

# Cifrado (encriptación) de la base de datos

**BASE100**

BASE 100, S.A.  
[www.base100.com](http://www.base100.com)

## Índice

<b>1. CIFRADO DE LA BASE DE DATOS .....</b>	<b>3</b>
1.1 ACCESO A BASES DE DATOS CIFRADAS .....	3
1.2 VARIABLES DE ENTORNO .....	4
<b>2. CREACIÓN DE UNA BASE DE DATOS CIFRADA .....</b>	<b>5</b>
<b>3. REPARACIÓN Y CHEQUEO DE BASES DE DATOS CIFRADAS .....</b>	<b>6</b>
3.1 COMANDO TREPIDX .....	6
3.2 COMANDO TCHKIDX.....	7

© Copyright BASE 100, S.A. Todos los derechos reservados. Ninguna parte de este documento puede ser reproducida ni transmitida por medio alguno sin permiso previo por escrito del titular del copyright. Todos los productos citados en este documento son marcas registradas o marcas comerciales registradas de sus respectivos propietarios.

[NT\_cifradobd\_v1]

## 1. Cifrado de la base de datos

---

La versión 3.6 upgrade 10 del gestor de base de datos CTSQL incluye una nueva funcionalidad que permite tanto cifrar (*encrypt*) una base de datos existente como crear una base de datos cifrada (*encriptada*).

El comando que permite ejecutar esta nueva funcionalidad de cifrado es **ttcryptdb**.

El directorio de instalación de esta utilidad depende del sistema operativo empleado:

- En Linux/Unix: \$TRANSDIR/bin.
- En Windows: Directorio bin del directorio de instalación de MultiWay.

Sintaxis:

```
./ttcryptdb -source <source DDBB> -dest <dest DDBB>  
[-password <dbatabase password>]
```

Parámetros:

-source	Path completo de la base de datos que se va a cifrar (incluido el directorio dbs). Este parámetro es obligatorio.
-dest	Path completo donde se creará la base de datos cifrada (incluido el directorio dbs). Este parámetro es obligatorio.
-password	Contraseña de protección de acceso a la base de datos. El valor de este parámetro debe ser alfanumérico y de 16 bytes como máximo.  No se trata de una palabra de acceso a la base de datos cifrada, pero cuando se acceda a ésta será necesario indicar su valor en la variable de entorno DBCRYPTPWD. Aunque no es obligatorio, se recomienda utilizarla.

Ejemplos:

```
./ttcryptdb -source /home/ctl/desencrip.dbs  
-dest /home/ctl/encrip.dbs  
./ttcryptdb -source /home/ctl/desencrip.dbs  
-dest /home/ctl/encrip.dbs -password mipasswd
```

El comando **ttcryptdb** sólo cifrará tablas que se encuentren dentro del directorio de la base de datos o en un directorio hijo. Si hubiese tablas de esa base de datos cuyos ficheros DAT e IDX estuviesen en un directorio distinto al de la base de datos, dichas tablas no serían cifradas.

### 1.1 Acceso a bases de datos cifradas

En el motor CTSQL se ha implementado una nueva funcionalidad que permite crear y acceder a bases de datos cifradas.

El motor CTSQL es totalmente compatible con las bases de datos existentes que no están cifradas, además de permitir la creación y el acceso a base de datos no cifradas.

Por su parte, el Monitor es incompatible con la versión 1.0, ya que la estructura de la memoria compartida es diferente.

La versión 3.6 upgrade 10 del gestor CTSQL necesita la versión 1.1 de Monitor.

## 1.2 Variables de entorno

### CRYPTDB

Esta variable de entorno indica al gestor de base de datos si la base de datos a la que se va a conectar está cifrada o no.

Se define en el fichero de configuración del CTSQL (ctsql.ini) si la conexión es cliente-servidor. Si la conexión es local se define en la sección [Environment] del fichero de configuración de Cosmos (cosmos.ini).

Sus posibles valores son: YES y NO.

Si no se define el gestor de base de datos se asume que la base de datos no está cifrada.

Si se asigna el valor YES y la base de datos no está cifrada, CTSQL mostrará el siguiente mensaje de error: "Base de datos no encontrada o sin permiso del sistema", y lo mismo sucederá si el valor de la variable es NO con la base de datos cifrada.

### DBCRYPTPWD

Esta variable de entorno indica al gestor de la base de datos cuál es la contraseña de protección de acceso a la base de datos cifrada a la que se va a acceder.

Si la base de datos ha sido creada con el comando **ttcryptdb**, el valor de la variable de entorno DBCRYPTPWD debe coincidir con el indicado en el parámetro "-password" de dicho comando. Esta variable de entorno no se deberá definir si no se ha indicado este parámetro en el proceso de creación de la base de datos.

Se debe definir en el fichero de configuración del gestor de base de datos (ctsql.ini), aunque también podrá definirse en el fichero de configuración del cliente dentro de la sección donde se define la conexión.

Si no se define esta variable de entorno, CTSQL intentará acceder a la base de datos cifrada sin palabra de acceso (en caso de que se hubiese definido la variable de entorno CRYPTDB=YES).

## **2. Creación de una base de datos cifrada**

---

La creación de una base de datos cifrada se podrá realizar desde las siguientes utilidades:

- Sql-Interactivo, tanto en versiones para Unix/Linux como Windows.
- Repositorio de Cosmos.
- Programa desarrollado en Cosmos.
- MultiBase para Windows sólo podrá crear bases de datos cifradas en conexiones cliente-servidor.

Al crear una base de datos cifrada, es recomendable definir la variable de entorno DBCRYPTPWD en el servidor (en el fichero de configuración ctsql.ini). Por razones de seguridad, no es recomendable definirla en el cliente.

En caso de que se indique en el cliente, lo más seguro es definirla desde dentro de la aplicación.

### 3. Reparación y chequeo de bases de datos cifradas

---

Los comandos encargados de los procesos de reparación y chequeo de bases de datos cifradas son respectivamente **trepidx** y **tchkidx**.

En la versión para Windows se han añadido dos parámetros nuevos: **-cryptdb** y **-dbcryptpwd**.

#### 3.1 Comando trepidx

Sintaxis:

```
./trepidx -v <database> <table>|<-all> [-ndl|-dl] [-y] [-cryptdb]
[-dbcryptpwd <cryptpassword>]
```

Parámetros:

-v	Muestra la versión del comando y su “upgrade”.
database	Nombre de la base de datos, sin extensión “.dbs”, a la que pertenece(n) la(s) tabla(s) a reparar. Esta base de datos tiene que encontrarse en el directorio en curso o bien en cualquiera de los directorios indicados en la variable de entorno DBPATH.  En el caso de no tener export, la variable de entorno DBPATH deberá indicar la ruta completa de la base de datos con la extensión “.dbs”.
table	Nombre de la tabla de la que se desea comprobar la integridad de sus índices con respecto a los datos. Esta tabla debe pertenecer a la base de datos indicada en el parámetro anterior.
-all	Indica que se repararán todas las tablas de la base de datos especificada en el parámetro anterior.
-ndl	Indica que el/los índice/s de la/s tabla/s no tiene/n límite de duplicados.
-dl	Indica que el/los índice/s de la/s tabla/s sí tiene/n límite de duplicados.
-y	No pregunta nada al operador.
-cryptdb	Indica si la base de datos que se va a reparar está cifrada o no.
-dbcryptpwd <cryptpassword>	Contraseña de cifrado de la base de datos. Si se incluye este parámetro no es necesario indicar “-cryptdb”, ya que asume que la base de datos está cifrada.

En la versión para Unix/Linux, la forma de ejecución no ha variado, simplemente se tendrán que añadir a la exportación de variables de entorno que se hacía en versiones anteriores las dos nuevas que se han creado para poder realizar el proceso de cifrado: **CRYPTDB** y **DBCRYPTPWD**.

Sintaxis:

```
./trepidx <database> <table>|<-all> [-ndl|-dl] [-y]
```

Ejemplo:

```
export CRYPTDB=YES
export DBCRYPTPWD=mipasswd
[ctl@rhel150 bin]$ ./trepidx/home/bbdd/encrip.dbs clientes
```

En este ejemplo, el comando **trepidx** accede a la base de datos “encrip.dbs”, que según las variables de entorno CRYPTDB y DBCRYPTPWD es una base de datos cifrada y con contraseña “mipasswd”, y repara la tabla “clientes”.

## 3.2 Comando tchkidx

Sintaxis:

```
tchkidx -v <database> [<table>|-all] [-ndl|-dl] [-y] [-cryptdb]
[-dbcryptpwd <cryptpassword>]
```

Parámetros:

-v	Muestra la versión del comando y su “upgrade”.
database	Nombre de la base de datos, sin extensión “.dbs”, a la que pertenece(n) la(s) tabla(s) a chequear. Esta base de datos tiene que encontrarse en el directorio en curso o bien en cualquiera de los directorios indicados en la variable de entorno DBPATH. En el caso de no tener export, la variable de entorno DBPATH deberá indicar la ruta completa de la base de datos con la extensión “.dbs”.
table	Nombre de la tabla de la que se desea comprobar la integridad de sus índices con respecto a los datos. Esta tabla debe pertenecer a la base de datos indicada en el parámetro anterior.
-all	Indica que se chequearán todas las tablas de la base de datos especificada en el parámetro anterior.
-y	No pregunta nada al usuario.
-ndl	Indica que el/los índice/s de la/s tabla/s no tiene/n límite de duplicados.
-dl	Indica que el/los índice/s de la/s tabla/s sí tiene/n límite de duplicados.
-cryptdb	Indica si la base de datos que se va a chequear está cifrada o no.
-dbcryptpwd <cryptpassword>	Indica la contraseña de cifrado de la base de datos. Si se incluye este parámetro no es necesario indicar “-cryptdb”, ya que asume que la base de datos está cifrada.

En la versión para Unix/Linux, en lugar de indicar los dos últimos parámetros habrá que exportar las variables de entorno CRYPTDB y DBCRYPTPWD.

Sintaxis:

```
tchkidx -v <database> [<table>|-all] [-ndl|-dl]
```

Los comandos **trepidx** y **tchkidx** son totalmente compatibles con bases de datos existentes que no estén cifradas.