aunque también sea posible hacerlo manualmente.
- ✓ **Seguridad de acceso al código**: Se puede especificar que una pieza de código tenga permisos de lectura de archivos pero no de escritura. Es posible aplicar distintos niveles de seguridad al código.
- ✓ **Despliegue**: Por medio de los ensamblados resulta mucho más fácil el desarrollo de aplicaciones distribuidas y el mantenimiento de las mismas. El Framework realiza esta tarea de forma automática mejorando el rendimiento y asegurando el funcionamiento correcto de todas las aplicaciones.

3.2 DIAGRAMA DE COMPONENTES DEL SISTEMA "SINORGES"

Las clases definidas para la implementación de la propuesta, son agrupadas en diferentes componentes para un mejor encapsulamiento y una mejor distribución de las funcionalidades. Cada uno de estos componentes coincide, en la mayor parte de los casos, con ficheros de código fuente, cuya integración conforman el sistema en su conjunto. Se elaboró un Diagrama de Componentes, mostrado en la *Fig.*

17, para representar gráficamente cada uno de los componentes utilizados y las relaciones entre ellos (Jacobson, I et al. 2000).

Las clases definidas para la implementación de la propuesta, son agrupadas en diferentes paquetes para un mejor encapsulamiento y una mejor distribución de las funcionalidades.

Entrada del Modelo de Datos: Permite crear relaciones para construir el modelo de datos relacional a través de la inserción de atributos y dependencias funcionales. Manipula las funciones generales para el diseño de la entrada del modelo de datos como son: inserción, eliminación, modificación de los datos que corresponden al modelo y del modelo en general, funciones gráficas específicas para cada relación como: mover, redimensionar, ocultar componentes de la relación. Este módulo también tiene como función cargar scripts realizados en los SGBD: *PostgreSQL, Microsoft SQL Server y MySQL* o en el *Embarcadero ER/ Studio.*

Proceso de Normalización: Módulo de validación de la estructura lógica de los datos, detecta el nivel de normalización del modelo de datos, desde 2FN hasta FNBC, notificándole al usuario la localización exacta donde se encuentra el problema para llegar a niveles superiores de normalización. Este módulo permite además la corrección de la estructura lógica del modelo aplicando el proceso de normalización hasta FNBC utilizando las dos técnicas que existen: Descomposición y Síntesis.

Generación de Scripts: A partir del esquema o modelo de datos normalizado genera las sentencias de código SQL correspondientes, almacenándolas en un fichero con extensión ***.sql*** para los SGBD: *PostgreSQL, Microsoft SQL Server y MySQL*.

Generación del Modelo Físico de BD: Cuenta con las funcionalidades necesarias para transformar el modelo lógico de datos a su correspondiente esquema físico de BD para los SGBD *PostgreSQL, Microsoft SQL Server y MySQL* a partir del modelo de datos normalizado. Este módulo realiza validaciones con respecto a la integridad de los datos introducidos por el usuario, dígase restricciones de dominios, reglas de integridad de entidades y regla de integridad referencial. Dejando lista la BD para ser manipulada por los usuarios a través del propio SGBD.

Visualización de Algoritmos de Normalización: Cuenta con las funcionalidades mínimas desarrolladas a través de mecanismos sencillos usando la técnica de visualización de algoritmos

descritas en el capítulo dos. Logrando a través de este módulo que nuestros estudiantes de la UCI conozcan y adquieran nuevos conocimientos referentes a esta teoría de normalización. Entre los algoritmos que se describen se encuentran:

- ✓ Cálculo de Cierre de Descriptores de un esquema de relaciones.
- ✓ Cálculo del Recubrimiento Minimal del conjunto de DF de un esquema.
- ✓ Equivalencia entre dos conjuntos de DF.
- ✓ Cálculo de las Claves Primarias de un esquema de relación.

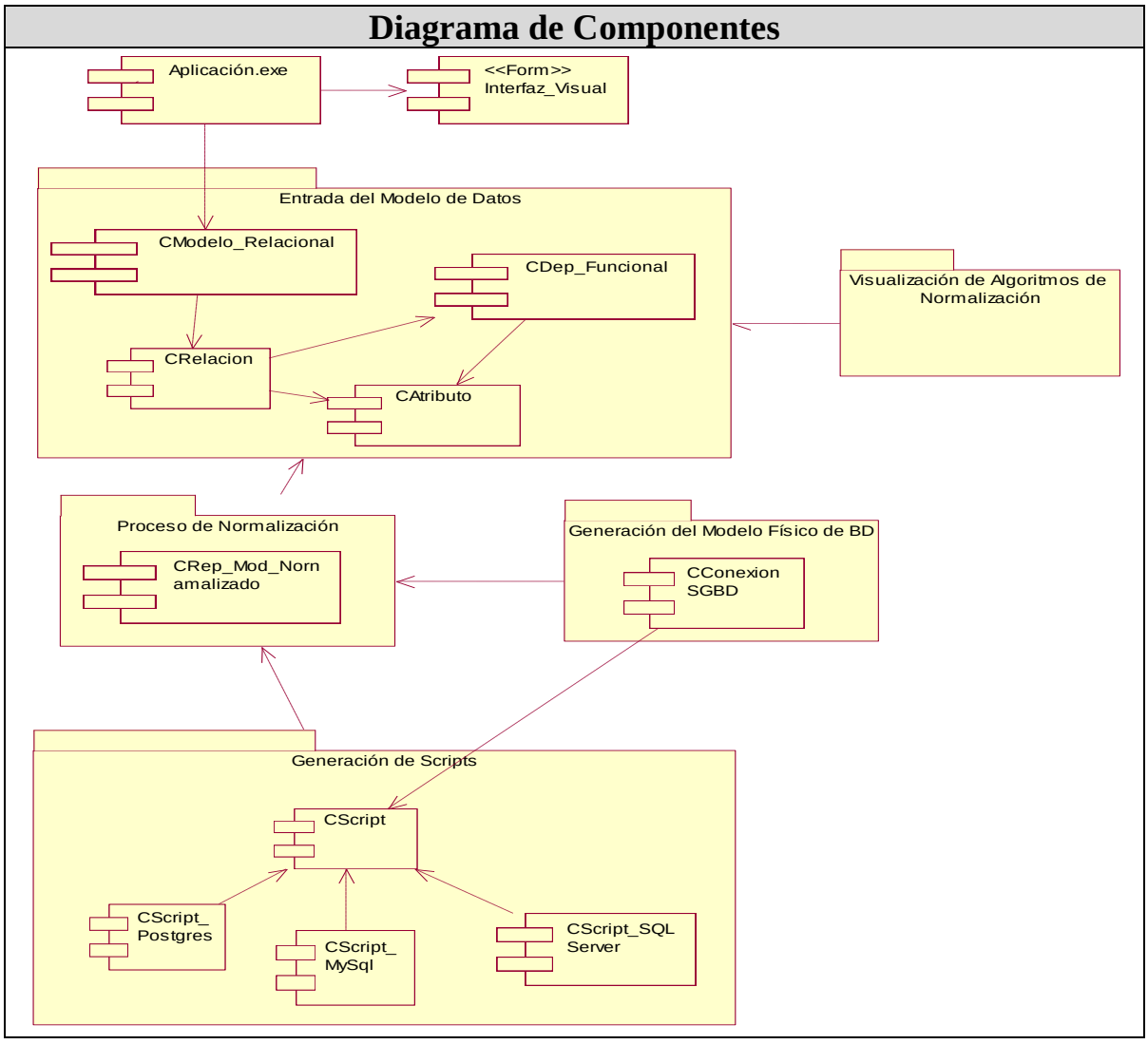


Fig. 17 Representación del Diagrama de Componentes del Sistema "SINORGES"

3. 3 VALIDACIÓN Y PRUEBA

3.3.1 VALIDACIÓN DE LA PROPUESTA

Se aplicó una encuesta a la mayoría de las entidades de la UCI, entre ellas: las diez facultades docentes, el Centro de Consultoría y el Centro de Tecnologías de Almacenamientos y Análisis de Datos (CENTALAD) de la Universidad. Entre los encuestados de cada una de estas entidades se encontraban jefes de polos, líderes de proyectos, diseñadores de bases de datos y otros (profesores de la asignatura de Sistemas de Bases de datos, analistas y programadores), a continuación se muestran en la *Tabla 3.1* los resultados obtenidos después de su procesamiento, teniendo en cuenta los aspectos evaluados en el cuestionario y las respuestas obtenidas en cada aspecto.

Aspectos del Cuestionario / Respuestas		**Siempre**	**Nunca**	**En Ocasión**	**Si**	**No**	**He oído hablar**
P #1	Necesidad de la utilización de la técnica de Normalización de BDR	68	0	32			
P #2	Necesidad de Integrar la Normalización con los SGBD				97	3	
P #3	Existencia de herramientas para la Normalización de BDR				11	74	15
P #4	Existencia de herramientas para la Modelación de BDR e Integración con los SGBD				74	17	9
P #5	Existencia de herramientas que Integren la Normalización con los SGBD				9	84	7
Total de Encuestados		**100**					

Tabla 3.1 Resultados obtenidos en la Encuesta realizada a Especialista de la Computación en la UCI

Realizando un análisis más profundo de las respuestas obtenidas, fundamentalmente para las respuestas *afirmativas* y las herramientas mencionadas que se utilizaron para confirmar dichas afirmaciones para la ***pregunta #3*** y ***pregunta #5*** de la encuesta. Se detectó lo siguiente:

- ✓ De un total de 9 afirmaciones **(Sí)** en la pregunta #5, **representando un 9% del total de encuestado con esta respuesta**:
  - El 55,5% (5 especialistas de estos) coincidieron con la herramienta CASE "Erwin".

- El 11,1% (1 especialista de estos) mencionó la herramienta "Oracle Warehouse Builder".
- El 33,3% (3 especialistas de estos) mencionó la herramienta "Embarcadero ER/Studio".

✓ De 7 respuestas **(He oído hablar de alguna(s))** en la pregunta #5, **representando un 7% del total de encuestado con esta respuesta:**

- El 28.57% (2 especialistas de estos) corroboró su respuesta con la herramienta CASE para UML: "*Visual Paradigm*".
- El 71,42% (el resto de ellos) no fundamentó su respuesta.

✓ De un total de 11 afirmaciones **(Sí)** en la pregunta #3, **representando un 11% del total de encuestado con esta respuesta:**

- El 45,45% (5 especialistas de estos) corroboró su respuesta con la herramienta CASE "Erwin".
- El 9,09% (1 especialista de estos) mencionó la herramienta CASE para UML: Rational Rose Enterprise.
- El 45,45% (5 especialistas de ellos) mencionó la herramienta CASE "Embarcadero ER/Studio".

✓ De 15 respuestas **(He oído hablar de alguna(s))** en la pregunta #3, **representando un 15% del total de encuestado con esta respuesta.**

- El 86,66% (13 especialistas de ellos) no fundamentaron su respuesta.
- El 6,66% (1 especialista de ellos) mencionó el SGBD *PostgreSQL*.
- El 6,66% (1 especialista de ellos) mencionó la herramienta CASE "Erwin".

El autor de la presente investigación comenta para el caso de la pregunta #5 que ninguna de las herramientas mencionadas por los encuestados garantiza la integración del proceso de normalización de BDR con los SGBD, al igual ocurre con las herramientas mencionadas para garantizar la automatización del proceso de normalización en la pregunta #3. Evidenciándose además por más del 65% de los especialistas encuestados criterios favorables que demuestran la importancia del proceso de normalización en el diseño de BDR y la necesidad de su integración con los SGBD, justificadas en los resultados de la pregunta #1 y pregunta #2, como se ha abordado en el curso de la investigación.

3.3.2 PRUEBA DEL SISTEMA PROPUESTO "SINORGES"

Las pruebas de software eran consideradas anteriormente "*sólo una actividad que realizaba el desarrollador para encontrar fallas en sus productos; con el paso de los años se ha determinado la*

importancia que tienen para garantizar el tiempo, el costo y la calidad del producto, de tal forma que actualmente son procesos cuyo propósito principal es evaluar en todo momento la generación del software respecto de los requerimientos establecidos al inicio" (Ramírez, 2008).

RUP, como Metodología de Desarrollo de Software (MDS), establece que se deben realizar pruebas exhaustivas a las funcionalidades ya implementadas con el propósito de no dejar pasar posibles detalles obviados en las descripciones textuales de CU del sistema o posibles errores de la aplicación ya culminada (Jacobson, I et al. 2000). La prueba a realizarse para comprobar estos objetivos descritos por RUP es "*Aceptación basada en la técnica de Caja Negra"*. Dicha técnica comprende la realización de "una prueba del comportamiento observable externamente del sistema". Para llevar a cabo ésta, se debe establecer un *plan de prueba*[1] conformado por: *los requerimientos que se desea probar, las estrategias a aplicar, la evaluación y el cronograma de las pruebas seleccionadas, además de los casos de prueba diseñados.*

3.3.2.1 REQUISITOS A PROBAR

Los requerimientos a probar son:

- ✓ Normalizar modelo de relación a FNBC por descomposición.
- ✓ Crear scripts en lenguaje SQL asociado al SGBD seleccionado por el usuario.
- ✓ Exportar el modelo físico normalizado al SGBD seleccionado por el usuario.
- ✓ Calcular las Claves Primarias de una relación.

3.3.2.2 CASOS DE PRUEBA DISEÑADOS

Un ***caso de prueba*** es un conjunto de entradas de prueba, condiciones de ejecución, y resultados esperados, desarrollados para un objetivo concreto, tal como, probar un camino concreto a través de un caso de uso, o verificar que se cumple un requisito específico. Los defectos hallados se analizan para localizar el problema. Después dichos problemas se priorizan y se corrigen por orden de importancia (Jacobson, I et al. 2000).

Los casos de prueba diseñados están enmarcados en probar los casos de usos del sistema críticos que a su vez son los escenarios de las pruebas a realizar. Estos casos de prueba encierran el resultado de la

1 Un plan de prueba no es más que la ruta a seguir para desarrollar las pruebas del software de una forma organizada, exigente y eficiente bajo restricciones explícitas de alta calidad.

interacción de los actores con el sistema y del cumplimiento de las especificaciones del caso de uso como tal.

Prueba de Aceptación para el Caso de Uso: Normalizar a FNBC

Caso de Prueba de Aceptación (CPA)
CU: Normalizar a FNBC
Nombre del CPA: CP1_ Normalizar modelo de relación a FNBC.
Descripción: El CP1 inicia cuando se desea descomponer el modelo de relación en el nivel de FNBC. Para realizarlo se cuenta con un modelo de datos al cual se le aplican los algoritmos de determinación de las claves primarias, cálculo del recubrimiento minimal de su conjunto de DF, validación del nivel de normalización de la relación y en caso de no encontrarse en FNBC, normaliza a FNBC; mostrándole una ventana con una vista previa del resultado de la descomposición, donde al aceptarlo se aplica la descomposición y se transforma el modelo de relación original al actual.
Condiciones de Ejecución: Se insertan datos válidos del modelo, incluyendo su conjunto de atributos y dependencias funcionales**.**
Resultado Esperado: El modelo de relación fue corregido descomponiéndose satisfactoriamente en el nivel de normalización FNBC y actualizado en el sistema.
Evaluación de la Prueba: Satisfactoria

Tabla 3.2 CP1_ Normalizar modelo de relación a FNBC

Prueba de Aceptación para el Caso de Uso: Generar Script

Caso de Prueba de Aceptación (CPA)
CU: Generar Script
Nombre del CPA: CP2_ Crear scripts en lenguaje SQL asociado al SGBD seleccionado por el usuario.
Descripción: El CP2 inicia cuando se desea crear el fichero ***.sql*** del modelo de datos normalizado que se está analizando. Para realizarlo el usuario selecciona el SGBD para el cual desea generar el script y los tipos de datos asociados de cada atributo que compone el modelo. Una vez introducido todos los datos se procede a generar el Script y luego en caso de ser solicitado por el usuario puede observarlo si lo desea.
Condiciones de Ejecución: Se debe seleccionar el SGBD y los tipos de datos asociados de cada atributo que compone el modelo de datos**.**
Resultado Esperado: Fue creado satisfactoriamente el script, almacenándolo en un fichero ***.sql***.
Evaluación de la Prueba: Satisfactoria

Tabla 3.3 CP2_Crear scripts en lenguaje SQL asociado al SGBD seleccionado por el usuario

Prueba de Aceptación para el Caso de Uso: Exportar Base de Datos

Caso de Prueba de Aceptación (CPA)
CU: Exportar Base de Datos
Nombre del CPA: CP3_ Exportar el modelo físico normalizado al SGBD seleccionado por el usuario.
Descripción: El CP3 inicia cuando se desea exportar automáticamente el modelo físico normalizado a los SGBD. Para realizarlo el usuario selecciona el SGBD donde desea exportar la BD y los tipos de datos asociados de cada atributo que compone el modelo. Una vez llenado todos los datos, ya es capaz de conectarse y exportar el modelo físico de BD al SGBD introduciendo el nombre del servidor de BD, usuario, contraseña y puerto (para el caso del SGBD PostgreSQL y MySQL).
Condiciones de Ejecución: Se debe seleccionar el SGBD y los tipos de datos asociados de cada atributo que compone el modelo de datos**.** Luego introducir el nombre del servidor de BD, usuario, contraseña y puerto (para el caso del SGBD PostgreSQL y MySQL).
Resultado Esperado: Fue exportado satisfactoriamente el modelo de datos deseado.
Evaluación de la Prueba: Satisfactoria

Tabla 3.4 CP3_ Exportar el modelo físico normalizado al SGBD seleccionado por el usuario

Prueba de Aceptación para el Caso de Uso: Calcular Clave Primaria

Caso de Prueba de Aceptación (CPA)
CU: Calcular Clave Primaria
Nombre del CPA: CP4_ Calcular las Claves Primarias de una relación.
Descripción: El CP4 inicia cuando se desea obtener la(s) clave(s) primaria(s) de una relación del modelo de datos introducido. Para realizarlo el usuario selecciona el nombre de la relación que pertenece al modelo donde desea calcular el conjunto clave, una vez seleccionada la relación, entonces ya es capaz de calcularla automáticamente.
Condiciones de Ejecución: Se debe seleccionar el nombre de la relación.
Resultado Esperado: Fue calculada correctamente la(s) clave(s) primaría(s) de la relación seleccionada.
Evaluación de la Prueba: Satisfactoria.

Tabla 3.5 CP4_ Calcular las Claves Primarias de una relación

3. 4 ANÁLISIS DE VIABILIDAD DE LA PROPUESTA

La fase de análisis de viabilidad comprende tres dimensiones:

Viabilidad Económica: Desde el punto de vista económico no hay limitaciones para desarrollar de manera satisfactoria nuestra investigación. Los recursos que necesitamos están al alcance de

cualquier universidad cubana y los beneficios monetarios que recibiríamos por la investigación, si fuéramos a venderla en el mercado internacional, harían que fuese muy rentable.

Los recursos materiales e imprescindibles que necesitamos son:

- ✓ Una computadora con requerimientos de Hardware que oscilan alrededor de la media de los requerimientos Hardware que tienen las computadoras en nuestras universidades.
- ✓ Los siguientes software: *Microsoft .Net Framework 3.5* y *SharpDevelop 3.0.*

Viabilidad Tecnológica: La tecnología que proponemos está bien estandarizada en el mercado internacional. Existen diferentes versiones de la herramienta SharpDevelop que están completamente insertados en el Mundo Actual.

En nuestra universidad muchas facultades utilizan el lenguaje C# sobre el IDE Microsoft Visual Studio.NET, donde el entorno de trabajo es muy similar al SharpDevelop 3.0, por lo que podemos constar con personal calificado para operar dicha tecnología y la infraestructura de apoyo necesaria.

Viabilidad Organizacional: La solución que proponemos está acorde con la cultura organizacional que existe en nuestra universidad, por lo que no habrá problemas con la adaptación a la misma

3. 5 CONCLUSIONES DEL CAPÍTULO

Al concluir este capítulo, después de mencionar las consideraciones que se tuvieron en cuenta para la selección de las herramientas que se utilizaron para desarrollar el sistema, mostrar de manera general, organizados por paquetes la arquitectura que fue usada para su implementación, hacer un pequeño análisis de la viabilidad económica, tecnológica y comercial de la propuesta, llevar a cabo un conjunto de pruebas realizadas al sistema mediantes diferentes casos de estudios seleccionados, y analizar los resultados obtenidos en la encuesta aplicada en la UCI podemos concluir que:

- ✓ La organización en paquetes de los elementos desarrollados en "SINORGES" permite un mejor encapsulamiento de sus funcionalidades y un proceso de implementación más organizado y productivo.

- ✓ Los resultados que se obtuvieron de la encuesta realizada demuestran la validez de la situación problemática planteada por el autor de la investigación y evidencia la importancia que requiere la integración del proceso de normalización de BDR con los SGBD.

- ✓ Existe un marcado desconocimiento de las herramientas que permiten normalizar modelos de datos en nuestra universidad, los cuales sin duda está influenciado por la pobre vinculación de dichas herramientas al proceso productivo.

Conclusiones Generales

Conclusiones Generales

Al concluir el desarrollo de la presente investigación, se ha llegado a las siguientes conclusiones:

- ✓ La integración del proceso de normalización con los SGBD permite desarrollar modelos robustos, eficientes, con mayor nivel de flexibilidad ante cambios introducidos en el negocio y favorece la interrelación de la etapa de diseño con la etapa de implementación del modelo de datos, demostrándose la importancia de la normalización en el entorno productivo.

- ✓ La integración del proceso de normalización con los SGBD contribuye a la disminución del tiempo de desarrollo de software en soluciones informáticas que manipulan un gran volumen de información.

- ✓ La visualización de los algoritmos utilizados en el proceso de normalización de bases de datos relacionales, a través del sistema informático "SINORGES", contribuye a mejorar la asimilación de los contenidos teóricos de la asignatura "Sistemas de Bases de Datos" que se imparte en las carreras del perfil informático, particularmente en la UCI.

- ✓ El sistema "SINORGES" que se desarrolló puede utilizarse en cualquier empresa productora de software del país y en universidades cubanas donde se estudien carreras con un perfil informático.

Recomendaciones

Recomendaciones

Para dar continuidad al trabajo realizado el autor de la presente investigación se propone las siguientes recomendaciones:

- ✓ Agregarle nuevas funcionalidades al sistema que permitan elevar sus potencialidades de integración con diversas herramientas de desarrollo de software:
  - Generar el modelo de clases del modelo de base de datos normalizado para los diferentes lenguajes de programación como: Java, C++, C#.
  - Generar código HTML y XML del modelo de bases de datos normalizado.
  - Integrarse con el SGBD Oracle.
- ✓ Incorporar un módulo que permita el diseño del modelo conceptual del negocio (diagrama entidad-relación) y su transformación en el modelo relacional equivalente.
- ✓ Desarrollar un ***plugin***[1] sobre un entorno totalmente libre para el SGBD PostgreSQL, que cumpla con los requerimientos funcionales del sistema propuesto.
- ✓ Utilizar el sistema SINORGES en el proceso productivo y educativo de la UCI.

1 Es un complemento, conector o extensión. Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

Referencias Bibliográficas

1. **MANCOMUN. 24 de 07 de 2009.** *Centro de Referencias a Servicios de Software Libre.* http://www.mancomun.org/es

2. **César Domínguez Pérez, et al. (8-10 julio 2009).** *¿Todavía interesa normalizar bases de datos?.* XV JENUI. Barcelona. *h*ttp://upcommons.upc.edu/revistes/bitstream/2099/7884/6/p122.pdf

3. **Embarcadero Technologies. 30 de enero de 2009.** http://www.soluciones-ag.com/pdf_productos/Spanish_ER-Studio_Datasheet_2009.pdf

4. **«Proyecto Estratégico (2008-2012).»** (Selección de Contenidos – Noviembre 2008).

5. **Georgiev, Nikolay. September 2008.** *A Web-Based Environment for Learning Normalization of Relational Database Schemata.* Umea University. Department of Computing Science.

6. ***et al. (Abril de 2008)*** «*Cátedra de Base de Datos. Facultad de Informática. Universidad de Morón* .». http://bdatos.wordpress.com/base-de-conocimiento/oracle-perfomance/oracle-considerando-la-introduccion-de-redundancia-desnormalizacion

7. ***Ramírez, E. 6 de marzo de 2008.*** *«10 Mejores Prácticas para la Generación de Software.»*. SG Software Guru. Obtenido de http://www.sg.com.mx/content/view/684/99999999/

8. **Blog de Web a Medida. 08 de enero de 2008**. *Potenciado por Joomla.* http://webamedida.net

9. **Karakaya, Ali Yazici & Ziya. ICCS 2007.** *JMathNorm: A Database Normalization Tool Using Mathematica.* pp. 186–193, Springer-Verlag Berlin Heidelberg. Vol. Part II.

10. **Carlos García González, A. R. 2007.** *Diseño y validación estructural de esquemas conceptuales utilizando una herramienta CASE***.** Revista Cubana de Ciencias Informáticas , 1 (4), 72-81.

11. **Herrera, Norma. 2006.** *Organización de archivos y Bases de Datos I.* Argentina. Universidad Nacional de San Luis.

12. **Merril, P. 2006.** *Computers in education (Allyn & Bacon ed. Vol. 1):*128 p.

13. **Basik, J. et al. (2006).** *SV in teaching at Brown University. Machine learning, 15*.

14. **Yazici, A. & Karakaya, Z. 2006.** *Normalizing Relational Database Schemas Using Mathematica.* LNCS. Springer-Verlag Berlin Heidelberg . Vol.3992 (375-382).

15. **Hsiang-Jui Kung, Hui-Lien Tung. 2006** *A WEB-BASED TOOL to enhance teaching/learning database normalization.* Southern Association for Information Systems Conference .

16. **Martín, Oscar. 2006.** *Documentación de DBDesign Tools.* [En línea] http://sourceforge.net/projects/dbdesigntools/

17. **MITROVIC, Antonija. 2005.** *The Effect of Explaining on Learning: a Case Study with a Data Normalization Tutor*. University of Canterbury, New Zealand. Intelligent Computer Tutoring Group Department of Computer Science and Software Engineering. http://www.cosc.canterbury.ac.nz/tanja.mitrovic/NORMIT-AIED05.pdf

18. **Cabrera, Victoria Espinosa. Agosto del 2004**. Manejo de Información y Datos Numéricos, ITESO. [En línea] **http://iteso.mx/~alep/la_evolucion_historica.doc**

19. **Marzo de 2004.** «Sistemas de Gestión de Bases de Datos Relacionales.» http://www.elultimohogar.org/libros/Informatica/

20. **Batini, Carlos. 2004.** *Diseño conceptual de bases de datos.* pp. 3-4.

21. **2004.** Historia del Libro y Las Bibliotecas Antes de la Aparición de la Imprenta. [En línea] http://web.usal.es/~alar/Bibweb/Temario/Histlib.PDF

22. **Saavedra, Oscar. Octubre 2003** «*Las Normas de Competencia Informativa en Bibliotecas Académicas*. EBSCO Information Services.».

23. **Peter Rob, Carlos Coronel. 2003.** *Sistemas de bases de datos: diseño, implementación y administración.* 5ta edición. pp 7, 21-22, 30-31, 196-197

24. **Quiroz, Javier. 2003.** *«El modelo relacional de bases de datos.»* Boletín de Política Informática, nº 6.

25. **Kroenke, David M. 2003.** *Procesamiento de bases de datos: fundamentos, diseño e implementación.* Octava Edición.

26. **Graham, C. 6 de mayo 2002.** *"DBMS Software Market: Flat but Not Calm"* Gartner Group (www.gartner.com).

27. **Orallo, José Hernández. mayo 2002.** *La Disciplina de los Sistemas de Bases de Datos. Historia, Situación Actual y Perspectivas.* Dep. de Sistemas Informaticos y Computación.

28. **MITROVIC, Antonija. 2002.** *Supporting Self-Explanation in a Data Normalization Tutor.* University of Canterbury. Private Bag 4800, Christchurch, New Zealand .Intelligent Computer Tutoring Group. http://www.cs.usyd.edu.au/~aied/vol9/vol9_roger_Mitrovic.pdf

29. **SMITT. 2002.** *"Slowed DBMS Market Means Tight Competition"* http://www.advisor.com/Articles.nsf/aid/SMITT247

30. **IDG. 23 de mayo del 2001.** *"Oracle extends lead in database sales"* IDG News Service. http://www.itworld.com/AppDev/119/DataquestOracleexte413/

31. **Marqués, Mercedes. 10 de febrero del 2001.** *«Apuntes de Ficheros y Bases de Datos.»*

32. **Date, Christopher J. 2001.** *Introducción a los Sistemas de Bases de Datos.* Séptima Edición. pp 9-10, 845, 331-333, 350-351

33. **Leavitt, N. August 2000.** *"Whatever Happened to Object-Oriented Databases?"* Computer. Vol. 33

34. **Lawrence, A et al. (2000).** Empirically Evaluating the Use of Animations to Teach Algoritms.Georgia Institute of Technology.

35. **Jacobson, I et al. (2000).** *"El Proceso Unificado de Desarrollo de software"* . Addison-Wesley.

36. **María Mato Gracía, Rosa. Octubre 1999.** *Diseño de Bases de Datos.* pp 2

37. **Stonebraker, M et al. 1999.** *"Object-Relational DBMSs: Tracking the Next Great Wave".* Second Edition.

38. **Miguel, A et al. (1999).** "*Diseño de Bases de Datos Relacionales".* España: RA-MA Editorial.

39. **Miguel, A., M Piattini. 1997.** *"Fundamentos y modelos de bases de datos" Ra-Ma.* Madrid, España.

40. **Graham, Ian. 1996.** *Métodos Orientados a Objetos.* 2da Edición. pp 180.

41. **Carlos Batini, et al. (1992).** *Conceptual Database Design: An Entity – Relationship Approach.* Massachusetts.E.U.A : Addison-Wesley Publishing Company.

42. **Ullman, Jeffrey D. 1988.** *Principles of Database and Knowledge Base Systems.* Vol. I Computers Science Press.

43. **Baizán, María Covadonga Fernández. 1987.** *El modelo racional de datos: De los fundamentos a los modelos deductivos.* Ediciones Díaz de Santos.

44. **(CODASYL), April 1971.** *Codasyl Data Base Task Group Report, ACM.* New York.

45. *Enciclopedia libre de C# y .NET. C# Online.NET (CSharp-Online.NET).* http://es.csharp-online.net

Anexos

ANEXOS I. ESPECIFICACIÓN DE LOS REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES DEL SISTEMA "SINORGES"

REQUISITOS FUNCIONALES

1. Crear una nueva relación
2. Insertar el conjunto de atributos
3. Insertar el conjunto de DF
4. Eliminar una o todas las relaciones.
5. Eliminar atributo(s).
6. Eliminar dependencia(s) funcional(es).
7. Modificar o editar el conjunto de atributos y DF.
8. Calcular el Cierre de Descriptores de una relación.
9. Calcular las Claves Primarias de una relación.
10. Determinar el Recubrimiento Minimal del conjunto de DF de una relación.
11. Determinar si una DF pertenece al conjunto de DF de una relación.
12. Determinar si dos conjunto de DF son equivalentes.
13. Determinar en que forma normal se encuentra una relación o el modelo relacional.
14. Normalizar modelo de relación a 2FN por descomposición.
15. Normalizar modelo de relación a 3FN por descomposición.
16. Normalizar modelo de relación a 3FN por síntesis de Bernstein.
17. Normalizar modelo de relación a FNBC por descomposición.
18. Visualización del algoritmo del Cálculo de Cierre de Descriptores de una relación.
19. Visualización del algoritmo del Cálculo de las Claves Primarias de una relación.
20. Visualización del algoritmo del Cálculo del Recubrimiento Minimal del conjunto de DF de una relación.
21. Visualización del algoritmo que determina la equivalencia entre dos conjuntos de DF.
22. Crear scripts en lenguaje SQL, asociado al SGBD seleccionado por el usuario.
23. Exportar el modelo físico de BD normalizado al SGBD seleccionado por el usuario.
24. Salvar el modelo relacional en un fichero con extensión propia.
25. Abrir un modelo relacional generado por el sistema.

26. Importar modelos de datos generados por los SGBD que son compatibles con el sistema y por la herramienta “Embarcadero ER/ Studio”.

REQUISITOS NO FUNCIONALES

Los requerimientos no funcionales garantizan propiedades o cualidades del producto a desarrollar, las que hacen al producto atractivo, usable, rápido o confiable.

Requerimientos de apariencia o interfaz externa

El sistema “SINORGES” brinda una interfaz amigable, legible, simple de usar, interactivo, con un diseño semejante a las herramientas CASE utilizadas para el diseño de modelos conceptuales de bases de datos relacionales.

Requerimientos de usabilidad

El sistema “SINORGES” puede ser usado por estudiantes que hayan recibido la asignatura “Sistema de Bases de Datos”, por diseñadores o especialistas de BD, pues deben tener una noción acerca de la teoría de BDR, específicamente en los procesos de normalización, para los usuarios finales con estas características la utilización y la apropiación del conocimiento brindado en el módulo de aprendizaje del sistema será inmediato. El usuario debe tener algún conocimiento básico de la sintaxis del lenguaje de consulta, así como, saber interactuar con los diferentes SGBD que se abordan en la propuesta.

Requerimientos de Rendimiento

El sistema “SINORGES” es totalmente interactivo, por lo que requiere un tiempo de respuesta aceptable a las acciones del usuario (normalización, visualización paso a paso, interacción con los SGBD, etc.)

Requerimientos de Portabilidad

La primera característica del sistema “SINORGES” que permite asegurar la portabilidad del mismo es el estar implementado bajo el paradigma de Programación Orientada a Objetos. Es un estilo de trabajo que garantiza, entre otras cosas, la portabilidad de los sistemas que sean desarrollados bajo esta técnica.

Al estar desarrollado sobre la plataforma .NET, en el lenguaje de programación C#, el cual según Microsoft es un lenguaje de programación con la potencia de C, la productividad de Visual Basic y la

elegancia de Java, brinda mayor potencialidad en todos los aspectos que favorecen la portabilidad (CSharp-Online.NET):

Independencia de lenguaje

Todos los lenguajes que conforman los estándares .NET podrán inter-operar entre sí, de forma totalmente transparente, las clases podrán ser heredadas entre unos lenguajes y otros, y se podrá disfrutar de polimorfismo entre lenguajes. Por ejemplo, si se tiene una clase en C#, esta clase podrá ser heredada y utilizada en Visual Basic o JScript o en cualquier lenguaje .NET. Todo es posible por medio de una de las características de .NET llamado Common Type System (CTS). También tiene la cualidad de poder incluir más lenguajes a la plataforma.

Multiplataforma

Cuando un programa es compilado, no es compilado en un archivo ejecutable sino en un lenguaje intermedio llamado "Lenguaje Intermedio" (IL) el cual podrá ser ejecutado por el Common Language Runtime (CLR) en la plataforma en que el CLR esté disponible. Los sistemas operativos Windows XP o superiores incluyen el CLR nativamente y SuSE Linux 9.3 o superior planea incorporar el CLR (Mono) en su distribución, lo que quiere decir que un programa .NET podrá ser compilado y ejecutado en cualquiera de estas plataformas, o en cualquier plataforma que incluya un CLR.

Requisitos de soporte

El sistema "SINORGES" fue diseñado teniendo en cuenta la posibilidad de mejorar sus funciones, realización de cambios y agregación de nuevos servicios.

Requisitos de ayuda y documentación en línea

El sistema "SINORGES" cuenta en el módulo de aprendizaje con descripciones de ayuda que especifican en cada momento los pasos u operaciones que se realizan por cada algoritmo representado, mostrando además los resultados que se van adquiriendo.

Requisitos de Interoperabilidad

El sistema "SINORGES" es capaz de conectarse a los siguientes SGBD: *PostgreSQL*, *Microsoft SQL Server*, *MySQL*, a través del módulo "generación de scripts", cargando automáticamente la BD

normalizada del esquema relacional introducido por el usuario. Además permite cargar modelos físicos de BD realizados en los SGBD mencionados y en la herramienta CASE "Embarcadero ER / Studio".

Requisitos de software

"SINORGES" está implementado en el lenguaje *C#* sobre el sistema operativo *Windows*, se utilizó como Entorno de Desarrollo Integrado (IDE) de software para su realización ***SharpDevelop 3.0***. Para poder ser usado sólo requiere *Microsoft .Net Framework 3.5*.

Requisitos de hardware

Se limita a las restricciones de hardware para la instalación de Microsoft .Net Framework 3.5:

- ✓ Sistemas operativos compatibles: Windows Server 2003; Windows Server 2008; Windows Vista; Windows XP
- ✓ Procesador: procesador Pentium a 400 MHz o equivalente (mínimo); procesador Pentium a 1 GHz o equivalente (recomendado)
- ✓ RAM: 96 MB (mínimo); 256 MB (recomendado)
- ✓ Disco duro: se pueden necesitar hasta 500 MB de espacio disponible
- ✓ Unidad de CD o DVD: no se necesita
- ✓ Pantalla: 800 x 600, 256 colores (mínimo); color de alta densidad de 1024 x 768, 32 bits (recomendado).

ANEXOS II. ENCUESTA A ESPECIALISTAS DE LA COMPUTACIÓN EN LA UCI.

Con el objetivo de lograr resultados reales en la presente encuesta, les pedimos responder el siguiente cuestionario con la sinceridad que amerita.

Seleccione la categoría a la que pertenece.

___ Jefes de Polos
___Líder de Proyectos
___Diseñadores de Bases de Datos
___Otros

Preguntas de la Encuesta:

1. ¿Considera usted necesario la aplicación de las técnicas de normalización de bases de datos relacionales en el diseño de un sistema informático?
   _____ Siempre _____En ocasiones _____Nunca

2. ¿Considera usted necesario la ***integración***[1] del objetivo reflejado en la pregunta #1 con los sistemas gestores de bases de datos más utilizados?
   ____ Sí ____ No

3. ¿Conoce usted alguna herramienta informática capaz de automatizar el proceso de normalización de bases de datos relacionales? En caso afirmativo menciónelas.

   ____ Sí ____ He oído hablar de alguna(s) ____ No

1. _____ 4. _____
2. _____ 5._____
3. _____ 6. _____

4. ¿Conoce usted alguna herramienta informática que permita modelar bases de datos relacionales y al mismo tiempo integrarse con algunos sistemas gestores de bases de datos relacionales? En caso afirmativo menciónelas.

   ____ Sí ____ He oído hablar de alguna(s) ____ No

1. _____ 4. _____
2. _____ 5. _____
3. _____ 6. _____

5. ¿Conoce usted alguna herramienta informática que permita integrar los objetivos resaltados en la pregunta # 1 y # 2? En caso afirmativo menciónelas.
   ____ Sí ____ He oído hablar de alguna(s) ____ No

1. _____ 4. _____
2. _____ 5. _____
3. _____ 6. _____

1 Definiendo integración en este contexto por: **"Capacidad de fundir los beneficios de la normalización con las potencialidades de los sistemas gestores de bases de datos".**

Glosario de Términos

SGI. - **Sistemas de Gestión de la Información**. Es una colección de datos debidamente recopilados y estructurados, que proporcionan información sobre una parcela de la realidad.

SGBD. - **Sistemas Gestores o de Gestión de Bases de Datos**. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

BDR.- **Bases de Datos Relacional**. Es una base de datos que se percibe por los usuarios como una colección de tablas.

Complejidad Temporal.- Es una métrica teórica que se aplica a los algoritmos para obtener la idea del tiempo que consumen para resolver un problema.

RUP.- **Rational Unified Process (Proceso Unificado del Desarrollo de Software)**. Metodología de desarrollo de software creada por IBM Rational para el desarrollo y construcción de software basado íntegramente en el Lenguaje Unificado de Modelado (UML) que soporte a la metodología.

Herramientas CASE.- **Computer Aided Software Engineering.** Aplicaciones Informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

IBM.- **International Business Machines**. El mayor fabricante de ordenadores del mundo. IBM es el inventor del PC y con sus sistemas medios (AS/400) y sus grandes ordenadores (mainframes) han revolucionado el mundo de la empresa.

CLR.- **Common Language Runtime.** Es el componente de máquina virtual de la plataforma .Net de Microsoft. Es la implementación del estándar Common Language Infrastructure (CLI) que define un ambiente de ejecución para los códigos de los programas. El CLR ejecuta una forma de código intermedio (bytecode) llamada: Common Intermediate Language (CIL).

MSIL. - **Common Intermediate Language** (CIL, pronunciado "sil" o "kil"), anteriormente llamado Microsoft Intermediate Language o MSIL es el lenguaje de programación legible por humanos de más bajo nivel en el Common Language Infrastructure y en el .NET Framework. Los lenguajes del .NET Framework compilan a CIL. CIL es un lenguaje ensamblador orientado a objetos, y está basado en pilas Es ejecutado por una máquina virtual.