

## Algoritmos de Inserción Segura para Esquemas Conceptuales de Bases de Datos.

Gonzalo Rojas D.

### Resumen

En el presente artículo se presenta una propuesta para abordar el problema de la conservación de la consistencia de bases de datos, que consiste en la generación directa de algoritmos de inserción a partir de especificaciones formales de restricciones de integridad a nivel conceptual, utilizando el modelo de datos CCER (Varas, 1998). Se presenta una arquitectura de sistemas de bases de datos que incluye el conjunto de algoritmos, sus mecanismos de obtención y de mapeamiento a niveles más cercanos a la implementación. Finalmente, se ejemplifica la obtención de los algoritmos, mediante una restricción particular.

**Palabras Clave:** Bases de Datos, Consistencia, Restricciones de Integridad, Modelamiento Conceptual, Algoritmos de Inserción, Arquitectura de Sistema de Bases de Datos, Dependencia entre Tipos de Interrelación.

### 1 Introducción.

El problema de la conservación de la consistencia de las bases de datos está estrechamente ligado al deficiente manejo de las restricciones de integridad que la realidad representada impone a los datos.

Los modelos tradicionalmente utilizados en el diseño de bases de datos, como el MER (Chen, 1976) y el Relacional (De Miguel y Piattini, 1993), son capaces de expresar una cantidad muy limitada de restricciones, no permitiendo la especificación formal de muchas otras. Esta carencia de expresividad obliga a que estas últimas sólo puedan ser implementadas en etapas tardías del desarrollo, o bien, que sean completamente ignoradas, permitiendo que los usuarios modifiquen los datos sin preocuparse por su cumplimiento y, por ende, poniendo en riesgo la consistencia e integridad de la base de datos.

En este escenario, resulta difícil especificar formalmente algoritmos para la manipulación de la base de datos, de modo que sean portables y que su actualización sea de bajo costo, debido a que las restricciones de integridad que éstos deben considerar son implementadas con un alto grado de dependencia de la plataforma.

En el presente artículo, se plantea la propuesta de conservación de la consistencia de una base de datos durante el ingreso de datos, mediante la utilización de algoritmos generados directamente desde el esquema conceptual de la base de datos, incluyendo éste especificaciones formales de las restricciones de integridad que tales procedimientos deben considerar. La propuesta se basa en la utilización del modelo CCER (Varas, 1998) que, mediante la especificación formal de estructuras y de un conjunto importante de restricciones de integridad, otorga los recursos necesarios para generar completamente los métodos de inserción. A partir de la especificación de cada una de las restricciones expresables por el modelo, se obtiene un algoritmo de inserción de datos que obligan a su cumplimiento, cerrando cualquier posibilidad de introducción de inconsistencias. El conjunto de algoritmos generados a partir de este mapeamiento (Base de Algoritmos de Inserción Segura), regirá la definición de los procedimientos transaccionales de inserción.

La consideración de restricciones de integridad en los algoritmos de inserción de datos permitirá, desde un comienzo, asegurar el cumplimiento de éstas y mantener la consistencia de la base de datos. La portabilidad de los procedimientos transaccionales se verá aumentada, por la dependencia formal de los algoritmos en que se basan.

Eventuales actualizaciones de los algoritmos obtenidos, producto de modificaciones y depuraciones en los esquemas de datos, implicarán un menor costo que las modificaciones de procedimientos construidos en las etapas postreras del desarrollo de la base de datos.

En este artículo, se amplía la propuesta presentada por el autor (Rojas y Varas, 2000), incorporando una descripción de la arquitectura del sistema de bases de datos en el que la Base de Algoritmos se inserta y un ejemplo de mapeamiento a algoritmos de inserción de una restricción que involucra tanto a Tipos de Entidad como a Tipos de Interrelación.

En la sección 2 de este artículo, se describe brevemente la problemática que plantea a los diseñadores de bases de datos el problema del manejo de restricciones de integridad. En la sección siguiente, se presentan los principales enfoques de investigación para abordar este problema. En la sección 4, se expone la propuesta de obtención de algoritmos de inserción segura, además de la presentación de un modelo de arquitectura de bases de datos, donde se representan los mecanismos de obtención de estos algoritmos en el nivel conceptual y su relación con otros niveles del sistema. En la sección 5, se presenta la especificación formal que plantea el modelo CCER de una restricción de integridad particular, con el fin de describir, en la sección 6, el proceso de obtención de los algoritmos de inserción correspondientes a la restricción. Finalmente, en la sección 7, se muestran algunas conclusiones de lo aquí planteado.

## **2 El problema de manejo de restricciones de integridad.**

La conservación de la consistencia de una base de datos es un requisito fundamental para que la información contenida en ella refleje lo más certeramente posible la realidad modelada. Para la consecución de este objetivo, uno de los aspectos más importantes a considerar durante la realización de transacciones a una base de datos (inserción, actualización o eliminación de datos), es el cumplimiento de las restricciones de integridad.

Las restricciones de integridad son reglas que una base de datos debe satisfacer en todo momento, para poder representar fielmente una determinada realidad (Varas, 1998).

Durante el diseño conceptual de una base de datos, se realizan grandes esfuerzos por capturar todas aquellas restricciones que permiten mantener estados consistentes. El resultado de esta actividad puede ser tan simple como el establecimiento de cardinalidades y dominios, o incluir una cantidad de conocimiento altamente complejo, abarcando una parte significativa del entendimiento del dominio del problema.

Sólo una pequeña parte de estos esfuerzos se incluyen en el diseño conceptual de la base de datos. Uno de los principales problemas para la consideración de las restricciones de integridad en esta etapa radica en la dificultad de expresar tal conocimiento formalmente, para que de tal especificación puedan derivarse múltiples implementaciones particulares.

Los productos comerciales soportan restricciones muy simples, como clave primaria e integridad referencial. El conocimiento restante queda en manos de los diseñadores de aplicación, así es que la conservación de la consistencia se vuelve más un problema de ingeniería de software que una propiedad inherente del sistema de base de datos (Ceri, Fraternali, Paraboschi y Tanca, 1994).

## **3 Enfoques adoptados para la solución del problema.**

Para abordar este problema, las principales labores de investigación, según Ceri et al (1994), han adoptado al menos una de las siguientes perspectivas, considerándose éstas ya sea como alternativas o bien como complementarias:

- a. a. Apoyo conceptual y tecnológico diseñado específicamente para la conservación de consistencia, independientemente del modelo de datos y de la representación física de éstos. La investigación en los campos de dependencia de bases de datos, ejecución y evaluación de restricciones es la base de esta perspectiva.

- b. b. Enriquecer los modelos de datos, lo que permite incluir varios tipos de restricciones de integridad directamente en los esquemas.

Un ejemplo de los trabajos basados en la primera perspectiva es el expuesto en Ceri et al (1994), donde se propone, en lugar de realizar rollbacks a las transacciones que generan inconsistencias, que el sistema de bases de datos reaccione en forma autónoma a la introducción de inconsistencias, gatillando un conjunto de acciones reparadoras capaces de eliminar progresivamente la violación de restricciones de integridad hasta que se alcance un estado consistente. Esta solución y, por lo general, las soluciones basadas en este enfoque son muy costosas en términos de procesamiento computacional y complejidad, pues implican diseñar complejos algoritmos de detección de inconsistencias y comparación de información (Celle y Bertossi, 1994).

La propuesta de este artículo se basa en la segunda perspectiva, a partir del enriquecimiento del modelo de datos CCER (Varas, 1998), un modelo conceptual de bases de datos con mayor capacidad expresiva de restricciones de integridad, en comparación a los modelos tradicionalmente utilizados, como el MER y sus extensiones (Chen, 1976; Batini, Ceri y Navathe, 1994; Korth y Silberschatz, 1993) y el Relacional (De Miguel y Piattini, 1993).

El modelo CCER, acrónimo de Constraint Centered Entity-Relationship Model (Modelo Entidad Interrelación centrado en Restricciones), permite especificar formalmente una amplia variedad de restricciones de integridad. Nace de la necesidad de integrar más conceptos que los considerados en los modelos tradicionales, con interpretaciones únicas y no ambiguas de la estructura y restricciones de sus datos. Las características de este modelo permiten una generación directa de métodos de alta portabilidad que implementen las restricciones de integridad de los datos.

## **4 Propuesta de solución: Base de Algoritmos de Inserción Segura.**

### **4.1 Presentación de la propuesta.**

Se propone un mecanismo de mapeamiento para la generación directa de procedimientos algorítmicos de inserción de datos, que obligue al cumplimiento de las restricciones de integridad expresables en el modelo CCER.

Se plantea la asociación de un algoritmo de inserción a cada restricción de integridad del modelo, generando un conjunto de procedimientos denominado *Base de Algoritmos de Inserción Segura para CCER* (BAIS-CCER). Este conjunto podrá ser utilizado por aplicaciones que manipulen la base de datos, implementado como servicios de inserción.

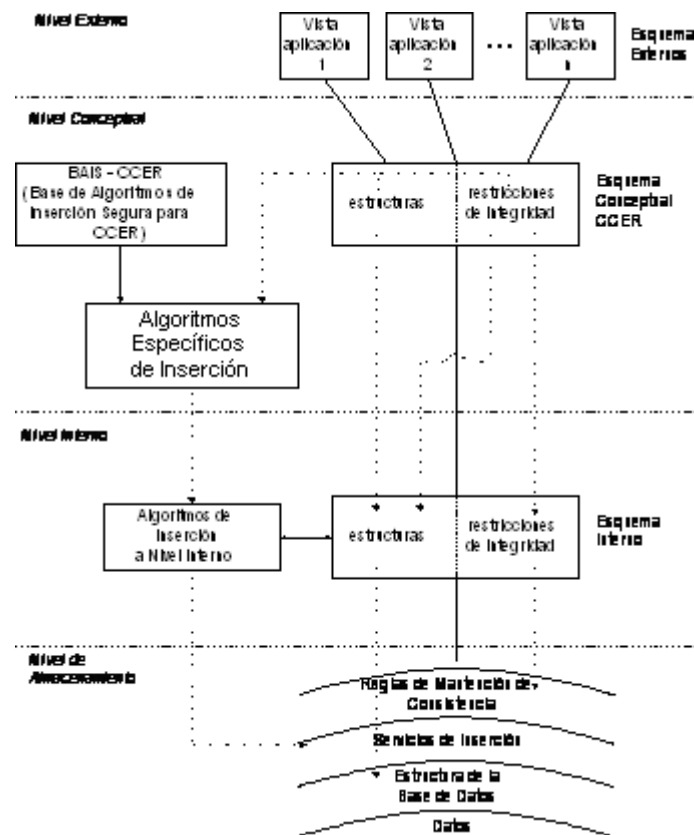
Los algoritmos generados aseguran el cumplimiento de sus correspondientes restricciones, antes y durante la inserción de datos, no otorgando al usuario la posibilidad de violarlas. Con esto, se pretende generar una base de datos "consistente por inserción", es decir, la utilización de estos procedimientos asegurará la mantención de la consistencia de la base de datos durante el ingreso de datos.

La complejidad de la tarea de asociar a una restricción de integridad un algoritmo de inserción que la implemente, radica tanto en el análisis de las condiciones que tal restricción impone a los datos que se ingresarán, como en la gran variedad de manifestaciones que una misma restricción puede tener en problemas de la vida real.

Para la elaboración de los algoritmos de inserción que conformarán la BAIS, se asume que los esquemas de datos que servirán como entrada están correctamente diseñados, es decir, no existen conflictos entre las restricciones de integridad que incluyan. Los algoritmos serán expresados en un lenguaje independiente de la plataforma de implementación, para asegurar su portabilidad.

## 4.2 Propuesta de Arquitectura del Sistema de Bases de Datos.

El siguiente esquema muestra la arquitectura del sistema de bases de datos en el cual está inserta la Base de Algoritmos de Inserción Segura:



**Figura 1. Arquitectura del Sistema de Bases de Datos con Base de Algoritmos de Inserción Segura.**

El nivel superior (*Nivel Externo*) contiene las vistas de las aplicaciones de usuarios que accesan los datos de la Base de Datos, expresadas en el diagrama como Esquemas Externos.

En el *Nivel Conceptual*, se diseña el *Esquema Conceptual* en CCER de la base de datos, que contiene las estructuras y restricciones de integridad expresadas en forma independiente del Sistema de Gestión de Bases de Datos particular, mediante los formalismos gráfico y textual del Modelo CCER.

A partir del mismo modelo, se generó la *Base de Algoritmos de Inserción Segura para CCER* (BAIS-CCER), un conjunto de algoritmos que permiten insertar datos sin violar las restricciones de integridad de un esquema en particular. De este conjunto de algoritmos, y a través de un mapeamiento de estructuras y restricciones del esquema conceptual, se obtienen los Algoritmos Específicos de Inserción Segura, que son los algoritmos que permitirán realizar inserciones en la base de datos específica que se está modelando.

Las estructuras definidas en el nivel conceptual son mapeadas a estructuras en el Nivel Interno, donde se establece la estructura de almacenamiento y recuperación de datos. Las restricciones del nivel conceptual se mapean a estructuras y restricciones de este nivel. Todas las estructuras y restricciones resultantes del mapeamiento conceptual-interno están comprendidas en el llamado Esquema Interno.

Los Algoritmos Específicos de Inserción Segura (nivel conceptual) se mapean al nivel interno, obteniendo los *Algoritmos de Inserción de Nivel Interno*, mapeamiento que depende de las

estructuras y restricciones obtenidas en este nivel.

El *Nivel de Almacenamiento de Datos* se organiza en un conjunto de capas, siendo la más interna la correspondiente a los Datos y la más externa a las Reglas de Mantenimiento de Consistencia. El dinamismo o variabilidad de la información almacenada explica este orden, variabilidad que aumenta desde las capas internas hacia las externas.

Las *Reglas de Mantenimiento de Consistencia* corresponden a las hasta aquí llamadas restricciones de integridad, pero almacenadas de acuerdo al SGBD particular. Se obtienen mediante mapeamiento desde las restricciones de integridad de nivel interno. Caso análogo lo constituye la Estructura de la Base de Datos, mapeada desde las estructuras del esquema interno.

Finalmente, los Algoritmos de Inserción de Nivel Interno son mapeados al nivel de almacenamiento, dando origen a los *Servicios de Inserción*, que son los procedimientos que llevarán a cabo la inserción de los datos. Estos Servicios ocupan una capa intermedia, bajo las Reglas de Mantenimiento de Consistencia y sobre la Estructura de la Base de Datos. Esto se explica por el hecho que cualquier modificación en tales reglas, que corresponden a la capa con mayor variabilidad dentro del nivel de almacenamiento, implicará una modificación de los Servicios de Inserción, no así de la Estructura de la Base de Datos.

## 5 Restricción de Dependencia entre Tipos de Interrelación.

Para ejemplificar la obtención de algoritmos de inserción segura a partir de la especificación formal de restricciones de integridad, se considerará el caso de la restricción de Dependencia entre Tipos de Interrelación.

La Dependencia entre Tipos de Interrelación es una restricción de integridad que puede ser expresada a nivel conceptual utilizando el modelo CCER.

### 5.1 Especificación formal de la restricción.

La restricción de Dependencia entre Tipos de Interrelación (Varas, 1998, 2000) se expresa a través de la siguiente función:

```

dep:  $R \times TE \rightarrow L$  ; con

R el conjunto de los Tipos de Interrelación;

TE el conjunto de los Tipos de Entidad; y

 $L = \{ \wedge, \vee, \perp \}$ 

 $dep(R, TE) = op$ 

con

 $R = \{R_1, \dots, R_n\}$  ; y

 $TE = \{TE_a, \dots, TE_z\}$ 

```

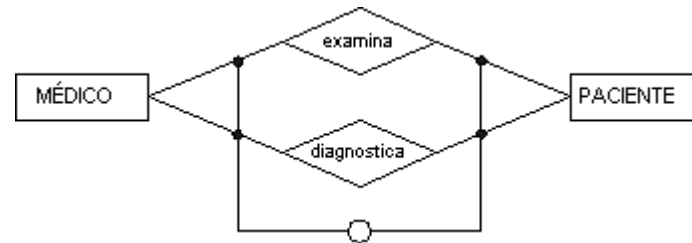
Si  $op = \wedge$ , el conjunto de tipos de interrelación R es conjuntivo con respecto al conjunto de tipos de entidad TE, es decir, toda interrelación presente en algún  $R_i$  ( $i \in \{1, \dots, n\}$ ) debe estar presente en todos los otros  $R_j$  en R ( $j \in \{1, \dots, n\}; j \neq i$ ).

Si  $op = \vee$ , el conjunto R es disyuntivo con respecto al conjunto TE, es decir, si una interrelación está presente en algún  $R_i$  ( $i \in \{1, \dots, n\}$ ), puede o no estar presente en los otros  $R_j$  en R ( $j \in \{1, \dots, n\}; j \neq i$ ).

$\{1,..n\}; j i)$ .

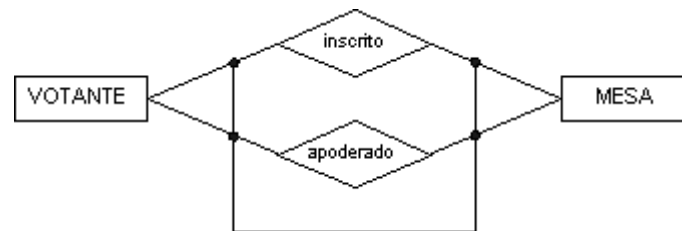
Si  $op = \oplus$ , se dice que el conjunto R es exclusivo con respecto al conjunto TE, es decir, si una interrelación está presente en algún  $R_i$  ( $i \in \{1,..,n\}$ ), no debe estar presente en ninguno de los otros  $R_j$  en R ( $j \in \{1,..n\}; j \neq i$ ).

## 5.2 Ejemplos de aplicación.



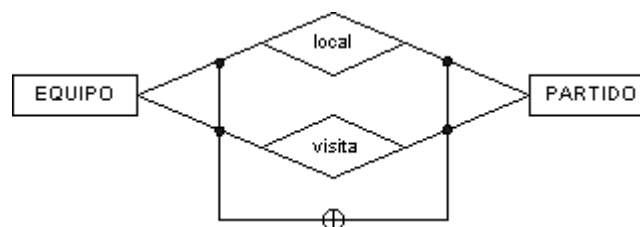
**Figura 2. Ejemplo de Dependencia en Conjunción entre Tipos de Interrelación.**

En el problema de la figura 2, el conjunto  $R = \{\text{examina}, \text{diagnostica}\}$  es *conjuntivo* con respecto a  $TE = \{\text{MÉDICO}, \text{PACIENTE}\}$ , pues un médico que examina a un paciente necesariamente establece un diagnóstico de su estado de salud y viceversa (un diagnóstico siempre va acompañado de un examen previo). De acuerdo a esto, cualquier interrelación médico-paciente presente en uno de los Tipos de Interrelación pertenecientes a R necesariamente está presente en el otro.



**Figura 3. Ejemplo de Dependencia en Disyunción entre Tipos de Interrelación.**

En la figura 3, el conjunto  $R = \{\text{inscrito}, \text{apoderado}\}$  es *disyuntivo* con respecto a  $TE = \{\text{VOTANTE}, \text{MESA}\}$ , puesto que un votante puede estar inscrito en una mesa de votación y ser apoderado de esa mesa, estar inscrito sin ser apoderado de su mesa, o bien, ser sólo apoderado de esa mesa, pudiendo estar inscrito en otra. Es decir, una interrelación votante-mesa puede estar presente en cualquiera de los dos Tipos de Interrelación, en ambos o en ninguno. Nótese que la dependencia en disyunción entre Tipos de Interrelación *no es restrictiva*.



**Figura 4. Ejemplo de Dependencia en Exclusión entre Tipos de Interrelación.**

Finalmente, en la figura 4, el conjunto  $R = \{\text{local}, \text{visita}\}$  es *exclusivo* con respecto a  $TE = \{\text{EQUIPO}, \text{PARTIDO}\}$ , puesto que un equipo debe jugar un partido como local o como visita, no pudiendo estar presente una misma interrelación equipo-partido tanto en "local" como en "visita".

## 6 Algoritmos para Dependencia entre Tipos de Interrelación.

El proceso de diseño de los algoritmos de la BAIS-CCER consiste en un estudio, a partir de ejemplos reales de modelamiento conceptual, de distintas manifestaciones de la restricción a mapear. Este estudio arroja todas las condiciones de cumplimiento de la restricción y las posibles fuentes de inconsistencias en el proceso de inserción de datos en las estructuras asociadas a ella. Los pasos de cada algoritmo buscan entonces bloquear estas fuentes y cumplir con las condiciones de la restricción.

Para el diseño de los algoritmos correspondientes a la Dependencia entre Tipos de Interrelación, se consideraron los casos de dependencia en conjunción y exclusión. La disyunción, como se señaló, no tiene carácter restrictivo.

### 6.1 Definiciones previas.

El *espectro* de un Tipo de Interrelación es el conjunto de Tipos de Entidad que relaciona.

El *grado* de un Tipo de Interrelación es la cardinalidad de su Espectro, es decir, el número de Tipos de Entidad que relaciona.

De acuerdo a las definiciones mostradas en el apartado 5.1, para todo  $R_i$   $R$ , con  $i \in \{1, \dots, n\}$ , se tiene que:

$$\text{Espectro}(R_i) = \{TE_1, TE_2, \dots, TE_a, \dots, TE_z, \dots, TE_{g_i}\}$$

con  $g_i = \text{Grado}(R_i)$ ;  $1 \leq a \leq g_i$

El *Subespectro* de un Tipo de Interrelación es el conjunto de atributos que lo componen, sin considerar aquéllos aportados por los Tipos de Entidad que relaciona (Varas, 2000).

Se define el *Espectro Independiente* del Tipo de Interrelación  $R_i$   $R$ , con  $i \in \{1, \dots, n\}$ , como el conjunto de los Tipos de Entidad del Espectro de  $R_i$  que no pertenecen al conjunto  $TE$ .

Para todo  $R_i$   $R$ , con  $i \in \{1, \dots, n\}$ , se tiene que:

$$\text{Espind}(R_i) = \text{Espectro}(R_i) - TE = \{TE_1, \dots, TE_{a-1}, TE_{z+1}, \dots, TE_{g_i}\}$$

### 6.2 Funciones auxiliares.

Las siguientes son funciones auxiliares, invocadas por los algoritmos correspondientes a esta restricción:

" ref\_integrity ( $e_1, \dots, e_n, TE_1, \dots, TE_n$ )

Función booleana que chequea la existencia actual de las entidades  $e_i$  en sus respectivos Tipos de Entidad  $TE_i$ ,  $i=1..n$

" inserted ( $e_1, \dots, e_n, R_x$ ), con  $1 \leq n \leq \text{grado}(R_x)$

Función booleana que retorna el valor true si existe alguna interrelación, actualmente inserta en el Tipo de Interrelación  $R_x$ , que agrupe las entidades  $e_1, \dots, e_n$ ; en caso contrario, retorna el valor false.



" insertar<sub>ti</sub> ( $e_1, \dots, e_{g_x}, R_x$ ), con  $g_x = \text{grado}(R_x)$

Procedimiento que gatilla la inserción de una interrelación que agrupa las entidades  $e_1, \dots, e_{g_x}$  en el Tipo de Interrelación  $R_x$ . En este procedimiento, se forma el elemento a insertar, con los valores identificadores de tales entidades, yuxtapuestos con los valores de atributos pertenecientes al Subespectro de  $R_x$ .

### 6.3 Algoritmo para Dependencia en Conjunción de Tipos de Interrelación.

Las condiciones de cumplimiento de esta restricción se han representado en su algoritmo de inserción, por medio de la ejecución de los siguientes pasos:

- Se realiza un chequeo de integridad referencial, para asegurar que todas las entidades que se quieren agrupar en el Tipo de Interrelación  $R_x$  ya estén insertas en sus respectivos Tipos de Entidad. Si esto no ocurre, se rechaza la transacción.
- Se realiza el chequeo de unicidad, para asegurar que la interrelación (ya validada) no ha sido previamente insertada en  $R_x$ . Si esto ya ha ocurrido, se rechaza la transacción, para evitar duplicidad de información.
- Cumplidas satisfactoriamente las condiciones anteriores, se inserta la interrelación en  $R_x$ , habiendo el usuario previamente ingresado los valores de los atributos pertenecientes al SubEspectro de  $R_x$ .
- Finalmente, para cumplir con la conjunción, se agrupan las mismas entidades de la interrelación en los otros Tipos de Interrelación pertenecientes al conjunto  $R$ , previo ingreso de las entidades que no pertenecen a la conjunción y a los atributos correspondientes a cada Tipo de Interrelación. Se asume que, habiéndose utilizado este mismo algoritmo para la inserción en todos los Tipos de Interrelación de  $R$ , si la interrelación ya estuviese presente en uno de ellos, lo estaría también en todos los demás, incluyendo en  $R_x$ . Esto no es posible, por el chequeo de unicidad ya realizado.

El Algoritmo de Inserción Segura para la Dependencia en Conjunción de Tipos de Interrelación es el siguiente:

```

procedure insertion_conjdep (e1, ... , egx:entidad ; Rx: R){

  ref_int := ref_integrity (e1, ... , egx, Espectro(Rx));

  if not ref_int then

    error ('interrelación con al menos una entidad no insertada')

  else if inserted (e1, ...., egx , Rx) then

    error (' interrelación ya presente en Rx')

  else{

    insertar_ti (e1, ... , egx, Rx);

    for k:=1 to |R| do

      if k ≠ x then {

        gk := Grado (Rk)

        for j:=1 to gk do

          if (j<a) or (j>z) then

```



```
fj := read (entidad de TEj Espind(Rk));  
  
ref_int2 := ref_integrity (f1,...,fa-1,fz+1,...,fgk, Espind(Rk));  
  
if not ref_int2 then  
  
    error ('al menos una de las entidades no ha sido insertada')  
  
else  
  
    insertar_ti (f1,...,fa-1, ea, ..., ez, fz+1,...,fgk, Rx); }  
  
}
```

## 6.4 Algoritmo para Dependencia en Exclusión de Tipos de Interrelación.