

ERECASE: UNA HERRAMIENTA DE AYUDA A LA MODELACIÓN DE ESQUEMAS CONCEPTUALES GLOBALES.

Autores: MSc. Abel Rodríguez Morffí ¹, Dra. Luisa González González ², Lic. Leonel Cabrera Tamayo, MSc. Alberto Morell Pérez ³. Universidad Central “Marta Abreu” de Las Villas.

¹ arm@cei.uclv.edu.cu, ² luisagon@cei.uclv.edu.cu, ³ amorellp@cei.uclv.edu.cu.

RESUMEN

Con el objetivo de dar un enfoque más sistemático a la resolución de problemas de modelación de esquemas conceptuales globales mediante el empleo de herramientas CASE se usa ampliamente el modelo Entidad-Relación Extendido (ERE). En el presente trabajo se hace un estudio del modelo ER y sus extensiones, y las herramientas más populares en Cuba para el diseño de Bases de Datos que usan este modelo. Se diseñó e implementó una herramienta que se incluye en un sistema integrado de ayuda al diseño de Bases de Datos Distribuidas, desarrollada por el grupo de investigación al que pertenecen los autores de este trabajo, que brinda un amplio conjunto de construcciones del modelo ERE y aplica diferentes validaciones a éste.

PALABRAS CLAVES

Modelación conceptual, herramientas CASE.

I. INTRODUCCIÓN

En los últimos tiempos ha existido un gran desarrollo en las aplicaciones de Bases de Batos, por esto su diseño se ha convertido en una actividad muy común realizada por cualquier persona aunque no siempre con la preparación necesaria. El diseño de bases de datos está fuertemente influido por la elección de un modelo adecuado para representar los datos. Estos modelos, al igual que los lenguajes de programación, tienen un conjunto determinado de construcciones que son usadas para la descripción de datos y funciones. Además, estas construcciones tienen una representación gráfica que permite al diseñador construir diagramas e imágenes que son más fáciles de leer y comprender. Se ha creado una gran variedad de herramientas de diseño por computadoras, muchas de ellas soportan una representación gráfica de los esquemas de datos y funciones. Entre las capacidades gráficas que las mismas ofrecen se encuentran: edición del modelo usando el ratón, el uso de múltiples ventanas para seguir diferentes aspectos del proceso de diseño, etc.

Se han propuesto varios modelos conceptuales como alternativas, entre estos se encuentran los modelos semánticos de datos, el modelo estructurado y el modelo funcional. Todos estos modelos conceptuales están basados en el uso de mecanismos de abstracción por lo cual es posible definir correspondencias entre ellos. En particular el modelo Entidad-Relación (ER) emergió como la principal estructura formal para la representación conceptual de datos, convirtiéndose y estableciéndose como un estándar internacional. Este modelo es el más usado para el diseño conceptual de Bases de Datos. Fue presentado por Peter Chen en 1976 y se ha hecho muy popular; muchas conferencias y talleres han sido organizadas sobre las aplicaciones del modelo ER en el diseño de Bases de Batos y en el de software en general. En 1988 fue elegido como modelo estándar para los *Information Resource Dictionary Systems (IRDS)* y debido a su gran utilidad, muchas herramientas de diseño de Bases de Datos utilizan sus conceptos.

En los intentos de dar un enfoque más sistemático a la resolución de problemas de modelación se ha enfatizado en la automatización mediante el empleo de herramientas CASE. Si bien es cierto que algunas de ellas adoptan

enfoque más avanzados también lo es que muchas no pasan de ser simples utilidades para dibujar. En general no disponen siquiera de un soporte metodológico, o no son lo suficientemente estrictas en su aplicación, con lo cual el diseñador no encuentra el camino correcto para realizar su tarea [ATK96]. Además, los modelos que generalmente soportan son modelos lógicos, que suelen incluir bastante consideraciones físicas, aunque la notación gráfica empleada sea un dialecto del modelo ER. En este trabajo se incluye un conjunto de construcciones de este modelo para lograr una mejor expresividad en el diagrama, la realización de validaciones en un esquema, la transformación de un esquema al modelo relacional, y la generación de código SQL para la creación de la Base de Datos física en un Sistema Manejador de Bases de Datos (SMBD) determinado. El propósito de la construcción de esta herramienta es su inclusión en el Sistema Integrado de Ayuda al Diseño de Bases de Datos Distribuidas, desarrollado por el grupo de investigación en Bases de Datos del Centro de Estudios de Informática de la Universidad Central “Marta Abreu” de Las Villas, Cuba.

II. MARCO DE DESARROLLO

II.1. Herramientas CASE para la modelación de bases de datos más utilizadas en Cuba.

El término ingeniería de software asistida por computadoras (CASE, del inglés *Computer Aided Software Engineering*) nació como resultado de la necesidad de organizar el desarrollo de sistemas informáticos. Las primeras herramientas CASE creadas datan del año 1982 y fueron enfocadas principalmente en el apoyo a la construcción de diagramas. La idea básica de la tecnología CASE es proporcionar un conjunto integrado de herramientas que automaticen el desarrollo del software y el trabajo de mantenimiento. Se evalúan tres herramientas CASE actuales: DeZign, ERwin y DDS, basándose en las facilidades que brindan, las construcciones del modelo ER que soportan y las validaciones realizadas al modelo.

II.1.1. DeZign, Versión 2.3.1 Octubre 2000: “DeZign for Databases” es una herramienta de desarrollo de Bases de Datos de la compañía Datanamic (llamada Heraut Solutions hasta el 3 de Abril del 2001). En ella el diseñador crea un diagrama ER insertando, de forma gráfica, entidades e interrelaciones y genera automáticamente esquemas SQL para los SMBD más populares. Las construcciones que provee para crear un diagrama son solamente los conjuntos de entidades y las interrelaciones. Para los conjuntos de entidades, DeZign permite especificar su nombre y atributos. En el caso de las interrelaciones, son dibujadas entre las entidades y pueden incluir un nombre, una descripción y las cardinalidades máximas y mínimas. Los modelos creados en DeZign son más bien modelos lógicos que modelos conceptuales ya que las llaves extranjeras son especificadas en el diagrama cuando se crea una interrelación Uno-Muchos, además permite la especificación de índices, elementos que son completamente ajenos al modelo ER. Para los atributos, sólo permitidos para los conjuntos de entidades e interrelaciones Muchos-Muchos, se detallan su dominio o tipo de dato, si es llave primaria, si permite valores nulos, si es requerido, las restricciones a ser chequeadas, etc. En esta herramienta el usuario puede definir dominios (tipos definidos por el usuario) que pueden ser aplicados a los atributos. Cuando se modifica un dominio se actualizan globalmente los atributos asociados a éste, esto permite una forma muy sencilla de estandarizar las características de los datos y modificar el modelo de forma consistente cuando se necesita hacer cambios.

DeZign muestra la información del modelo en varios niveles. Estos niveles de visualización o detalle pueden ser usados en varias etapas de desarrollo del modelo y expresan la información en abstracto o en varios niveles de detalle. Existen cuatro modos de visualización diferentes: nivel de entidad, nivel de llave primaria, nivel de atributo y nivel de llave primaria con atributos únicos. La herramienta soporta los niveles de datos lógicos y físicos de un mismo modelo usando para esto la migración automática de la llave extranjera en tiempo de diseño.

Una modificación a una propiedad de una relación o a una llave primaria afecta automáticamente las llaves extranjeras y puede ser visible, en el diseño, en un modo de visualización más detallado.

“DeZign for Databases” genera automáticamente los esquemas de una Base de Datos de forma directa desde el diagrama ER. Se puede usar el mismo diagrama para generar esquemas para distintos tipos de Bases de Datos. Para esto se debe especificar con anterioridad el SMBD (Oracle, MS Access 95/97/2000, Informix, Sybase, Interbase, DB2, SQL Anywhere, Ingres, MySQL, MS SQL Server, ANSI Level 2, PostgreSQL). Estos esquemas incluyen las sentencias “Create Table” y los detalles sobre la estructura de la tabla. Dependiendo del esquema seleccionado la herramienta generará las llaves primarias, las llaves extranjeras, restricciones de chequeo, restricciones de atributo único, etc. DeZign no realiza ningún tipo de validación del diagrama construido, por lo que a la hora de generar los esquemas de las Bases de Datos pueden existir errores que pasan inadvertidos al diseñador.

Otras facilidades de DeZign son la generación de reportes del diseño y el control de versiones. Los reportes pueden ser generados con distintos niveles de detalle y en formato Html o texto. Con el control de versiones se puede regresar a una versión anterior del diagrama, estas versiones son automáticamente guardadas por DeZign cada vez que se guarda el proyecto.

II.1.2. ERwin, Versión 3.5. Año 1998, es una herramienta de modelación de datos comercializada por Logics Works, Inc. Fue desarrollado para soportar los estándares de modelación IDEF1X (desarrollado por la fuerza aérea de los Estados Unidos y es usado por varias agencias del gobierno, la industria aeroespacial y financiera) e IE (Information Engineering, fue desarrollado por James Martin, Clive Finkelstein y otros y es ampliamente difundido en una variedad de industrias), los cuales son variantes de modelo ER y usan diferentes símbolos para representar las construcciones del modelo. ERwin brinda para la modelación las construcciones de conjuntos de entidades, interrelaciones de identificación y no-identificación, interrelaciones muchos-muchos y subconjuntos. Una interrelación de identificación indica que la llave extranjera pasa a formar parte de la llave primaria del esquema del conjunto de entidades hija y en una de no-identificación que la llave extranjera no pasa a formar parte de la llave primaria. En el caso de los atributos, estos son especificados para cada entidad, se les asigna un dominio que es transformado al correspondiente tipo de dato en el SMBD seleccionado como destino a la hora de generar el esquema. A las interrelaciones no es posible especificarle atributos.

ERwin permite a partir de un mismo diagrama generar esquemas físicos para varios SMBD. Los esquemas son generados usando las sentencias “Create Table” correspondientes a cada SMBD seleccionado y además genera restricciones, vistas, índices, etc. Algunos de los SMBD que soporta son: DB2, ORACLE, Ingres, INFORMIX, SQL Server, SYBASE, dBASE III y IV, Access, Paradox, etc.

También soporta los niveles de datos lógicos y físicos de un mismo modelo, se puede cambiar fácilmente del modelo lógico al físico con sólo seleccionar una opción de la barra de herramientas. El modelo lógico soporta las necesidades del diseñador para representar la información y definir las reglas o restricciones del problema. El modelo físico soporta las necesidades del administrador de Bases de Datos quien se centra en la implementación física del modelo. Además, brinda facilidades para hacer la ingeniería inversa, generando un modelo a partir de una Base de Datos física. Durante este proceso primeramente se captura la información necesaria en la Base de Datos o en un fichero script, como son: tablas, columnas, relaciones, disparadores, procedimientos almacenados, reglas de validación, etc., luego crea el diagrama del modelo físico usando dicha información. Se usa ODBC (Open Database Connectivity) para acceder a estas Bases de Datos.

Al trabajar con ERwin el usuario puede obtener una documentación del sistema a crear que puede ser usado por el equipo de diseño para definir los requerimientos del sistema y para comunicarse entre ellos y con los usuarios finales.

II.1.3. Database Design Studio (DDS), Versión de evaluación 1.09.0 Año 2000, es una herramienta creada por la compañía Chilli Source la cual provee de un ambiente para el diseño, la creación, la consulta y edición de Bases de Datos. El DDS brinda, para la creación del diagrama ER las siguientes construcciones: conjunto de entidades fuertes, conjunto de entidades débiles, interrelaciones binarias, especialización, generalización. Para las interrelaciones ternarias, no proporcionan una construcción especial, pues plantean que este tipo de interrelación

se puede modelar usando una entidad débil conectada a tres entidades. Los atributos son adicionados a cada conjunto de entidades y están restringidas a contener al menos una llave. En el caso de las entidades débiles la llave es opcional, las asociaciones Muchos-Muchos también pueden tener atributos. Los tipos de datos usados para crear los atributos son tipos internos del DDS que a la hora de generar los esquemas son transformados a los tipos específicos de un SMBD determinado. A las interrelaciones se les especifica la cardinalidad máxima y mínima.

El esquema de la Base de Datos diseñada es generado usando las sentencias “Create Table” correspondientes a cada SMBD seleccionado. Además genera otros archivos, dependiendo también del tipo de SMBD, estos son “Load” y “Drop”. Los SMBD que soporta son: ANSI, Ingres, InterBase, Informix, SQLBase, MicroSQL, MSSQL, MSVB, MSVB-Data, Oracle, Java, Access, MySQL.

El DDS brinda otras facilidades al diseñador: permite realizar la documentación del modelo a través de reportes tanto del modelo conceptual como del lógico. Para los diseñadores que prefieran trabajar con Java tiene la opción de generar el código de la Base de Datos en este lenguaje y especificar una máquina virtual para compilarlo y ejecutarlo desde el mismo DDS. Una herramienta muy útil del DDS es el DataStore, una Base de Datos creada y mantenida por el DDS cuyo objetivo principal es proveer al diseñador de un ambiente enmarcado completamente en la herramienta para darle valores a las tablas de las Bases de Datos en construcción y hacer pruebas de su diseño utilizando el SQLConsole, una aplicación que viene junto con el DDS, que permite acceder a cualquier Base de Datos usando una conexión ODBC y hacer consultas sobre la misma.

Las herramientas descritas anteriormente constituyen un claro ejemplo del estado actual de las herramientas de diseño de Bases de Datos en el mercado cubano del software. Como se pudo apreciar, estas herramientas más allá de permitir la generación de código para múltiples Manejadores de Bases de Datos y brindar una gran cantidad de reportes, en los aspectos gráficos, entiéndase representación del diagrama ER, en la interfaz con el usuario y en la aplicación de validaciones a los esquemas, pueden ser mejorados.

En el caso de las construcciones que ofrecen, algunos de ellos se limitan a permitir las entidades y las interrelaciones (por ejemplo el DeZign), no dejando la posibilidad de agregarle atributos a estas últimas, salvo en algunas excepciones. El DDS es el que más se acerca a un modelo ER característico, pero aún le faltan varias construcciones que le den mayor expresividad al modelo y que le permitan al diseñador representar con más facilidad las particularidades de su problema. La inclusión de elementos de los modelos lógicos y físicos en el diseño pueden confundir a un diseñador inexperto que quizás sólo tenga conocimientos básicos del modelo ER y no le sea sencillo enfrentarse a conceptos como el de migración de llaves o índices.

Aunque el propósito principal de la herramienta de diseño creada como parte de este trabajo es el de ser insertada en una herramienta más general para el diseño de Bases de Datos Distribuidas, las facilidades con que cuenta, como son: contar con un amplio conjunto de construcciones del modelo ER Extendido que permitan obtener una mayor expresividad en el esquema conceptual; que sea capaz de validar la estructura del esquema y las restricciones de integridad del modelo; y que esté dotada de una interfaz sencilla, cómoda y adaptable a las características o gustos de un usuario; le permiten ser usada como una herramienta independiente, y llegar a ser de mucha utilidad tanto a los diseñadores de Bases de Datos como a cualquier usuario en general.

III. ARQUITECTURA DE LA HERRAMIENTA PROPUESTA

Las herramientas avanzadas de diseño deben satisfacer los siguientes requerimientos:

1. Interfaz de usuario avanzada y poderosa que debe ser consistente en la pantalla, reconocer diseños anteriores y usar un lenguaje gráfico.

2. Flexibilidad y alcance global son rasgos importantes de los sistemas de diseño. La herramienta debe apoyar el proceso de diseño completo, tener integrados editores para los diseños del esquema y del flujo de datos, analizadores, sintetizadores y transformadores.
3. Robusta, estar bien integrada y tener un eficiente apoyo teórico. Debe ser eficiente en la adquisición de las semánticas, no sólo de los gráficos. Las decisiones de diseño pueden ser discutidas con el usuario. Se deben generar alternativas. El sistema tiene la capacidad de detectar la falta de información y ayudar a arreglar resultados erróneos. También puede mostrar diferentes versiones del mismo esquema y trabajar de igual forma con uno malo que con uno bueno.
4. Extensible en múltiples direcciones.

La herramienta no requiere que el usuario entienda la teoría, las restricciones de la implementación y los problemas de programación al realizar un diseño. Un diseñador inexperto puede crear un diseño correcto usando el sistema.

III.1. Módulos de la herramienta de diseño:

Editor gráfico: Le permite al usuario especificar gráficamente la estructura de una aplicación, las restricciones que son válidas en la aplicación dada (y los procesos, operaciones y transacciones que sean necesarios). Esta extensión requiere un soporte para gráficos avanzado y sencillo. El editor permite al diseñador representar la información del diseño, éste desarrolla la estructura, las semánticas estáticas y dinámicas, y las operaciones de la aplicación.

Personalización: La interfaz de usuario es adaptada al nivel, habilidades e intenciones del diseñador.

Persistencia: El sistema de diseño almacena versiones del diseño actual, de anteriores o ejemplos. Estos pueden ser parciales o totalmente incluidos dentro del diseño actual y pueden ser usados para cambiar partes de éste.

Traductor: Esta componente traduce los resultados del proceso de diseño a un modelo lógico o físico y puede generar determinado código que puede ser usado en otras aplicaciones.

La arquitectura interna de ERECASE consta de tres módulos fundamentales; el editor gráfico, el validador y el traductor. Estos módulos interactúan entre sí durante el proceso de diseño de una Base de Datos. Véase la figura 1.

a, El módulo editor gráfico tiene tres componentes fundamentales:

- ✧ Plataforma del diagrama ER. Su función es contener las construcciones del diagrama ER y brindar el espacio necesario para su creación.
- ✧ Editor de la plataforma. Es el encargado de manipular las funciones generales del diseño como la creación de nuevas construcciones, el copiado, pegado y eliminación; dibuja la rejilla de la plataforma para la ubicación de las construcciones; también es el encargado de notificar los eventos a los editores de las construcciones y actualizar el explorador del modelo.
- ✧ Editores de los objetos gráficos. Son los que realizan las funciones gráficas específicas para cada construcción, entiéndase mover, redimensionar, invocar el editor de propiedades correspondiente, construir el menú contextual, etc.

b, El validador del modelo ER. Es el encargado de aplicar las reglas de validación a un esquema conceptual determinado determinando si éste es correcto o no.

c, Las partes fundamentales del módulo traductor son:

- ✧ Transformador del modelo ER al relacional. Genera a partir de un esquema conceptual previamente validado y correcto el modelo lógico correspondiente.

- ❖ Generador de código SQL. Construye las sentencias de creación de tablas a partir de un modelo relacional previamente generado.

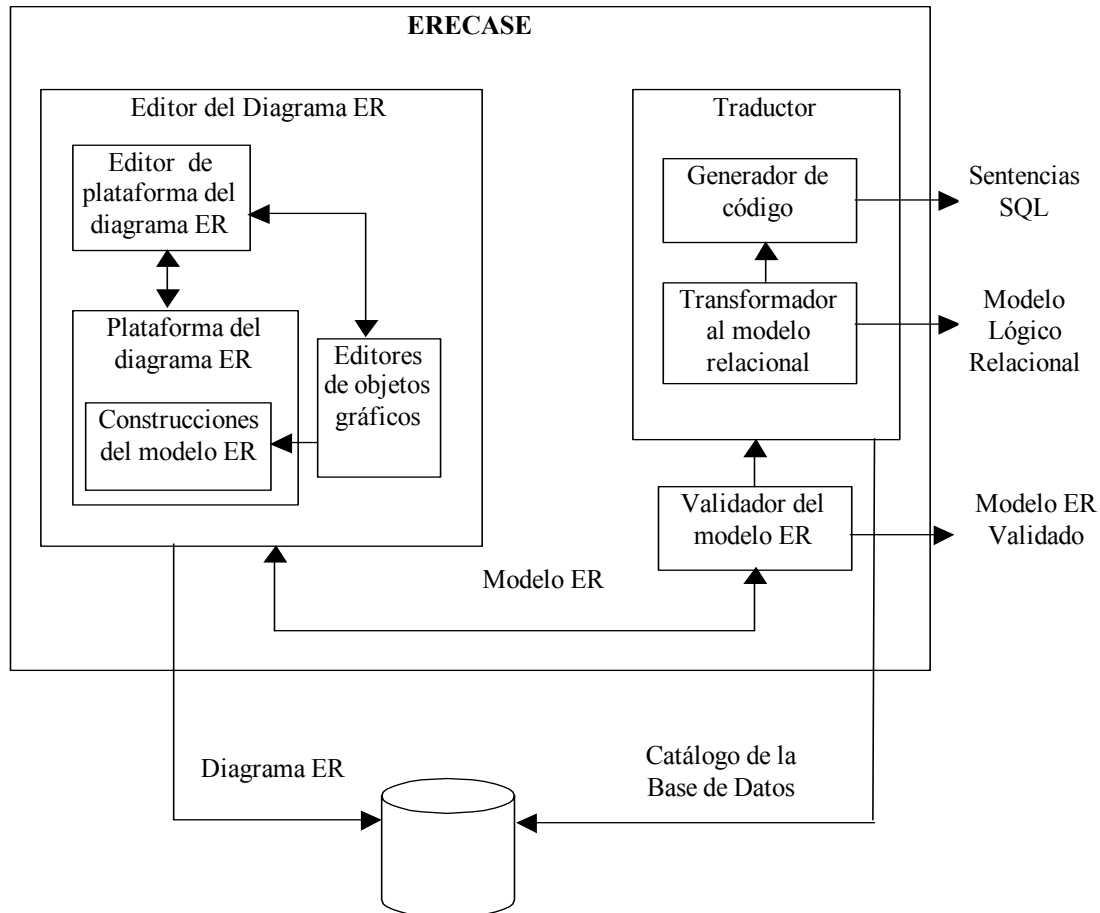


Figura 1. Arquitectura interna de ERECASE

III.2. Construcciones que ofrece ERECASE para la creación del diagrama ER.

ERECASE es una herramienta gráfica que apoya la creación de diagramas Entidad-Relación, la forma más ampliamente aceptada de definir el modelo conceptual de un sistema de Bases de Datos, y la transformación automática del modelo conceptual al modelo lógico equivalente (modelo relacional). A través del uso de interfaces gráficas, la herramienta le permite al usuario editar las propiedades de los conjuntos de entidades, interrelaciones y otras construcciones del modelo ER Extendido (ERE).

Para la creación del modelo conceptual se ofrece en esta herramienta un conjunto variado de construcciones del modelo ERE. Dichas construcciones son:

- ❖ Conjunto de entidades fuertes y débiles.
- ❖ Asociaciones recursivas, binarias, ternarias y múltiples.
- ❖ Interrelaciones ISA, ID, TypeOf.
- ❖ Generalizaciones.
- ❖ Agregaciones.

Para representar el comportamiento de las construcciones se diseñaron dos jerarquías de clases paralelas, en una de estas jerarquías se representa el comportamiento interno de las construcciones o las características propias del modelo ER; en la otra se representa el comportamiento visual de las construcciones o las características del diagrama ER. Para el uso de estas dos jerarquías por otros módulos sin tener que conocer las características propias de cada una, se crearon un conjunto de interfaces que resumen el comportamiento de las construcciones del modelo ER, cada clase implementa las interfaces respectivas y así para otros módulos una clase, ya sea visual o no, tiene el mismo comportamiento. Una clase visual delega la implementación de los métodos de las interfaces del modelo ER que soporta, en la clase correspondiente que representa su comportamiento interno

ERECASE tiene la capacidad de determinar si un esquema conceptual determinado es correcto o no, para esto aplica varios tipos de chequeos al esquema como son los de: unicidad de nombres, llave primaria, identificación de conjuntos de entidades débiles, conjunto de entidades huérfanas, validez estructural de las asociaciones recursivas, y ciclos de pertenencia de conjuntos de entidades débiles. La validez estructural de un diagrama ER consiste en que las construcciones de un diagrama ER determinado no contengan contradicciones entre ellas. Un diagrama ER con al menos una contradicción de este tipo es estructuralmente inválido. Un diagrama ER es estructuralmente válido solamente cuando consideraciones simultáneas de todas las restricciones estructurales impuestas en el modelo no implican una inconsistencia lógica en cualquiera de sus posibles estados, es semánticamente válido solamente cuando una y cada una de las interrelaciones representan el concepto del modelador del dominio de aplicación [DUL99]. Como la validez semántica depende de la aplicación no se pueden definir criterios de validación generales.

Para el caso de las asociaciones recursivas, su validez estructural es abordada en [DUL99] en el que se exponen y demuestran varias reglas, teoremas y corolarios que permiten definir si una asociación recursiva es estructuralmente válida o no. La herramienta que se construyó para este trabajo valida las asociaciones recursivas utilizando estas reglas.

Los conjuntos de entidades débiles (ED) en el diagrama ER pueden ser transformados de dos formas distintas, como atributo de otra entidad o permanecer como una entidad del modelo. En el primer caso la ED pasa a ser atributo de la entidad más cercana, ya sea directa o indirectamente, que se haya decidido permanezca como una entidad. En el segundo caso, las llaves primarias de las entidades padre de la ED pasan a formar parte de la llave de esta. La semántica y manipulación de los conjuntos de entidades débiles puede complicarse cuando tienen más de un propietario, o cuando están interrelacionados con otros conjuntos de entidades, o cuando el, o los propietarios, son conjuntos de entidades débiles. Como consecuencia, un modelo ER puede incluir complicadas redes de pertenencia. Por ejemplo, pueden haber múltiples caminos de identificación entre un conjunto de entidades débiles y su propietario, pueden existir ciclos de pertenencia, etc. La comprensión de estas estructuras se convierte en algo muy difícil, la semántica poco evidente torna el diagrama incoherente y muchas herramientas de diseño que verifican la consistencia o validez de los esquemas no soportan estos casos, al igual ocurre a la hora de transformar al modelo relacional. Por esto, en este trabajo y dado que la presencia de conjuntos de entidades débiles en un modelo ER es importante, para la representación del problema en las etapas tempranas del diseño, se decidió aplicar un algoritmo para transformar los conjuntos de entidades débiles en un modelo ER determinado. Un algoritmo para eliminar/convertir entidades débiles se expone en [SHO99].

III.3. Transformador del modelo ER al relacional.

ERECASE sigue las reglas generales [ELM97] para la transformación del modelo ER a modelo relacional. El transformador del modelo lógico es el encargado de aplicar estas reglas a un modelo ER determinado y devolver el modelo lógico resultante. El modelo lógico constituye una lista de esquemas, que a su vez contienen una lista de campos. Para la implementación del modelo lógico se creó un conjunto de clases e interfaces que definen sus funciones principales. Estas reglas para la transformación del modelo ER a modelo relacional son implementadas

en una clase que, a partir de un modelo ER especificado mediante un método de preparación, genera el modelo lógico equivalente. Este método de esta clase se debe invocar antes de la ejecución de la transformación para especificar el modelo ER fuente y el modelo lógico destino. Adicionalmente hay una función que indica si el modelo lógico fue generado satisfactoriamente o si hubo algún error durante el proceso.

El generador de código SQL es el encargado de, a partir de un modelo lógico determinado, generar las sentencias de creación de tablas. Una clase implementa esta función a través de un método de generación, el código resultante de la generación se muestra usando otra clase construida para tal efecto. Se integran todos los módulos en una clase que también brinda la interfaz necesaria para la comunicación de la herramienta con el diseñador. Esta contiene todos los controles y opciones que permiten al diseñador interactuar con la herramienta, también es la encargada de cada vez que el diseñador abra un nuevo modelo, crear una plataforma para éste y el editor correspondiente.

VI. CONCLUSIONES

Se realizó un estudio de tres de las herramientas CASE comerciales más populares en Cuba, y se encontraron aspectos mejorables tales como: poca variedad de construcciones del modelo ER ofrecidas, inclusión de elementos de los modelos lógicos y físicos en los esquemas conceptuales, y realización de pocas o ninguna validación a los esquemas creados.

Como resultado del presente trabajo se obtuvo una herramienta CASE para el diseño de Bases de Datos relacionales usando como modelo de datos conceptual el ER Extendido. A esta herramienta se le llamó ERECASE y se caracteriza por:

- ◆ Proveer un amplio conjunto de construcciones del modelo ER que brinda para la creación de un esquema conceptual, lo que permite lograr una mayor expresividad en el diagrama.
- ◆ Lograr una gran aproximación a la representación más popular de las construcciones en el diagrama ER.
- ◆ Realizar validaciones a un esquema determinado reduciendo el número de errores que pueda cometer el diseñador de la Base de Datos.
- ◆ Transformar al modelo lógico del esquema conceptual y la posibilidad de la intervención del diseñador en este proceso tomando decisiones que pueden ser dependientes del universo de discurso.
- ◆ Generar las sentencias para la creación de las tablas de la Base de datos usando como sistema manejador objetivo el SQL Server.

V. BIBLIOGRAFÍA Y REFERENCIAS BIBLIOGRÁFICAS

- [ATK96] Atkins, C. Prescription or Description: Some observations on the conceptual modeling process. Proceedings of International Conference on Software Engineering: Education and Practice, 1996.
- [BAT92] Batini, C., Ceri, S. y Navathe, S.B. Conceptual Database Design: an Entity-Relationship approach. The Benjamin/Cummings Publishing Company, Inc. 1992.
- [CUA99] Cuadra, D., Nieto, C. y Martínez, P.; de Miguel, A. Control de restricciones de cardinalidad en una metodología de desarrollo de Bases de Datos Relacionales. Novática Jul/Ago 1999 no 140 pp 28-33.
- [DAT90] Date, C. J. Introducción a los Sistemas de Bases de Datos. Vol-1 Quinta Edición. Addison-Wesley Iberoamericana 1993. (1990) Wilmington, Delaware, E.U.A.
- [DUL99] Dullea, J. y Song, Il-Ycol. A Taxonomy of Recursive Relationships and Their Structural Validity in ER Modeling. The proceeding of 16th International Conference on the Entity-Relationship Approach. pp 384-398.
- [ELM97] Elmasri, R. y Navathe, S. B. Sistemas de Bases de Datos. Conceptos fundamentales. Segunda Edición. Addison-Wesley Iberoamericana, Wilmington, Delaware, E.U.A, 1997.

[SHO99] Shoval, P. y Balaban, M. Resolving de “Weak Status” of a Weak Entity Types in Entity Relationship Schemas. The proceeding of 16th International Conference on the Entity-Relationship Approach. pp 369-383.

[SIL98] Silberschatz, A., Korth. H. F. y Sudarshan, S. Database System Concepts. McGraw-Hill. Madrid, España, 1998.

ANEXO. INTERFAZ DE ERECASE

