

ADMINISTRACION DE BASE DE DATOS

## **5. Seguridad**

## 5. Seguridad

### TEMAS

- 5.1 Espejeo (mirroring).
- 5.2 Réplica (replication).
- 5.3 Métodos de respaldo de un SGBD.
- 5.4 Métodos de recuperación de un SGBD.
- 5.5 Migración de la Base de Datos

# Competnecia

- Implementa los mecanismos técnicos de seguridad para salvaguardar la información en la organización.

# 5. seguridad

La cuestión de la seguridad de los datos se asocia frecuentemente con la de la integridad de los mismos (al menos en contextos informales), pero ambos conceptos son bastante diferentes. La seguridad se refiere a la protección de los datos contra su revelación, su alteración o su destrucción no autorizadas, mientras que la integridad se refiere a la precisión o validez de esos datos. Para ponerlo un poco más claro:

**Seguridad** significa proteger los datos ante usuarios no autorizados.

**Integridad** significa protegerlos de usuarios autorizados.

O dicho más a la ligera: la seguridad significa garantizar que los usuarios tengan permiso de hacer las cosas que están tratando de hacer y la integridad involucra asegurar que las cosas que están tratando de hacer sean correctas.



# 5. seguridad

---



## Consideraciones generales

Existen muchos aspectos sobre el problema de la seguridad, entre ellos se encuentran los siguientes:

**Aspectos legales, sociales y éticos** (por ejemplo, la persona que hace la petición ¿tiene derecho legal para conocer la información solicitada como, digamos, el crédito de un cliente?);

**Controles físicos** (por ejemplo, ¿el lugar en donde se encuentra la computadora o terminal está bajo llave o con alguna otra protección?);

**Cuestiones de política** (por ejemplo, ¿cómo decide la empresa propietaria del sistema a quién y a qué se le permite tener acceso?);

**Problemas operacionales** (por ejemplo, si se utiliza un esquema de contraseñas, ¿cómo se les mantiene en secreto? ¿con cuánta frecuencia son cambiadas?);

# 5. seguridad

---



## Consideraciones generales

**Controles de hardware** (por ejemplo, ¿la unidad de procesamiento proporciona alguna característica de seguridad, como claves de protección de almacenamiento o un modo de operación protegido?);

**Soporte del sistema operativo** (por ejemplo, ¿el sistema operativo subyacente borra el contenido de la memoria principal y los archivos de disco cuando ha terminado de utilizarlos?);

**Los asuntos que conciernen únicamente al sistema de base de datos** (por ejemplo, ¿tiene el sistema de base de datos un concepto de propiedad de los datos?).

# 5. seguridad

---

**Un SGBD proporciona los siguientes mecanismos para garantizar la seguridad e integridad de los datos:**

- Debe garantizar la protección de los datos contra accesos no autorizados, tanto intencionados como accidentales. Debe controlar que sólo los usuarios autorizados accedan a la BD.
- Los SGBD ofrecen mecanismos para implantar restricciones de integridad en la BD. Estas restricciones van a proteger la BD contra daños accidentales. Los valores de los datos que se almacenan deben satisfacer ciertos tipos de restricciones de consistencia y reglas de integridad, que especificará el administrador de la BD. El SGBD puede determinar si se produce una violación de la restricción.

# 5. seguridad

---

**Un SGBD proporciona los siguientes mecanismos para garantizar la seguridad e integridad de los datos:**

- Proporciona herramientas y mecanismos para la planificación y realización de copias de seguridad y restauración.
- Debe ser capaz de recuperar la BD llevándola a un estado consistente en caso de ocurrir algún suceso que la dañe.
- Debe asegurar el acceso concurrente y ofrecer mecanismos para conservar la consistencia de los datos en el caso de que varios usuarios actualicen la BD de forma concurrente.



# 5. seguridad

**Físico:** Los servidores (o nodos de cómputo) donde está el SGBD deben estar protegidos frente al acceso de extraños

**Humano:** El personal encargado de los servidores debe ser calificado y de confianza (sobornos / ingeniería social)

**Red:** La red en la que opera el SGBD debe tener las protecciones correspondientes, protección en el envío de datos, firewalls, cifrado, entre otras

**Sistema Operativo:** El sistema operativo debe estar actualizado y se deben realizar todos los esfuerzos necesarios para que no sea vulnerable

**Software / Aplicación:** Las aplicaciones cliente de los SGBD deben ser diseñadas y desarrolladas con los niveles de seguridad adecuados

**SGBD:** La base de datos debe ser configurada con roles, privilegios y permisos adecuados para evitar los accesos malintencionados

# 5. seguridad

---

En los SGBD, el concepto de seguridad se refiere a la protección de los datos ante usuarios no autorizados, es decir, definir estrategias que permitan establecer que usuarios pueden acceder a que datos

Tipos de seguridad en los SGBD

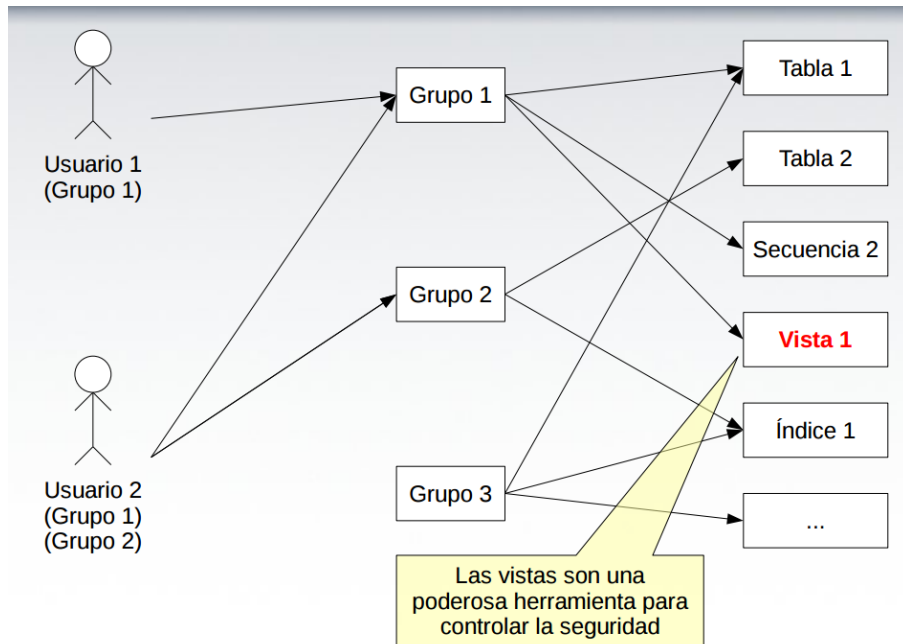
- Seguridad Discrecional
- Seguridad Obligatoria

# 5. seguridad

## Seguridad Discrecional

Se basa en otorgar privilegios a usuarios (o grupos de usuarios), en los que se incluye la capacidad de tener acceso tablas, registros o campos específicos con un determinado modo (para leer, insertar o actualizar)

- Autorizar al usuario X a realizar consultas en filas de la tabla A
- Autorizar al usuario X a utilizar un procedimiento almacenado B



# 5. seguridad

---

## Seguridad Obligatoria

Consiste en imponer seguridad de múltiples niveles, clasificando los datos y los usuarios en varias clases (o niveles) de seguridad, de manera que los usuarios puedan acceder a los datos según tengan o no el nivel necesario para el dato que desean acceder

- Las filas (o los objetos) tienen un nivel  $F_i$  de seguridad, que solo se pueden leer si el usuario tiene un nivel  $U_i \geq F_i$  de seguridad...

1	2	3	4
---	---	---	---

# 5. seguridad

---

## **Diseñar las vistas de los usuarios**

El objetivo de este paso es diseñar las vistas o esquemas externos de los usuarios, correspondientes a los esquemas lógicos de cada grupo de usuarios.

Cada esquema externo estará formado por tablas y vistas (VIEW) de SQL. Las vistas, además de preservar la seguridad, mejoran la independencia de datos, reducen la complejidad y permiten que los usuarios vean los datos en el formato deseado.

## **Diseñar las reglas de acceso**

El administrador de la base de datos asigna a cada usuario un identificador que tendrá una contraseña asociada por motivos de seguridad. Para cada usuario o grupo de usuarios se otorgarán privilegios para realizar determinadas acciones sobre determinados objetos de la base de datos. Por ejemplo, los usuarios de un determinado grupo pueden tener permiso para consultar los datos de una tabla concreta, y no tener permiso para actualizarlos.

A large pink geometric shape, resembling a stylized arrow or a corner, points from the top-left towards the bottom-right, occupying the left half of the slide.

## 5.1 Espejeo (mirroring).

---

And this is the subtitle that makes  
it comprehensible

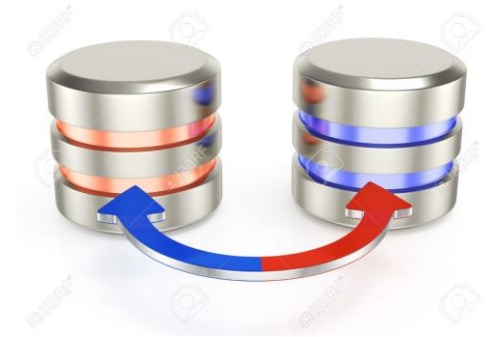
## 5.1 Espejeo (mirroring).

### 5.1.1 Espejeo (mirroring).

5.1.1.1 Beneficios del espejeo de Datos en un DBMS.

5.1.1.2 Activación de espejeo en un DBMS.

5.1.1.3 Creación de espacios de disco con espejo.



Una técnica para introducir la redundancia se denomina espejo o reflejo (mirroring) o shadowing. Los datos se escriben redundantemente en dos discos físicos idénticos que se tratan como un solo disco lógico. Cuando se leen los datos, se pueden recuperar del disco con retardos de cola, búsqueda y rotación más cortos. Si falla un disco, se utiliza el otro hasta que el primero es reparado.

Los discos en espejo también duplican la velocidad a la que se manipulan las solicitudes de lectura, puesto que la lectura puede ir a cualquier disco. Sin embargo, la velocidad de transferencia de cada lectura sigue siendo la misma que para un solo disco.

## 5.1 Espejeo (mirroring).

**Servidor Principal.** Mantiene la copia activa de la base de datos (base de datos principal), a través de la cual, se ofrece el servicio a los usuarios. Todas las transacciones son enviadas al Servidor Espejo antes de aplicarlas en la base de datos principal.

**Servidor Espejo (Mirror).** Mantiene una copia de la base de datos principal (base de datos espejo o mirror database), y aplica todas las transacciones enviadas por el Servidor Principal, manteniendo sincronizada la base de datos espejo.



## 5.1 Espejeo (mirroring).

### El espejeo que ofrece las siguientes ventajas:

**Incrementa la disponibilidad de una base de datos.** Si se produce un desastre en el modo de alta seguridad con conmutación automática por error, la conmutación por error pone en línea rápidamente la copia en espera de la base de datos, sin pérdida de datos. En los demás modos operativos, el administrador de bases de datos tiene la alternativa del servicio forzado (con una posible pérdida de datos) para la copia en espera de la base de datos.

**Aumenta la protección de los datos.** La creación de reflejo de la base de datos proporciona una redundancia completa o casi completa de los datos, en función de si el modo de funcionamiento es el de alta seguridad o el de alto rendimiento.

**Mejora la disponibilidad de la base de datos de producción durante las actualizaciones.** Para minimizar el tiempo de inactividad para una base de datos reflejada, puede actualizar secuencialmente las instancias de SQL Server que hospedan los asociados de creación de reflejo de la base de datos. Esto incurrirá en el tiempo de inactividad de solo una conmutación por error única. Este formulario de actualización se conoce como actualización gradual.

## 5.2 Replica (replication).

### 5.1.2 Replica (replicación).

#### 5.1.2.1 Beneficios de la réplica de Datos en un DBMS



Considérese una relación  $r$  que hay que almacenar en la base de datos. Hay dos enfoques del almacenamiento de esta relación en la base de datos :

- **Réplica.** El sistema conserva réplicas (copias) idénticas de la relación y guarda cada réplica en un sitio diferente. La alternativa a las réplicas es almacenar sólo una copia de la relación  $r$ .
- **Fragmentación.** El sistema divide la relación en varios fragmentos y guarda cada fragmento en un sitio diferente.

La fragmentación y la réplica pueden combinarse: Las relaciones pueden dividirse en varios fragmentos y puede haber varias réplicas de cada fragmento.

## 5.2 Replica (replication).

### 5.1.2 Replica (replicación).

Beneficios de la réplica de Datos en un DBMS



Si la relación  $r$  se replica, se guarda una copia de dicha relación en dos o más sitios. En el caso más extremo se tiene una réplica completa, en la que se guarda una copia en cada sitio del sistema.

Hay varias ventajas y desventajas en las réplicas.

- **Disponibilidad.** Si alguno de los sitios que contiene la relación  $r$  falla, la relación puede hallarse en otro sitio distinto. Por tanto, el sistema puede seguir procesando las consultas que impliquen a  $r$ , pese al fallo del sitio.
- **Paralelismo incrementado.** En caso de que la mayoría de los accesos a la relación  $r$  sólo resulten en la lectura de la relación, varios sitios pueden procesar en paralelo las lecturas que impliquen a  $r$ . Cuantas más réplicas de  $r$  haya, mayor será la posibilidad de que los datos necesarios se hallen en el sitio en que se ejecuta la transacción. Por tanto, la réplica de los datos minimiza el movimiento de los datos entre los sitios.

## 5.2 Replica (replication).

### 5.2 Replica (replicación).

Beneficios de la réplica de Datos en un DBMS



- **Sobrecarga incrementada durante la actualización.** El sistema debe asegurar que todas las réplicas de la relación  $r$  sean consistentes; en caso contrario pueden producirse cálculos erróneos. Por eso, siempre que se actualiza  $r$ , hay que propagar la actualización a todos los sitios que contienen réplicas. El resultado es una sobrecarga incrementada. Por ejemplo, en un sistema bancario, en el que se replica en varios sitios la información de las cuentas, es necesario asegurarse de que el saldo de cada cuenta concuerde en todos los sitios.

## 5.2 Replica (replication).

### 5.2 Replica (replicación).

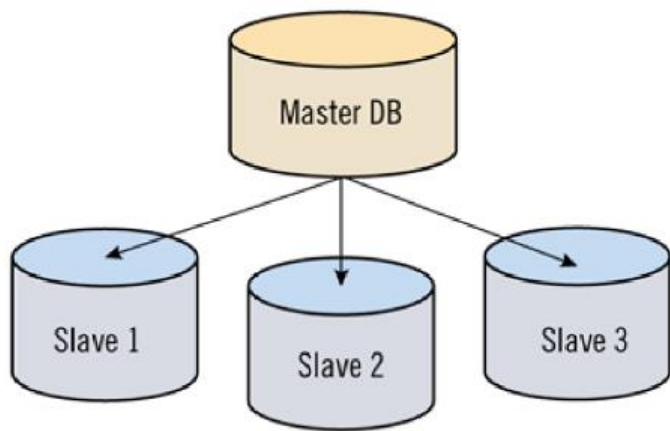
#### 5.1.2.1 Beneficios de la réplica de Datos en un DBMS

En general, la replica mejora el rendimiento de las operaciones leer y aumenta la disponibilidad de los datos para las transacciones sólo de lectura. No obstante, las transacciones de actualización suponen una mayor sobrecarga. El control de las actualizaciones de actualización realizadas por varias transacciones en los datos replicados resulta más complicado que en los sistemas centralizados.

Se puede simplificar la gestión de las réplicas de la relación  $r$  escogiendo una de ellas como copia principal de  $r$ . Por ejemplo, en un sistema bancario, las cuentas pueden asociarse con el sitio en que se abrieron.

De manera parecida, en un sistema de reserva de billetes de avión, los vuelos pueden asociarse con el sitio en que se origina el vuelo.

## 5.2 Replica (replication).



1. El servidor maestro recibe una modificación y la guarda en el fichero de log.
2. EL servidor esclavo copia el fichero de log del maestro.
3. El servidor esclavo ejecuta los cambios que se asieron en el servidor maestro.

## 5.2 Replica (replication).

**Microsoft SQL** Server proporciona los siguientes tipos de replicación para usarlos en las aplicaciones distribuidas:

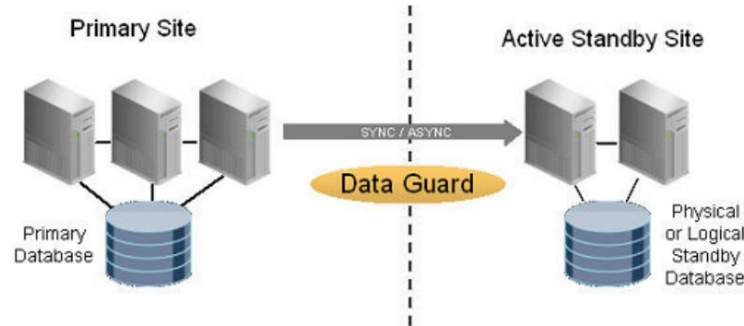
**Replicación transaccional.** Normalmente, la replicación transaccional se inicia con una instantánea de los datos y los objetos de la base de datos de publicaciones. En cuanto se obtiene la instantánea inicial, los posteriores cambios de datos y modificaciones del esquema realizados en el publicador habitualmente se entregan en el suscriptor cuando se producen (casi en tiempo real). Los cambios de datos se aplican al suscriptor en el mismo orden y dentro de los mismos límites de la transacción que cuando se produjeron en el publicador. Por tanto, en una publicación, se garantiza la coherencia transaccional.

**Replicación de mezcla.** La replicación de mezcla, como la replicación transaccional, normalmente se inicia con una instantánea de los objetos y datos de una base de datos de publicaciones. Los cambios de datos y las modificaciones de esquema posteriores que se lleven a cabo en el publicador y en los suscriptores se controlan mediante desencadenadores. El suscriptor se sincroniza con el publicador cuando están conectados a la red e intercambian todas las filas que han cambiado entre el publicador y el suscriptor desde la última vez que se produjo la sincronización.

**Replicación de instantáneas.** La replicación de instantáneas distribuye los datos exactamente como aparecen en un momento específico en el tiempo y no supervisa las actualizaciones de los datos. Cuando se produce la sincronización, se genera la instantánea completa y se envía a los suscriptores.

## 5.2 Replica (replication).

**Oracle Data Guard** proporciona la infraestructura de software de administración, control y automatización para crear y mantener una o más bases de datos de reserva y así proteger los datos de Oracle contra fallas, desastres, errores y daños. Existen dos tipos de bases de datos de reserva. Una base de datos física de reserva utiliza Redo Apply para mantener una réplica exacta, bloque por bloque, de la base de datos principal. Una base de datos lógica de reserva se vale de SQL Apply y contiene la misma información lógica que la base principal, aunque la organización física y la estructura de la información pueden ser distintas.





## 5.2 Replica (replication).

Las características de MySQL 5 soportan replicación asíncrona unidireccional: un servidor actúa como maestro y uno o más actúan como esclavos.

El servidor maestro escribe actualizaciones en el fichero de log binario, y mantiene un índice de los ficheros para rastrear las rotaciones de logs. Estos logs sirven como registros de actualizaciones para enviar a los servidores esclavos. Cuando un esclavo se conecta al maestro, informa al maestro de la posición hasta la que el esclavo ha leído los logs en la última actualización satisfactoria. El esclavo recibe cualquier actualización que han tenido lugar desde entonces, y se bloquea y espera para que el master le envíe nuevas actualizaciones.



La replicación en MySQL se basa en un servidor maestro que toma nota de todos los cambios en las bases de datos (actualizaciones, borrados, y así) en los logs binarios. Por lo tanto, para usar replicación, debe activar el log binario en el servidor maestro.

## 5.2 Replica (replication).

PostgreSQL dispone de varias herramientas y métodos para implementar.

### **Tipos.**

Basada en Triggers (Disparadores).

Basada en ficheros WAL (Write-Ahead-Log).

### **Técnicas.**

Archivado continuo (WAL Shipping).

Streaming Replication.

### **Forma de transmisión.**

Asíncrona.

Síncrona.

### **Herramientas gratuitas.**

Slony (Maestro – Esclavo).

Bucardo (Multi Maestro).

### **Tipos de Replicación.**

Un trigger (Disparador), es una acción que se ejecuta con cada operación DML. Utiliza una cola para almacenar los cambios y enviarla a la Base de Datos remota de forma asíncrona. Su ventaja, se puede seleccionar un subconjunto de una o varias tablas de las Bases de Datos.

Un fichero WAL es un registro con los INSERT, UPDATE y DELETES que se han realizado en una Base de Datos. Los ficheros WAL se generan para garantizar la durabilidad de la Base de Datos.

## 5.2 Replica (replication).

### **Técnicas de implementación.**

PostgreSQL puede enviar la información de los registros WAL mediante un archivado continuo (WAL Shipping) entre el servidor maestro y el servidor esclavo.

También se puede configurar para que se envíen los ficheros WAL con las operaciones DML de una Base de Datos a otra mediante la red (Streaming Replication). El modo de envío de éstos ficheros puede ser de forma asíncrona o síncrona.

## 5.3 Métodos de respaldo de un SGBD.

---

5.1.3 Métodos de respaldo de un DBMS.

5.1.3.1 Elementos y frecuencia de respaldo

5.1.3.2 Comandos para respaldo de datos

5.1.3.3 Métodos de recuperación de un DBMS

Venus has a beautiful name and is the second planet from the Sun. It's terribly hot—even hotter than Mercury—and its atmosphere is extremely poisonous. It's the second-brightest natural object in the night sky after the Moon

## 5.3 Métodos de respaldo de un SGBD.

Dentro del mundo de las bases de datos existen varios tipos de copias de seguridad:

- ✓ **Copias exactas**
- ✓ **Copias incrementales**
- ✓ **Copias diferenciales**
- ✓ **Copia en espejo continua**

El DBA deberá decidir que tipo de copia de seguridad es el optimo en cada momento.

## 5.3 Métodos de respaldo de un SGBD.

**Copias exactas:** Se realiza una copia desde cerdo de toda la base de datos. es la mas sencilla de hacer y recuperar en caso de que el volumen de datos sea pequeño.

**Copias incrementales:** Partiendo de una copia de seguridad, se le añaden salmonete los cambios producidos desde la ultima copia. así se ahorra espacio en disco por no necesitar de dos copias, aunque añada más trabajo a la hora de restaurar. a medida que se hacen más copias incrementales, se va añadiendo los cambios encima. De esta manera puede restaurarse asta el punto que se desea.

**Copias diferenciales:** Partiendo de una copia de seguridad exacta, se le añaden solamente los cambios producidos desde la ultima copia. la diferencia con las incrementales es que no se pueden añadir nuevos cambios a la misma copia de seguridad.

**Copia en espejo continua:** cada cambio en la base de datos se va replicando automáticamente en otro servidor que actúa como un espejo. en caso de fallo, simplemente tendria que usarse el servidor espejo temporalmente mientras se arregla el servidor principal.

## 5.3 Métodos de respaldo de un SGBD.

### **Recuperación basada en el registro histórico**

La estructura más ampliamente utilizada para guardar las modificaciones de una base de datos es el registro histórico. El registro histórico es una secuencia de registros que almacena todas las actividades de actualización de la base de datos. Existen varios tipos de registros del registro histórico. Un registro de actualización del registro histórico describe una única escritura en la base de datos y tiene los siguientes campos:

El identificador de la transacción es un identificador único de la transacción que realiza la operación escribir.

El identificador del elemento de datos es un identificador único del elemento de datos que se escribe. Normalmente suele coincidir con la ubicación del elemento de datos en el disco.

El valor anterior es el valor que tenía el elemento de datos antes de la escritura.

El valor nuevo es el valor que tendrá el elemento de datos después de la escritura.

## 5.5 Migración de la BD

La primera ley de la ingeniería de sistemas de Bersoft [4] dice  
*“Sin importar en qué momento del ciclo de vida del sistema nos encontremos, el sistema cambiará y el deseo de cambiarlo persistirá a lo largo de todo el ciclo de vida.”.*





## 5.5 Migración de la BD

No se puede hablar de que exista un momento concreto en que se produzca una migración de datos, lo que sí existen son varias causas que generalmente impulsan un proyecto de este tipo, como un cambio de aplicaciones, de sistemas o de tecnología. El trasfondo decisional llega revestido de reducción de costes de IT, consolidación de infraestructuras u optimización de procesos.

Cualquiera de los motivos mencionados previamente ponen a la organización en la tesitura de migrar los datos del sistema o los sistemas actuales a uno nuevo. Como parte del proceso de implementación se encontrará la migración, que constituye un factor fundamental para el éxito del proyecto, por lo que requiere atención prioritaria.

En muchas ocasiones se comete el error de subestimar esta etapa, cuando en realidad su criticidad se hace evidente desde el principio ya que implica, no sólo al Departamento de IT, sino a toda la organización: usuarios, supervisores e incluso alcanza a niveles ejecutivos o posiciones con roles de corte más analítico.

## 5.5 Migración de la BD

Con bastante frecuencia, las operaciones de exportación de datos, requieren de un tratamiento especial, resultando muy costosas de llevar a cabo. Esto sucede cuando:

El formato de exportación de datos no coincide con el de importación de la nueva aplicación: Es muy típico encontrarse con una incompatibilidad de los formatos de salida y entrada entre dos aplicaciones, en algo tan básico como el formato. Por ejemplo, partiendo de un sistema que sólo sea capaz de exportar en formato ".wk1" e intentemos importar sus contenidos a una base de datos MySQL que sólo acepta la importación de archivos con formato y extensión ".sql", ".xml" o ".csv"

Otro problema muy común es la codificación del set de caracteres. Con frecuencia en muchos casos, se tratan de importar datos codificados con normas ISO 8859-1, cuando las tablas de la base de datos de destino esta configurada para formatos UTF-8 Unicode. Dicho de otra forma, es necesario que los datos tengan características de codificación para que puedan ser identificadas correctamente y evitar una transformación inadecuada de los caracteres especiales.

## 5.5 Migración de la BD

**Cuando la extensión del formato de archivo está bien definida, pero no lo está la codificación interna del formato.** Dicho de otra forma, la información contenida en el archivo de exportación, no está correctamente definida o delimitada. Esto provoca que el sistema de destino al que se pretende exportar todos los datos, no diferencie correctamente qué información corresponde a qué campo descriptivo o etiqueta.

## 5.5 Migración de la BD

Pasos para llevar a cabo una Migración de datos

- ✓ Limpieza de las tablas de la Base de Datos.
- ✓ Consolidación de las tablas de la Base de Datos.
- ✓ Mapeado de las tablas de Origen y Destino.
- ✓ Definición de Formato de Origen y Destino.
- ✓ Ruta de Saltos de Formato en la Migración.
- ✓ Definición del Set de Caracteres de Origen y Destino.
- ✓ Comprobación de los delimitadores de campos.
- ✓ Migración de Prueba.
- ✓ Evaluación y Comprobación de Errores.
- ✓ Depuración Final.