

## Control de cambios y gestión de la configuración de la base de datos

### *Changes control and configuration management on databases*

Erik de la Vega García

Universidad de las Ciencias Informáticas

[edelavega@uci.cu](mailto:edelavega@uci.cu)

#### **Resumen**

Durante el proceso de desarrollo de software la gestión de la configuración es uno de los procesos que requiere esfuerzo durante todo el ciclo. Realizar una correcta planificación de los procesos y el control estricto sobre los cambios es necesario para lograr un producto de calidad y disminuir los riesgos a lo largo del desarrollo del proyecto. La gestión de la configuración es primordial para todo proyecto informático y debe ser aplicado en todas las áreas de desarrollo, incluyendo por supuesto las bases de datos.

En las bases de datos la gestión de la configuración es un tema complejo, sobre todo en aquellos proyectos que tienen un equipo de desarrollo grande. Se debe controlar cuidadosamente cada cambio en la base de datos, almacenar cada versión tanto a nivel de objeto como de la base de datos en su conjunto y documentar correctamente cada uno de los elementos de la base de datos. En las bases de datos cada cambio debe ser analizado con mucho cuidado y documentar hasta el más mínimo detalle. La gestión de la configuración y el control de cambios en las bases de datos son esencialmente necesarios e importantes.

**Palabras clave:** Base de datos, control de cambios, gestión de la configuración, versión.

#### **Abstract**

*Configuration management is a process that requires effort throughout all software development process. Perform proper planning processes and strict control over the changes is necessary to achieve a quality product and reduce the risks along the project's development. The configuration management is essential for all IT projects and should be implemented in all areas of development, including of course the databases.*

*In databases configuration management is a complex issue, especially in those projects that have a large development team. In the databases every change should be monitor, storing each version of objects and of the whole database, and properly documenting each element on the database. In databases changes should be analyzed very carefully and documenting evens the smallest detail. The configuration management and control of changes in databases is essentially necessary and important.*

**Key words:** *Configuration management, databases, changes control, version.*

#### **Introducción**

El control de cambios es tan necesario en el desarrollo de un sistema informático, como el sistema mismo. Un control estricto y una gestión de la configuración eficiente, disminuye los riesgos en el proyecto y las posibilidades de éxito aumentan considerablemente. En las bases de datos, la gestión de la configuración es muy compleja pues intervienen muchos factores, el control de los cambios en la programación, la estructura de cada uno de los objetos y el control sobre los datos que inicialmente deben estar en las tablas para cada versión. En otras palabras hay que controlar tanto los DML como los DDL ejecutados en la base de datos, sin descartar cualquier cambio.

## Desarrollo

### Gestión de la configuración

Gestión de la configuración es el conjunto de procesos destinados a asegurar la validez de todo producto obtenido durante cualquiera de las etapas del desarrollo de un sistema de información a través del estricto control de los cambios realizados sobre los mismos y de la disponibilidad constante de una versión estable de cada elemento para toda persona involucrada en el citado desarrollo. Estos dos elementos (control de cambios y control de versiones de todos los elementos del sistema informático) facilitan también el mantenimiento de los sistemas al proporcionar una imagen detallada del sistema en cada etapa del desarrollo. La gestión de la configuración se realiza durante todas las fases del desarrollo de un sistema de información, incluyendo el mantenimiento y control de cambios, una vez realizada la puesta en producción. (Wikimedia Foundation, Inc., 2008).

En los sistemas informáticos que usan bases de datos resulta aún más difícil la gestión de la configuración pues es necesario implementar dos métodos: uno para el control de las versiones en las bases de datos y otro para el del código y documentación de la aplicación. Cada versión de una base de datos contiene todos los objetos necesarios para la creación desde cero de una base de datos con esa versión, por lo tanto la versión incluye, tablas, procedimientos almacenados, paquetes, triggers, funciones, índices, tipos de datos, java source, usuarios, roles, etc. Obtener el script de la versión actual de la base de datos es muy fácil.

### Filosofía para los cambios en la base de datos

Los cambios en las bases de datos son inherentes al desarrollo y no se pueden evitar. Llevar un estricto control de cambios y una buena gestión de la configuración de cada base de datos es tarea de primer orden y su incorrecto funcionamiento puede provocar retrasos considerables en el desarrollo o consecuencias catastróficas a la consistencia de los datos cuando se realiza un cambio en el ambiente real.

El control de los cambios de las bases de datos y el manejo de las versiones debe incluir cada una de las etapas del desarrollo y garantizar una estricta y completa sincronización en cuanto a estructura de cada una de las bases de datos del sistema. La estrategia se basa en llevar versiones de las bases de datos por cada versión estable, considerando cada uno de los cambios necesarios que se deben realizar para la puesta en marcha de una nueva versión.

Tres ambientes son involucrados en el proceso, pues es necesario tener en cuenta cada uno de ellos cuando se aplican cambios en la base de datos. Cada ambiente consiste en la en una réplica similar a la del ambiente real en término de arquitectura, y son estructuralmente independientes. Deben existir al menos tres bases de datos diferentes una para cada etapa.

El siguiente esquema muestra cada una de los ambientes y el peso en cuanto a esfuerzo que recibe la gestión de cambios de las bases de datos en cada una.

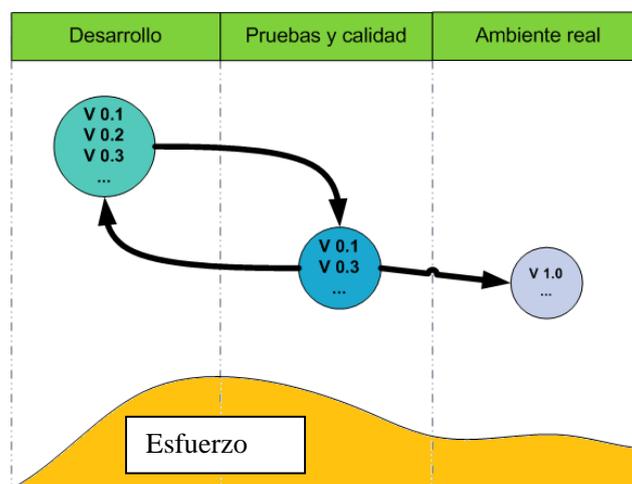


Fig. 1. Gestión de la configuración y esfuerzo destinado durante cada etapa.

En la medida que aumenta el proceso de desarrollo es necesario intensificar el control de los cambios en las bases de datos, cada vez más cerca de la versión final el esfuerzo disminuye pero nunca llega a cero pues aún en las bases de datos pudieran ocurrir cambios necesarios. No todas las versiones que se controlan en cada etapa pasa a la etapa superior, dado que el proceso de desarrollo del software puede ser extenso y conllevar a varios ciclos dentro de la misma etapa.

### Control de los cambios durante el proceso de desarrollo

Durante la etapa de desarrollo, que no es necesariamente la etapa de desarrollo del software, sino también cada una de los momentos que es imprescindible regresar a hacer cambios debido a correcciones o soluciones de problemas, siempre el proceso del control de cambio tiene dos partes importantes, en primer lugar cada cambio realizado dentro de la base de datos ya sea DDL sobre los objetos o DML sobre las tablas nomencladoras debe ser registrado en un sistema de control de versiones especialmente diseñado para la etapa de desarrollo.

Para ello se pueden usar facilidades que tienen los mismos gestores de bases de datos para crear triggers que se levanten ante cualquier evento en la base de datos. La mayoría de los gestores de bases de datos en la actualidad tienen la potencialidad de crear triggers que se disparan ante los eventos DDL en las bases de datos, permitiendo desarrollar la estrategia que se expone a continuación.

En primer lugar es necesario recordar que cada elemento que se crea, modifica o elimina en las bases de datos se realiza mediante un comando SQL que se ejecuta en el servidor. Por lo tanto se puede usar ese SQL para llevar el registro de los cambios que han ocurrido en los objetos.

El modelo de la gestión de cambios basada en el propio gestor está compuesta por dos partes fundamentales triggers que notifican los cambios realizados, tabla para el almacenaje de los objetos modificados, la versión, la fecha, hash del comando SQL que lo crea y en caso necesario el “DDL CREATE” ejecutado. Esta Tabla recibe el nombre de tabla Tbl\_control\_version. De forma simplificada y simple el proceso es el siguiente; un trigger notifica los cambios ocurridos y un procedimiento almacenado se encarga de realizar el resto, que consiste en preguntar si el objeto en cuestión está versionado anteriormente, de ser así se compara los hash del “DDL CREATE” con el almacenado en la tabla Tbl\_control\_version, si son diferentes se registra una nueva versión para el objeto. Si por otra parte el objeto no está versionado pues simplemente se registra la primera versión.

El siguiente diagrama expone el proceso de captura de los cambios para un sistema de control de cambios basado en el gestor de bases de datos.

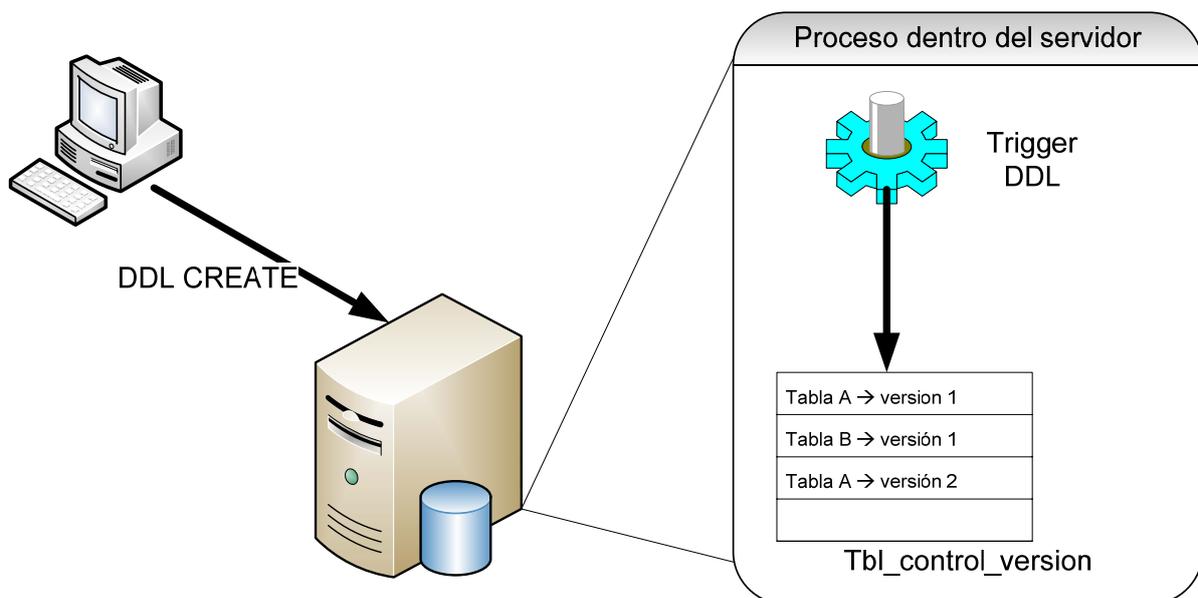


Fig. 2. Control de cambios basado en el propio gestor de BD.

Durante la etapa de desarrollo no se puede olvidar el control de las versiones realizando una extracción completa de la todos los objetos de la base de datos, pues aunque no se escale a una etapa superior como puede ser la prueba de la versión correspondiente, si se debe tener en cuenta la cantidad de cambios realizados, la versión alcanzada de la aplicación y en correspondencia obtener una versión global de la base de datos.

### Flujo del proceso de control de cambios

Cuando una versión de la aplicación es liberada del desarrollo, lleva implícita una versión de la base de datos. En el siguiente paso la versión de la aplicación deberá pasar el control de calidad y por consiguiente su base de datos correspondiente. Para la actualización de las bases de datos donde se realizan las pruebas primero se extrae la versión final del desarrollo de cada una de las bases de datos y después de realizar el procesamiento correspondiente del que se obtiene el script de cambio para cada base de datos, por último se aplica el script de cambio en las bases de datos de calidad. De esta forma se garantiza que la versión final del desarrollo y la versión para las pruebas de calidad de las bases de datos sean las mismas.

El procedimiento para aplicar los cambios es similar al de calidad, pero este solo se realizará teniendo en cuenta que no se perderán datos o se provocará datos en los mismos. Cada una de las bases de datos del sistema tienen su versión registrada en el servidor para la gestión de los cambios de manera que solo es necesario comparar cada una de las diferentes versiones de las bases de datos reales con la versión liberada en calidad para obtener cada uno de los script de sincronización.

El siguiente esquema muestra cada una de las etapas para el control de los cambios en un ambiente Cliente-Centro de Datos, cada uno con sus estructuras de bases de datos independientes.

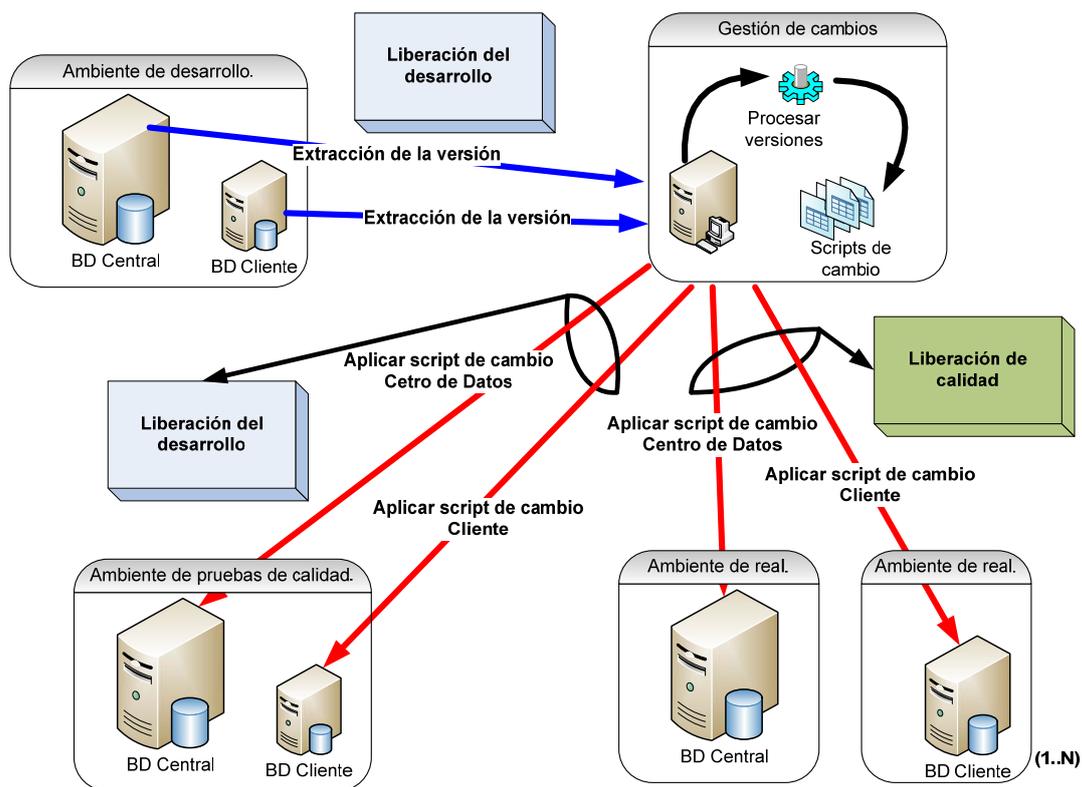


Fig. 3. Flujo del proceso de control de cambio.

El servidor de gestión de cambios almacena el historial y cada una de las versiones actuales de las bases de datos, en cada uno de los ambientes. Por lo que es posible regresar la base de datos a cualquier versión anterior y hacer análisis de los cambios realizados en el transcurso del proyecto.

## Control de versiones

Cada versión de las bases de datos es almacenada de manera que sea posible regresar cualquier base de datos a un estado anterior. A cada base de datos que se crea se le extrae la versión de manera que quede almacenada en términos de script de creación de cada uno de los objetos que la conforman.

La comparación entre la versión nueva (Versión N) y la versión actual existente (Versión N-1) determinará el script de actualización de tal forma que no se afecte la consistencia de los datos almacenados. De igual manera se puede regresar a una versión anterior de la base de datos comparando las versiones en cuestión y aplicando el script de cambio.

El siguiente esquema muestra la obtención del script de cambio para una nueva versión de las bases de datos.

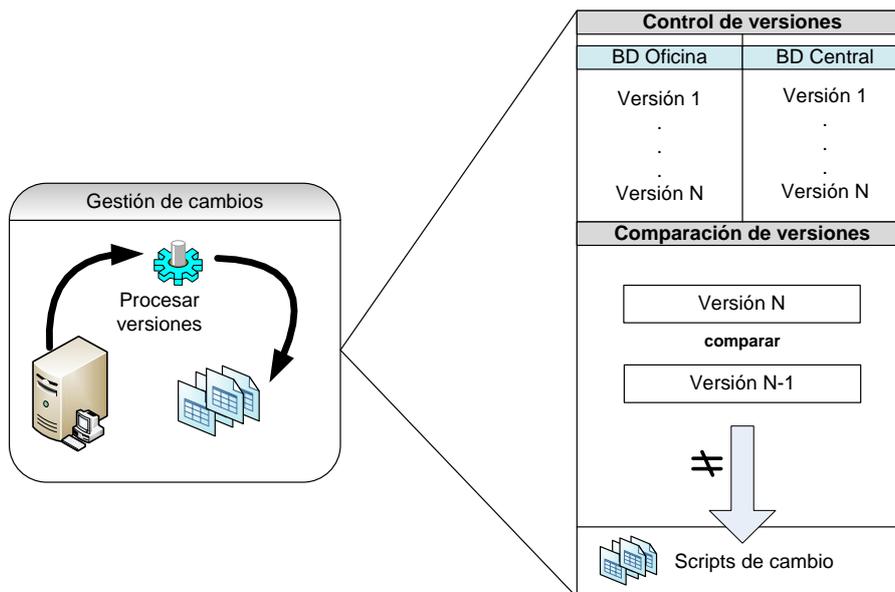


Fig. 4. Flujo para obtener el script de cambio de una versión a otra.

Las versiones son extraídas mediante tareas programadas o a solicitud del responsable de la gestión de la configuración. Cada versión es almacenada y sirve de respaldo para cada etapa del desarrollo.

## Documentación del código dentro de la base de datos.

La documentación de cada elemento del sistema es muy necesaria y sobre todo el código, permitiendo que los miembros del equipo y terceras personas puedan comprender lo que se quiere con cada método. Documentar dentro del mismo código ayuda a recordar y analizar cada método sin necesidad que buscar en la documentación del sistema.

Con el propósito de desarrollar la documentación de una forma estándar para todos los desarrolladores se implementó un sistema de normas y estructuras documentales que permiten documentar cada uno de los elementos programados en la base de datos, tales como, procedimientos almacenados, triggers, paquetes y funciones; y basado en estas normas y estructuras se desarrolló una herramienta que pueda generar una documentación, para cada uno de los objetos de la base de datos.

En el encabezado de cada procedimiento, trigger y función y por cada uno de los elementos en cada paquete, así como uno para la documentación de la funcionalidad del paquete debe aparecer un comentario en forma de XML que describe cada método. La estructura es simple de manera que sea fácil su comprensión. La estructura "Comentario" indica que comienza un bloque de comentarios para el objeto, el bloque "Resumen" es opcional y se utiliza para encabezar cada comentario en la documentación, mientras que el bloque "Descripción" contiene la información completa de cada método. Cada comentario debe estar como

comentario en el código, lo cual no afecta el rendimiento de las operaciones, en el caso de Oracle los comentarios se expresan entre estos símbolos: `/* */` ejemplo: `/*esto es un comentario*/`.

Esta estructura puede ser variable en dependencia del tipo de documentación y la información que se quiera recoger de cada objeto, se puede definir todos los tags necesarios, por ejemplo incluir el autor, para el caso en que el desarrollo sea en equipos.

En el siguiente recuadro aparece un ejemplo de la documentación de un procedimiento almacenado.

```
/*
<Comentario>
  <Resumen>
    Procedimiento para insertar una persona.
  </Resumen>
  <Descripcion>
    Procedimiento que inserta en las tablas TPersona, TDatosOpcionales...
  </Descripcion>
</Comentario>
*/
```

### Herramienta para extraer y generar la documentación

Una herramienta fácil de implementar y que pueda convertir la documentación escrita en el código en un reporte para la gestión documental como complemento inherente a la gestión de la configuración, se describe a continuación.

Utilizando la estructura descrita anteriormente se puede extraer de la base de datos cada uno de estos comentarios y conformar un documento XML de manera que contenga los elementos necesarios para el reporte final.

Una vez estructurado correctamente el XML podemos transformarlos usando XSLT o XSL-FO permitiendo crear reportes y documentos afines al código, creando de forma simple y rápida la documentación de los elementos programados en la base de datos, tarea que resulta a veces muy tediosa y difícil, sobre todo en grandes equipos de desarrollo.

El siguiente diagrama explica el flujo de creación de un reporte para los objetos programados de la base de datos.

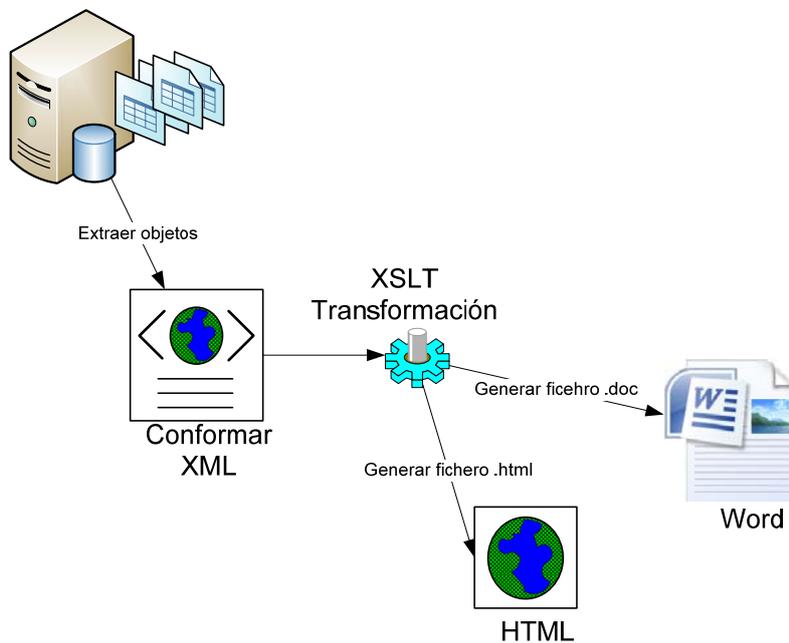


Fig. 5. Flujo para la creación de la documentación de los objetos programados en la base de datos.

El resultado de este proceso es obtener un documento Word, PDF, HTML, etc., que nos permita imprimir la documentación de los objetos programados en la base de datos. Un sitio de gestión de proyecto puede contener cada uno de estos elementos permitiendo el acceso rápido a la información por parte de los desarrolladores y del personal, teniendo la información actualizada con lo que existe en el código real.

## Conclusiones

La gestión de la configuración es un tema complejo pero si se toma con responsabilidad y las tareas se acometen según un plan definido se puede lograr resultados extraordinarios. En primer lugar tener versiones de cada uno de los objetos de la base de datos, y de la base de datos en su conjunto, permite regresar en cada momento a una versión anterior o resguardar todo el flujo de trabajo dentro de la base de datos. En segundo lugar poder extender el modelo de la etapa del desarrollo a todas las etapas siguientes y por consiguiente mantener el control de sobre los cambios en la base de datos. Por último, pero no menos importante, manejar la documentación de los objetos dentro de las bases de datos y generar reportes flexibles y amigables permite obtener de forma rápida la información necesaria de cada objeto programado dentro de la base de datos.

## Referencias Bibliográficas

**Alapati, Sam R y Watson, John. 2005.** *Expert Oracle Database 10g Administration*. s.l. : Apress, 2005. 1590594517.

**Keyes, Jessica. 2004.** *Software Configuration Management*. s.l. : CRC Press, 2004. 0849319765.

**Teorey, Toby J., Lightstone, Sam y Nadeau, Tom. 2005.** *Database Modeling & Design: Logical Design*. s.l. : Academic Press, 2005. 0126853525.

**Wikimedia Foundation, Inc. 2008.** Gestión de la configuración. *Wikipedia*. [En línea] 25 de febrero de 2008. [Citado el: 22 de Abril de 2008.] [http://es.wikipedia.org/wiki/Configuration\\_management](http://es.wikipedia.org/wiki/Configuration_management).

—. 2007. Modelo relacional. *Wikipedia*. [En línea] 8 de Abril de 2007. [Citado el: 21 de Abril de 2007.] [http://es.wikipedia.org/wiki/Modelo\\_relacional](http://es.wikipedia.org/wiki/Modelo_relacional).