

Universidad de las Ciencias Informáticas

Facultad 10



*Módulo de extracción, transformación y réplica de datos del
Sistema para el Aseguramiento de la Actividad de Estructura y
Composición de las FAR versión 3.*

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor(es):

Julio Rodríguez Bermúdez

Tutor(es): 1 Tte. Raciél Garrido Torres

Consultante:

Tte. Alejandro Abiague Nápoles



"Vamos a ser fortísimos en la Computación (Informática), como ya lo estamos siendo en la Medicina, y no solo para beneficio de nuestro pueblo, sino de toda la Humanidad. Será también esta una poderosísima fuerza científica, económica y política del país en toda nuestra lucha por desarrollarnos."

Fidel Castro

DECLARACIÓN DE AUTORÍA

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Julio Rodríguez Bermúdez

Firma del Autor

Tte. Raciel Garrido Torres

Firma del Tutor

DATOS DEL CONTACTO

Teniente Ingeniero Raciél Garrido Torres graduado en la Universidad de las Ciencias Informáticas (UCI) en el año 2008 con diploma de oro. Desde su graduación está trabajando en la UCI, desempeñándose como Jefe de Línea de Estructura y Composición en la unidad 1722.

Email: rgarrido@uci.cu

Teniente Ingeniero Alejandro Abiague Nápoles graduado en la Universidad de las Ciencias Informáticas (UCI) en el año 2008. Se desempeña como especialista general en la Unidad de Compatibilización Integración y Desarrollo para la Defensa (UCID).

Email: aabiague@uci.cu

A mi madre y mi padre por siempre estar conmigo, darme su apoyo incondicional en todo momento y darme fuerzas para seguir adelante.

A mis otros padres, Mame y Cheo por todo el amor y cariño que siempre me han dado.

A mi hermano Xiovany por haberme dado una sobrina tan linda.

A Yakelyn que me quiere como un hermanito.

A mi flaquita Yumi por quererme tanto y por soportar todas mis pesadeces.

A toda mi familia de sangre que no menciono nombres para que no se me quede ninguno porque son muchos.

A mi otra familia en especial a Mita y Nana por darme parte de este cuerpo.

A mi tutor Raciél que más que un tutor fue mi compañero de tesis.

A mi piquete Dunia, Yane, Tato y Ernesto por compartir tantos momentos juntos en este tiempo a pesar de que dicen que soy un pesado y un sangrón.

A mi primo Yuri que no somos primos realmente pero nos queremos como si lo fuésemos.

A mis compañeros de aula por compartir este tiempo de universidad.

A mis amigos del barrio que me acompañan desde pequeño.

A mis amigos de la universidad.

A todos los que de una forma u otra me han apoyado en la realización de este trabajo.

A todos muchas gracias.

A mis padres que son la razón de mi ser.

A Mame y Cheo que siempre han estado conmigo.

A mis hermanos Yakeilyn y Xiovany que los quiero mucho.

Resumen

El presente trabajo se desarrolló en la Unidad de Compatibilización, Integración y Desarrollo para la Defensa (UCID) de la Universidad de las Ciencias Informáticas, centro en el cual se desarrollan una gran cantidad de proyectos para el Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR).

El objetivo principal de este trabajo es lograr una migración y réplica de datos sin perder los existentes. Específicamente se trata de seleccionar una herramienta que sea capaz de desarrollar la migración y réplica de información para la versión 3 del Sistema para el Aseguramiento de la Actividad de Estructura y Composición de las FAR.

Palabras claves:

Herramienta, proyectos, migración, réplica.

Índice

Introducción	1
Capítulo 1: Fundamentación teórica	3
1.1. Introducción	3
1.2. Extracción, transformación y carga de datos (ETL)	3
1.2.1. Extracción	3
1.2.2. Transformación.....	3
1.2.3. Carga.....	4
1.3. Migración de datos	4
1.3.1. Técnicas de Migración de Datos	5
1.3.2. ¿Por qué migrar datos?	5
1.3.3. Necesidad de migrar datos en las FAR	5
1.4. Réplica de datos	6
1.4.1. Ambientes de replicación	6
1.4.2. Conflictos de la replicación de datos.....	7
1.5. Sincronización.....	8
1.6. Tecnologías actuales.....	8
1.6.1. Software Propietario	8
1.6.2. Tendencia al Software Libre	8
1.6.3. Sistemas Gestores de Bases de Datos	9
1.6.4. SQL Server.....	10
1.6.5. PostgreSQL.....	10
1.6.6. PGAdmin III.....	11
1.6.7. PHP	11
1.6.8. Servidor web.....	12

1.6.9.	Apache	12
1.6.10.	WampServer	12
1.6.11.	Navegador web.....	12
1.6.12.	Mozilla Firefox	13
1.6.13.	Notepad++	13
1.6.14.	Proceso de Desarrollo y Gestión de Software	14
1.7.	Análisis de las herramientas ETL	15
1.7.1.	Talend Open Studio	15
1.7.2.	Bitool.....	16
1.7.3.	Benetl.....	16
1.7.4.	Scriptella	17
1.7.5.	RDBTool	17
1.8.	Conclusiones.....	18
Capítulo 2: Requerimientos	19
2.1.	Introducción	19
2.2.	Situación problemática.....	19
2.3.	Información que se maneja	19
2.4.	Funcionamiento de la herramienta RDBTool	20
2.5.	Requerimientos que debe cumplir un sistema de réplica.....	23
2.6.	Captura de requisitos	24
2.7.	Conclusiones.....	24
Capítulo 3: Análisis de la solución y prueba	25
3.1.	Introducción	25
3.2.	Procedimiento para migrar y replicar la base de datos	25
3.3.	Dependencia entre las tablas	25

3.3.1. Dependencias de la tabla nom_aristaeav.....	26
3.3.2. Dependencias de la tabla nom_calificador_cargo	26
3.3.3. Dependencias de la tabla nom_campoestruc	26
3.3.4. Dependencias de la tabla nom_grupocomple.....	27
3.3.5. Dependencias de la tabla nom_organo.....	27
3.3.6. Dependencias de la tabla nom_salario.....	28
3.3.7. Dependencias de la tabla nom_valor_defecto.....	29
3.3.8. Dependencias de la tabla nom_categcivil	29
3.3.9. Dependencias de la tabla nom_cargomilitar.....	29
3.3.10. Dependencias de la tabla nom_cargocivil	30
3.3.11. Dependencias de la tabla nom_gradomilit.....	31
3.3.12. Dependencias de la tabla nom_gradomarina	32
3.3.13. Dependencias de la tabla nom_tecnica.....	32
3.3.14. Dependencias de la tabla nom_modulotecnica	33
3.3.15. Dependencias de la tabla nom_filaestruc	34
3.3.16. Dependencias de la tabla nom_valorestruc	34
3.4. Deferred.....	35
3.4.1. Tablas con llaves foráneas referenciadas hacia ellas mismas.	35
3.5. Orden de los esquemas	37
3.6. Orden de las tablas	37
3.6.1. Ejemplo de tablas sin dependencia funcional	37
3.6.2. Ejemplo de tablas con dependencia funcional.....	38
3.7. Filtros	39
3.8. Migrar campo dominiofila	39
3.9. Modelo de datos.....	40

3.10.	Descripción textual de las tablas	41
3.11.	Diagrama de despliegue	53
3.12.	Prueba.....	55
3.13.	Conclusiones.....	57
Conclusiones	58
Recomendaciones	59
Referencias Bibliográficas	60
Bibliografía	63
Glosario de términos	65

Introducción

Desde finales del siglo XX, el mundo ha alcanzado un gran desarrollo en la informatización con la generación de nuevas tecnologías y el desarrollo de la Internet. Cuba está consciente, de que una sociedad para ser más eficaz, eficiente y competitiva debe aplicar la informatización en todas sus esferas y procesos, como parte del perfeccionamiento continuo al cual está sometida. Las Fuerzas Armadas Revolucionarias (FAR) como pilar de la sociedad, se traza una estrategia de informatización siguiendo los lineamientos del software libre, para asegurar las tareas principales de la defensa del país. El perfeccionamiento continuo de las FAR se ejecuta en el Departamento de Organización, generando un gran cúmulo de información, el cual informatiza sus procesos en un sistema para gestionar plantillas por proyectos denominado Sistema Informativo para el Apoyo a la Actividad de Estructura y Composición (SIDEC) en su versión 2.0 el cual está soportado por tecnologías propietarias desarrollado a principio del siglo XXI.

Con el surgimiento de la Universidad de las Ciencias Informáticas (UCI) se logran insertar cadetes de las FAR con el fin de convertirlos en oficiales especialistas en informática. Unos años más tarde se decide realizar un centro dentro de la UCI para el desarrollo de software en las FAR llamado Unidad de Compatibilización, Integración y Desarrollo para la Defensa (UCID).

Para alcanzar la soberanía tecnológica y dar solución a nuevos requerimientos se crea en el UCID un nuevo proyecto de desarrollo de la nueva versión del SIDEC v3, que profile los objetivos generales de las FAR respecto al software libre. Con el desarrollo del SIDEC v3 se hace necesaria la migración de la información existente hacia la nueva versión desarrollada, evitando su pérdida o un gasto de esfuerzo en agruparla nuevamente y una vez migrada, realizar réplicas a las distintas estructuras organizativas de menor nivel con que cuenta las FAR. Ver anexo 1.

La gran diversidad de información que se maneja es totalmente confidencial, es por ello que se requiere de mucha disponibilidad, calidad y seguridad por parte de las aplicaciones informáticas destinadas a la informatización de los procesos.

Teniendo en cuenta lo expresado anteriormente, se plantea el siguiente **problema a resolver**:

¿Cómo garantizar persistencia de la información almacenada anteriormente en la versión 2 del Sistema para el Aseguramiento de la Actividad de Estructura y Composición de las FAR (SIDEC) en la nueva versión del sistema?

De esta forma, se define el **objeto de estudio** como las tecnologías de extracción, transformación y réplica de datos; y el **campo de acción** como la transformación y réplica de datos del SIDEC versión 2 para la migración a la versión 3.

Para la realización de esta investigación se ha planteado la siguiente **Idea a defender**: Si se desarrolla la migración y réplica de información para la versión 3 del Sistema para el Aseguramiento de la Actividad de Estructura y Composición de las FAR, entonces se contará con una solución integral para asegurar la persistencia de los datos anteriormente almacenados en la versión 2.

Basado en lo expuesto se define como **objetivo general** de esta investigación: Desarrollar la migración y réplica de información para la versión 3 del Sistema para el Aseguramiento de la Actividad de Estructura y Composición de las FAR.

Para lograr el Objetivo general se proponen los siguientes **objetivos específicos**:

- Realizar una investigación de los sistemas de extracción, transformación y réplica de datos.
- Realizar un análisis de la información a migrar y replicar por el sistema.
- Desarrollar la transformación y réplica de los datos.
- Realizar pruebas a la solución.

La estructura del documento está constituida de la siguiente manera:

En el Capítulo 1. Fundamentación teórica: Se realiza un estudio de los principales sistemas de extracción, transformación, carga y réplica de datos, se describen las herramientas y tecnologías a utilizar para el desarrollo del trabajo.

En el Capítulo 2. Requerimientos: Se realiza un análisis de los requerimientos a tener en cuenta para realizar la réplica de datos.

En el Capítulo 3. Análisis de la solución y prueba: Se realiza una descripción detallada de todo el proceso que se llevó a cabo para la realización de la réplica y se describen las pruebas realizadas a la solución.

Capítulo 1: Fundamentación teórica

1.1.Introducción

En el presente capítulo se brinda una panorámica actual de las soluciones existentes del problema planteado, tanto a nivel internacional como en la Fuerzas Armadas Revolucionarias (FAR) de Cuba. Se fundamentan las principales herramientas, metodologías de desarrollo y software utilizados en la solución propuesta del problema y se describen las tecnologías actuales de desarrollo utilizados para dar solución a la propuesta.

1.2.Extracción, transformación y carga de datos (ETL)

ETL son siglas en inglés (Extract-Transform-Load) que significan Extraer, Transformar y Cargar. Proceso que organiza el flujo de los datos entre diferentes sistemas en una organización y aporta los métodos y herramientas necesarias para mover datos desde múltiples fuentes a un almacén de datos, reformatearlos, limpiarlos y cargarlos en otra base de datos, data mart ó bodega de datos. La idea es que una aplicación ETL lea los datos primarios de unas bases de datos de sistemas principales, realice transformación, validación, el proceso cualitativo, filtración y al final escriba datos en el almacén y en este momento los datos son disponibles para analizar por los usuarios. (1)

1.2.1. Extracción

Consiste en extraer los datos desde los sistemas de origen. Analiza los datos extraídos, lo que resulta un chequeo que verifica si estos datos cumplen la pauta o estructura que se esperaba y los convierte a un formato listo para iniciar el proceso de transformación. De no cumplirse esto, los datos son rechazados. Un requerimiento importante en esta fase es que cause un impacto mínimo en el sistema origen, lo que quiere decir que si los datos a extraer son muchos, el sistema de origen podría ralentizarse o colapsar provocando que no pueda utilizarse con normalidad.

1.2.2. Transformación

En esta fase se aplican una serie de reglas de negocio o funciones sobre los datos extraídos para convertirlos en datos que serán cargados. Algunas de las transformaciones que se pueden realizar son las siguientes:

- Seleccionar ciertas columnas para su carga (por ejemplo, que no se carguen las columnas con valores nulos).
- Traducir códigos (por ejemplo, si el origen tiene almacenado una “H” para Hombre y una “M” para Mujer, pero el destino tiene que guardar un “1” para Hombre y un “2” para Mujer).

- Codificar los valores libres (por ejemplo, convertir Hombre en “H” o Sr en “1”).
- Generar campos clave en el destino.
- Trasponer o pivotar (girar columnas en filas o viceversa).
- Validar los datos (si están bien, se entregan a la otra fase, si no, se ejecutan políticas de tratamiento de excepciones).

1.2.3. Carga

La fase de carga es el momento en el cual los datos de la fase anterior (transformación) son cargados en el sistema destino, sobrescribiendo la información antigua por datos nuevos o manteniendo un historial de los registros, esto depende del tipo de base de dato.

Existen dos formas básicas de desarrollar el proceso de carga:

- **Acumulación simple:** consiste en realizar un resumen de todas las transacciones comprendidas en el periodo de tiempo seleccionado y transportar el resultado como una única transacción hacia la BD.
- **Rolling:** se almacena información resumida en varios niveles, correspondientes a distintas agrupaciones de la unidad de tiempo o diferentes niveles jerárquicos en alguna o varias de las dimensiones de la magnitud almacenada.

1.3. Migración de datos

Cuando se cambia una base de datos o una aplicación existente a una nueva versión es necesario preservar los datos en esta última, para ello se utiliza la migración de datos, proceso por el cual grandes magnitudes de datos son trasladados de sistemas existentes hacia nuevos sistemas.

La migración de datos se realiza mediante las siguientes etapas:

Levantamiento de Información: contempla la definición de los datos que requiere el nuevo sistema y la identificación de la fuente de origen.

Análisis de Calidad: verifica la calidad de los datos para establecer los tiempos reales de las etapas de Transformación, Limpieza y Mapeo de Datos.

Transformación/Mapeo de datos: contempla la definición de los diccionarios con las reglas de transformación y el mapeo de campo entre las fuentes de origen y destino.

Limpieza de Datos: contempla la depuración de los datos en las fuentes de origen.

Construcción ETCL: se realiza la construcción de los extractores, transformadores e inyectores de datos.

Pruebas: se realizan las pruebas funcionales de la migración de datos, verificando el éxito de la misma.

Simulación de cargas: se pretende identificar los inconvenientes que se podrían tener en las cargas reales, a través de la simulación del proceso real.

Cargas Reales: se realiza la migración real de los datos del antiguo al nuevo sistema.

Revisión de Resultados: se realiza la revisión final de la información cargada al nuevo sistema.(2) Ver anexo 2.

1.3.1. Técnicas de Migración de Datos

- **Planeación:** Lo más importante al migrar una base de datos es llevar a cabo un proceso de planeación y análisis del trabajo, puesto que aunque pareciera tomarse algún tiempo adicional, éste será retribuido en el éxito de la operación y menos costos por errores de datos. Es importante que esto sea aplicado cuando la base de datos destino está en producción.
- **Contador de registros:** Si la migración se realiza de forma manual, mediante alguna consulta de inserción es recomendable inicializar un contador para cada registro insertado con éxito y otro para los no insertados, así obviamente, la suma de ambos debe ser igual a los registros originales.
- **Mapeado de Tipos de datos:** Algunas plataformas no soportan algunos tipos de datos, así que es necesario planificar el mapeo de los campos en la nueva base de datos.
- **Restricciones y Triggers:** Antes de iniciar la migración de la BD, es recomendable deshabilitar los Triggers y/o restricciones.
- **Codificación de Caracteres:** Cuando el copiado se realiza de forma automática, es necesario identificar la codificación de caracteres que la BD destino espera, pues así evitaremos el reemplazo automático de caracteres o en su caso, pérdida de los mismos.

1.3.2. ¿Por qué migrar datos?

Existen varios motivos para hacer una migración:

- Mejorar el desempeño de la base de datos.
- Cumplir con nuevos requerimientos de usuario, de la aplicación o políticas de seguridad.
- Realizar actualización de versiones.
- Realizar estandarización de la tecnología de información en la organización.
- Reducir los costos que se pueden tener al cambiar por Software Libre.
- El aumento en el volumen de datos.

1.3.3. Necesidad de migrar datos en las FAR

Actualmente en nuestro país, el MINFAR tiene implementado el Sistema para el Aseguramiento de la Actividad de Estructura y Composición de las FAR, el mismo está montado sobre SQL Server, un gestor de base de datos propietario muy costoso. Esto lleva a la necesidad de migrar la base de datos existente a PostgreSQL, un sistema de gestión de base de datos sobre una plataforma libre, obteniendo como resultado no solo reducir los costos sino también actualizar el sistema hacia una nueva versión y lograr una estandarización de la tecnología utilizada por las diferentes estructuras organizativas de las FAR.

1.4. Réplica de datos

La replicación de datos consiste en el transporte de datos entre dos o más servidores, permitiendo que ciertos datos de la base de datos estén almacenados en más de un sitio, y así aumentar la disponibilidad de los datos y mejorar el rendimiento de las consultas globales.

La réplica de datos puede aplicarse de dos maneras diferentes de acuerdo a las necesidades propias de las personas o instituciones que la lleven a cabo, estas son:

Maestro-Esclavo: También es conocido como de solo lectura, porque permite a un solo sitio (maestro) realizar consultas de escritura sobre los demás, mientras estos solo pueden hacer consultas de lectura (esclavos). Ver anexo 3.

Maestro-Maestro: También llamado par-a-par o réplica de camino de n, porque permite múltiples sitios (maestros), actuando como pares iguales. Cada sitio es un sitio maestro, y se comunica con otros sitios maestros. Esta capacidad tiene también un severo impacto en el desempeño debido a la necesidad de sincronizar los cambios entre todas las partes que intervienen en la réplica. Este tipo de entorno puede ser usado para mantener sitios recuperables ante posibles desastres o caídas, así como para proveer sistemas con alta disponibilidad y para balancear la carga de consultas a través de las distintas ubicaciones. Ver anexo 4.

1.4.1. Ambientes de replicación

Un ambiente de replicación es una configuración de dos o más sitios mediante un escenario par a par. Cada sitio es un par que contiene un motor de replicación y una BD simple o compartida. El motor de replicación puede residir en la misma computadora que la BD asociada o en una computadora separada. En cada caso, la BD debe ser accesible por el cliente del motor de replicación. Cada sitio almacena solo los datos que requieren los usuarios locales. En segundo plano, el motor de replicación gestiona los cambios realizados a la BD sincronizando las actualizaciones de los datos con otros sitios activos en la red. Una red de replicación puede ser de los siguientes tipos:

- **Homogénea:** Se replican datos entre BD con gestores y plataformas del mismo tipo (ej. SQLServer + Windows <-> SQLServer + Windows).
- **Homogénea con diferentes plataformas:** Se replican datos entre BD con gestores del mismo tipo y plataformas diferentes (ej. PostgreSQL + Windows <-> PostgreSQL + Linux).
- **Heterogénea:** Se replican los datos entre BD con gestores diferentes y sin importar la plataforma (ej. SQLServer + Windows <-> PostgreSQL + Linux).

Una red de replicación requiere una estructura básica TCP/IP que posibilite una comunicación efectiva y eficiente entre todos los sitios. La red de replicación por sí misma constituye estructuralmente una red virtual que se coloca por encima de la red física. (3)

1.4.2. Conflictos de la replicación de datos

Al realizar una réplica de datos hay que tener en cuenta varios tipos de conflictos, los mismos deben ser considerados y tratados.

- **Conflicto de actualización:** ocurre cuando dos transacciones de distintos sitios actualizan el mismo registro en forma cercana de tiempo.
Se resuelve priorizando los servidores, o considerando correcta la más nueva o vieja de acuerdo al tiempo, o garantizando que cada registro sea manipulado por un único servidor.
- **Conflicto de unicidad:** ocurre por ejemplo cuando dos transacciones de distintos sitios intentan insertar un registro con el mismo valor de la llave primaria (identificador de cada tupla).
Se resuelve brindando un rango distinto de números de llaves primarias para cada servidor, o agregando a la llave el identificador del servidor, o replicando en tablas separadas.
- **Conflicto de supresión:** es cuando una transacción intenta borrar un registro y otra intenta modificarlo o borrarlo.
Una posible solución es que desde los sitios replicados no se puedan borrar físicamente los datos, estos se marcarían y eliminarían posteriormente mediante una función ejecutada periódicamente en el servidor.
- **Conflicto de orden:** puede ocurrir en ambientes con tres o más sitios maestros cuando uno X está bloqueado y las transacciones de modificación siguen propagándose a los otros, al final las modificaciones debieron propagarse al sitio X en orden distinto a como ocurrió en los demás, provocando conflictos.

Suele resolverse aplicando distintas prioridades a los sitios maestros para ordenar las transacciones. (4)

1.5.Sincronización

En muchos casos, en los sistemas de información, los datos se administran de forma independiente por múltiples aplicaciones o BD. Sin embargo, se hace necesario mantener la coherencia entre dichos sistemas. La necesidad de la sincronización de los datos puede ser permanente (sincronización entre sistemas operativos) o temporal (durante una migración de datos). (5)

1.6.Tecnologías actuales

1.6.1. Software Propietario

El software propietario, también conocido como software privativo o de código cerrado, es cualquier programa informático que no cumple los criterios de la Fundación para el Software Libre. El usuario final debe pagar para utilizarlo, no tiene acceso al código fuente del programa, ni el derecho de copiarlo, modificarlo o estudiarlo.

En nuestro país, los software propietarios más utilizados son los creados por Microsoft, actualmente sólo por conceptos de licencias se tienen que pagar 119.00 USD por el Sistema Operativo Windows XP Profesional, y 255.00 USD por la Suite de Microsoft Office 2007.

1.6.2. Tendencia al Software Libre

El software libre es una cuestión de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Más precisamente, significa que los usuarios de programas tienen las cuatro libertades esenciales.

1. La libertad de ejecutar el programa, para cualquier propósito (libertad 0).
2. La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
3. La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).
4. La libertad de distribuir copias de sus versiones modificadas a terceros (la 3ª libertad). Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios. El acceso al código fuente es una condición necesaria para ello.

Un programa es software libre si los usuarios tienen todas esas libertades. Entonces, debería ser libre de redistribuir copias, tanto con o sin modificaciones, ya sea gratis o cobrando una tarifa por distribución, a cualquiera en cualquier parte. El ser libre de hacer estas cosas significa, entre otras cosas, que no tiene que pedir o pagar el permiso. (6)

1.6.2.1. Licencias del Software Libre

Una licencia es la autorización formal que da el autor de algún proyecto, a un interesado que quiera ejercer cualquier tarea sobre el mismo. Desde el punto de vista del Software Libre, existen muchas variantes de licencias:

Licencia Pública General del proyecto GNU (GPL): creada por la Free Software Foundation (FSF), permite la redistribución del código binario y fuente, realizar modificaciones sin restricción alguna y está pensada para asegurar la libertad del código en todo momento, ya que un programa publicado y licenciado bajo sus condiciones nunca podrá ser hecho software propietario.

Licencia Pública General de Librerías del proyecto GNU (LGPL): permite el uso de programas libres con software propietario. El programa en sí se redistribuye como si estuviera bajo la licencia GPL, pero se permite la integración con cualquier otro software sin prácticamente ninguna limitación. Un programa con licencia LGPL puede pasarse a la licencia GPL, pero no al contrario.

CopyLeft: es un método general para hacer un programa (u otro tipo de trabajo) libre, exigiendo que todas las versiones modificadas y extendidas del mismo sean también libres. (7)

1.6.2.2. Open Source

Traducido literalmente como "Código Abierto". Es un término que empezó a utilizarse en 1998 por algunos usuarios de la comunidad del software libre, usándolo como reemplazo al ambiguo nombre original, en inglés, del software libre (free software), que no significaba exactamente lo que se pretendía (free significa a la vez "gratis" y "libre"). El significado obvio del término "código abierto" es "se puede mirar el código fuente", permite que varios programadores puedan leer, modificar y redistribuir el código fuente de un programa, por lo que ese programa evoluciona constantemente. Los usuarios lo mejoran, lo adaptan y corrigen sus errores a una velocidad impresionantemente mayor a la aplicada en el desarrollo de software convencional o cerrado, dando como resultado la producción de un mejor software. (8)

1.6.3. Sistemas Gestores de Bases de Datos

Se define una base de datos como un conjunto de información relacionada que se encuentra agrupada o estructurada en una computadora. Un Sistema Gestor de Base de Datos (SGBD) o DBMS (Database Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. Existen numerosos SGBD, algunos libres y otros comerciales entre los que se destacan por su uso y prestaciones:

✚ Sistemas comerciales.

- Microsoft Access, Microsoft SQL Server, MAGIC, WindowBase, Progress (DBMS), Oracle, DB2, Sybase.

✚ Sistemas libres.

- PostgreSQL, MySQL, SQLite.

1.6.4. SQL Server

SQLServer es un Sistema Gestor de Base de Datos (SGDB) relacional de software propietario basado en el lenguaje Transact-SQL que usa la arquitectura Cliente/Servidor.

Sus principales características son:

- Integración con internet: incluye compatibilidad integrada con XML. Admite características como English Query y el servicio Microsoft Search para incorporar consultas descriptivas y sólidas funciones de búsqueda en aplicaciones Web.
- Escalabilidad y disponibilidad: se puede utilizar en un intervalo de plataformas desde equipos portátiles que ejecutan Microsoft Windows® 98 por medio de grandes servidores con varios procesadores que ejecutan Microsoft Windows 2000, Data Center.
- Características de bases de datos corporativas: protege la integridad de los datos a la vez que minimiza la carga de trabajo que supone la administración de miles de usuarios modificando la base de datos simultáneamente.
- Facilidad de instalación, distribución y utilización: incluye un conjunto de herramientas administrativas y de desarrollo que mejora el proceso de instalación, distribución, administración y uso de SQL Server en varios sitios.
- Almacenamiento de datos: incluye herramientas para extraer y analizar datos de resumen para el procesamiento analítico en línea y herramientas para diseñar gráficamente las bases de datos y analizar los datos mediante preguntas en inglés. (9)

1.6.5. PostgreSQL

PostgreSQL es un potente Sistema Gestor de Base de Datos (SGBD) relacional de software libre orientado a objetos, liberado bajo la licencia BSD. Esta dirigido por una comunidad de desarrolladores y organizaciones comerciales llamada PostgreSQL Global Development Group (PGDG).

Sus principales ventajas son:

- Estable.

- Alto rendimiento.
- Gran flexibilidad.
- Se puede extender su funcionalidad.
- Varias interfaces de programación. ODBC, JDBC, C/C++, SQL Embebido, Tcl/Tk, Perl, Python, PHP.

1.6.6. PGAdmin III

PGAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad. (10)

1.6.7. PHP

El **PHP** (Hypertext Preprocessor), es un lenguaje "open source" interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Está diseñado originalmente para la creación de páginas web dinámicas. Algunas de sus características son:

- Es similar en sintaxis a C y Perl.
- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Es multiplataforma, funciona en todas las plataformas que soporten apache.

- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- No soporta directamente punteros, como el C, de forma que no existen los problemas de depuración provocados por estos.
- Tiene manejo de excepciones. (11)

Ver anexo 5.

1.6.8. Servidor web

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada). (12)

Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML; éstas incluyen scripts CGI, seguridad SSL y páginas activas del servidor (ASP).

1.6.9. Apache

Apache es un servidor web gratuito, potente desarrollado por la Apache Software Foundation y que nos ofrece un servicio estable y sencillo de mantener y configurar.

Sus características son:

- Multiplataforma.
- Open-source.
- Implementa el protocolo HTTP.
- Es usado para enviar páginas web estáticas y dinámicas en la World Wide Web.
- Es capaz de utilizar lenguajes como PHP, TCL, Python. (13)

1.6.10. WampServer

Es un Servidor Web gratuito. Su nombre se lo debe a las utilidades que incorpora (**W**indows **A**pache **M**ySQL **P**HP), las cuales serán de suma utilidad en la configuración y modificación del servidor. El uso de un WAMP permite servir páginas HTML a internet, además de poder gestionar datos en ellas, al mismo tiempo, un WAMP proporciona lenguajes de programación para desarrollar aplicaciones web. LAMP es el sistema análogo que corre bajo ambiente Linux. (14)

1.6.11. Navegador web

Un navegador web (del inglés, *web browser*) es un programa que permite visualizar la información que contiene una página web. Su funcionalidad básica es que permite visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet. Cualquier navegador actual permite mostrar o ejecutar gráficos, secuencias de vídeo, sonido, animaciones y programas diversos además del texto y los hipervínculos o enlaces. (15)

1.6.12. Mozilla Firefox

Mozilla Firefox es un excelente navegador completamente gratuito. El código fuente de Firefox está disponible libremente bajo la triple licencia de Mozilla como un programa libre y de código abierto. Posee barra de direcciones inteligente, Identificación instantánea del sitio web, Anti-malware, Anti-phishing, Control de contenido, Programas antivirus, Gestor de contraseñas, Actualización automática, Bloqueador de ventanas emergentes, Gestor de complementos, Más de 5000 formas de personalizarlo, Gestor de descargas, Corrector ortográfico, Restauración de sesiones, Sugerencias de búsqueda, Búsqueda en la web integrada. (16)

1.6.13. Notepad++

Notepad++ es un editor gratuito de código fuente y un sustituto de Notepad que soporta varios lenguajes. Su uso está regulado por la licencia GPL. Basado en un potente componente de edición llamado Scintilla, Notepad++ está escrito en C++ y usa directamente la API de Win32 y STL, lo que asegura una gran velocidad de ejecución y un menor tamaño del programa.

Los lenguajes de programación que admite son:

- ✚ C, C++, Java, C#, XML, HTML, PHP, CSS, Java Script, ASP, VB/VBS, SQL, Pascal, Perl, Python, Matlab. (17)

1.6.14. Proceso de Desarrollo y Gestión de Software

Para el desarrollo de la propuesta de solución se opta por el Proceso de Desarrollo y Gestión de Proyectos de Software. Este proceso es un híbrido de varias metodologías ágiles como RUP, XP y SCRUM que surge como resultado de varios años de desarrollo de los trabajadores del UCID con el objetivo de desarrollar con más calidad los proyectos de la UM según las características de los mismos. Consta de cinco fases secuenciales (Inicio, Modelación, Construcción, Explotación Experimental y Despliegue).

1.6.14.1. Lenguaje Unificado de Modelado (UML)

Es el lenguaje de modelado de sistemas de software más conocido en la actualidad. Se utiliza para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Incluye aspectos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Algunos de estos componentes son:

- Diagramas de estructura estática.
- Diagramas de comportamiento.
- Diagramas de implementación.

1.6.14.2. Lenguaje de Modelado de Procesos del Negocio (BPM)

BPM es un paradigma para la administración de procesos de negocio. Mediante BPM se persigue el modelado de las actividades de negocio para lograr una mejor administración, automatización y optimización.

Algunos ejemplos de aplicaciones de BPM o motivos de implantación son las que siguen:

- Implantación o mejora del sistema de calidad.
- Comprensión y mejora de procesos.
- Reingeniería de procesos.
- Optimización de tareas.
- Establecimiento de metodología a nivel de empresa.
- Control y seguimiento de los procesos organizativos.

1.6.14.3. Visual Paradigm

Es una herramienta UML profesional que ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Entre sus principales características están:

- Soporte de UML versión 2.1
- Interoperabilidad con modelos UML2.
- Ingeniería directa e inversa.
- Generación de modelo y código.
- Importación y exportación de ficheros XML. (18)

1.7. Análisis de las herramientas ETL

1.7.1. Talend Open Studio

Es una solución Open Source que permite la sincronización o replicación de bases de datos, migración de datos y la transformación y carga de datos complejos. Ofrece conectores técnicos y de negocios nativos para todos los entornos.

Algunos ejemplos de estos conectores son:

- Archivos: CSV, Excel, XML, LDIF.
- Bases de datos: MySQL, MS SQL Server, DB2, Oracle, Ingres, PostgreSQL, Sybase, MS Access, Informix, Firebird.

Sus principales características son:

- Permite la réplica en una sola dirección.
- Procesa los datos en el servidor origen.
- Combina vistas gráficas con técnicas.
- Se ejecuta en Windows, Unix y Linux.
- Genera componentes como Perl, Java o SQL nativo de un código para ETL.
- El código fuente Java/Eclipse está disponible para su descarga y personalización.

Desventajas:

- No existen productos complementarios de calidad de datos y gestión de metadatos.
- Necesita de Java Database Connectivity (API que permite la realización de operaciones sobre bases de datos) para acceder a las fuentes.

- Al estar financiado por una firma de capital privado, existe un mínimo riesgo de que si se deja de inyectar capital puede provocarse el parón en las mejoras del producto y la no compatibilidad con nuevas versiones de bases de datos.
- No tiene proceso automático de separación y redistribución de datos lo cual puede generar cuellos de botella. (19)

Ver anexo 6.

1.7.2. Bitool

Consiste en un software especializado en carga de datos (ETL), que permite extraer información de cualquier origen de datos (AS400, Oracle, SQL Server), transformarlo a través de lenguajes de programación sencilla (SQL, VBScript, Componentes ActiveX, Componentes Visuales) y cargarlo a cualquier motor de base de datos. (20)

Características:

- Propietario.
- Permite conectarse a cualquier motor de base de datos sin importar su ubicación (Internet, Red LAN, Red WAN) ni versión de la BD, entre los que también incluyen archivos de texto y hojas de cálculo.
- Es 100% visual. amigable e intuitivo que reduce notablemente la curva de aprendizaje.
- Emplea tecnología "Drag & Drop" para crear poderosos proyectos de carga, no necesita introducir ninguna línea de código. (21) Ver anexo 7.

1.7.3. Benetl

Es una herramienta ETL libre desarrollada en el 2007. Para utilizarla se necesita Java Runtime 1.6 (JRE), una base de datos PostgreSQL 8.4 con el apoyo de plpgsql o una base de datos MySQL 5.4 con la función OLD_PASSWORD de inicio de sesión de usuario.

Sus principales características son:

- Corre sobre Windows y Linux.
- Permite trabajar con fuentes de datos .csv, .txt y .xls.
- Propone una forma automática para recuperar los datos, utilizando las tareas programadas del sistema.
- Permite recuperar fácilmente la información de archivos planos de datos, filtrado de ellos y organización de ellos, de acuerdo a la fecha (y hora) y la entidad. (22) Ver anexo 8.

1.7.4. Scriptella

Es una herramienta ETL de código abierto (open source) desarrollada en Java.

Sus principales características son:

- Requiere JDK o JRE versión 5.0 o posterior para funcionar.
- Interoperabilidad con LDAP, JDBC, XML y otras fuentes de datos.
- Ejecución de scripts escritos en SQL, Java Script, JEXL.
- Velocidad.
- Realiza operaciones de importación y exportación de XML y texto a CSV. (23) Ver anexo 9.

1.7.5. RDBTool

Permite la incorporación de plantillas de generación de scripts, la incorporación de nuevos proveedores de esquemas a través de la extracción de esquemas de base de datos relacionales, usa el estándar OleDb para nombrar los datos, utiliza el DOM implementado en la librería RDBCore.dll que permite mapear un modelo relacional.

Ventajas sobre las demás herramientas:

- Bases de datos: SQL Server, Oracle, PostgreSQL.
- Facilidad de uso.
- Interfaz agradable para el usuario.
- Migra todas las tablas a la vez y con gran rapidez.
- Multiplataforma.
- Libre.
- Permite la réplica en ambas direcciones.
- La réplica produce sobrecarga mínima en el servidor destino.

Ver anexo 10.

1.7.5.1. Replication Schema Tool

Se encuentra integrada con la herramienta RDBTool. Permite el mapeo de las tablas de la BD origen con las tablas de la BD destino. Ver anexo 11.

1.7.5.2. DataSynchronizer

Realiza el proceso de sincronización entre dos servidores, es decir desde la BD local a la BD remota y el proceso inverso. Los tipos de gestores que utiliza se pueden configurar en el fichero DataSynchronizer.exe.config.xml. Ver anexo 12.

1.8.Conclusiones

En este capítulo se han introducido conceptos indispensables para la comprensión del proceso de migración y réplica de datos con el uso de las tecnologías actuales. Se realizó un análisis de las diferentes herramientas ETL existentes y se propuso como solución la herramienta RDBTool y sus componentes, se utilizó el Proceso de Desarrollo y Gestión de Proyectos de Software, como lenguaje de modelado UML y BPM y como herramienta para la modelación, Visual Paradigm.

Capítulo 2: Requerimientos

2.1. Introducción

En este capítulo se hace la descripción del funcionamiento de la herramienta propuesta para dar solución al problema actual. Se especifican y describen los requerimientos que debe cumplir un sistema de réplica.

2.2. Situación problemática

El Ministerio de las Fuerzas Armadas Revolucionarias y más específico el Departamento de Organización, es el encargado de llevar a cabo la gestión de plantillas por proyectos de las FAR a través de un sistema denominado Sistema Informativo para el Apoyo a la Actividad de Estructura y Composición (SIDEC) en su versión 2.0. Este sistema está desarrollado hace varios años por lo que toda la información es almacenada en bases de datos soportadas sobre SQLServer. Actualmente con el desarrollo de la informatización en todo el país y la aparición del software libre, se está desarrollando una nueva versión del sistema que cumpla las expectativas creadas por la FAR. Para continuar con la ejecución de este proyecto se hace necesario realizar una transformación de los datos hacia el nuevo gestor de bases de datos seleccionado, sin que estos se pierdan.

La gestión de toda esta información es un proceso nada sencillo que depende sobre todas las cosas de la actualidad que tenga la información que se manipule para lograr el éxito, pues se puede gestionar información no actualizada lo que provocaría que se tenga que repetir el proceso de gestión una vez actualizada la información, por no contar con mecanismos que se encarguen de esto y causaría retrasos en la toma de decisiones.

Este proceso de intercambio de información se lleva a cabo de una manera ineficiente, pues no se cuenta con mecanismos o sistemas capaces de efectuarlo con la mayor efectividad y aceptación posible.

2.3. Información que se maneja

Al MINFAR le urge la necesidad de tener el control de todas las entidades del país por lo que con este componente se realiza un flujo de información muy amplio ya que desde las estructuras organizativas de más bajo nivel se replican hacia el servidor central la información que identifica a cada entidad para de ese modo el MINFAR llevar un control estricto de cada entidad.

2.4. Funcionamiento de la herramienta RDBTool

En el capítulo 1 se concluyó que la herramienta que se va a utilizar para realizar la migración y réplica de datos es el RDBTool y sus componentes. Esta herramienta es una aplicación Desktop que permite su configuración en cualquier entorno.

Su funcionamiento básico es el siguiente:

Para soportar los procesos de replicación necesarios se implementó un mecanismo que se encarga de realizar la sincronización de los datos entre los servidores. Este mecanismo posee solo dos procesos, los cuales se describen a continuación.

a) Captura o detección de cambios.

Para lograr la sincronización entre dos sitios inicialmente es necesario detectar qué información ha cambiado desde la última vez que se sincronizó. Para esto se definió la siguiente estructura:

1- Se crea una tabla de control para cada tabla de la BD que interviene en el proceso de replicación.

En estas tablas de control se almacena información de cada tupla actualizada:

- Id de la tupla
- Fecha de la última actualización
- Código de la operación(0 para INSERT o UPDATE, 1 para DELETE)
- Sitio en que se produce la actualización.

Esta información es la mínima indispensable para identificar los cambios realizados sobre las tablas de la BD, de modo que siempre se cuenta con la última actualización. Por ejemplo, si sobre una tabla se realizan 100 actualizaciones sobre la misma tupla, a los efectos de la sincronización solo sería necesario contar con la última operación para aplicarla en el sitio destino.

2- Para cada tabla de la BD que interviene en el proceso de replicación se asocia un disparador (trigger) que se encarga de registrar en la tabla de control correspondiente la actualización realizada. La ejecución de este disparador se valida en dependencia del usuario que realiza la actualización, para de esta forma aportar un grado de seguridad al mecanismo.

3- Se crea en cada sitio una tabla para registrar información sobre la fecha de sincronización del sitio con respecto a los demás. De esta forma, entre los sitios A y B, todas las actualizaciones registradas en las tablas de control con fecha posterior a la fecha de sincronización de los dos servidores serán tenidas en cuenta, no así las que se hayan producido en una fecha anterior, debido a que estas últimas ya fueron sincronizadas en algún momento.

b) Aplicación de cambios o sincronización.

Una vez que se tienen identificados cuales son los datos que se deben sincronizar, se utiliza una aplicación externa para llevar a cabo este proceso. La aplicación enlaza temporalmente los dos sitios por medio de una conexión peer-to-peer, capturando en el sitio Fuente los datos actualizados y aplicando los cambios necesarios en el sitio Destino a través de llamadas a procedimientos almacenados. Una vez que culmina este proceso de sincronización se cierra la sesión de replicación liberando de esta forma los recursos y minimizando el impacto sobre el servidor. Debido al sentido bidireccional del proceso, la aplicación intercambia los sitios (Fuente-Destino, Destino-Fuente), posibilitando de esta forma que en una sola corrida se sincronicen los servidores en ambos sentidos.

Alcance del mecanismo

Es importante resaltar que este mecanismo de replicación es independiente del gestor sobre el que estén funcionando las BD. Debido a la simplicidad de sus procesos, para adaptarlo a algún gestor determinado solo sería necesario modificar la forma en que se crean los objetos de la BD necesarios (tablas de control, disparadores y procedimientos almacenados). De esta forma, se evidencia que está soportada la replicación entre servidores con cualquier gestor, e incluso entre servidores con gestores distintos.

Otro rasgo distintivo es la replicación en cascada; es decir, dada una cadena de servidores se posibilita la sincronización de uno a otro de forma escalonada. Por ejemplo, un sistema cuenta con la siguiente estructura: un servidor central donde se almacena toda la información del sistema para cuestiones administrativas y de toma de decisiones; un conjunto de servidores regionales que abarcan parte de la información del sistema, pero no cuentan con aquella que va dirigida a los ejecutivos ni con la información de otra región; y un conjunto de servidores locales destinados a usuarios específicos. Los servidores regionales se sincronizan con las actualizaciones realizadas en el servidor central, y a su vez los servidores locales se actualizan a partir de las actualizaciones de dichos servidores regionales. Todo este proceso se hace de forma automática sin necesidad de realizar modificaciones en las aplicaciones regionales ni locales.

Si al caso anterior se une el hecho de que está soportada la replicación bidireccional, se podrían definir ambientes de replicación con cualquier estructura, incluyendo esquemas circulares. Ver anexo 13.

A modo resumen, estas son las principales características del mecanismo:

- Replicación bidireccional basada en sesión (tiempo de conexión).
- Basada en registro de cambio, no basada en colas.
Solo la última actualización es enviada, evitando transmisiones redundantes.
- Peer-to-peer.
Modelo "actualizar en cualquier sitio". No existe la noción "Master/Slave".

- Topología flexible y escalable.
- Soporte para ambientes heterogéneos.
Sincronización entre gestores de cualquier tipo.
- Soporte de particionamiento horizontal de los datos.
- Garantiza integridad de los datos.
Proceso transaccional. Identificación y registro de conflictos.
- Implementado sobre plataforma .NET.

Algoritmo

A continuación se describe el algoritmo implementado para capturar los cambios y realizar la sincronización.

Capturar cambios en sitio Fuente:

1. Se actualiza una tupla por medio de la aplicación en una tabla de la BD.
2. El usuario de la aplicación tiene permiso para replicar, por lo que se dispara el trigger de control en la tabla.
3. Por medio del trigger se registra el cambio en la tabla de control correspondiente:
 - Si la operación de actualización es INSERT o UPDATE se inserta una tupla correspondiente en la tabla de control, o se actualiza si ya está, estableciendo el código de la operación en 0 y la fecha de actualización en null.
 - Si la operación de actualización es DELETE se actualiza la tupla correspondiente en la tabla de control, estableciendo el código de la operación en 1 y la fecha de actualización en null.
 - Se establece el campo SOURCE de la tabla de control con el código del servidor donde se produjo la actualización. Ver anexo 14.

Sincronizar datos entre sitios Fuente y Destino:

1. El servicio de replicación carga desde un fichero XML toda la configuración necesaria para llevar a cabo la replicación. Como parte de esta configuración se encuentran las tablas que intervienen en el proceso, ordenadas de acuerdo a sus restricciones de integridad (por orden de inserción). Así como el conjunto de reglas (filtros) necesarias para realizar el particionamiento a las tablas que lo requieran.
2. El servicio de replicación establece una conexión peer-to-peer entre los sitios Fuente y Destino a partir de un fichero de configuración XML que contiene los datos y las cadenas de conexión de estos sitios.
3. Para identificar los cambios a replicar en el sitio Fuente de las tablas del sistema, se identifican de cada tabla de control correspondiente las tuplas que contengan actualizaciones

cuya fecha sea superior a la fecha de sincronización del sitio Fuente con respecto del sitio Destino (todas las actualizaciones que se realizaron después de la última sincronización). Para el caso de una tabla que replica en ambos sentidos, solo se toman las actualizaciones cuyo origen (campo SOURCE) no sea el mismo que el sitio Destino, pues se supone que este cambio fue producto de una sincronización previa.

4. Una vez identificadas las actualizaciones a replicar en las tablas de control, se obtienen las tuplas reales de las tablas del sistema por medio de los campos ID. Estos datos se le pasan a los procedimientos almacenados que se encuentran en el sitio Destino, los cuales se encargan de aplicar los cambios en dependencia del código de la última operación (para 0: se trata de insertar la tupla, si ya está se actualiza; para 1: se borra la tupla).
5. El proceso del paso 4 se realiza para cada una de las tablas del sistema que replican, garantizando la transaccionabilidad, de modo que en caso de fallo en la conexión o conflicto de actualización, no se produzcan inconsistencia en los datos.
6. Al culminar se actualiza en ambos sitios la tabla de control del servidor, con la fecha actual relativa a cada sitio.
7. El servicio de replicación en una misma corrida separa en dos hilos de ejecución las sincronizaciones Local-Remoto Remoto-Local, de esta forma, garantizando el sentido bidireccional. El mismo se configura para que se ejecute cada cierto tiempo de forma automática.
8. En caso de producirse un conflicto de actualización (ej. violación de llave foránea) se para el proceso y se registra en un fichero LOG el conflicto en sí, el cual debe ser resuelto por el administrador de la BD antes de la próxima corrida. (24)

2.5. Requerimientos que debe cumplir un sistema de réplica

- Debe permitir que la réplica se realice entre dos o más base de datos.
- Debe mantener la coherencia transaccional.
- Debe tener una latencia baja. Las actualizaciones en el servidor origen deben llegar al servidor destino con rapidez.
- El proceso de réplica debe producir una sobrecarga mínima en el servidor de origen.
- El sistema debería tener un alto rendimiento si la replicación de cambios incrementales es necesaria: debería procesar la replicación de un gran número de transacciones.

- El sistema debe permitir la réplica de los datos en ambas direcciones.
- Los datos requeridos en los servidores de destino pueden ser un subconjunto de los datos disponibles en el servidor de origen.
- Debe permitir que se realicen filtros.
- Debe existir atomicidad: Todas las operaciones de la transacción son ejecutadas por completo, o no se ejecuta ninguna de ellas (si se ejecuta la transacción, se hace hasta el final).
- El sistema debe permitir la sincronización entre las tablas.

2.6. Captura de requisitos

Durante la captura de requisitos desarrollada por los especialistas del MINFAR se obtuvo como resultado lo siguiente:

1. Seleccionar una herramienta.
 - La herramienta seleccionada debe funcionar sobre el Sistema Operativo Linux.
 - Debe permitir la réplica desde y hacia más de dos servidores.
 - Debe permitir la réplica en ambas direcciones.
 - Debe permitir que se realicen filtros a los datos.
 - Debe permitir procesar la replicación de un gran número de transacciones.
2. Entregar la documentación con el funcionamiento de la herramienta.

2.7. Conclusiones

En este capítulo se definieron los requerimientos generales que debe cumplir un sistema de réplica, se describieron los requisitos para realizar la réplica de los datos. Partiendo de todas las características que se han expuesto, se está en condiciones de realizar la migración y réplica de los datos siguiendo sus especificidades.

Capítulo 3: Análisis de la solución y prueba

3.1. Introducción

En este capítulo se describe el procedimiento para realizar la migración y réplica de la base de datos haciendo uso de la herramienta RDBTool. Se analizan todas las dependencias que existen entre las tablas de la base de datos y se describe brevemente, las pruebas realizadas.

3.2. Procedimiento para migrar y replicar la base de datos

Para realizar la migración y réplica de la información almacenada en la base de datos se realizan los siguientes pasos:

1. Ejecutar la aplicación RDBTool.exe.
2. Realizar la conexión a la base de datos. (Ver anexo 15)
3. Adicionar las tablas a migrar. (Ver anexo 16)
4. Generar scripts y salvar proyecto. (Ver anexo 17)
5. Ejecutar la aplicación ReplicationSchemaTool.exe. (Ver anexo 18)
6. Adicionar las tablas origen. (Ver anexo 19)
7. Adicionar las tablas destino. (Ver anexo 20)
8. Realizar el mapeo de las tablas. (Ver anexo 21 y 22)
9. Salvar proyecto.
10. Configurar el archivo xml DataSynchronizer.exe.config. (Ver anexo 23)
11. Configurar el archivo xml DB_Schema_A y DB_Schema_B. (Ver anexo 24)
12. Ejecutar los scripts generados anteriormente, en los servidores.
13. Configurar tablas control_synch de los servidores. (Ver anexo 25)
14. Ejecutar DataSynchronizer.exe.

3.3. Dependencia entre las tablas

Para realizar la réplica correcta de la base de datos primero se tiene en cuenta las dependencias que existen entre las tablas, esto evita que se produzcan una serie de errores.

3.3.1. Dependencias de la tabla nom_aristaeav

En la tabla nom_aristaeav se guarda como llave foránea, el identificador (idnomeav) de la tabla nom_nomencladoreavestruc.

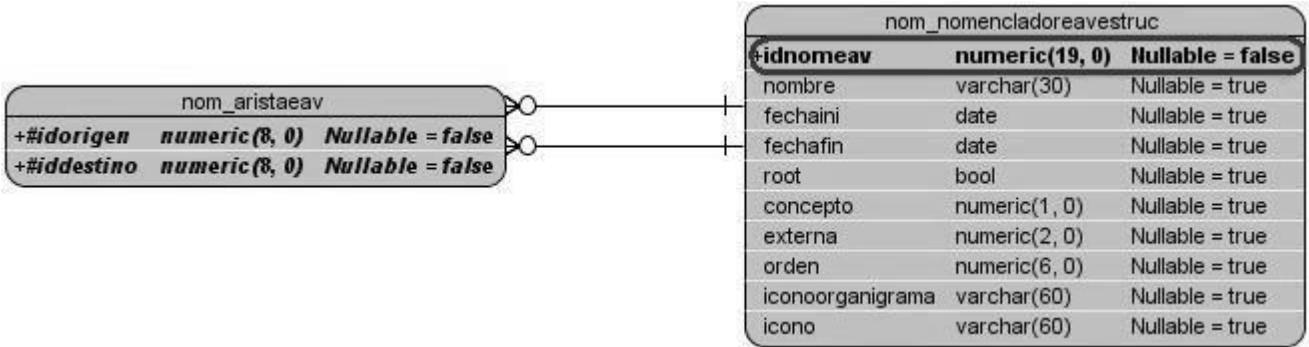


Figura 1: Dependencia de la tabla nom_aristaeav.

3.3.2. Dependencias de la tabla nom_calificador_cargo

En la tabla nom_calificador_cargo se guardan como llaves foráneas, el identificador (idtipocalificador) de la tabla nom_tipo_calificador y el identificador (idcategocup) de la tabla nom_categocup.

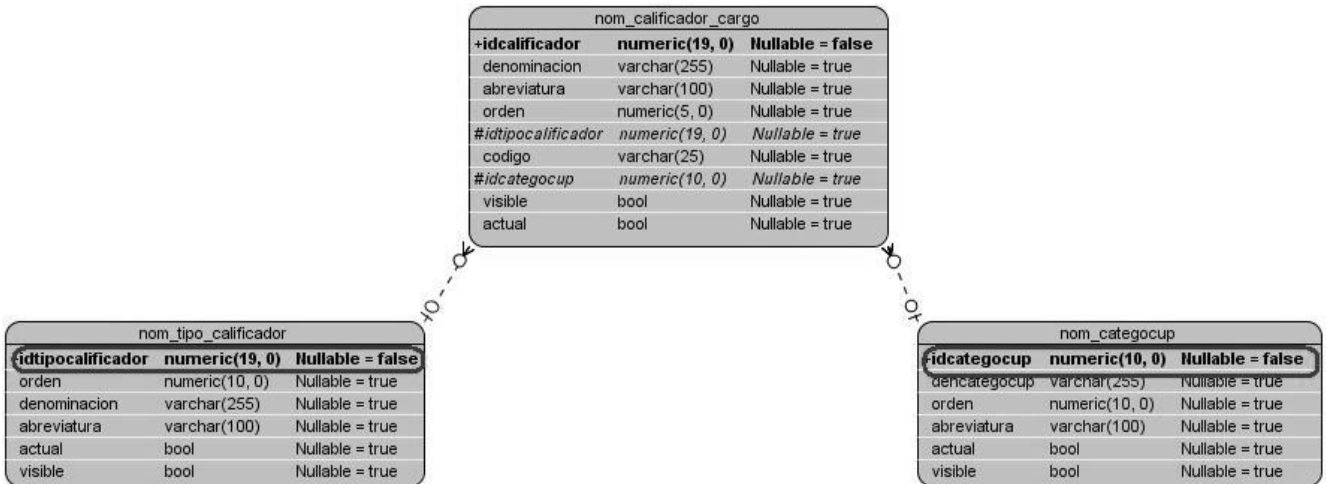


Figura 2: Dependencia de la tabla nom_calificador_cargo.

3.3.3. Dependencias de la tabla nom_campoestruc

En la tabla nom_campoestruc se guarda como llave foránea, el identificador (idnomeav) de la tabla nom_nomencladoreavestruc.

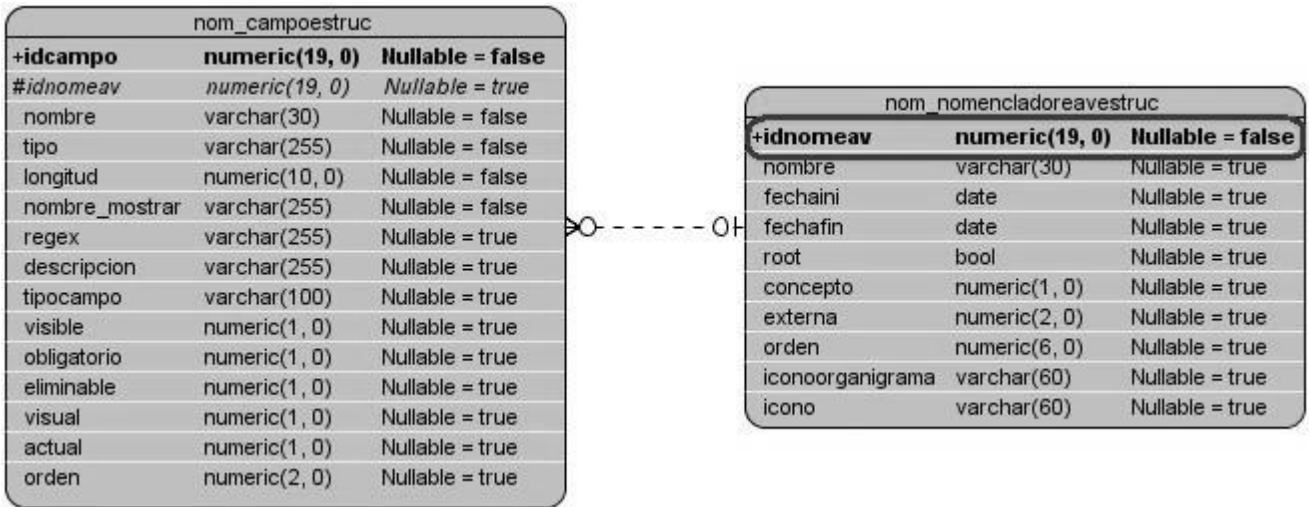


Figura 3: Dependencia de la tabla nom_campoestruc.

3.3.4. Dependencias de la tabla nom_grupoComple

En la tabla nom_grupoComple se guarda como llave foránea, el identificador (idescalasalarial) de la tabla nom_escalasalarial.

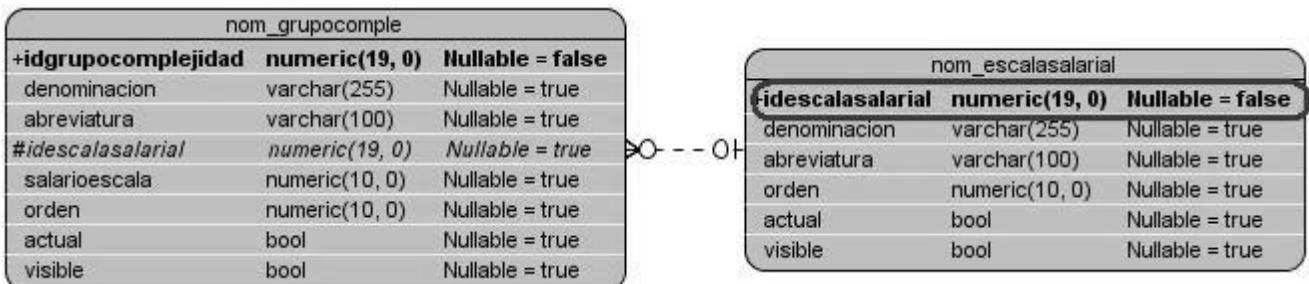


Figura 4: Dependencia de la tabla nom_grupoComple.

3.3.5. Dependencias de la tabla nom_organO

En la tabla nom_organO se guardan como llaves foráneas, el identificador (idnomeav) de la tabla nom_nomencladoreavestruc, el identificador (idnivelestr) de la tabla nom_nivelestr y el identificador (idespecialidad) de la tabla nom_especialidad.

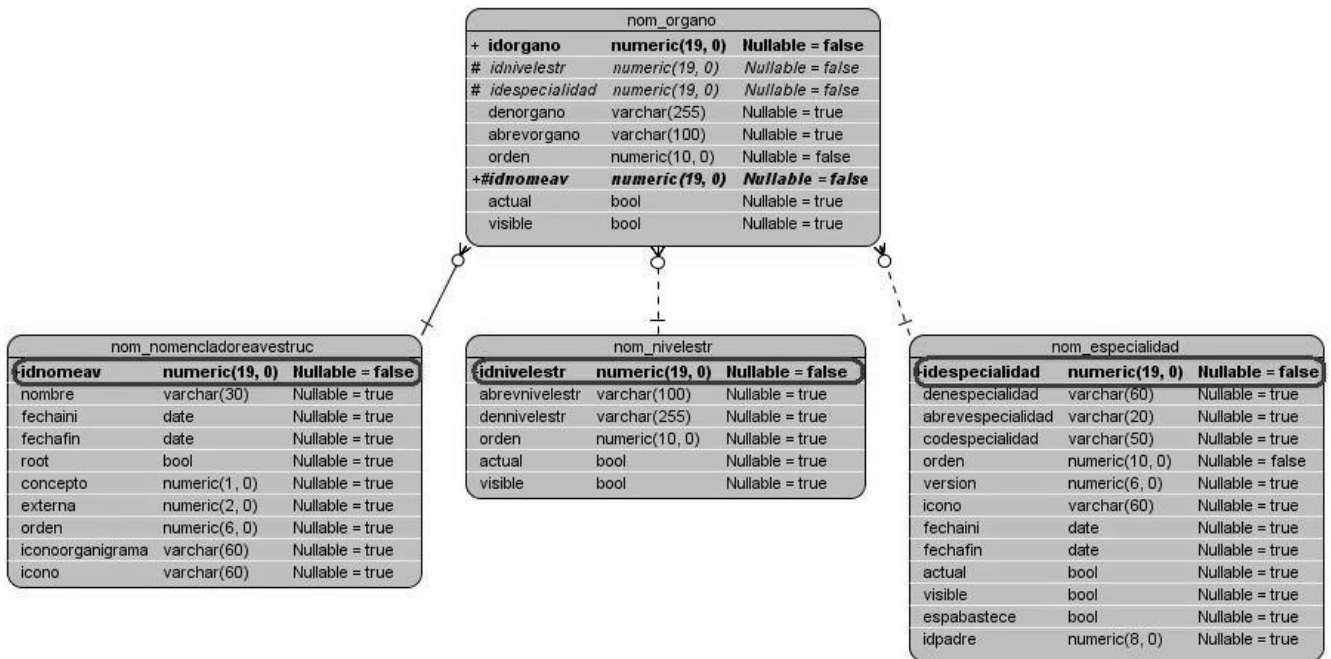


Figura 5: Dependencia de la tabla nom_organo.

3.3.6. Dependencias de la tabla nom_salario

En la tabla nom_salario se guardan como llaves foráneas, el identificador (idgrupocomplejidad) de la tabla nom_grupoComple y el identificador (idescalasalariar) de la tabla nom_escalasalariar.

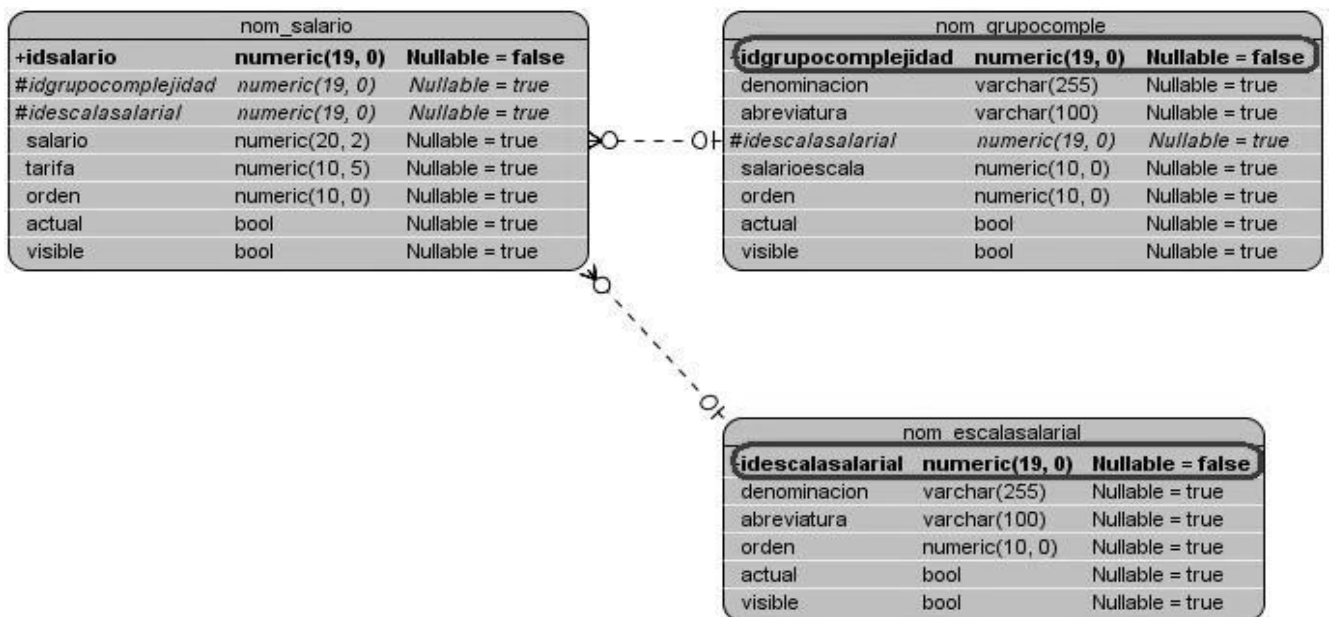


Figura 6: Dependencia de la tabla nom_salario.

3.3.7. Dependencias de la tabla nom_valor_defecto

En la tabla nom_valor_defecto se guarda como llave foránea, el identificador (idcampo) de la tabla nom_campoestruc.

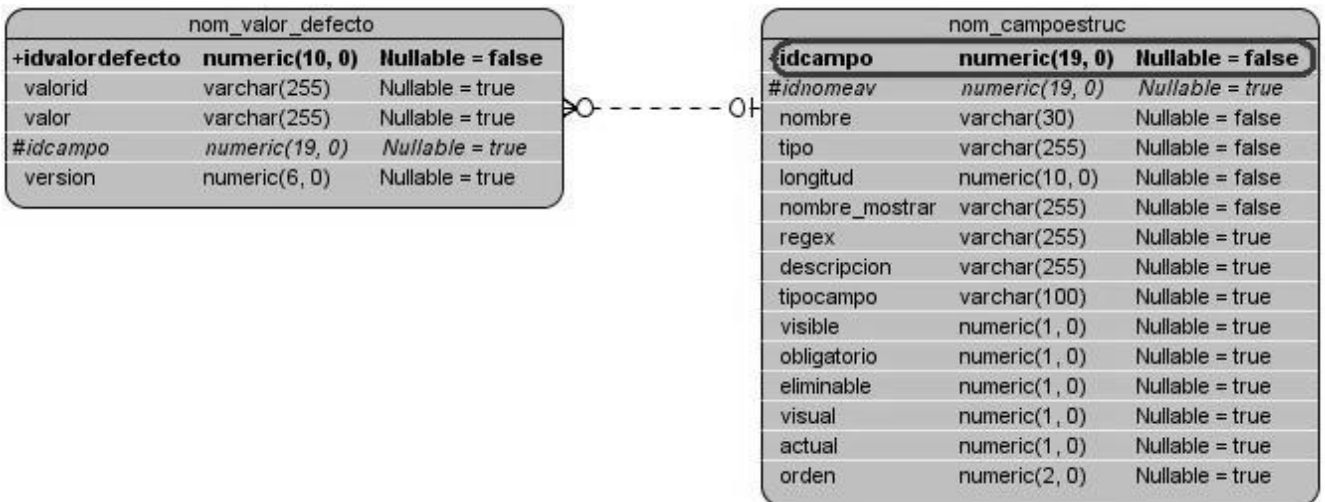


Figura 7: Dependencia de la tabla nom_valor_defecto.

3.3.8. Dependencias de la tabla nom_categcivil

En la tabla nom_categcivil se guarda como llave foránea, el identificador (idcategmilitar) de la tabla nom_categmilitar.

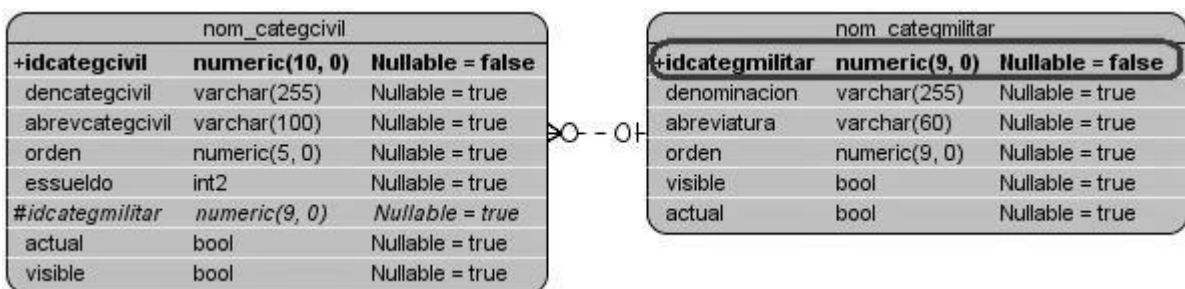


Figura 8: Dependencia de la tabla nom_categcivil.

3.3.9. Dependencias de la tabla nom_cargomilitar

En la tabla nom_cargomilitar se guardan como llaves foráneas, el identificador (idcalificador) de la tabla nom_calificador_cargo, el identificador (idcategmilitar) de la tabla nom_categmilitar, el

identificador (idespecialidad) de la tabla nom_especialidad y el identificador (idfamiliacargo) de la tabla nom_familiacargo.

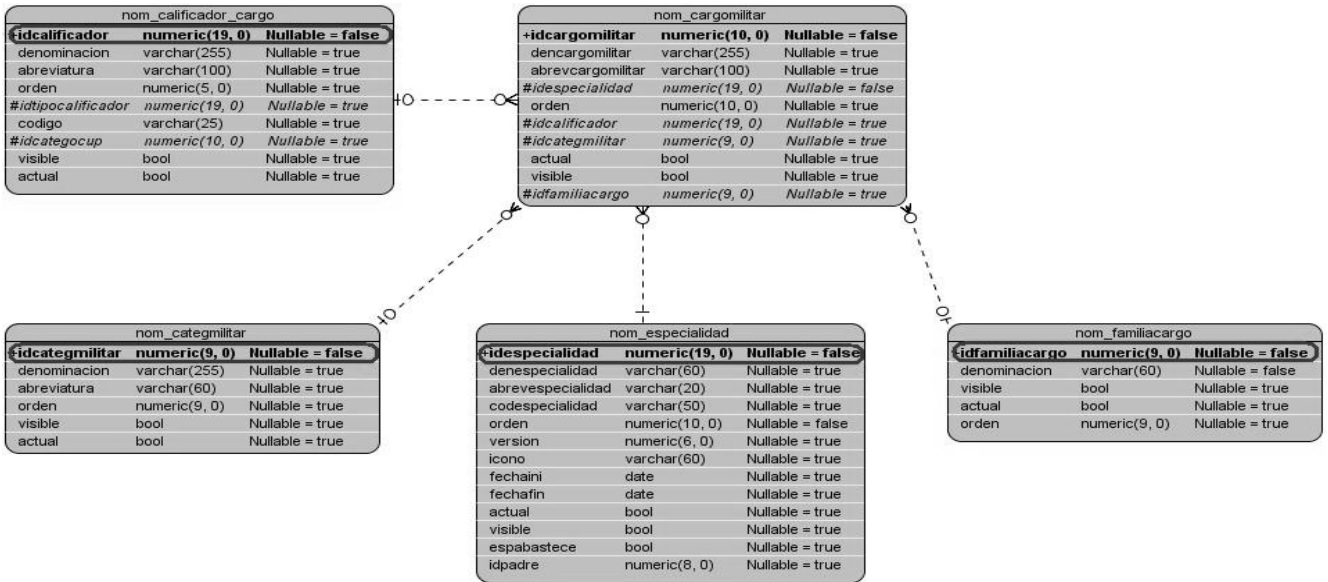


Figura 9: Dependencia de la tabla nom_cargomilitar.

3.3.10. Dependencias de la tabla nom_cargocivil

En la tabla `nom_cargocivil` se guardan como llaves foráneas, el identificador (`idcalificador`) de la tabla `nom_calificador_cargo`, el identificador (`idcategcivil`) de la tabla `nom_categcivil`, el identificador (`idespecialidad`) de la tabla `nom_especialidad`, el identificador (`idgrupocomplejidad`) de la tabla `nom_grupoComple`, el identificador (`idnivelutilizacion`) de la tabla `nom_nivel_utilizacion` y el identificador (`idcategocup`) de la tabla `nom_categocup`.

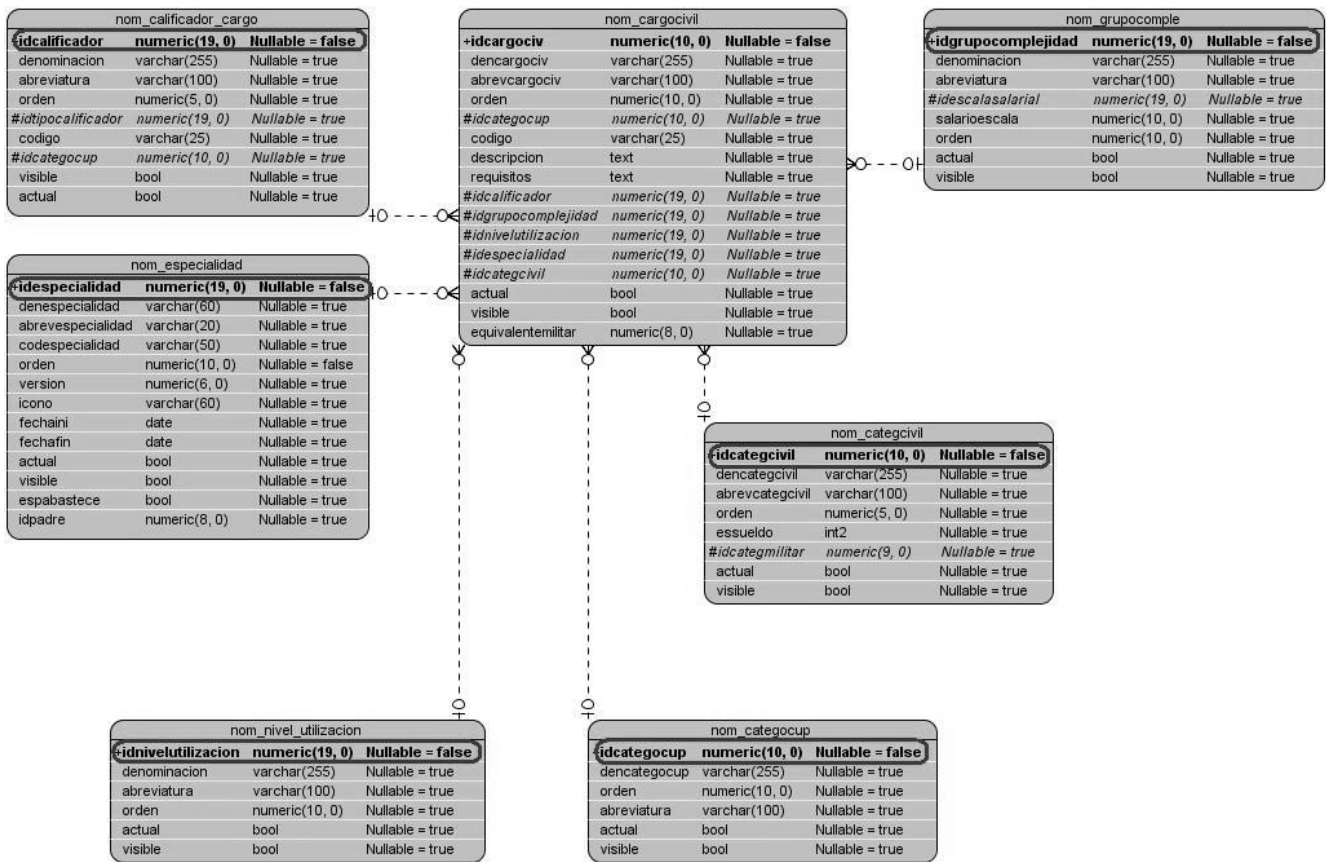


Figura 10: Dependencia de la tabla `nom_cargocivil`.

3.3.11. Dependencias de la tabla `nom_gradomilit`

En la tabla `nom_gradomilit` se guardan como llaves foráneas, el identificador (`idgsubcateg`) de la tabla `nom_gsubcateg`, el campo (`anterior`) de la tabla `nom_gradomilit` y el campo (`sucesor`) de la tabla `nom_gradomilit`.

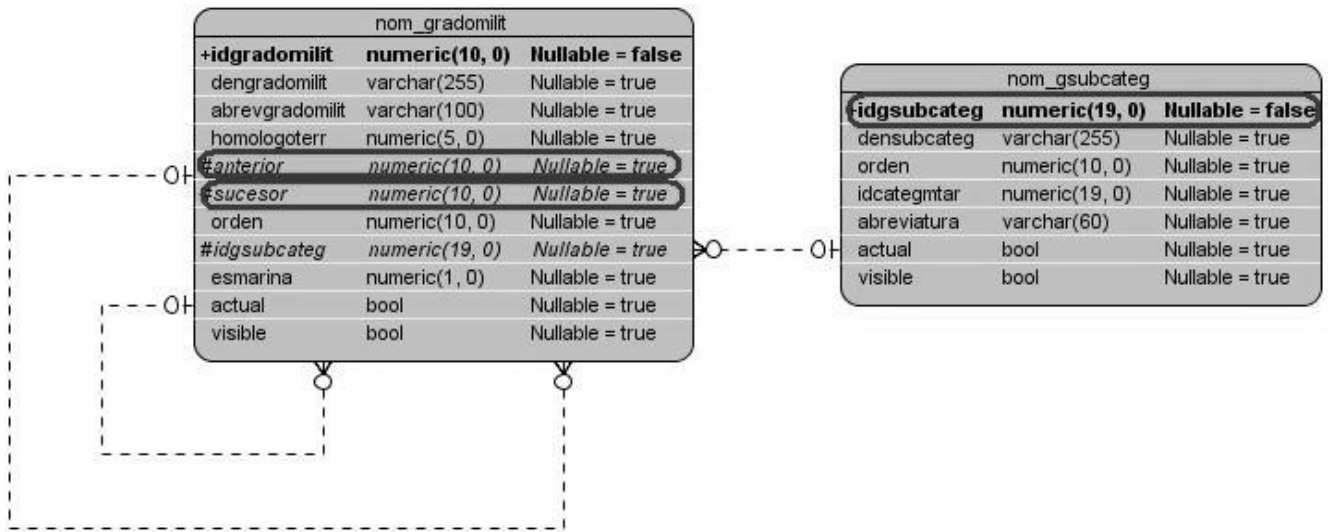


Figura 11: Dependencia de la tabla nom_gradomilit.

3.3.12. Dependencias de la tabla nom_gradomarina

En la tabla nom_gradomarina se guarda como llave foránea, el identificador (idgradomilit) de la tabla nom_gradomilit.

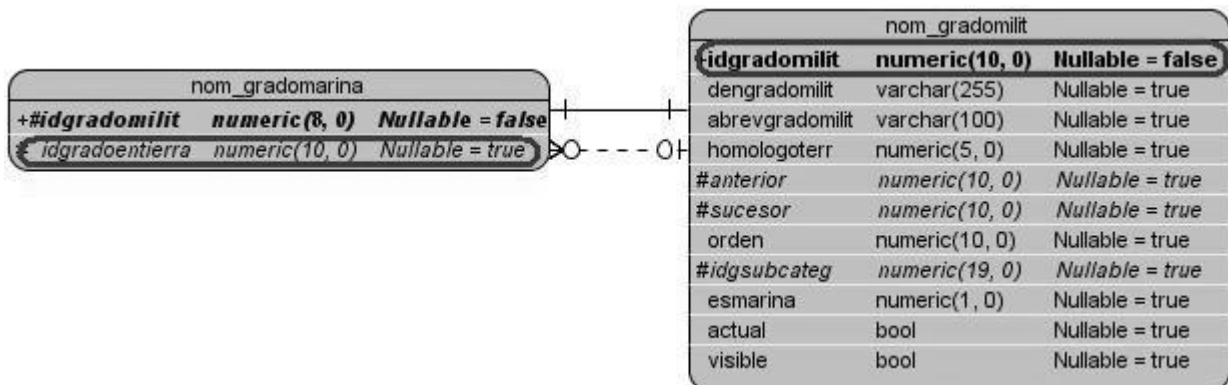


Figura 12: Dependencia de la tabla nom_gradomarina.

3.3.13. Dependencias de la tabla nom_tecnica

En la tabla nom_tecnica se guardan como llaves foráneas, el identificador (idplanifica) de la tabla nom_planificatecnica, el identificador (idsituaciontecnica) de la tabla nom_situaciontecnica, el identificador (idmaqtecnica) de la tabla nom_maqtecnica, el identificador (idnomordtecnica) de la tabla nom_ordtecnica y el identificador (idtipotecnica) de la tabla nom_tipotecnica.

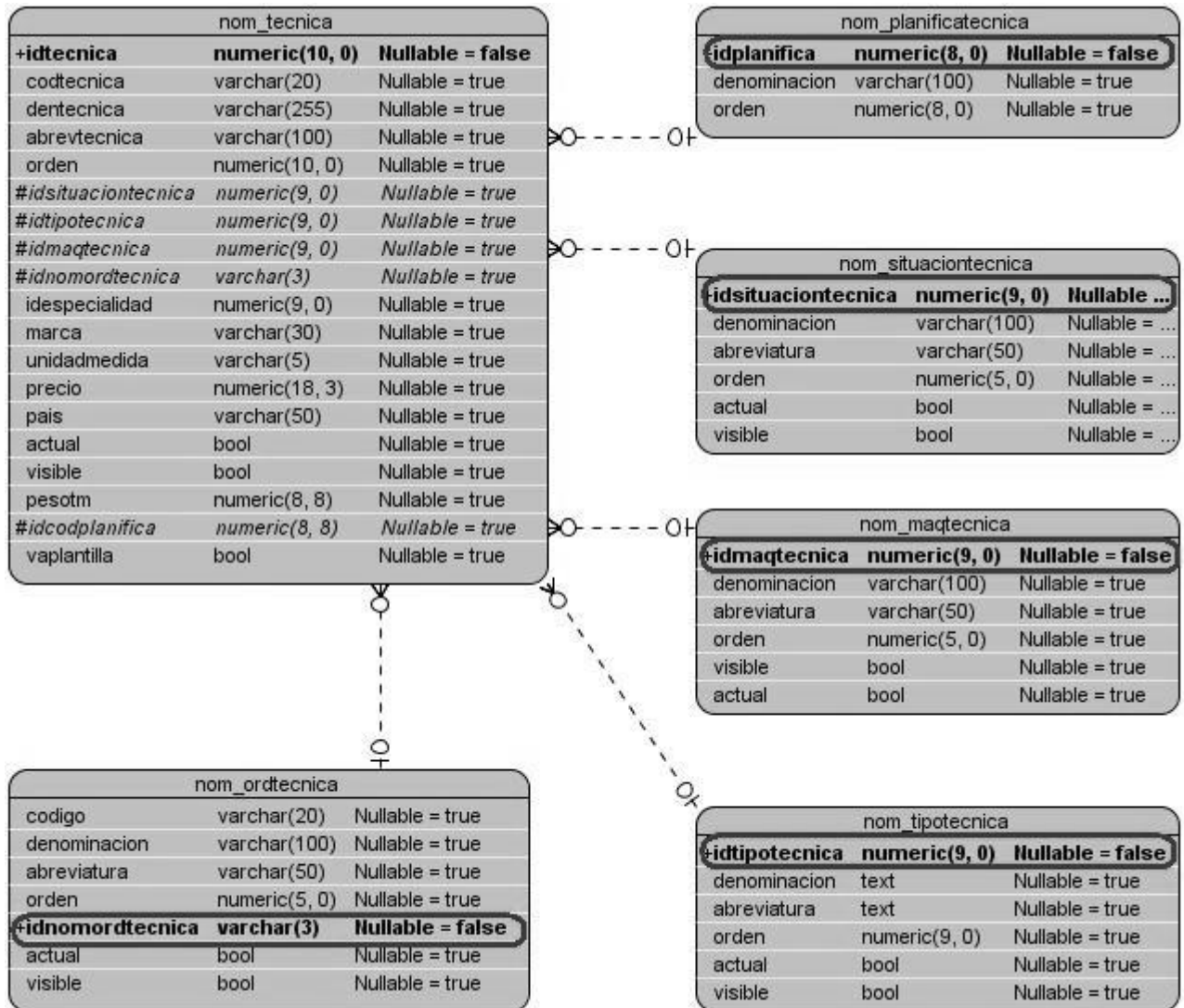


Figura 13: Dependencia de la tabla nom_tecnica.

3.3.14. Dependencias de la tabla nom_modulotecnica

En la tabla nom_modulotecnica se guardan como llaves foráneas, el identificador (idtecnica) de la tabla nom_tecnica, y el identificador (idmodulo) de la tabla nom_modulo.

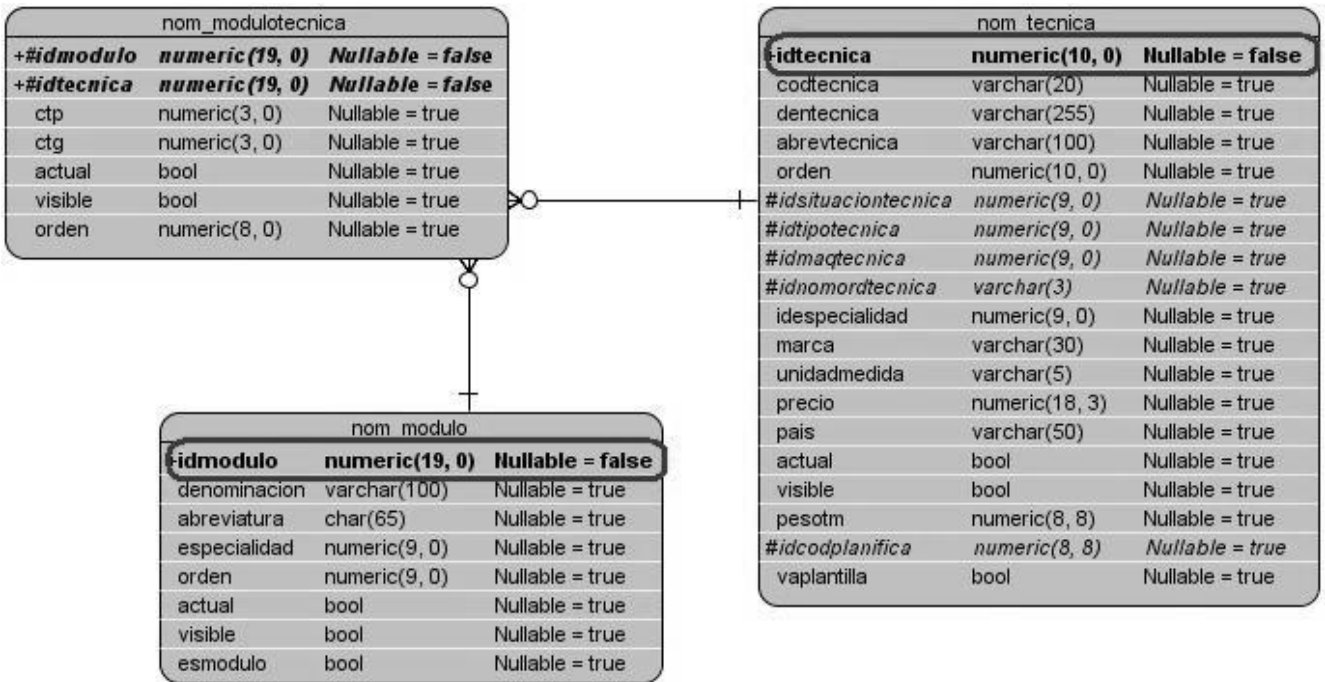


Figura 14: Dependencia de la tabla nom_modulotecnica.

3.3.15. Dependencias de la tabla nom_filaestruc

En la tabla nom_filaestruc se guardan como llaves foráneas, el identificador (idnomeav) de la tabla nom_nomencladoreaveestruc, y el campo (idpadre) de la tabla nom_filaestruc.

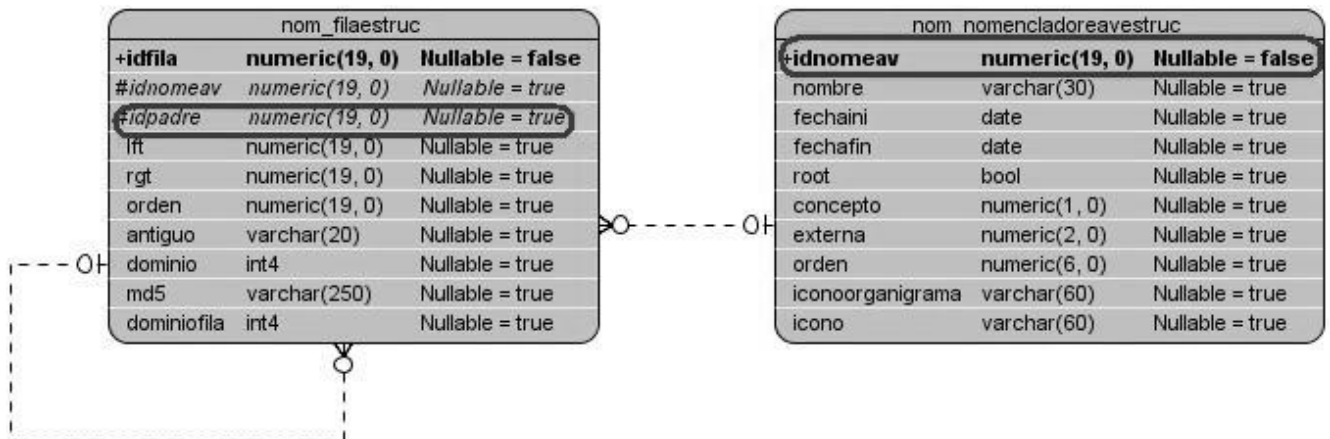


Figura 15: Dependencia de la tabla nom_filaestruc.

3.3.16. Dependencias de la tabla nom_valoreestruc

En la tabla nom_valoreestruc se guardan como llaves foráneas, el identificador (idcampo) de la tabla nom_campoestruc, y el identificador (idfila) de la tabla nom_filaestruc.

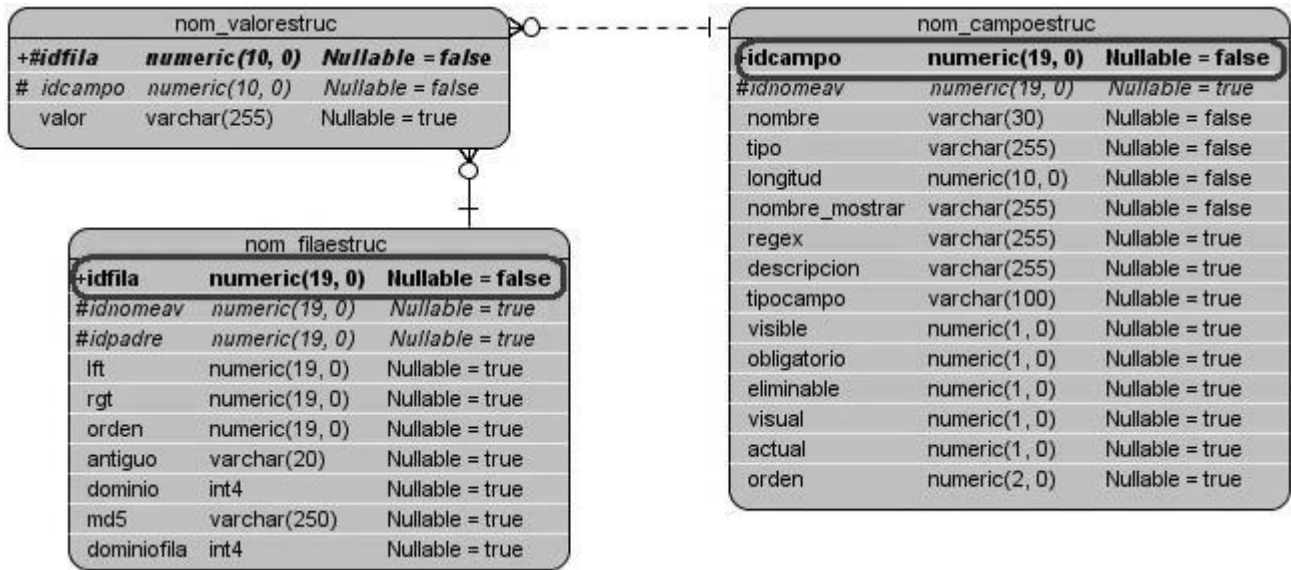


Figura 16: Dependencia de la tabla nom_valorestruc.

3.4. Deferred

Deferred es una propiedad o característica que poseen las llaves foráneas que hacen referencia hacia la misma tabla. Cuando se realiza una réplica de datos es necesario activar esta propiedad a todas las tablas que lo necesiten, esto evita que ocurran errores y se interrumpa la réplica.

3.4.1. Tablas con llaves foráneas referenciadas hacia ellas mismas.

Tabla 3.1. Llaves foráneas en mod_nomencladores

<i>mod_nomencladores</i>	
Tablas	Llave Foránea
nom_centrocostopatrimonial	nom_centrocostopatrimonial_idcentrocostopatrimonial
nom_centrocostopresupuestario	nom_centrocostopresupuestario_idcentrocostopresupuestario
nom_dpa	nom_dpa_iddpa
nom_elementogastopatrimonial	nom_elementogastopatrimonial_idelementogastopatrimonial
nom_especialidad	nom_especialidad_idespecialidad
nom_parrafofiscal	nom_parrafofiscal_idparrafofiscal
nom_ufc	nom_ufc_idufc

nom_elementogastopresupuestario	nom_elementogastopresupuestario_idelementogastopresupuestario
---------------------------------	---------------------------------------------------------------

Tabla 3.2. Llaves foráneas en mod_estructuracomp

<i>mod_estructuracomp</i>	
Tablas	Llave Foránea
dat_subordinacion	fk_padre_fila
nom_gradomilit	fk_anterior
nom_gradomilit	fk_sucesor
nom_filaestruc	fk_padre

Para activar esta condición se ejecutan los scripts correspondientes.

Ejemplo:

```
ALTER TABLE "mod_estructuracomp"."nom_filaestruc"
  DROP CONSTRAINT "fk_padre" RESTRICT;
ALTER TABLE "mod_estructuracomp"."nom_filaestruc"
  ADD CONSTRAINT "fk_padre" FOREIGN KEY ("idpadre")
  REFERENCES "mod_estructuracomp"."nom_filaestruc"("idfila")
  ON DELETE CASCADE
  ON UPDATE CASCADE
  DEFERRABLE
  INITIALLY DEFERRED;
```

Figura 17: Script para activar la propiedad Deferred

3.5. Orden de los esquemas

Luego de haber seleccionado correctamente las tablas con dependencias funcionales, se establece el orden en que se van a replicar los esquemas de la base de datos, quedando de la siguiente forma:

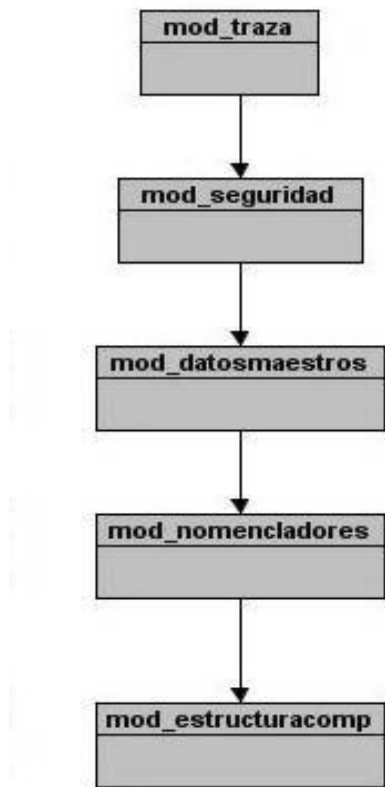


Figura 18: Orden de los esquemas.

3.6. Orden de las tablas

Entre las tablas de cada uno de los esquemas también se establece un orden, evitando que ocurra algún error de integridad referencial. Primero se replican aquellas tablas que no dependen de ninguna otra y luego las demás según la cantidad de dependencias que tenga.

3.6.1. Ejemplo de tablas sin dependencia funcional

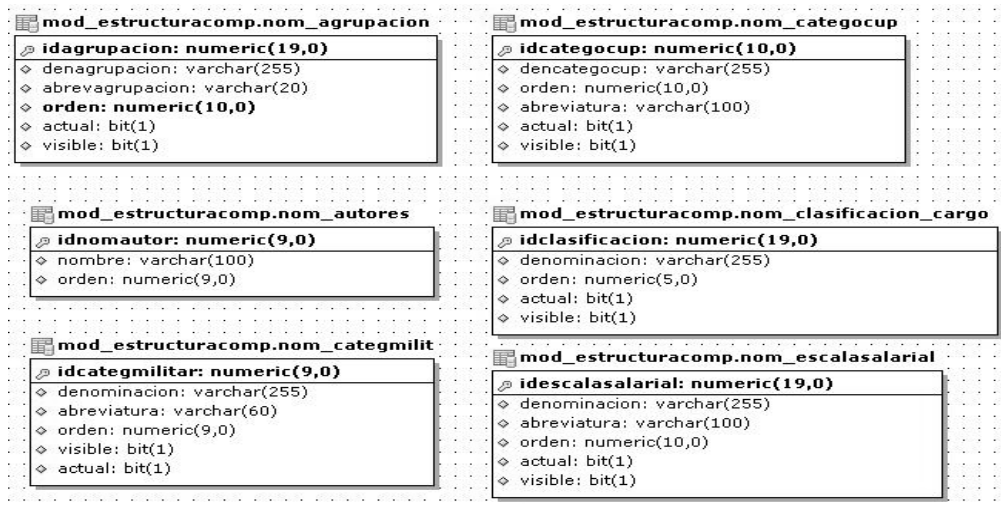


Figura 19: Tablas sin dependencia funcional del esquema `mod_estructuracomp`.

3.6.2. Ejemplo de tablas con dependencia funcional

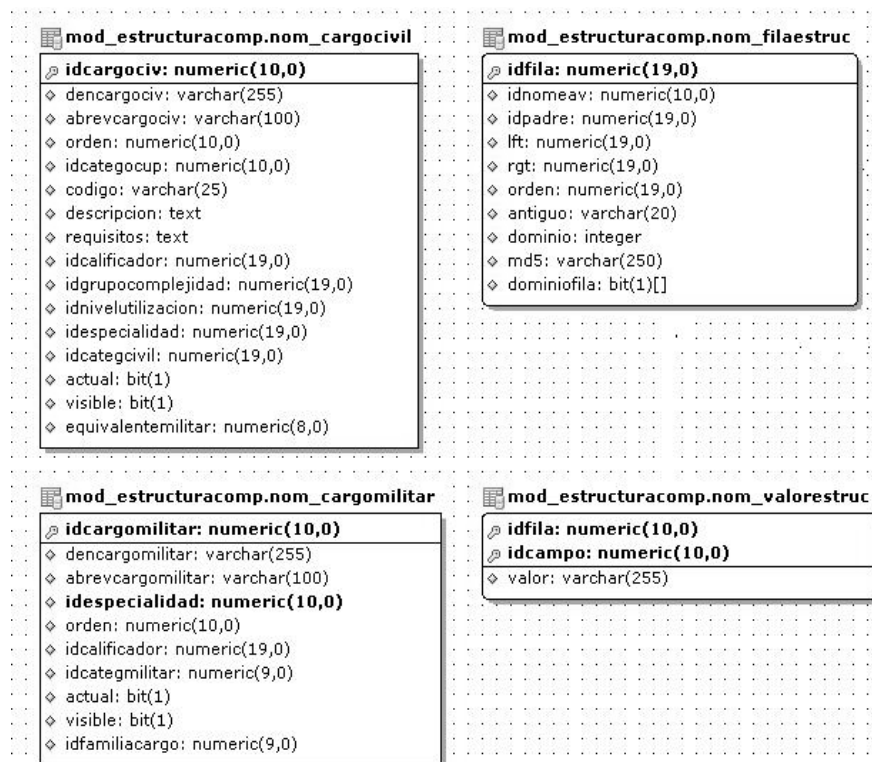


Figura 20: Tablas con dependencia funcional del esquema `mod_estructuracomp`.

3.7. Filtros

Existen una serie de reglas o filtros que pueden ser sobre tipos de operaciones y de reglas SQL que tienen que cumplir los datos para ser replicados. Los tipos de operaciones que se pueden filtrar son las eliminaciones de tuplas y las modificaciones y/o inserciones. El filtro según una regla SQL simplemente replica las tuplas que cumplan con la regla creada. En este caso estos filtros están definidos por los dominios asignados a cada BD.

Tabla 3.3. Dominios asignados a cada BD.

Denominación	Dominio
E.OCC	1
E.CTRAL	2
E.OTAL	3
DIV.TANQUE	4

Algunos filtros creados:

- ✚ dominiofila[1] = 1:: bit
- ✚ {idestructura} in (select idfila from mod_estructuracomp.nom_filaestruc where dominiofila[1] = 1 :: bit (1))
- ✚ {idcargo} in (select idfila from mod_estructuracomp.nom_filaestruc where dominiofila[1] = 1 :: bit (1))
- ✚ {idfila} in (select idfila from mod_estructuracomp.nom_filaestruc where dominiofila[1] = 1 :: bit (1)).

3.8. Migrar campo dominiofila

El campo dominiofila es de tipo bit(1)[] y se encuentra en la tabla nom_filaestruc del esquema mod_estructuracomp. La herramienta seleccionada para dar solución a la problemática existente no es capaz de replicar este tipo de dato por lo que es necesario realizar la réplica implementando una función PHP, la cual se ejecuta en un servidor web a través del navegador Mozilla Firefox.

```

<?php
    $IDDOMINIO = 2;
    define('PG_esquema', 'mod_estructuracomp');
    define('PG_host', 'localhost');
    define('PG_bd', 'SIDEC');
    define('PG_usuario', 'estructura' );
    define('PG_passwd', 'estructuraerp2009');

    define('PG_esquemaD', 'mod_estructuracomp');
    define('PG_hostD', 'localhost');
    define('PG_bdD', 'destino');
    define('PG_usuarioD', 'estructura' );
    define('PG_passwdD', 'estructuraerp2009');

    $ConFuente= pg_connect('host='.PG_host.' port=5432 dbname='.PG_bd.' user='.PG_usuario.' password='.PG_passwd)
    or die('Error conectando a postgres '.pg_errormessage());
    $sql = "Select idfila,dominiofila from mod_estructuracomp.nom_filaestruc where dominiofila[$IDDOMINIO] = 1::bit";
    $resultBit = pg_query($ConFuente,$sql);
    $bit = array();

    $ConDestino = pg_connect('host='.PG_hostD.' port=5432 dbname='.PG_bdD.' user='.PG_usuarioD.' password='.PG_passwdD)
    or die('Error conectando a postgres '.pg_errormessage());
    while ($filaEntidad = pg_fetch_array($resultBit)) {
        $sqlActualizar = "UPDATE mod_estructuracomp.nom_filaestruc
            set
                dominiofila = '{ $filaEntidad['dominiofila']}'
                where idfila = { $filaEntidad['idfila']}";
        pg_query($ConDestino,$sqlActualizar) or die(pg_errormessage());
    }
    echo 'Completado';
?>

```

Figura 21: Función migrar dominiofila.

3.9. Modelo de datos

Un modelo de datos es un conjunto de conceptos, reglas y convenciones que nos permiten describir la estructura de una base de datos.

Por lo general, un modelo de datos presenta dos sublenguajes: un **Lenguaje de Definición de Datos o DDL (Data Definition Language)**, cuya función es describir, de una forma abstracta, las estructuras de datos y las restricciones de integridad; y un **Lenguaje de Manipulación de Datos o DML (Data Manipulation Language)**, que se orienta a describir las operaciones de manipulación de los datos.

En el modelo de datos se definen:

- Objetos (entidades, relaciones,...)
- Asociaciones entre objetos (interrelaciones,...)
- Propiedades de los objetos (atributos, campos,...)
- Dominios (conjunto de valores del que se toman los valores para las propiedades, por ejemplo el dominio de los días de la semana para una propiedad día de nacimiento.)
- Restricciones (Limitaciones impuestas por el modelo de datos que estamos utilizando (inherentes) o definidas por el diseñador para representar fielmente el mundo real (de integridad o semánticas).)

Ver anexo 26.

3.10. Descripción textual de las tablas

Tabla 3.2. Descripción de la tabla nom_agrupacion

Nombre: nom_agrupacion		
Descripción: agrupaciones		
Atributo:	Tipo:	Descripción:
idagrupacion	numérico	Identificador de la tabla
denagrupacion	texto	Denominación
abrevagrupacion	texto	Abreviatura
orden	numérico	Orden de la agrupación
actual	bit	
visible	bit	

Tabla 3.3. Descripción de la tabla nom_aristaeav

Nombre: nom_aristaeav		
Descripción: aristaeav		
Atributo:	Tipo:	Descripción:
idorigen	numérico	Identificador del origen de la arista que une dos niveles de estructuras
iddestino	numérico	Identificador del destino de la arista que une dos niveles de estructuras

Tabla 3.4. Descripción de la tabla nom_calificador_cargo

Nombre: nom_calificador_cargo		
Descripción: calificadores		
Atributo:	Tipo:	Descripción:
idcalificador	numérico	Identificador de la tabla
denominacion	texto	Denominación
abreviatura	texto	Abreviatura
orden	numérico	Orden que ocupa en la tabla
idtipocalificador	numérico	Identificador del calificador

codigo	texto	Código del calificador
idcategocup	numérico	Identificador de cateogocup
visible	bit	
actual	bit	

Tabla 3.5. Descripción de la tabla nom_campoestruc

Nombre: nom_campoestruc		
Descripción:		
Atributo:	Tipo:	Descripción:
idcampo	numérico	Identificador de la tabla
idnomeav	numérico	Identificador de valor eav correspondiente
nombre	texto	Nombre del nomenclador
tipo	texto	Tipo del campo de la estructura
longitud	numérico	Longitud del campo de la estructura
nombre_mostrar	texto	Nombre a mostrar en el campo de la estructura
regex	texto	Expresión regular
descripcion	texto	Descripción
tipocampo	texto	Tipo del campo(combobox, un textbox, etc.)
visible	numérico	Si es visible o no el campo
obligatorio	numérico	Si es obligatorio o no
eliminable	numérico	
visual	numérico	
actual	numérico	
orden	numérico	Orden que ocupa en la tabla

Tabla 3.6. Descripción de la tabla nom_cargocivil

Nombre: nom_cargocivil		
Descripción: cargos civiles		
Atributo:	Tipo:	Descripción:

idcargociv	numérico	Identificador de la tabla
dencargociv	texto	Denominación
abrevcargociv	texto	Abreviatura
orden	numérico	Orden que ocupa en la tabla
idcategocup	numérico	
descripcion	texto	Descripción del cargo
requisitos	texto	Guarda los requisitos que debe cumplir para ocupar el cargo.
idcalificador	numérico	Identificador de calificador
idgrupocomplejidad	numérico	Identificador de grupo complejidad.
idnivelutilizacion	numérico	Identificador de nivel de utilización.
idespecialidad	numérico	Identificador de especialidad.
idcategcivil	numérico	Identificador de categoría civil.
codigo	numérico	Guarda el código de la estructura.
actual	bit	
visible	bit	

Tabla 3.7. Descripción de la tabla nom_categcivil

Nombre: nom_categcivil		
Descripción: categoría civil		
Atributo:	Tipo:	Descripción:
idcategcivil	numérico	Identificador de la tabla
dencategcivil	texto	Denominación
abrevcategcivil	texto	Abreviatura
essueldo	bit	Para saber si la categoría civil lleva nivel funcional
idcategmilitar	numérico	Identificador de categoría militar
orden	numérico	Orden que ocupa en la tabla
actual	bit	
visible	bit	

Tabla 3.8. Descripción de la tabla nom_cargomilitar

Nombre: nom_cargomilitar		
Descripción: cargo militar		
Atributo:	Tipo:	Descripción:
idcargomilitar	numérico	Identificador de la tabla
dencargomilitar	texto	Denominación
abrevcargomilitar	texto	Abreviatura
idespecialidad	numérico	Identificador de especialidad
idcalificador	numérico	Identificador de calificador
idcategmilitar	numérico	Identificador de categoría militar
idfamiliacargo	numérico	Identificador de familia de cargos
orden	numérico	Orden que ocupa en la tabla
actual	bit	
visible	bit	

Tabla 3.9. Descripción de la tabla nom_categmilitar

Nombre: nom_categmilitar		
Descripción: categoría militar		
Atributo:	Tipo:	Descripción:
idcategmilitar	numérico	Identificador de la tabla
denominacion	texto	Denominación
abreviatura	texto	Abreviatura
orden	numérico	Orden que ocupa en la tabla
actual	bit	
visible	bit	

Tabla 3.10. Descripción de la tabla nom_categocup

Nombre: nom_categocup		
Descripción: categoría ocupacional		

Atributo:	Tipo:	Descripción:
idcategocup	numérico	Identificador de la tabla
dencategocup	texto	Denominación
abreviatura	texto	Abreviatura
orden	numérico	Orden que ocupa en la tabla
actual	bit	
visible	bit	

Tabla 3.11. Descripción de la tabla nom_clasificacion_cargo

Nombre: nom_clasificacion_cargo		
Descripción: clasificación de cargos		
Atributo:	Tipo:	Descripción:
idclasificacion	numérico	Identificador de la tabla
denominacion	texto	Denominación
orden	numérico	Orden de la estructura
actual	bit	
visible	bit	

Tabla 3.12. Descripción de la tabla nom_clasificacionunidad

Nombre: nom_clasificacionunidad		
Descripción: clasificación de unidades		
Atributo:	Tipo:	Descripción:
idclasificacionunidad	numérico	Identificador de la tabla
dencategocup	texto	Denominación
abreviatura	texto	Abreviatura
orden	numérico	Orden que ocupa en la tabla
actual	bit	
visible	bit	

Tabla 3.13. Descripción de la tabla nom_clasificaciondocumento

Nombre: nom_clasificaciondocumento		
Descripción: clasificación de documento		

Atributo:	Tipo:	Descripción:
idclasificaciondocumento	numérico	Identificador de la tabla
denominacion	texto	Denominación
machon	texto	machón
orden	numérico	Orden que ocupa en la tabla
actual	bit	
visible	bit	

Tabla 3.14. Descripción de la tabla nom_dominio

Nombre: nom_dominio		
Descripción: dominio de las estructuras		
Atributo:	Tipo:	Descripción:
iddominio	numérico	Identificador del dominio
denominacion	texto	Denominación
dominiostring	texto	Cadena del dominio
descripcion	texto	Descripción
seguridad	entero	
dominio	arreglo de bit	
idpadre	numérico	Identificador del padre
lft	numérico	Izquierdo del árbol
rgt	numérico	Derecho del árbol
dominiob	bit	

Tabla 3.15. Descripción de la tabla nom_escalasalarial

Nombre: nom_escalasalarial		
Descripción: escala salarial		
Atributo:	Tipo:	Descripción:
idescalasalarial	numérico	Identificador de la escala salarial
denominacion	texto	Denominación
abreviatura	texto	Abreviatura
orden	numérico	Orden de la escala salarial
actual	bit	

visible	bit	
---------	-----	--

Tabla 3.16. Descripción de la tabla nom_especialidad

Nombre: nom_especialidad		
Descripción: especialidades		
Atributo:	Tipo:	Descripción:
idespecialidad	numérico	Identificador de la especialidad
denespecialidad	texto	Denominación
abrevespecialidad	texto	Abreviatura
codespecialidad	texto	Código de la especialidad
version	numérico	
orden	numérico	Orden que ocupa en la tabla
icono	texto	
fechaini	fecha	Fecha inicio
fechafin	fecha	Fecha fin
actual	bit	
visible	bit	
espabastece	bit	
idpadre	número	

Tabla 3.17. Descripción de la tabla nom_modulotecnica

Nombre: nom_modulotecnica		
Descripción: componentes o modulo-técnica		
Atributo:	Tipo:	Descripción:
idmodulo	numérico	Identificador del modulo
idtecnica	numérico	Identificador de la técnica
ctp	numérico	Cantidad en tiempo de paz
ctg	numérico	Cantidad en tiempo de guerra
orden	numérico	Orden que ocupa en la tabla
actual	bit	
visible	bit	

Tabla 3.18. Descripción de la tabla nom_familiacargo

Nombre: nom_familiacargo		
Descripción: familia de cargos		
Atributo:	Tipo:	Descripción:
idfamiliacargo	numérico	Identificador de la tabla
denominacion	texto	Denominación
orden	numérico	Orden que ocupa en la tabla
actual	bit	
visible	bit	

Tabla 3.19. Descripción de la tabla nom_filaestruc

Nombre: nom_filaestruc		
Descripción: firma		
Atributo:	Tipo:	Descripción:
idfila	numérico	Identificador de la tabla
idnomeav	numérico	Identificador del eav correspondiente
idpadre	numérico	Identificador del padre
lft	numérico	Izquierdo del árbol
rgt	numérico	Derecho del árbol
orden	numérico	Orden que ocupa en la tabla
antiguo	texto	
dominio	entero	
md5	texto	
dominiofila	arreglo de bit	Dominio de la fila

Tabla 3.20. Descripción de la tabla nom_firmas

Nombre: nom_firmas		
Descripción: firma		
Atributo:	Tipo:	Descripción:
idfirma	numérico	Identificador de firma
idgrado	texto	Identificador de grado

nombre	texto	nombre
aprueba	numérico	Jefe que aprueba
cometario	Texto	comentario
cargo	texto	Cargo que ocupa
orden	numérico	Orden que ocupa en la tabla
actual	bit	
visible	bit	

Tabla 3.21. Descripción de la tabla nom_gradomarina

Nombre: nom_gradomarina		
Descripción: grados de la marina		
Atributo:	Tipo:	Descripción:
idgradomilit	numérico	Identificador de la tabla
idgradoentierra	numérico	Identificador de gradomilit

Tabla 3.22. Descripción de la tabla nom_gradomilit

Nombre: nom_gradomilit		
Descripción: grados militares		
Atributo:	Tipo:	Descripción:
idgradomilit	numérico	Identificador de la tabla
dengradomilit	texto	Denominación
abrevgradomilit	texto	Abreviatura
homologoterr	numérico	Identificador del grado homologo territorial
anterior	numérico	Identificador del grado anterior
sucesor	numérico	Identificador del grado sucesor
orden	numérico	Orden del grado militar
idgsubcateg	numérico	Identificador del grado de la subcategoría
esmarina	numérico	Si el grado pertenece a la marina o no
actual	bit	
visible	bit	

Tabla 3.23. Descripción de la tabla nom_grupocomple

Nombre: nom_grupocomple		
Descripción: complejidad		
Atributo:	Tipo:	Descripción:
idgrupocomplejidad	numérico	Identificador de la tabla
denominacion	texto	Denominación
abreviatura	texto	Abreviatura
idescalasalarial	numérico	Identificador de escala salarial
salarioescala	numérico	Salario de la escala
orden	numérico	Orden de los grupos de complejidad
actual	bit	
visible	bit	

Tabla 3.24. Descripción de la tabla nom_gsubcateg

Nombre: nom_gsubcateg		
Descripción: subcategorías		
Atributo:	Tipo:	Descripción:
idgsubcateg	numérico	Identificador de la tabla
densubcateg	texto	Denominación
orden	numérico	Orden de la subcategoría
idcategmtar	numérico	Identificador de categmilitar
abreviatura	texto	Abreviatura
actual	bit	
visible	bit	

Tabla 3.25. Descripción de la tabla nom_maqtecnica

Nombre: nom_maqtecnica		
Descripción: máquina base		
Atributo:	Tipo:	Descripción:
idmaqtecnica	numérico	Identificador de la tabla

denominacion	texto	Denominación
abreviatura	texto	Abreviatura
orden	numérico	Orden de la máquina base
visible	bit	
actual	bit	

Tabla 3.26. Descripción de la tabla nom_modulo

Nombre: nom_modulo		
Descripción: módulos		
Atributo:	Tipo:	Descripción:
idmodulo	numérico	Identificador de la tabla
denominacion	texto	Denominación
abreviatura	texto	Abreviatura
especialidad	numérico	Identificador de la especialidad
orden	numérico	Orden del módulo
actual	bit	
visible	bit	
esmodulo	bit	Si es módulo o no

Tabla 3.27. Descripción de la tabla nom_modulotecnica

Nombre: nom_modulotecnica		
Descripción: módulo y técnica		
Atributo:	Tipo:	Descripción:
idmodulo	numérico	Identificador de la tabla
idtecnica	numérico	Identificador de técnica
ctp	numérico	Cantidad en tiempo paz
ctg	numérico	Cantidad en tiempo guerra
orden	numérico	Orden que ocupa en la tabla
actual	bit	
visible	bit	

Tabla 3.28. Descripción de la tabla nom_nivel_utilizacion

Nombre: nom_nivel_utilizacion		
Descripción: niveles		
Atributo:	Tipo:	Descripción:
idnivelutilizacion	numérico	Identificador de la tabla
denominacion	texto	Denominación
abreviatura	texto	Abreviatura
orden	numérico	Orden del nivel de utilización
actual	bit	
visible	bit	

Tabla 3.29. Descripción de la tabla nom_nivelestr

Nombre: nom_nivelestr		
Descripción: nivele estructural		
Atributo:	Tipo:	Descripción:
idnivelestr	numérico	Identificador de la tabla
abrevnivelestr	texto	Abreviatura
dennivelestr	texto	Denominación
orden	numérico	Orden del nomenclador
actual	bit	
visible	bit	

Tabla 3.30. Descripción de la tabla nom_nomencladoreavestruc

Nombre: nom_nomencladoreavestruc		
Descripción: nomenclador eav		
Atributo:	Tipo:	Descripción:
idnomeav	numérico	Identificador de la tabla
nombre	texto	Nombre de la estructura referenciada
fechaini	fecha	Fecha inicio del uso del nomenclador
fechafin	fecha	Fecha fin del uso del nomenclador

root	bit	Se usa si es usada la recursividad dentro de las mismas tablas
concepto	numérico	Si la estructura es concepto o no
externa	numérico	Si la estructura es externa o no
orden	numérico	Orden del nomenclador
iconoorganigrama	texto	
icono	texto	

Tabla 3.31. Descripción de la tabla nom_ordtecnica

Nombre: nom_ordtecnica		
Descripción:		
Atributo:	Tipo:	Descripción:
codigo	texto	Código
denominacion	texto	Denominación
abreviatura	texto	Abreviatura
orden	numérico	Orden del nomenclador
idnomordtecnica	texto	Identificador de la tabla
actual	bit	
visible	bit	

Para consultar el resto de las descripciones de las tablas, ver anexo 27.

3.11. Diagrama de despliegue

Un diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos.

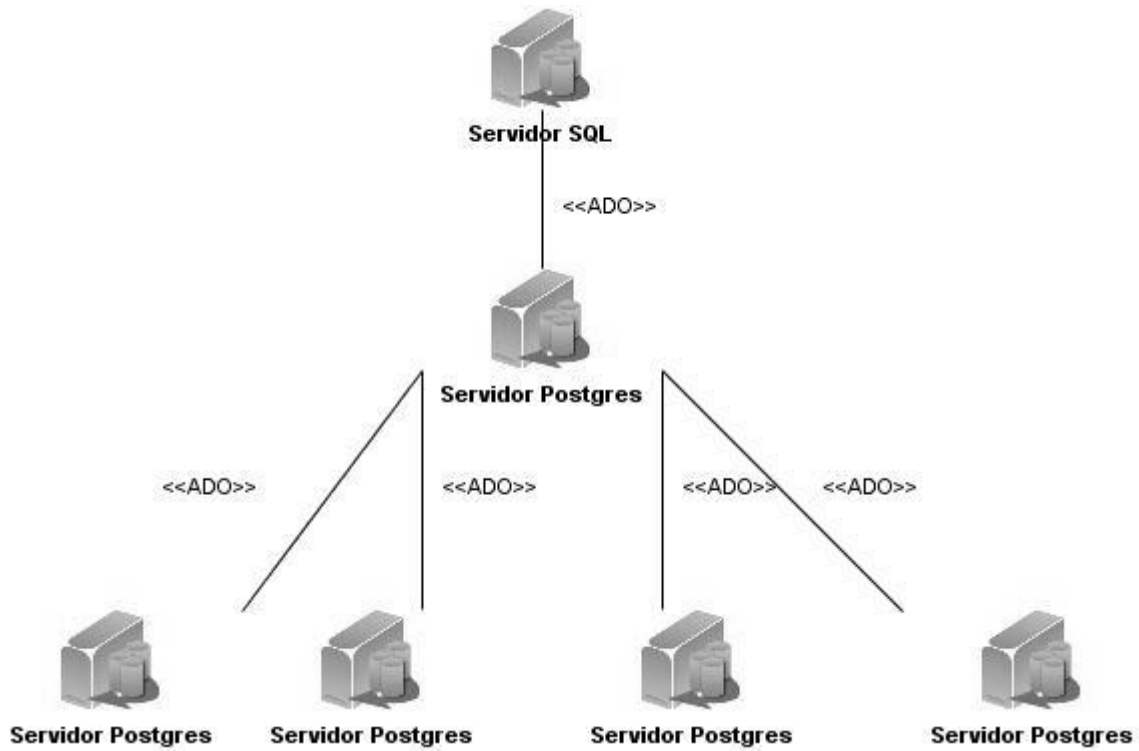


Figura 22: Diagrama de despliegue1.

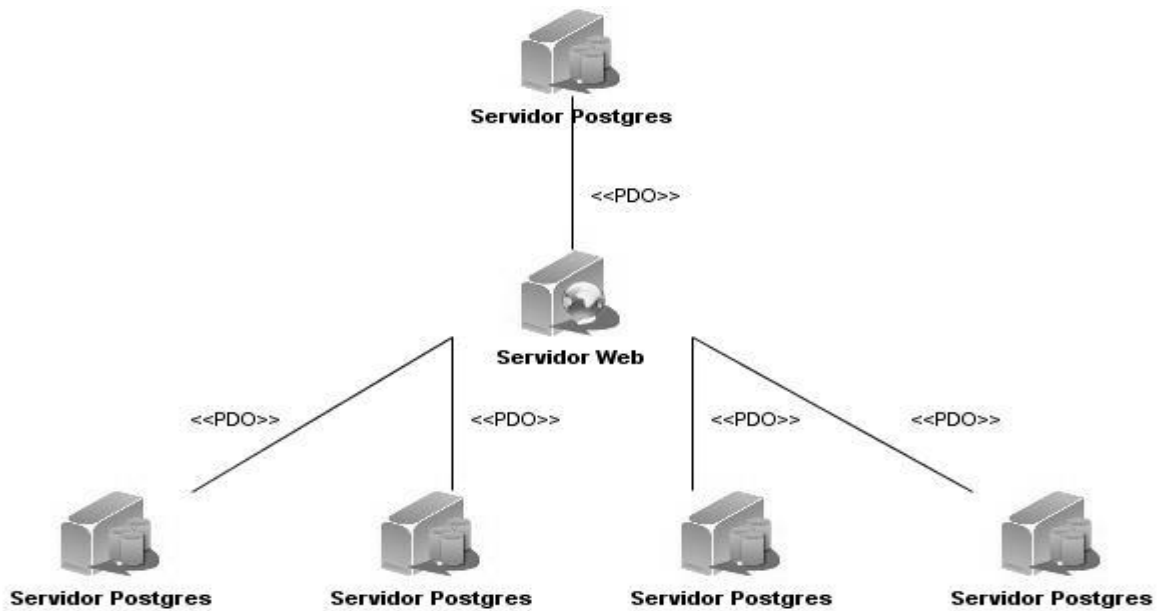


Figura 23: Diagrama de despliegue2.

3.12. Prueba

La prueba de un software es un proceso que corre en paralelo al proceso de desarrollo y que se realiza por el convencimiento de que todo sistema debe ser “revisado” con el objetivo de establecer el nivel de calidad requerido.

En este caso, el proceso de prueba se realizó con el objetivo de detectar insatisfacción de los requerimientos planteados y corregir estos errores.

La primera etapa de prueba se realizó en el centro UCID, tomándose como muestra una copia de la base de datos SIDEC versión 2.0. El proceso se realizó primeramente desde un servidor origen hacia un servidor destino y se puso en práctica en varias bases de datos del centro. Luego de comprobar que los resultados fueron buenos, se decidió realizar el proceso hacia varias bases de datos destino obteniéndose buenos resultados.

La segunda etapa se realizó en el MINFAR por especialistas, tomándose como muestra servidores destinados a esa función. Al principio se detectaron algunos errores:

- Algunas tablas y campos de la base de datos no coincidieron.

```

Error:
12/04/2010 12:22:00 AM
Npgsql.NpgsqlException:
relation "replica.tabla" does not exist
Severity: ERROR
Code: 42P01
at DataReplication.SchemaReplicator.Replicate()
at DataReplication.DbSynchronizer.e()
    
```

Figura 24: Error del sistema

- Faltaron algunos scripts necesarios para la realización de la réplica.

Ejemplo:

```
ALTER TABLE "mod_estructuracomp"."nom_gradomilit"
ADD CONSTRAINT "fk_anterior" FOREIGN KEY ("anterior")
REFERENCES "mod_estructuracomp"."nom_gradomilit" ("idgradomilit")
ON DELETE NO ACTION
ON UPDATE NO ACTION
DEFERRABLE
INITIALLY DEFERRED;
```

Figura 25: Script necesario por la réplica

Las pruebas realizadas fueron las siguientes:

1. Cálculo del tiempo que demora en realizarse la réplica.

Tabla 3.45: Pruebas de tiempo

Tabla	Cant. Tuplas	Tiempo(s)
nom_dominio	11	6
nom_especialidad	65	9-10
nom_filaestruc	168278	2700
las tres tablas	168354	2709

2. Insertar, modificar y eliminar datos de la base de datos origen y una vez realizado el proceso de réplica verificar que esta información estuviese también en el destino.

Todo esto a través de la ejecución de consultas SQL y conectando el SIDEC a la base de datos de cada dominio para asegurar que el usuario solo accediera a la información correcta.

Ejemplo:

- **Select idcargo from mod_estructuracomp.dat_cargo where idcargo in (select idfila from mod_estructuracomp.nom_filaestruc where dominiofila[10] = 1 :: bit (1))**
- **Select idcargociv from mod_estructuracomp.dat_cargocivil where idcargo in (select idfila from mod_estructuracomp.nom_filaestruc where dominiofila[10] = 1 :: bit (1))**

Finalmente, se corrigieron los errores, las pruebas dieron resultados positivos y se puso en marcha el sistema de réplica.

3.13. Conclusiones

En este capítulo se realizó una descripción de todo el proceso de migración y réplica de los datos como parte de la solución a la problemática existente. Se analizaron además el modelo de datos y el diagrama de despliegue y se explicó el proceso de prueba realizado al sistema.

Conclusiones

A partir de los objetivos planteados y el trabajo realizado en esta investigación, se arribó a las siguientes conclusiones:

- Se realizó un estudio de los conceptos indispensables para la comprensión del proceso de migración y réplica de datos con el uso de las tecnologías actuales.
- Se logró solucionar el problema a resolver que dio origen a la investigación, pues se cumplió el objetivo general trazado, al establecer una estrategia eficiente para la migración y réplica de datos del Sistema para el Aseguramiento de la Actividad de Estructura y Composición de las FAR.
- Se realizó la transformación y réplica de los datos, obteniendo resultados satisfactorios.

Recomendaciones

Como resultado de la investigación y elementos a tener en cuenta, se hacen las siguientes recomendaciones:

- Añadir a los tipos de datos que soporta la herramienta, el tipo de dato bit [], para mejor funcionalidad de la misma.
- Continuar investigando las herramientas que surgen al paso de los días y el desarrollo de la tecnología en el tema de la migración y réplica de datos, y complementar el sistema con estos avances en versiones posteriores.

Referencias Bibliográficas

1. Proceso ETL . *ETL-Tools.Info*. [En línea] GolilInfo company, 2010. HYPERLINK "http://etl-tools.info/es/bi/proceso_etl.htm." http://etl-tools.info/es/bi/proceso_etl.htm.
2. **Herrera, Alvaro A.** Migración de datos. *mailxmail.com*. [En línea] 2007. HYPERLINK "http://www.mailxmail.com/curso-migracion-datos/migracion-datos." <http://www.mailxmail.com/curso-migracion-datos/migracion-datos>.
3. **Pupo, Rosell y González, Yenier.** *Implementacion del componente replica de base de datos para Akademos v2.0*. Ciudad de la Habana : s.n., 2009.
4. **Rodríguez, Nancy y Suárez, Mercedes.** *Replicación de Datos*. 2007.
5. Sincronización de Datos. *Talend open data solution*. [En línea] Talend, 2008. HYPERLINK "http://es.talend.com/solutions-data-integration/data-synchronization.php." <http://es.talend.com/solutions-data-integration/data-synchronization.php>.
6. La Definición de Software Libre. *GNU Operating System*. [En línea] 2010. HYPERLINK "http://www.gnu.org/philosophy/free-sw.es.html" <http://www.gnu.org/philosophy/free-sw.es.html> .
7. **Van Der Henst, Christian.** Licencias libres de Software (II). *maestros del web*. [En línea] 2008. HYPERLINK "http://www.maestrosdelweb.com/editorial/licencias-libres-de-software-ii/" <http://www.maestrosdelweb.com/editorial/licencias-libres-de-software-ii/> .
8. Qué es el Open Source. *AltaVoz*. [En línea] AltaVoz S.A. HYPERLINK "http://www.altavoz.net/prontus_altavoz/site/artic/20060304/pags/20060304151415.html" http://www.altavoz.net/prontus_altavoz/site/artic/20060304/pags/20060304151415.html .
9. **Carrillo, Alma N, Cervantes, Freddy y Gutiérrez, Miguel.** SQL SERVER. *Universidad Autónoma del Estado de Morelos*. [En línea] 2010. HYPERLINK "http://www.uaem.mx/posgrado/mcruz/cursos/miic/sql.pdf" <http://www.uaem.mx/posgrado/mcruz/cursos/miic/sql.pdf> .
10. PgAdmin III. *Guía documentada para Ubuntu*. [En línea] 2008. HYPERLINK "http://www.guia-ubuntu.org/index.php?title=PgAdmin_III" http://www.guia-ubuntu.org/index.php?title=PgAdmin_III .

11. **Rodas, Raul.** Características de PHP. *LinuxCentro.net*. [En línea] 2007. HYPERLINK
"http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP"
<http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP> .
12. Conceptos básicos del servidor web. *Cibernetia*. [En línea] 2010. HYPERLINK
"http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php"
http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php .
13. **Programacionphp.net.** mailxmail.com. [En línea] 2006. HYPERLINK
"http://www.mailxmail.com/curso-configuracion-apache/que-es-apache-que-alternativas-hay-mercado"
<http://www.mailxmail.com/curso-configuracion-apache/que-es-apache-que-alternativas-hay-mercado> .
14. Presentation. *WampServer*. [En línea] 2009. HYPERLINK
"http://www.wampserver.com/en/presentation.php" <http://www.wampserver.com/en/presentation.php> .
15. Qué es un Navegador Web . *Básico y Fácil*. [En línea] 2010. HYPERLINK
"http://basicoyfacil.wordpress.com/2008/10/23/que-es-un-navegador-web/"
<http://basicoyfacil.wordpress.com/2008/10/23/que-es-un-navegador-web/> .
16. **DingoZOE.** Qué es Mozilla Firefox. *Firefox Peru*. [En línea] 2009. HYPERLINK
"http://firefoxperu.blogspot.com/2005/12/qu-es-mozilla-firefox.html"
<http://firefoxperu.blogspot.com/2005/12/qu-es-mozilla-firefox.html> .
17. Notepad++ 5.6.8. *FileHippo.com*. [En línea] 2010. HYPERLINK
"http://www.filehippo.com/es/download_notepad/" http://www.filehippo.com/es/download_notepad/ .
18. **Álvaro, Moreno Aguilera.** Herramientas ETL de código abierto . [En línea] 2008. HYPERLINK
"http://webcache.googleusercontent.com/search?q=cache:C8I2fpUJbyUJ:svn2.assembla.com/svn/bbdd_dd/Pr
esentaciones/Kettle%2520Vs%2520Talend.pptx+caracteristicas+de+scriptella&cd=4&hl=es&ct=clnk&gl=cu"
http://webcache.googleusercontent.com/search?q=cache:C8I2fpUJbyUJ:svn2.assembla.com/svn/bbdd_dd/Presentaciones/Kettle%2520Vs%2520Talend.pptx+caracteristicas+de+scriptella&cd=4&hl=es&ct=clnk&gl=cu .
19. RCE. *Red de Cooperación Empresarial*. [En línea] 2005. HYPERLINK
"http://redtt.apte.org/rib/view_od?od_id=3763" http://redtt.apte.org/rib/view_od?od_id=3763 .

20. **SAC, Empresa ARSON Group.** Bitool Planificador de procesos de Cargas. [En línea] 2005. HYPERLINK "http://www.bitool.com/bitool.htm" <http://www.bitool.com/bitool.htm> .
21. **Benoît, Carpentier.** Benetl is a free ETL tool. [En línea] 2007. HYPERLINK "http://www.benetl.net/" <http://www.benetl.net/> .
22. **Team., Scriptella Project.** Scriptella. [En línea] 2006-2009. HYPERLINK "http://scriptella.javaforge.com/" <http://scriptella.javaforge.com/> .
23. Visual Paradigm for UML. *Sitio de Descargas de Software.* [En línea] 2007. HYPERLINK "http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/" http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/ .
24. **Abiague, Alejandro y Gómez, Erich M.** *Solución para la Réplica de Datos del ERP.* Universidad de las Ciencias Informáticas, s.l. : 2008.

Bibliografía

- AltaVoz. ¿Qué es el Open Source? 2006. Available from world wide web: http://www.altavoz.net/prontus_altavoz/antialone.html?page=http://www.altavoz.net/prontus_altavoz/site/artic/20060304/pags/20060304151415.html
- Álvarez S. Sistemas gestores de bases de datos. 2007. Available from world wide web: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>
- Chovas G. Tutoriales, manuales de UML. 2007. Available from world wide web: <http://manuales.astalaweb.com/Manuales/UML.asp>
- Corona C. S. Asegure una migración exitosa de bases de datos. 2006. Available from world wide web: <http://www.enterate.unam.mx/Articulos/2006/agosto/migracion.htm>
- ETL para Analytics. 2008. Available from world wide web: <http://es.talend.com/solutions-data-integration/etl-for-analytics.php>
- Fernández J. Modelo de Datos. Available from world wide web: <http://aurea.es/wp-content/uploads/modelodedatos.pdf>
- FSF. Licencias. 2009. Available from world wide web: <http://www.gnu.org/licenses/licenses.es.html>
- FSF. ¿Qué es el copyleft?. 2009. Available from world wide web: <http://www.gnu.org/copyleft/copyleft.es.html>
- GEA. Tipos de Migración. Available from world wide web: http://www.encyclopedia-aragonesa.com/voz.asp?voz_id=8796#
- González C. D. Lenguaje PHP. 2010. Available from world wide web: <http://www.usabilidadweb.com.ar/php.php>
- González U. M. SQL SERVER. Available from world wide web: <http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml>
- Griñán J.A. Navegadores Web. 2010. Available from world wide web: http://www.juntadeandalucia.es/averroes/recursos_informaticos/curso_internet/curso_nvweb.htm

- Introducción a un BPM. Available from world wide web: <http://www.willydev.net/InsiteCreation/v1.0/WillyCrawler/2008.06.08.Introduccion%20a%20un%20BP M.pdf>
- Johan. Lenguaje de modelado unificado. 2009. Available from world wide web: <http://www.gratisblog.com/index.php?itemid=130343>
- Migración de datos. 2008. Available from world wide web: <http://es.talend.com/solutions-data-integration/data-migration.php>
- Monge R. *Sistemas Distribuidos de Computación*. [Valparaíso], 2005 Available from World Wide Web: http://148.208.239.12/dep/sada/carreras/Ingenieria%20en%20Sistemas%20Computacionales/5to%20Semestre/Taller%20de%20Base%20de%20Datos/taller_bd/replicacion.pdf
- Navegador web Firefox. 2008. Available from world wide web: <http://www.mozilla-europe.org/es/firefox/#feature-features>
- Nieto M. A. Arquitectura Maestro / Esclavo en Mysql. 2010. Available from world wide web: <http://miguelangelnieto.net/?action=view&url=arquitectura-maestro-esclavo-en-mysql>
- Open Source Initiative. The Open Source Definition. 2010. Available from world wide web: <http://www.opensource.org/docs/definition.php>
- ¿Qué es un servidor web? 2010. Available from world wide web: <http://www.masadelante.com/faqs/servidor-web>
- Significado y definición de Software Propietario. 2004. Available from world wide web: <http://www.mastermagazine.info/termino/6751.php>
- Sistema Gestor de base de datos SGBD. 2004. Available from world wide web: http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbid.php
- Talend. 2010. Available from world wide web: <http://es.talend.com/>
- Vega J. H, y Herrera R. C. Software de control remoto para aplicaciones del escritorio GNOME. 2004. Available from world wide web: http://gbtcr.chileforge.cl/info_web/informe_proyecto.html

Glosario de términos

Aplicación: A menudo, se refiere al programa que se está ejecutando y a los archivos y bases de datos con los que se trabaja.

Artefactos: En términos de ingeniería de software, los artefactos son productos tangibles del proceso.

Consulta: En términos de base de datos, es un código donde se describe una acción a ejecutar.

Esquema: Un esquema es la definición de una estructura (generalmente relaciones o tablas de una base de datos), es decir, determina la identidad de la relación y que tipo de información podrá ser almacenada dentro de ella; en otras palabras: son los metadatos de la relación.

GNU: Proyecto con el objetivo de crear un sistema operativo completamente libre: el sistema GNU. Se establecieron algunas motivaciones para realizar el proyecto GNU, entre las que destaca "volver al espíritu de cooperación que prevaleció en los tiempos iniciales de la comunidad de usuarios de computadoras".

Hipertexto: En informática, es el nombre que recibe el texto que, en la pantalla de una computadora, conduce al usuario a otro texto relacionado. La forma más habitual de hipertexto en documentos es la de hipervínculos o referencias cruzadas automáticas que van a otros documentos.

HTML (Lenguaje de Marcas de Hipertexto, HyperText Markup Language en Inglés): Es el lenguaje de marcado predominante para la construcción de páginas web.

Identificador: Nombran entidades. El concepto es análogo al de "nombre". Los identificadores se usan en prácticamente todos los sistemas de procesamiento de la información. Nombrar las entidades hace posible referirse a las mismas, lo cual es esencial para cualquier tipo de procesamiento simbólico.

Internet: Es un conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

IP (El Protocolo de Internet, en inglés Internet Protocol): Es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

JDK (Java Development Kit): es un software que provee herramientas de desarrollo para la creación de programas en java. Puede instalarse en una computadora local o en una unidad de red.

JRE (Java Runtime Environment): es un conjunto de utilidades que permite la ejecución de programas Java.

Java: es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90.

Llave: Identificador usado en registros de tablas dentro de una base de datos. Cuando identifica en forma única a un registro es llamada llave primaria.

Navegador web (del inglés, navigator o browser): Es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet.

Nodo: En informática, un nodo es "Punto de intersección o unión de varios elementos que confluyen en el mismo lugar". Ejemplo: En una red de ordenadores cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo.

Open source: Open Source es una marca de certificación propiedad de la Open Source Initiative. El 16 software OpenSource se define por la licencia que lo acompaña, que garantiza a cualquier persona el 17 derecho de usar, modificar y redistribuir el código libremente.

ODBC: (del inglés Open Database Connectivity Standard) Estándar de conectividad abierta de base de 19 datos.

Página web: Es una fuente de información adaptada para la WWW y accesible mediante un navegador de Internet. Esta información se presenta generalmente en formato HTML y puede contener hiperenlaces a otras páginas web.

Réplica: Es el proceso de duplicar o mas los datos existentes en una base de datos a otra(s). En muchos entornos se usan la réplica por espejo y la réplica por fragmentos de datos.

Servidor: En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Sincronización: Se refiere al proceso de propagación de los cambios en los datos y el esquema entre la fuente de datos y los destinos después de haber aplicado la instantánea inicial en el destino.

Software: Conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware).

Software: Se refiere al equipamiento lógico o soporte lógico de un computador digital, comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica.

SQL: Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar, de una forma sencilla, información de interés de una base de datos, así como también hacer cambios sobre la misma.

TCP/IP o familia de protocolos de Internet: Es un conjunto de protocolos de red en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. En ocasiones se le denomina “conjunto de protocolos TCP/IP”, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

TCP: Es un protocolo de comunicación orientado a conexión y fiable del nivel de transporte.

Transacción: Es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica. Cuando por alguna causa el sistema debe cancelar la transacción, empieza a deshacer las órdenes ejecutadas hasta dejar la base de datos en su estado inicial (llamado punto de integridad), como si la orden de la transacción nunca se hubiese realizado.

Trigger: Es un evento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE) en la base de datos.

Tupla: En la teoría de bases de datos, una tupla se define como una función finita que mapea (asocia unívocamente) los nombres con algunos valores.

W3C (World Wide Web Consortium): Es un consorcio internacional que produce estándares para la WWW.

World Wide Web o Red Global Mundial (WWW): Es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet. Con un navegador Web, un usuario visualiza páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

XML: *Extensible Markup Language* (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el W3C, que permite definir la gramática de lenguajes específicos.