

# METODOLÓGIA PARA LA CONSTRUCCIÓN DE UN MIGRADOR UNIVERSAL DE BASES DE DATOS

## BUILDING METHODOLOGY OF A UNIVERSAL DATABASE MIGRATOR

JAVIER FERNANDEZ

*Universidad Cooperativa de Colombia, Seccional Medellín, Director Centro de Investigaciones, ingenierojdf@gmail.com*

JUAN CAMILO GIRALDO

*Universidad Cooperativa de Colombia, Seccional Medellín, ingenierojcg@gmail.com*

Recibido para revisar mayo 4 de 2009, aceptado abril 21 de 2010, versión final mayo 21 de 2010

**RESUMEN:** El artículo trata sobre la propuesta y aplicación de una Metodología para la Construcción de un Migrador de Bases de Datos; el cual pretende constituirse en una herramienta útil para quienes evalúan la eficacia de los procesos de migración de datos entre múltiples plataformas, proceso este que viene constituyéndose en pilar de las organizaciones modernas.

La finalidad es mostrar el proceso de construcción de un sistema de migración de bases de datos para empresas de bienes y servicios, en el se desarrollan los antecedentes conceptuales e investigativos inherentes a los sistemas de migración, la metodología propuesta para la construcción del prototipo planteado, así como los diagramas de casos de uso y clases que dan cuenta del análisis y diseño que soporta la aplicación, también se muestra la interfaz grafica construida para el manejo de los proceso de migración de datos.

**PALABRAS CLAVE:** Bases de Datos, Metodología, Migración, Migrador.

**ABSTRACT:** This article talks about the proposal and application of a methodology to build a database migrator that will serve as a very useful tool for those who evaluate the efficiency of data migration processes between multiple platforms, these processes are now being identified as emerging pillars of modern organizations.

The main purpose is to show the building process of a database migration system geared towards consumer goods and services companies; it involves the development of the conceptual background and research involved on migration systems, the proposed methodology to build the suggested prototype, as well as the use case and class diagrams that demonstrate the analysis and design supported by the application, it also exposes the graphical interface built to manage the database migration processes.

**KEYWORDS:** Databases, Methodology, Migration, Migrator.

### 1. INTRODUCCIÓN

El presente trabajo muestra el proceso de construcción de un sistema de migración de bases de datos en empresas de bienes y servicios, el prototipo construido parte del reconocimiento de los desarrollos presentes en las técnicas y metodologías de migración de datos con el propósito de servir Como marco metodológico para futuros desarrollos

en el área; en él se explican las técnicas utilizadas en el análisis, diseño y programación, como son los diagramas de caso de uso y de objetos de UML para el proceso de análisis y diseño y la programación en Java para el desarrollo de las interfaces y la migración multiplataforma de datos, contribuyendo así al desarrollo de la

comunidad de administradores de bases de datos con una herramienta soportada para el refinamiento y especificaciones de calidad en los procesos de migración de datos.

En la segunda sección se muestran los antecedentes teóricos más relevantes como soporte al sistema propuesto, en la tercera sección se muestra la metodología utilizada en el proceso de construcción del prototipo, en la cuarta sección se muestra el funcionamiento del sistema propuesto y en la quinta sección se hace una descripción sobre la aplicación y proyección del sistema.

## 2. ANTECEDENTES

Una mirada al estado actual sobre los sistemas de migración de bases de datos nos lleva necesariamente a hacer una revisión de conceptos y desarrollos logrados en el marco de aplicación de este tipo de sistemas propuestos.

### 2.1. Antecedentes conceptuales

En la medida en que los sistemas computacionales de procesamiento de datos se hicieron más importantes, las empresas comenzaron a reconocer que la información era un recurso corporativo de valor considerable. Estas percibieron que los datos necesarios para contestar numerosas preguntas estaban disponibles en sus archivos de procesamiento de datos. Como consecuencia, comenzaron a presionar a los sistemas de información en cuanto a la utilización de la máxima potencia del computador para producir información a partir de los datos corporativos. Esto inició la demanda de los sistemas de bases de datos, los que garantizarían más efectivamente el acceso a los datos y su manipulación [1].

A mediados de los sesenta (60's) se introdujeron los primeros sistemas de bases de datos, cuyo fundamento era una estructura jerárquica de los datos. Estos sistemas permitieron la recuperación de múltiples registros asociados con un registro único de otro archivo. Inmediatamente después, se desarrollaron los sistemas de base de datos en redes que soportaron interrelaciones entre registros de archivos diferentes mucho más complejas. Ambos modelos de base de datos, el jerárquico y el de red,

requirieron el uso de punteros físico predefinidos para enlazar los registros relacionados [1].

En 1970, Codd revolucionó el pensamiento en la industria de las bases de datos. El enfoque de Codd proponía el acceso y la manipulación de los datos únicamente desde el punto de vista de sus características lógicas. Durante los años setenta y ochenta se desarrollaron numerosos sistemas de bases de datos relacionales y, en la actualidad, éstos dominan el mercado comercial. [1].

Las Bases de Datos Relacionales (RDBMS) dominaron el mercado de las bases de datos y de los sistemas de información de las organizaciones, permitiéndoles a éstas últimas automatizar sus procesos más importantes. La metodología más utilizada para modelar los datos de estas aplicaciones es la Entidad / relación y sus extensiones. Los tipos de datos que se modelan en las Bases de Datos relacionales son del tipo atómico, a los que se refieren frecuentemente como "simplemente estructurados". La tecnología logró gran aceptación desde mediados de la década de los 80's, pero las ventajas competitivas logradas con los RDBMS han disminuido con el transcurso del tiempo, fundamentalmente en los últimos cinco años. Para lograr ventajas competitivas en nuestros días las organizaciones necesitan aplicaciones del tipo Internet / intranet y un conjunto más extenso y complejo de tipos de datos que permitan el manejo de datos estructurados complejos, relaciones de composición y herencia así como la administración de imágenes, videos, sonidos, series de tiempo y datos geoespaciales [1].

Las tecnologías que surgieron para manejar estos nuevos tipos de datos y las relaciones más complejas que existen entre los mismos son las Bases de Datos Orientadas a Objetos (ODBMS) y las Bases de Datos Objeto Relacionales (ORDBMS) [2].

Los ORDBMS surgen como una respuesta a la incorporación de la tecnología de objetos en las Bases de Datos Relacionales y con ello

permiten el tratamiento de datos y relaciones complejas. Por ser una extensión de la tecnología relacional presenta dos ventajas frente a los ODBMS: que es compatible con la tecnología relacional y tiene un mejor soporte para aplicaciones voluminosas [2].

### *2.1.1 Bases de datos*

Una Base de Datos es una serie de tablas ordenadas, estructuradas y relacionadas entre sí que contienen información y facilitan el acceso a la misma según criterios específicos. Las bases de datos generalmente tienen índices asociados a alguna de sus columnas, de forma que el acceso sea lo más rápido posible.

Las bases de datos se utilizan para almacenar información y suministrar esta como servicios a muchas aplicaciones. Estas presentan características similares de software como es analizar y relacionar la información, permitir su organización y asociación en una plataforma en la cual se ejecutan aplicaciones en una organización.

Es importante destacar también el diseño e implementación que requieren las bases de datos ya que sin un conjunto de requisitos previos sería imposible organizar la información y representar las relaciones. [2]

Dentro de las bases de datos encontramos modelos lógicos como: el jerárquico, el de red, el relacional y el objetual, los cuales tienen ciertas ventajas de procesamiento y de negocios.

Existen otros tipos de bases de datos como documental, distribuidas, orientadas a objetos e hipermedia (que integra gráficos, sonidos y video) y tienen como función derivar, almacenar y procesar datos dentro de una información, al igual que crear consultas e informes rápidos y personalizados. Se pueden crear consultas para realizar diferentes operaciones con los datos obtenidos.

También se pueden utilizar para controlar las instrucciones necesarias, para seleccionar los registros y campos requeridos de una base de datos.

### *2.1.2 Bases de datos de fichero*

Las bases de datos de fichero plano consisten en ficheros de texto divididos en filas y columnas. Estas bases de datos son las más primitivas y quizás ni tan siquiera merezcan considerarse como tales. Pueden ser útiles para aplicaciones muy simples, pero no para aplicaciones medianas o complejas, debido a sus grandes limitaciones.

### *2.1.3 Bases de datos relacionales*

Las bases de datos relacionales son las más populares actualmente. Su nombre proviene de su gran ventaja sobre las bases de datos de fichero plano: la posibilidad de relacionar varias tablas de datos entre sí, compartiendo información y evitando la duplicidad y los problemas que ello conlleva (espacio de almacenamiento y redundancia). Existen numerosas bases de datos relacionales para distintas plataformas (Access, Paradox, Oracle, Sybase) y son ampliamente utilizadas. Sin embargo, tienen un punto débil: la mayoría de ellas no admite la incorporación de objetos multimedia tales como sonidos, imágenes o animaciones [3].

### *2.1.4 Bases de datos orientadas a objetos*

Es un manejador de base de datos que soporta el modelaje y la creación de datos como objetos. En la actualidad, las arquitecturas tradicionales de un sistema de información delegan la responsabilidad de la persistencia de los datos a un motor de base de datos relacional, mientras que en la parte lógica, y en general en la ejecución del sistema, se mantienen familias de objetos para encapsular las entradas y mantener el estado de las operaciones en dichos sistemas. Esto obliga a siempre buscar alternativas para extraer la información contenida en una familia de objetos y adaptarla de tal forma que se pueda almacenar el estado de éstos en tablas de un modelo relacional.

Las bases de datos O.O. hacen transparente este proceso, permitiendo hacer la

persistencia, consulta y reconstrucción de objetos de forma inmediata [4].

Las bases de datos orientadas a objetos incorporan el paradigma de la Orientación a Objetos (OO) a las bases de datos. La base de datos está constituida por objetos, que pueden ser de muy diversos tipos, y sobre los cuales se encuentran definidas unas operaciones. Las bases de datos orientadas a objetos pueden manejar información binaria (como objetos multimedia) de una forma eficiente. Su limitación suele residir en su especialización, ya que suelen estar diseñadas para un tipo particular de objetos (por ejemplo, una base de datos para un programa de CAD).

### *2.1.5 Bases de datos híbridas*

Las bases de datos híbridas combinan características de las bases de datos relacionales y las bases de datos orientadas a objetos. Manejan datos textuales y datos binarios, a los cuales se extienden las posibilidades de consulta. Es una tecnología reciente y aún existen pocas en el mercado.

### *2.1.6 Migración de Datos*

La seguridad e integridad de los datos son esenciales para el buen funcionamiento y la toma de decisiones dentro de una organización; por esta razón es indispensable contar con una Base de Datos confiable. Para realizar una migración de Bases de Datos, se debe realizar un estudio previo para poder identificar cuáles son los mejores pasos a seguir para cada migración en particular: analizando los riesgos, creando planes de contingencia, y tomando todos los recaudos necesarios para que la migración sea un éxito.

El proceso de la migración puede ser bastante complejo y, como hay tantas bases de datos distintas, resultaría difícil “dar una receta que funcione en todos los casos”. Además, aparte de la dificultad de transferir la información entre los dos sistemas gestores de base de datos, también nos influirá mucho en la complejidad del problema el tipo de los datos de las tablas que estamos utilizando. Por ejemplo, las fechas, los campos numéricos con decimales o los boléanos pueden dar problemas al pasar de un sistema a otro porque pueden almacenarse de maneras distintas o, en el caso de los números, con una precisión distinta.

Al actualizar a una nueva versión de una base de datos o de una aplicación, o al cambiar a un nuevo sistema, los datos necesitan ser preservados en este nuevo sistema. El propósito de la migración de datos es transferir datos existentes al nuevo ambiente. Es decir, transformar los datos a un formato conveniente para el nuevo sistema, mientras que se preserva la información presente en el viejo sistema.

### *2.1.7 Casos en los que es necesaria la migración de datos*

Hay circunstancias muy particulares que hacen necesaria la migración de la plataforma donde se encuentran los datos de una empresa. La migración la sugerirá el departamento técnico o de sistemas, también podrá ser sugerida luego de una consulta sobre algún problema específico. De todas formas, aquí se muestran algunos casos en los que es posible detectar, a priori, la necesidad de migrar:

- Se ha vuelto muy lenta la carga y consulta de datos
- Se debe mantener compatibilidad hacia arriba con otros productos
- El volumen de información que se maneja necesita mayor robustez de la base de datos
- Se desea mantener actualizados los productos
- Es necesario cambiar hacia un producto de base de datos sin licencia
- El actual Sistema Operativo -S.O.- esta desactualizado
- No es posible usar los programas nuevos
- Se necesita bajar los gastos en tecnología
- El Sistema Operativo actual no es confiable desde que se utiliza Internet
- Se desea armar una red
- Se han comprado nuevos equipos

### *2.1.8 ¿De qué consta el proceso de migración?*

Los procesos que involucra la migración varían dependiendo, entre otros factores, del producto actual, de si la decisión de la nueva base había sido tomada o si es necesario plantear diferentes alternativas, entre otros.

A grandes rasgos, el proceso de migración será así:

En primer lugar se analizará la factibilidad de la migración.

Si la migración es posible, se hace un backup de la información que contengan los equipos y luego se procede a realizar la migración.

Luego se recuperará la información anterior y por último se verificará que todo funcione adecuadamente así como la integridad de la información recuperada.

### *2.1.9 ¿Cuál es el resultado?*

Las migraciones traen como resultado el mejor aprovechamiento de la tecnología que se posea y la mayor velocidad de acceso a la información. En muchos casos y dependiendo del tipo de migración, también se logra mayor seguridad y versatilidad.

### *2.1.10 Migración universal*

La migración universal es el proceso de traspaso de información (datos) entre diferentes motores de bases de datos; al hablar de ‘universal’, nos referimos a la interacción que puede existir entre diferentes motores trabajando sobre diferentes plataformas.

Se desarrolla con el fin de suplir las necesidades puntuales que puede surgir en cada compañía ya que cada una de estas cuenta con diferentes motores de almacenaje de sus datos.

### *2.1.11 Superar Plataformas*

Un Migrador universal de Bases de datos busca superar las limitaciones que se tienen actualmente con los migradores que existen en el mercado, ya que actualmente es posible realizar migraciones entre diferentes motores de bases de datos siempre y cuando se trabaje sobre una misma plataforma.

El Migrador para bases de datos relacionales tiene una solución multiplataforma, la cual sea fácilmente adaptable a cualquier necesidad de migración. Se entiende como plataformas no solo el extenderse a diferentes sistemas operativos (Linux, Microsoft, etc.), sino también a diferentes motores de base de datos (Microsoft SQL Server, Oracle, Sybase,

Informix, MySQL, Rational Rose, Postgress, IBM DB 2, Microsoft Access).

## **3. METODOLOGIA**

El prototipo del Migrador Universal de Bases de Datos se implementó utilizando la tecnología del lenguaje de programación Java, puesto que es una herramienta que nos permite interactuar entre diferentes plataformas y aporta en gran cantidad a superar este límite que actualmente aqueja a los migradores existentes.

Java es un lenguaje de programación orientado a objetos el cual ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de los demás lenguajes de programación, al ser este un lenguaje más simple, usable y de fácil interacción.

Para el Proceso de conexión de la interfaz de migración con las bases de datos origen y destino se desarrollo sobre el API de Java conocido como JDBC (Java Database Connectivity) el algoritmo de migración propuesto, este API (Interfaz de Programación de Aplicaciones) permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede.

Con este API y con el algoritmo de migración propuesto se puede realizar cualquier tipo de tareas con la base de datos a las que tenga permiso: consulta, actualización, creación, modificación y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc.

Por último, tanto la interfaz en Java como la API y el algoritmo utilizado permiten crear las estructuras necesarias para el proceso de migración de datos como se explicara a continuación:

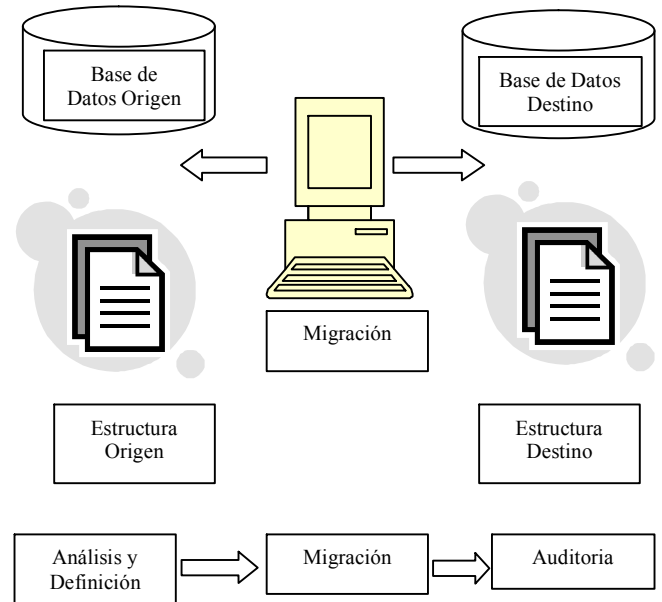
Para la construcción de un sistema de migración de bases de datos que logre unificar las plataformas de procesamiento de datos se hace necesario dar respuesta al siguiente interrogante investigativo ¿Existe una metodología unificada que permita migrar información entre múltiples plataformas de bases de datos sin perder la integridad de los datos?, al ser enfrentado este interrogante se ha propuesto un esquema de trabajo que permite brindar pautas para la construcción de una *MUPM* (Metodología Unificada de Procesos de Migración), este esquema viene representado a través de las siguientes fases:

Fase de análisis y definición de procesos de migración y reconocimiento del origen: Implementando un algoritmo de conexión a la base de datos destino se establece el “puente” entre el Migrador y el destino de datos.

En ella se define la estructura de los datos destino y las características técnicas de la plataforma hacia la cual se migraran los datos; al tener claramente definida la estructura destino se procede a la conexión del Migrador con la fuente de datos de destino.

Fase de migración: Implementando un algoritmo de reconocimiento de estructuras se procede al proceso de migración de los datos sobre el “puente” construido en la fase previas.

Una vez se ha establecido la conexión con el origen y el destino de los datos se procede a realizar la tercera fase de Migración de los Datos. Con las estructuras conectadas el aplicativo Migrador, hace un reconocimiento a ambas estructuras y procede a generar un archivo interno con los datos origen convertidos a la estructura destino, tras lo cual se descargaran en la plataforma destino de la migración los datos migrados y validados estructuralmente, como se puede apreciar en la grafica siguiente:



**Figura 1.** Esquema de Migración de la Metodología Propuesta  
**Figure 1.** Proposed Methodology Migration Schema

En este procesamiento de las bases de datos se soporta un fuerte esquema de control y análisis de datos, en el cual mediante técnicas propias de la administración de las bases de datos, se consolida la información que interviene en los procesos de migración.

Una vez soportado el esquema de migración, se procede a la fase de “Auditoría de la Información”.

Fase de auditoría de la información: El Migrador permite generar los informes de consistencia de la información migrada, con el fin de detectar posibles errores, como duplicidades o falta de datos.

Una vez presentado el esquema Macro del sistema propuesto, se procede, haciendo uso del UML – Lenguaje Unificado de Modelado-, a presentar el Modelo para la construcción del sistema Migrador de bases de datos propuesto.

3.1. Análisis y diseño

El sistema de Migración de Datos, como vimos anteriormente presenta un esquema general que se basa en los siguientes requerimientos funcionales:

- Hacer compatibles las plataformas de origen y destino de datos.
- Establecer la conexión con la fuente de datos.
- Capturar la información del origen de datos.
- Establecer la conexión con el destino de los datos.
- Migrar la información.
- Validar la información migrada
- Generar el informe de consistencia de la migración

Diagrama de Casos de Uso

Los casos de uso tratan sobre el funcionamiento del sistema en términos de sus requerimientos funcionales, ayudando en la comprensión de las necesidades del usuario y brindando información suficiente y necesaria sobre las interfaces de comunicación con el usuario final del sistema.

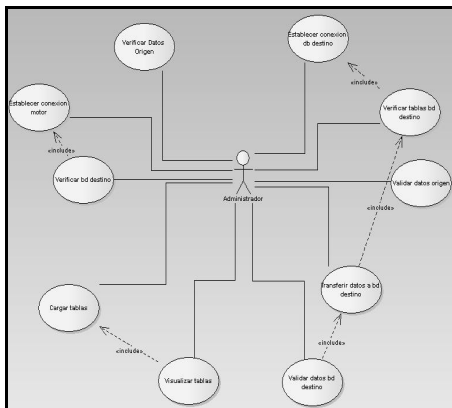


Figura 2. Diagrama de Casos de Uso del Sistema  
Figure 2. Use Case System Diagram

Diagrama de Clases

Las clases en UML representan las cosas, personas, documentos, maquinas o subsistemas que poseen características y realizan operaciones a través de sus relaciones al interior del sistema de información.

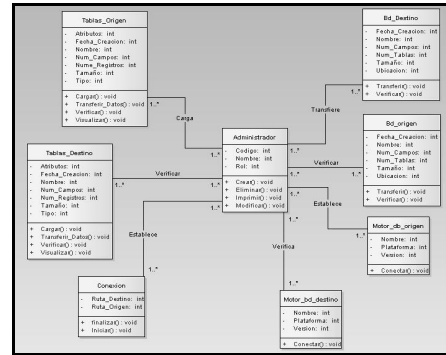


Figura 3. Diagrama de Clases del Sistema  
Figure 3. System Class Diagram

3.2. Construcción del Prototipo

Una vez superada la fase de análisis y diseño del sistema propuesto, se procede con el proceso de implementación o codificación.

Haciendo uso del Lenguaje de programación Java, se ha construido una interfaz o modulo del Migrador de Bases de Datos, como se muestra en el siguiente grafico:

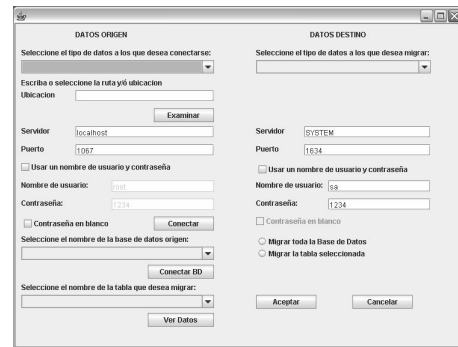
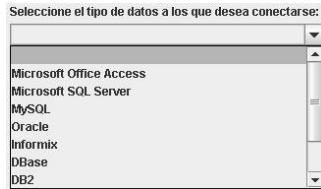


Figura 4. Representación del Sistema Migrador de Bases de Datos  
Figure 4. Illustration of the Database Migrator System

Al iniciar el programa se carga el anterior Frame con las siguientes funcionalidades:

**Selección del tipo de destino**

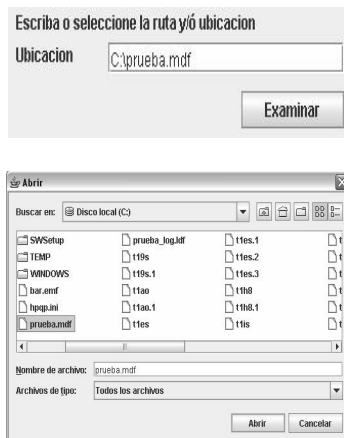
En esta sección debemos seleccionar el motor de base de datos al cual vamos a conectarnos, este motor es el que actualmente se tiene y sobre el que se va a ejecutar el proceso de migración.



**Figura 5.** Representación de la Opción de Selección del Tipo de Destino  
**Figure 5.** Illustration of the Destination Type Selection Option

**Selección de ruta y/o ubicación de la base de datos**

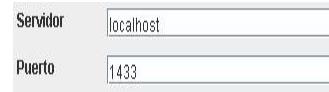
En esta sección ubicamos la ruta donde se encuentra nuestra base de datos a migrar, para esto, podemos escribir manualmente la ruta o presionar el botón examinar el cual nos traerá una ventana de selección como se puede observar a continuación.



**Figura 6.** Representación de la Opción de Selección de la Ruta y la Ubicación de la Base de Datos  
**Figure 6.** Illustration of the Database Route and Location Selection Option

**Seleccionar servidor y puerto de conexión**

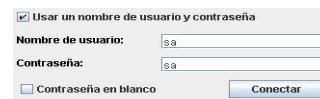
Se debe digitar la dirección y/o nombre del servidor donde se encuentra el motor a migrar, de igual manera el puerto que se está utilizando para la conexión al motor.



**Figura 7.** Representación de la Opción de Selección del Servidor y el Puerto de Conexión  
**Figure 7.** Illustration of the Server and Port Connection Selection Option

En esta sección marcamos el campo de “Usar un nombre de usuario y contraseña” siempre y cuando la base de datos lo requiera, si esta opción se encuentra deshabilitada se ingresaría con la autenticación del Sistema Operativo.

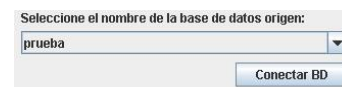
Por último, presionamos el botón conectar para realizar el proceso de conexión.



**Figura 8.** Representación de la Configuración de la Conexión  
**Figure 8.** Illustration of the Connection Configuration

**Selección de base de datos de origen**

Luego de establecer la conexión seleccionamos la base de datos que se va a migrar, para esto utilizamos el menú desplegable el cual nos va a mostrar las bases de datos disponibles. Luego continuamos presionando el botón Conectar BD para establecer la conexión de esta base de datos.

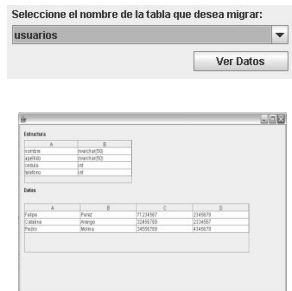


**Figura 9.** Representación de la Opción de Selección de la Base de Datos Origen  
**Figure 9.** Illustration of the Source Database Selection Option



### Selección de la(s) tabla(s) a migrar

En esta sección seleccionamos la tabla o tablas que vamos a migrar, al presionar sobre el botón “*Ver Datos*” podremos observar la estructura de la tabla así como los datos contenidos en esta.



**Figura 10.** Representación de la Opción de Selección de las Tablas a Migrar

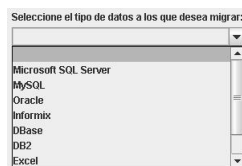
**Figure 10.** Illustration of the Migrating Tables Selection Option

### Selección destino

Esta sección se encuentra en el mismo Frame y es donde se seleccionara la base de datos destino donde quedaran almacenados los datos ya migrados.

### Selección de base de datos destino

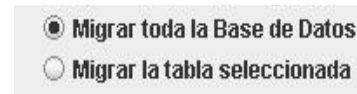
Seleccionamos el motor de base de datos al cual deseamos migrar la información, para esto contamos con una lista desplegable con el listado de motores disponibles.



**Figura 11.** Representación de la Opción de Selección de la Base de Datos Destino

**Figure 11.** Illustration of the Destination Database Selection Option.

Continuamos seleccionando una de las opciones de migración las cuales serian de migración de base de datos completa o migración de la tabla seleccionada.

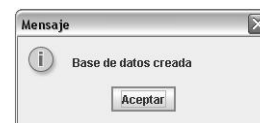


**Figura 12.** Representación de la Opción de Selección del Proceso de Migración

**Figure 12.** Illustration of the Migration Process Selection Option

Por último presionamos el botón aceptar el cual dará inicio al proceso de migración de la información.

Al finalizar el proceso de migración satisfactoriamente, nos aparecerá un mensaje confirmándonos la migración exitosa, “Base de datos creada”.



**Figura 13.** Representación del Cuadro de Dialogo del Resultado de la Migración

**Figure 13.** Illustration of the Migration Results Dialog Box

## 4. RESULTADOS

Durante el proceso de desarrollo se han ido evaluando permanentemente los elementos que entran a conformar el sistema propuesto, entre los eventos importantes que se han presentado durante este proceso de desarrollo del sistema tenemos:

Se realizo un analisis en terminos de determinacion de las necesidades que debera cubrir el proceso de migracion de datos para generar un esquema metodologico unificado como el propuesto.

Se han construido los diagramas mas representativos de UML para modelar el sistema propuesto, como con: el Diagrama de Casos de Uso y el diagrama de clases de los requerimientos funcionales mas representativos del sistema, haciendo una

revisión permanente sobre los cambios a realizar en el sistema.

Se ha construido un primer prototipo con el lenguaje de programación Java del sistema migrador de bases de datos.

Al compararse con migradores existentes en el medio como: My Great Data (migratedata) o las propuestas de Innovative Systems y Clever Components, encontramos que esta propuesta supera las anteriores en términos de: eficiencia del algoritmo para el reconocimiento de las múltiples estructuras de datos, completitud de la metodología soporte al uso del esquema de migración, portabilidad de la aplicación desarrollada ya que se puede correr en múltiples plataformas independientes del sistema operativo.

## 5. CONCLUSIONES

El proyecto desarrollado permitirá entre otros aspectos:

El Migrador universal como método, especifica y estandariza toda una secuencia de pasos lógicos para trasladar información entre diferentes motores de bases de datos instalados en algunos casos, en diferentes sistemas operativos.

El Migrador Universal, como proyecto dirigido desde la academia, potencializa la capacidad investigativa de los integrantes y los involucra en el ámbito de innovación al generar un prototipo que supera las capacidades de otros desarrollos creados para la misma área, que se limitan a una sola plataforma o sistema operativo y cuyos fines son específicos para una misma compañía.

Java como lenguaje de programación permite superar barreras que suponen diferentes plataformas y más específicamente los diferentes motores de bases de datos.

Para el proceso efectivo de migración de datos entre diferentes plataformas (sistemas operativos y motores) se hace necesario un algoritmo de verificación, el cual permite validar que el número de transacciones de una base de datos origen, sea el

mismo en la base de datos destino evitando así duplicidad de las mismas o traslado parcial de información, así como la validación de la estructura de los datos.

Una interfaz amigable, intuitiva y precisa, son elementos claves en el éxito del funcionamiento del prototipo ya que dirige a conexiones exitosas y en consecuencia traslado efectivo de la información.

El Migrador Universal de bases de datos, aportará a futuro mejoras en los diferentes procesos empresariales que involucren escalabilidad o actualizaciones de software y también ahorros económicos.

Tanto las herramientas de diseño utilizadas como las de programación han resultado adecuadas a los propósitos planteados en este proyecto, a un bajo costo, flexibles y adaptables, se ha podido construir un primer prototipo para el sistema propuesto.

## REFERENCIAS

- [1] GONZÁLEZ MARTÍN, ÓSCAR; "ARQUITECTURAS DE SISTEMAS DE BASES DE DATOS", Universidad de Castilla la Mancha, Escuela superior de informática, 1999/2000.
- [2] HERNÁNDEZ ORALLO, JOSÉ; "La Disciplina de los Sistemas de Bases de Datos. Historia, Situación Actual y Perspectivas", Departamento de Sistemas Informáticos i Computació, Universitat Politècnica de Valencia, 2002.
- [3] Codd, E. F; "The Relational Model for Database Management": Version 2. Reading, Mass. Los Altos, CA: Addison-Wesley Publishing, 1990.
- [4] CADAVID RENGIFO, HÉCTOR; "Bases de datos orientados a objetos", Maestría en ingeniería de Sistemas, Universidad Nacional de Colombia, 2007.