# Descubrimiento automático de equivalencias en esquemas de bases de datos relacionales

## Automatic discovery of equivalences in schemes of relational database

**Christian G. Arias I.**


## Resumen

La mayoría de las organizaciones en la actualidad cuenta con múltiples aplicaciones de software que permiten un manejo muy propio de sus respectivas bases de datos, lo cual es un verdadero reto a la hora de integrarlas en aplicaciones heterogéneas. Tarea que tiene muchos procesos: inicialmente, se identifica qué aplicaciones se desea o se necesita integrar, después se ubica la información común y se crea un mapa de equivalencias entre los diversos atributos y, finalmente, se especifican los mecanismos de sincronización.

Para ello, este trabajo plantea, mediante un modelo flexible y con base en los trabajos de Li y Clifton (1994) y Rizopoulus, la utilización de algoritmos genéticos para identificar los conjuntos de parámetros que ofrezcan mejores resultados a la hora de generar equivalencias en esquemas de bases de datos relacionales.


**Palabras clave:** sistemas inteligentes, redes neuronales, base de datos, integración de datos, clasificación automática.


## Abstract

Most organizations today have multiple software applications that facilitate the management of their databases, which is a real challenge when it comes to integrate heterogeneous applications. This task involves many different processes: first, to identify which applications are wanted or need to be integrated, then to find regular information and create a map of equivalences between the different attributes existing and finally, to specify synchronization mechanisms.

To this end, this paper proposes, by means of a flexible model and based on the work of Li and Clifton (1994) and Rizopoulus, the use of genetic algorithms to identify sets of parameters that provide better results when generating equivalences on schemes of relational database.


• Fecha de recepción del artículo: Junio de 2008 • Fecha de aceptación: Septiembre de 2008.

**CHRISTIAN G. ARIAS I.** Ingeniero de Sistemas y docente-investigador del programa de Ingeniería de Sistemas Universidad de San Buenaventura Cali. Integrante del grupo de investigación Lidis - Laboratorio de Investigación en el Desarrollo de Ingeniería de Software. Correo electrónico: *cgarias@usbcali.edu.co*

**Keywords:** Intelligent systems, neural networks, databases, data integration, automatic classification.

## Introducción

Gracias al rápido desarrollo de las ciencias de la computación y los sistemas de gestión de la información, el proceso de adquisición de tecnología informática por parte de las empresas se ha agilizado mucho. Dentro de este panorama nos encontramos que las compañías buscan sistematizar prioritariamente dos tipos de proceso: el primero, que involucra labores administrativas que tienen un alto desgaste operativo, y el segundo, que tiene que ver con el núcleo de su negocio, por ejemplo, si una empresa de mensajería compra un sistema de contabilidad y un sistema que administre la información de los paquetes a enviar –que al parecer son independientes y no comparten mucha información–, tendrá la necesidad de integrarlos al momento de adquirir nuevos sistemas.

Estos sistemas adquiridos en fechas distintas, bajo diferentes administraciones y para distintas necesidades, han sido concebidos de forma independiente, y difícilmente pueden trabajar de forma integrada. Sin embargo, estas "islas" de información comparten datos comunes que se manejan de forma distinta y tienen su propia copia del dato.

Esta situación es el punto central sobre el cual se enfoca el EAI[1]. Su objetivo principal es lograr la integración de aplicaciones empresariales a nivel de procesos y datos que generen valor y simplifiquen la coexistencia y operación de los diversos sistemas.

Para integrar los datos se debe considerar la forma de almacenamiento, la arquitectura, la conectividad, la semántica de los datos, etc. De forma simplificada dentro del proceso de integración se pueden distinguir las siguientes etapas: inicialmente se debe identificar qué aplicaciones se desea o necesita integrar, posteriormente, se identifica la información común con la cual se crea un mapa de equivalencias, y finalmente, se especifican los mecanismos de sincronización.

La etapa de identificación de equivalencias es por sí sola un proyecto que involucra una gran complejidad. De una correcta y completa identificación depende el éxito del proyecto. Normalmente esta etapa requiere una gran cantidad de tiempo y la participación activa de expertos en las estructuras de datos de las aplicaciones que se desean integrar. La automatización de esta tarea brindaría enormes beneficios al proceso de integración de datos y al proceso de EAI en general.

## Propuesta de solución

### Esquema general

La idea general es clasificar los atributos de una base de datos generando categorías a partir de sus características estructurales y sus valores, utilizando redes neuronales (Hilera, y otros, 1995; Nirmal & Liang, 1996). Una vez obtenidas las categorías, se procede a efectuar un análisis similar en la base de datos que se quiere comparar, con la diferencia que a esta última no se le efectuará una categorización (es decir, no se crearán categorías nuevas a partir de sus atributos), sino que se calculará la distancia de cada atributo en las categorías previamente creadas.

Cuando un atributo esté lo suficientemente cerca de una categoría, se marcará como candidato. Los atributos candidatos pasarán a una nueva fase en la cual, se realizará una comparación más detallada entre los valores de los atributos. De esta forma se confirmará si existe una equivalencia, y de ser así, se identificará su tipo.

Las parejas que superen esta segunda fase serán presentadas como resultados muy probables a los que sólo les resta la verificación humana.

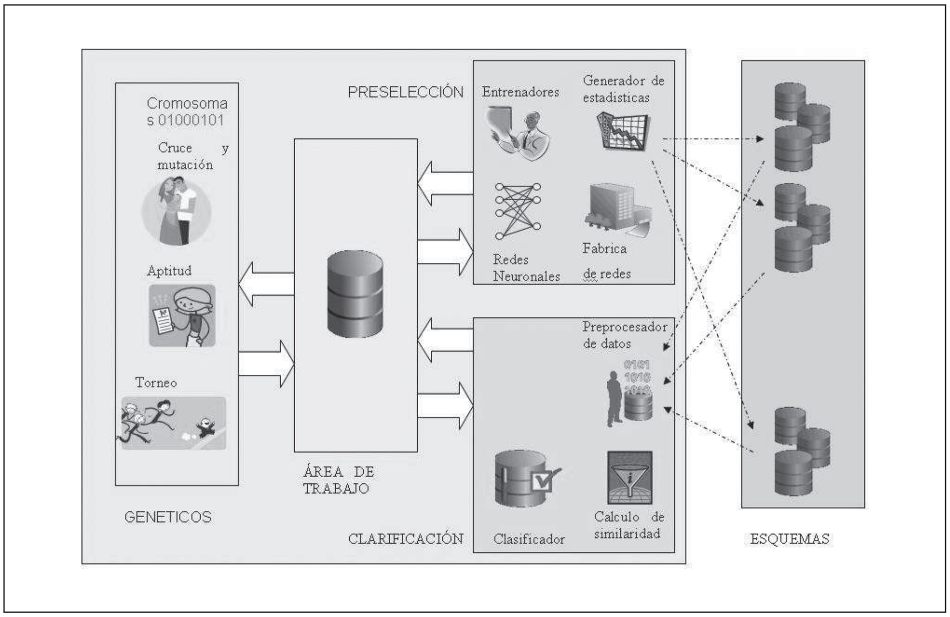
## Planteamiento del modelo

El modelo está conformado por las entradas definidas por el usuario, un área de trabajo (almacenamiento de resultados), y tres módulos de procesamiento (Ver figura 1):

- Módulo de preselección o generación de candidatos.

1. Enterprise Application Integration. Definido como el acceso compartido a datos y procesos a través de aplicaciones conectadas y fuentes de datos en la empresa.

**Figura 1**
**Arquitectura del sistema**




- Módulo de verificación y clasificación de equivalencias.
- Módulo genético.

## Entradas del sistema

### Esquemas de datos

Los esquemas o fuentes de datos son los grupos de estructuras sobre los cuales se desea generar las equivalencias. Estos esquemas deben aportar información de los siguientes parámetros:

- Nombre o dirección IP del servidor donde reside la base de datos.
- Puerto de escucha del servidor de base de datos.
- Nombre de la base de datos.
- Esquema.
- Usuario.
- Contraseña.

A partir de dicha información se generará el URL de conexión. La forma de dicho URL es similar entre los diversos motores aunque cada uno cuenta con sus propias características. Luego, se debe indicar qué esquema se empleará como pivote, es decir, cual de ellos se utilizará para crear las categorías iniciales. Ejemplo:

IP: 192.168.0.3
Puerto: 1521
Base de datos: ORCL
Esquema: HR
Contraseña: HR

### Conjunto de características a analizar

Se define una característica como una función que se puede aplicar sobre los valores que toma un atributo y retorna como un valor numérico.

Definiendo la característica C1 como la función promedio, al aplicarla sobre la columna salario, el resultado será: 1’200.000

| Salario |
|---|
| 2’000.000 |
| 1’450.000 |
| 550.000 |
| 800.000 |
| 1’200.000 |

También es posible especificar características que no tengan en cuenta los valores de los datos, sino las definiciones de los atributos. Ejemplo:

$$C2 = \begin{cases} \textit{Si\_tipoDato=Numérico} \longrightarrow \textit{retorne\_1} \\ \textit{Sino} \longrightarrow \textit{retorne\_0} \end{cases}$$

Para la columna salario el resultado será 1.

El objetivo de dichas características es que al aplicarlas nos brinden algún tipo de información sobre la naturaleza del atributo y de esta forma lograr una caracterización adecuada del conjunto de valores presentes. Algunas características pueden tomar mayor o menor sentido dependiendo del tipo de dato. Este modelo es flexible y permite agregar nuevas características a gusto del usuario. El modelo especifica que la característica debe definir claramente cómo se deben procesar cada uno de los tipos de datos permitidos (cadenas, números, fechas). En algunos casos se podrían transformar las cadenas y las fechas en números para calcular ciertas variables. Las fechas por ejemplo, se convierten en números de 8 dígitos que corresponden al formato aaaammdd (año-mes-día).

A continuación se brinda una breve reseña de las características que se definieron y utilizaron en las pruebas:

*Promedio de caracteres numéricos (PCN)*

Para generar este valor, a cada dato de la columna se le cuentan la cantidad de caracteres que estén incluidos en el conjunto {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, y posteriormente, se calcula el promedio. No se toman en cuenta los caracteres de puntuación como puntos ni comas. Esta característica tiene más sentido para los atributos tipo cadena, sin embargo, también se puede aplicar para los atributos de tipo numérico. En el caso de los datos tipo fecha se asigna la constante 8 (por representarse con 8 dígitos).

*Promedio de caracteres en blanco (PCB)*

Para generar este valor, a cada dato de la columna se le cuentan la cantidad de espacios en blanco, y posteriormente, se calcula el promedio. No se toman en cuenta los caracteres de puntuación como puntos ni comas. Esta característica tiene más sentido para los atributos tipo cadena. Para los datos tipo cadena y numérico se asigna la constante 0.

*Promedio de longitud (PLN)*

Este valor se genera contando en cada dato de la columna la cantidad de caracteres totales, y luego, se calcula el promedio. Esta característica tiene más sentido para los atributos tipo cadena, sin embargo, se aplica también a los campos numéricos. A los tipos de dato fecha se le asigna la constante 8.

*Promedio (PRM)*

Para generar este valor se calcula el promedio de los valores de los campos. Para los atributos tipos cadena se verifica si contiene sólo dígitos y se trata como un número. En el caso de las cadenas se convierten a un número, donde sus primeros cuatro dígitos representan el año, los dos siguientes representan el mes y los dos últimos representan el día. No se tiene en cuenta la fracción del día, es decir, horas y minutos. Se consideró convertir las fechas a su representación juliana pero el soporte de los motores de base de datos para hacer esta conversión es menor que el soporte para la conversión propuesta.

*Varianza (VRN)*

Para generar este valor se calcula la varianza de los valores de los campos, la que se define como un medidor de la dispersión de una variable aleatoria *(X),* respecto a su esperanza E[X]:

$$\mathrm{V}\,(X) = E\left[(X - E\,[X\,])^2\right]$$

El valor de esta variable puede ser muy grande, por esta razón se decidió disminuir su magnitud utilizando la función logaritmo natural.

*Coeficiente de variación (CV)*

Este valor se genera colocando la varianza de los campos. Para los tipos de datos cadena y fecha, se realiza la misma conversión que la utilizada para calcular el promedio.

*Tipo de dato cadena (TCD)*

Retorna 1, en caso que el tipo de dato sea cadena, de lo contrario, retorna 0.

*Tipo de dato numérico (TNM)*

Retorna 1, en caso que el tipo de dato sea numérico, de lo contrario retorna 0.

*Tipo de dato fecha (TFC)*

Retorna 1, en caso que el tipo de dato se fecha, de lo contrario, retorna 0.

El siguiente conjunto de datos se tomará como ejemplo para ilustrar el funcionamiento de la especificación de las características:

**Tabla 1**
**Valores de prueba para calcular el valor de sus características**

| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|---|---|---|---|---|---|---|---|
| 7369 | SMITH | CLERK | 7902 | 17/12/80 | 800 | | 20 |
| 7499 | ALLEN | SALESMAN | 7698 | 20/02/81 | 1600 | 300 | 30 |
| 7521 | WARD | SALESMAN | 7698 | 22/02/81 | 1250 | 500 | 30 |
| 7566 | JONES | MANAGER | 7839 | 02/04/81 | 2975 | | 20 |
| 7654 | MARTIN | SALESMAN | 7698 | 28/09/81 | 1250 | 1400 | 30 |
| 7698 | BLAKE | MANAGER | 7839 | 01/05/81 | 2850 | | 30 |
| 7782 | CLARK | MANAGER | 7839 | 09/06/81 | 2450 | | 10 |
| 7788 | SCOTT | ANALYST | 7566 | 19/04/87 | 3000 | | 20 |
| 7839 | KING | PRESIDENT | | 17/11/81 | 5000 | | 10 |
| 7844 | TURNER | SALESMAN | 7698 | 08/09/81 | 1500 | 0 | 30 |
| 7876 | ADAMS | CLERK | 7788 | 23/05/87 | 1100 | | 20 |
| 7900 | JAMES | CLERK | 7698 | 03/12/81 | 950 | | 30 |
| 7902 | FORD | ANALYST | 7566 | 03/12/81 | 3000 | | 20 |
| 7934 | MILLER | CLERK | 7782 | 23/01/82 | 1300 | | 10 |

**Tabla 2**
**Valores del análisis de características de los datos de la tabla 5**

| MPO | TDC | TNM | TFC | PCN | PCB | PLN | PRM | VRN | CV |
|---|---|---|---|---|---|---|---|---|---|
| PNO | 0 | 1 | 0 | 4 | 0 | 4 | 7726,57143 | 31789 | ,023075482 |
| AME | 1 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 |
| B | 1 | 0 | 0 | 0 | 0 | 7 | 0 | 0 | 0 |
| R | 0 | 1 | 0 | 4 | 0 | 4 | 7739,30769 | 10757 | ,013401026 |
| REDATE | 0 | 0 | 1 | 8 | 0 | 8 | 19819256,8 | 484629314 | 0 |
| L | 0 | 1 | 0 | 4 | 0 | 4 | 2073,21429 | 1398314 | ,570371925 |
| MM | 0 | 1 | 0 | 3 | 0 | 3 | 550 | 363333 | 1,09594796 |
| PTNO | 0 | 1 | 0 | 2 | 0 | 2 | 22,1428571 | 64 | ,362095876 |

El usuario del modelo debe definir de qué forma se logra medir una característica nueva, teniendo en cuenta los posibles tipos de datos. Se debe tener en cuenta que el resultado esperado siempre debe ser de tipo numérico.

## Área de trabajo

Como su nombre lo sugiere, este componente es un espacio físico donde se almacenará información generada por el proceso y albergará la información extraída por el generador de estadísticas, así, para cada atributo procesado se tendrán los valores obtenidos para cada característica analizada. También, almacenará la información generada por la red SOM[2] (Kohonen, 1990), respecto a las categorías formadas y los pesos del centro del *cluster* correspondiente.

## Módulo de preselección

Este componente se encarga de generar las parejas de atributos que presenten claros indicios de ser equivalentes y las pasa al módulo de verificación y clasificación. Incluye el uso de dos tipos de redes neuronales: SOM y Back-propagation. Las redes neuronales son muy apropiadas para este tipo de problemas ya que son fácilmente adaptables y toleran muy bien el ruido.

El módulo de preselección tiene los siguientes componentes:

*Generador (recolector) de estadísticas*

Este componente es el primero en participar dentro del proceso de este módulo. Para cada esquema suministrado selecciona todos los campos que cumplan con los tipos de datos permitidos y que no sean parte de una llave foránea, y a cada uno de ellos, le aplica las fórmulas especificadas por cada característica, obtiene el valor resultante y lo almacena para su posterior utilización. Pueden existir varias copias de este componente ejecutándose concurrentemente y cada uno analizando atributos distintos. Se debe procesar todos los esquemas de datos de igual forma, sin importar si es el esquema pivote o no. Los resultados obtenidos se almacenan en el área de trabajo.

*Generador de la red SOM*

Este componente se encargará de generar las redes auto-organizadas, las cuales realizarán el proceso de categorización de los atributos. Recibe como parámetros la cantidad de características que se debieron analizar (E) y el número de atributos procesados del esquema pivote(S). La red resultante tendrá (E) neuronas de entrada y (S) neuronas de salida. Se propone este componente debido a que el modelo permite especificar un número variable de características y en consecuencia la cantidad de neuronas de la capa de entrada sólo se puede conocer después de haber recibido esta información. Otro tanto, sucede con la capa de salida que depende del número de atributos procesados. Entonces, se tienen en cuenta sólo los atributos del esquema pivote al usar la red resultante, porque esta será la base para generar las categorías.

*Red SOM*

Este componente es generado por el componente anterior. Recibe como entrada el resultado del análisis de las características de los atributos del esquema pivote. La idea es que se genere máximo una categoría para cada atributo, aunque dependiendo de los datos ingresados es posible que varios atributos se enmarquen en la misma categoría, en consecuencia, el número de categorías formadas será menor a la cantidad total de atributos. Luego, permitimos que la red se entrene y forme sus categorías, una vez finalizado el proceso, se utiliza la red (sin aprendizaje) para verificar qué atributos están en una misma categoría y de esta forma calcular los pesos de los centros de los *cluster* correspondientes. Para las categorías que tengan un solo atributo, los pesos del centro del *cluster* serán iguales a los parámetros de entrada iniciales del atributo. Las categorías formadas y sus pesos correspondientes son almacenados en el área de trabajo para ser utilizadas posteriormente por la red *back-propagation*.

*Generador de la red* back-propagation

Este componente se encargará de generar las redes back-propagation, las cuales realizarán el proceso de calcular la cercanía de los atributos de las tablas de los esquemas no pivote a las categorías de los atributos del esquema pivote. Recibe como parámetros la cantidad de características que se analizaron (E) y el número de categorías generadas por la red SOM(S). La red resultante tendrá (E) neuronas de entrada, (S) neuronas de salida y, (E+S/2) neuronas en la capa oculta. Se propone este componente debido a que el modelo permite especificar un número variable de características, y en consecuencia, la cantidad de neuronas de la capa de entrada sólo se puede conocer después de haber re-

2. Self-Organizing Map.

cibido esta información. Otro tanto, sucede con la capa de salida que depende del número de categorías formadas por la red SOM.

*Red back-propagation*

Este componente es generado por el componente anterior. Su objetivo es calcular la cercanía de los atributos de los esquemas no pivote a las categorías generadas por la red SOM. Para desempeñar su función primero se debe entrenar la red, esto se realiza suministrando los conjuntos de datos de entrada y verificación. Los datos de entrada corresponderán a los valores de los pesos de los centros de los *clusters* de cada categoría y los datos esperados generan la activación de la neurona correspondiente. Una vez finalizado el entrenamiento la red se podrá utilizar para la identificación de posibles atributos equivalentes mediante el cálculo de cercanía a cada categoría. Los resultados obtenidos se almacenarán en el área de trabajo.

*Selector de pares*

Este componente es el encargado de determinar qué pares de atributos muestran indicios de ser equivalentes. Para desarrollar su tarea recibe como entrada un parámetro definido por el usuario que indica la distancia máxima a la que puede encontrarse un atributo de una categoría para ser considerado candidato, y recibe los resultados de las distancias de los atributos a cada categoría. Los pares candidatos serán almacenados en el área de trabajo.

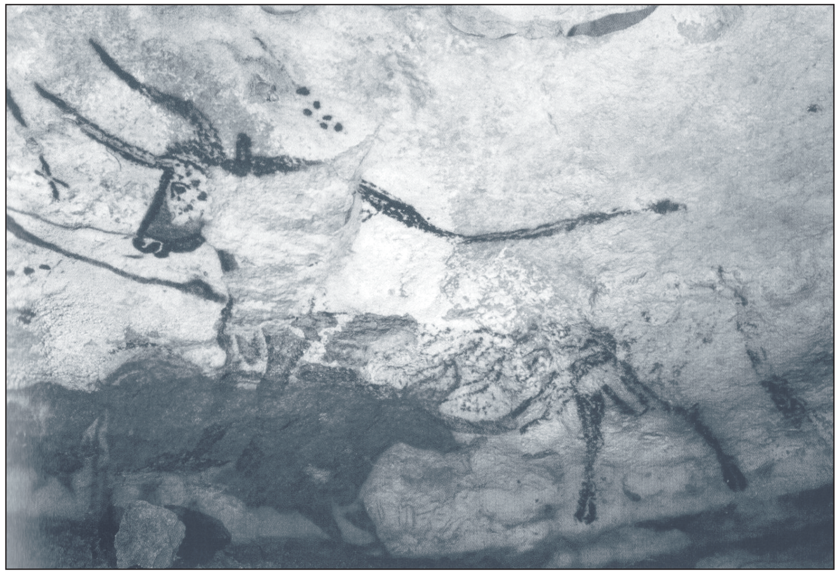
## Módulo de clarificación

Este componente se encarga de verificar si los pares de atributos preseleccionados por el módulo anterior son o no equivalentes y en qué grado. Esta tarea se efectúa mediante la comparación de ambos conjuntos de datos.

El módulo de de clarificación tiene los siguientes componentes:

*Extractor y preparador de datos*

Este componente se encarga de acceder al origen del atributo y seleccionar los datos almacenados, posteriormente, dependiendo del tipo de dato se lleva a cabo una normalización y ordenamiento para hacer más efectiva su posterior comparación, creando una estructura especial en el área de trabajo donde se guardan los datos después de su procesamiento.



*Período anterior al Magdaleniense, probablemente Solutrense. Uros. Dibujo a pincel y pintura de tinta plana. Sala de los Toros, cueva de Lascaux, Montignac.*

*Comparador unidireccional*

Este componente se encarga de verificar los grados de similaridad entre dos atributos teniendo en cuenta la dirección. Se utilizará las definiciones hechas por Larson (1989) y retomadas por Rizopoulos, (2004), excluyendo las equivalencias de incompatibilidad ya que se presume que esos casos se eliminarán en etapas previas. Explicándolo forma sencilla, se definirá la función S(X,Y) como el grado de similaridad de X con Y, que es distinto a S(Y,X), que es el grado de similaridad de Y con X. Al observar la fórmula la diferencia se hace evidente:

$$S(X,Y) = \frac{count\ (X \cap Y)}{count\ (X)} \quad S(Y,X) = \frac{count\ (Y \cap X)}{count\ (X)}$$

Los valores resultantes se encuentran en un rango entre 0 y 1. Los resultados obtenidos son almacenados en el área de trabajo.

*Consolidador de grados*

Toma como entrada los valores suministrados por el comparador y clasifica los resultados de la siguiente forma:

| | |
|---|---|
| Si S(X,Y) y S(Y,X) > E | Atributos equivalentes |
| Si S(X,Y) y S(Y,X) < D | Atributos disyuntos |
| Si S(X,Y) > E y S(Y,X) < E | Y incluye a X |
| Demás casos | Los atributo se intersectan |

Estos resultados son almacenados en el área de trabajo.

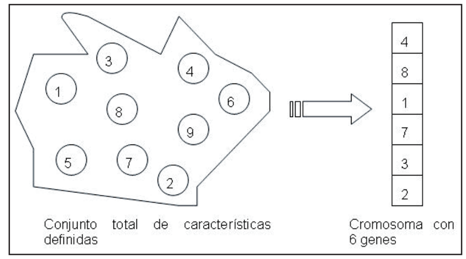
## Módulo genético

Este módulo implementa algoritmos genéticos. El primer paso que se debe dar para trabajar con esta técnica es modelar las posibles soluciones del problema en forma de un cromosoma. Los genes de nuestro cromosoma serán las variables cuyos valores queremos descubrir para lograr que el proceso sea más preciso.

En este modelo se plantea analizar una serie de características sobre cada uno de los atributos participantes y esta cantidad de características determinará el número de neuronas de la capa de entrada. Entre más variables se analicen más precisa será la clasificación, sin embargo, al aumentar el tamaño de este conjunto, si bien, las redes pueden lograr buenos resultados, el tiempo de convergencia será mayor. El ideal sería trabajar sólo con las características que fuesen más relevantes para lograr una adecuada clasificación. En efecto, al tener un gran número de características se hace difícil analizar la red e identificar cuáles fueron determinantes para encontrar la solución.

Aparece entonces un nuevo problema: ¿Cómo determinar cuáles son las características fundamentales para clasificar adecuadamente un atributo? Esta situación es la que se pretende resolver mediante la utilización del módulo genético. Para la ejecución de las pruebas se determinó que se buscarían las 6 características más relevantes, sin embargo, las pruebas pueden realizarse con cualquier otro número. Es importante señalar que también sería posible incluir en el cromosoma la elección de otras variables que participen en el proceso, además de las características de los atributos. Por ejemplo, el número de neuronas de la capa oculta de la red back-propagation, el número de iteraciones de entrenamiento de la red SOM, valor mínimo de similitud para que dos atributos sean considerados equivalentes o el valor máximo de similitud para ser considerados no equivalentes.

**Figura 2**
**Cromosoma de características**




El conjunto total de posibles soluciones está determinado por:

$$C\binom{N}{M}$$

Siendo (N), el número total de variables, que para este caso sería 9, ya que se han definido 9 características en total y (M) la cantidad de variables que queremos encontrar (en este caso 6). El total del espacio de soluciones será 84.

El módulo genético tiene los siguientes componentes:

*Generador de población inicial*

Este componente es el encargado de crear los individuos de la primera generación. Recibe como parámetro el número total de individuos a generar y las características del cromosoma, es decir, la longitud y posibles valores para cada gen (alelos). Crea de forma aleatoria el número de individuos solicitados y los envía al componente de evaluación de aptitud.

*Evaluador de aptitud*

Este componente se encarga de asignar una calificación a cada cromosoma dependiendo de su desempeño. Para determinar este valor se realiza el siguiente proceso para cada individuo: Se invoca el módulo de preselección utilizando los parámetros definidos por el cromosoma, este módulo genera las parejas candidatas de atributos equivalentes, las cuales son procesadas por el módulo clarificador que se encarga de definir el grado de similitud y el tipo de equivalencia. Una vez obtenidos estos valores se procede a compararlos con los resultados correctos y se generará una calificación dependiendo de la distancia existente entre ellos.

*Selector de sobrevivientes*

Este componente es el encargado de determinar qué cromosomas serán utilizados para el proceso de cruce y generación de descendencia. Recibe como parámetros la población de la generación actual y la cantidad de individuos que debe escoger. El método aplicado para determinar los sobrevivientes será el de la ruleta.

*Generador de descendencia*

Este componente se encarga de realizar los procesos de cruce y mutación. Recibe como parámetros los cromosomas supervivientes, el número total de descendientes que debe generar, y el porcentaje de cuántos de ellos deben ser producto de una mutación. Para el cruce, se escoge al azar una pareja de cromosomas de los cuales, se generan dos hijos, continuándose el proceso hasta completar la cantidad de hijos requeridos por cruce, posteriormente, elige al azar un cromosoma y genera un clon mutado, proceso que se realiza hasta completar la cantidad correspondiente de genes mutados.

*Controlador de evolución*

Este componente es el encargado de coordinar todo el proceso evolutivo e invocar a cada uno de los otros componentes. Al final de cada ciclo observa los resultados y decide cuándo detener el proceso. La condición de parada está compuesta por dos factores: la diferencia de calificación entre los mejores individuos de las últimas 2 generaciones y el valor de la función de aptitud del mejor individuo.

## Conclusiones

El desarrollo de este trabajo de investigación permite probar, más fácilmente, el efecto que tiene la utilización de nuevas características sobre un conjunto de datos a la hora de categorizarlos teniendo en cuenta la información suministrada por las demás características.

Existe un gran campo de acción para investigar y desarrollar nuevos mecanismos que permitan encontrar equivalencias semánticas de forma automática. La tendencia es utilizar métodos híbridos que incluyan heurísticas y técnicas de aprendizaje.

En este sentido, clasificar el tipo de equivalencia descubierta entre una pareja de atributos es de gran utilidad para el proceso de integración, ya que permite aplicar directamente la estrategia más adecuada para cada caso, sin requerir una exploración más profunda.

## Referencias


- HILERA GONZALEZ, José Ramón y MARTÍNEZ HERNANDO, Victor José (1995). *Redes neuronales artificiales: fundamentos y aplicaciones*. Estados Unidos de América: Addison-Wesley Iberoamericana. ISBN 020187895X.
- KOHONEN, Teuvo. *The self-organizin Map. Proceedings of the IEEE*. Vol. 78 (N° 9). Septiembre, 1990.
- LARSON, J., NAVATHE, S., and ELMASRI, R. (1989). *A theory of attribute equivalence in databases with application to schema integration.* IEEE Transactions on Software Engineering, .15(4):449–463.
- LI, Wen-Syan & Clifton, Chris (1994). *Semantic integration in Heterogeneous Databases Using Neural Networks*. Northwestern University. pp. 1-12. ISBN 1-55860-153-8.
- ———— (1995). *Semint: A system prototype for semantic integration in heterogeneous Databases.* Northwestern University. ISBN 0-89791-731-1.
- NIRMAL K., Bose & PING, Liang (1996). *Neural network fundamentals with graphs, algorithms and aplications*. Estados Unidos de América: McGraw-Hill. ISBN 0070066183.
- RIZOPOULOS, Nikos (2004). *Automatic discovery of semantic relationschips between schema elements*. Imperial College. pp. 3-8.
- YOUNG, M. (1989). *The Technical Writers Handbook*. University Science.