



# Ingeniería de Pandora FMS



<https://pandorafms.com/manual!/current/>

ent link:

[https://pandorafms.com/manual!/current/es/documentation/pandorafms/complex\\_environments\\_and\\_optimization/09\\_pandorafms\\_engineering](https://pandorafms.com/manual!/current/es/documentation/pandorafms/complex_environments_and_optimization/09_pandorafms_engineering)

06/10 14:36



# Ingeniería de Pandora FMS

## Diseño de la base de datos de Pandora FMS

Las **primeras versiones** de Pandora FMS, desde la 0.83 hasta la 1.1, estaban basadas en una idea muy sencilla: un dato, una inserción de la base de datos. Esto permitía al software realizar búsquedas fáciles, inserciones y otras operaciones rápidamente.

Aparte de todas las ventajas que suponía este desarrollo, había un inconveniente: la *escalabilidad* (crecimiento rápido sin afectar o afectar poco las operaciones y rutinas de trabajo). Este sistema tenía un límite definido en máximo número de módulos que pueda soportar, y con cierta cantidad de datos (más de 5 millones de elementos), el rendimiento se veía afectado.

Las soluciones basadas en *clúster* MySQL, por otro lado, son difíciles: aunque permiten gestionar una mayor carga, añaden algunos problemas y dificultades extra, y tampoco ofrecen una solución a largo plazo al problema de rendimiento con gran cantidad de datos.

Actualmente, Pandora FMS implementa una compactación de datos en *tiempo real* para cada inserción, además de realizar una compresión de datos basada en interpolación. Por otro lado, la **tarea de mantenimiento** permite borrar automáticamente los datos que sobrepasen cierta antigüedad.

El sistema de procesamiento de Pandora FMS almacena solamente datos «nuevos»: si entra un valor duplicado en el sistema, no se almacenará en la base de datos. Esto es muy útil para mantener la base de datos reducida, y funciona para todos los tipos de módulo de Pandora FMS (numérico, incremental, *booleano* y cadena de texto).

Estas modificaciones comportan grandes cambios a la hora de leer e interpretar los datos. En las últimas versiones de Pandora FMS, se ha rediseñado desde cero el motor gráfico para poder representar los datos rápidamente con el nuevo modelo de almacenamiento de datos.

Los mecanismos de compactación tienen, además, **ciertas implicaciones** a la hora de leer e interpretar los datos gráficamente: Actualmente se cuenta con menú de configuración gráfico el cual permite agregar percentiles, *datos en tiempo real*, cuándo se produjeron eventos y/o alertas, además de otras opciones.

Adicionalmente, Pandora FMS permite la disgregación total de componentes, de manera que se puede equilibrar la carga de procesado de ficheros de datos y ejecución de módulos de red en diferentes servidores.

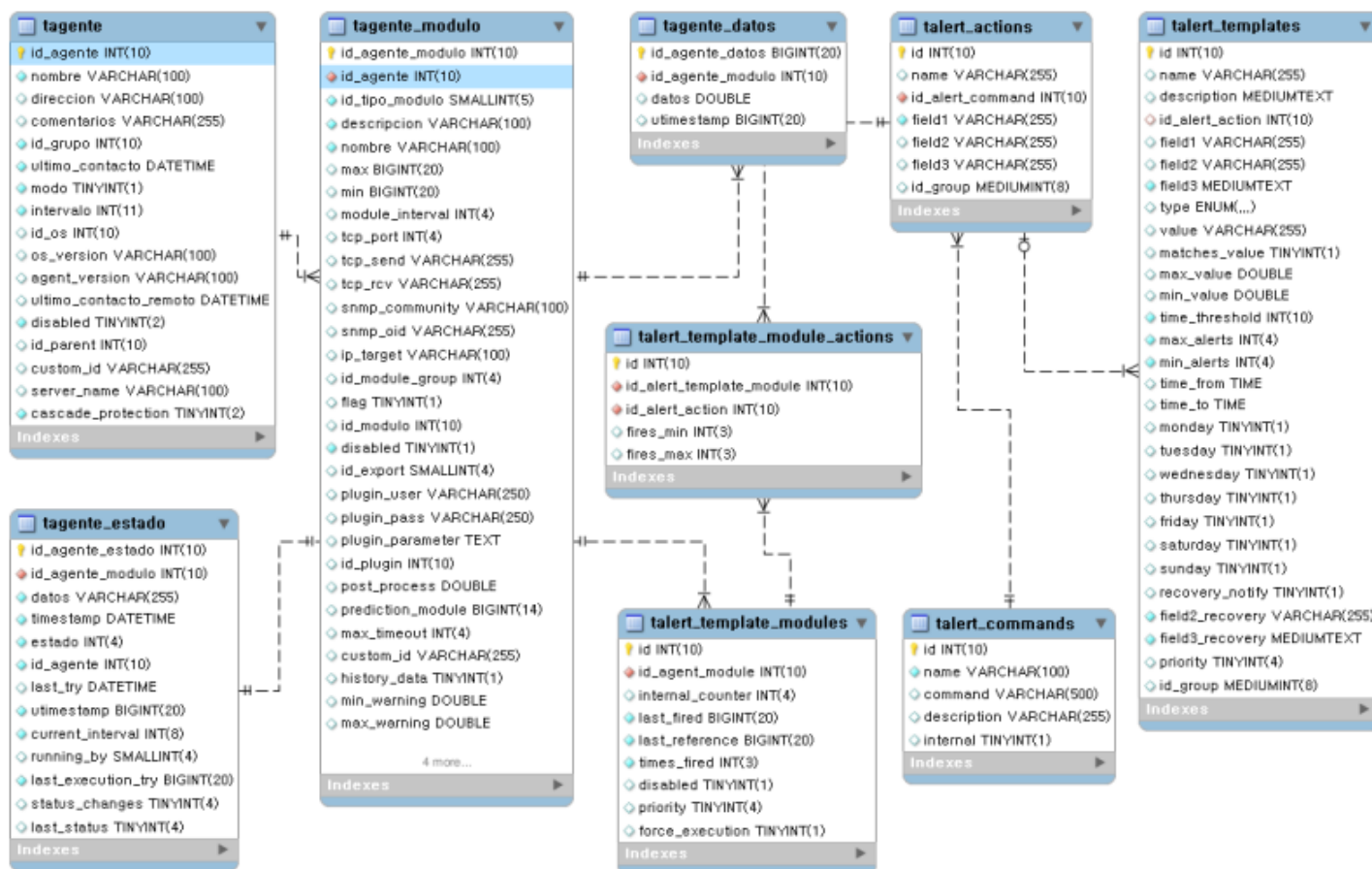
## Otros aspectos técnicos de BBDD

A lo largo de las actualizaciones del software, se han ido implementando mejoras en el modelo relacional de la base de datos de Pandora FMS. Uno de los cambios introducidos ha sido la indexación de información en base a *diferentes tipos de módulos*. De esta forma, Pandora FMS puede acceder mucho más rápido a la información, ya que esta se reparte en diferentes tablas.

Es posible realizar un particionado de las tablas (por marcas de tiempo) para mejorar aún más el rendimiento del acceso a los históricos de datos.

Además, factores como la representación numérica de las marcas de tiempo (`_timestamp_` formato UNIX), acelera las búsquedas de rangos de fecha, comparaciones de las mismas, etc. Este trabajo ha permitido una mejora considerable en los tiempos de búsqueda y en las inserciones.

## Tablas principales de la base de datos



## Compresión de datos en tiempo real

Para evitar sobrecargar la base de datos, el servidor realiza una sencilla compresión en tiempo de inserción. Un dato no se guarda en la base de datos a menos que sea distinto del anterior o haya una diferencia de más de 24 horas entre ambos.

Por ejemplo, suponiendo un intervalo aproximado de 1 hora, la secuencia 0,1,0,0,0,0,0,0,1,1,0,0 se almacena en la base de datos como 0,1,0,1,0. No se guardará otro 0 consecutivo a menos que hayan pasado 24 horas.

La compresión afecta en gran medida a los algoritmos de procesamiento de datos, tanto a las métricas como a las gráficas, *y es importante tener en cuenta que hay que rellenar los "huecos" provocados por la compresión.*

Teniendo en cuenta todo lo anterior, para realizar cálculos con los datos de un módulo dado un intervalo y una fecha inicial, hay que seguir los siguientes pasos:

- Buscar el dato anterior, fuera del intervalo y fecha dados. Si existiera, hay que traerlo al inicio del intervalo. Si no existiera anteriormente no habría datos.
- Buscar el siguiente dato, fuera del intervalo y fecha dados hasta un máximo igual al intervalo del módulo. Si existiera hay que traerlo al final del intervalo. Si no existiera, habría que prolongar el último valor disponible hasta el final del intervalo.
- Se recorren todos los datos, teniendo en cuenta que un dato es válido hasta que se reciba un dato distinto.

## Compactación de datos

Pandora FMS ha incluido un sistema para *compactar* la información de la base de datos. Este sistema está orientado a escenarios de tamaño pequeño y mediano ( de 250 a 500 agentes y con menos de 100.000 módulos) que desean tener un histórico de información extenso pero *perdiendo* detalles.

El mantenimiento de la base de datos de Pandora FMS que se ejecuta cada hora, y entre otras tareas de limpieza permite realizar una compactación de datos antiguos. Esta compactación usa una interpolación simple y lineal, al ser una interpolación, esto hace que se pierda detalle en esta información, pero seguirá siendo suficientemente informativa para la generación de informes y gráficas mensuales, anuales, etcétera.

En bases de datos de gran tamaño, este comportamiento puede ser bastante costoso en términos de rendimiento y tendría que ser desactivado; en su lugar, se recomienda optar por el modelo de base de datos histórica.

## Base de datos histórica

La **base de datos histórica** es una funcionalidad utilizada para almacenar toda la información pasada, que no se usa en vistas de días recientes, como por ejemplo los datos con una antigüedad de más de un mes. Esos datos se migran, de manera automática, a una base de datos diferente, que debe estar en un servidor distinto con un almacenamiento (disco) diferente al de la base de datos principal.

Cuando se muestre una gráfica o un informe con datos antiguos, Pandora FMS buscará los primeros días en la base de datos principal, y al llegar al punto en el que se migran a la base de datos histórica, pasará a buscar en ella. Gracias a ello el rendimiento se maximiza incluso al acumular un gran cantidad de información en el sistema.

## Configuración avanzada

La configuración por defecto de Pandora FMS no transfiere datos tipo cadena de texto string a la base de datos de histórico, no obstante si se ha modificado esta configuración y la base de datos de histórico está recibiendo este tipo de información es imprescindible que se configure su purgado, ya que de otro modo terminará ocupando demasiado, ocasionando grandes problemas y obteniendo un impacto negativo en el rendimiento.

Para configurar este parámetro se deberá ejecutar directamente una consulta en la base de datos para determinar los días tras los cuales se purgará esta información. La tabla en cuestión es tconfig y el campo string\_purge. Para establecer 30 días para el purgado de este tipo de información, se ejecutaría la siguiente query directamente sobre la base de datos de histórico:

```
UPDATE tconfig SET value = 30 WHERE token = "string_purge";
```

La base de datos es mantenida por un script llamado pandora\_db.pl, para comprobar que el mantenimiento de la base de datos se ejecuta correctamente se ejecuta el *script* de mantenimiento manualmente:

```
/usr/share/pandora_server/util/pandora_db.pl /etc/pandora/pandora_server.conf
```

No debería reportar ningún error. Si otra instancia está utilizando la base de datos, bien puede utilizar la opción -f que fuerza la ejecución; con el parámetro -p no realiza un compactado de los datos. Esto es especialmente útil en entornos de **Alta disponibilidad** (HA) con base de datos histórica, ya que el *script* se asegura de realizar en un orden y modo correcto los pasos necesarios para dichos componentes.

## Estados de los módulos en Pandora FMS

### ¿Cuándo se establece cada estado?

- Por un lado, cada módulo tiene en su configuración unos umbrales de *Advertencia* (WARNING) y *Crítico* (CRITICAL).
  - Estos umbrales definen los valores de sus datos para los que se activarán dichos estados.
  - Si el módulo proporciona datos fuera de estos umbrales, se considera que está en estado *Normal* (NORMAL).

- Cada módulo tiene, además, un intervalo de tiempo que establecerá la frecuencia con la que obtendrá los datos.
  - Ese intervalo será tenido en cuenta por la consola para recoger los datos.
  - Si el módulo lleva el doble de su intervalo sin recoger datos, se considera que ese módulo está en estado *Desconocido* (UNKNOWN).
- Por último, si el módulo tiene alertas configuradas y alguna de ellas ha sido disparada y no se ha validado, el módulo tendrá el correspondiente estado de *Alerta disparada*.

## Propagación y prioridad

Dentro de la organización de Pandora FMS, ciertos elementos dependen de otros, como es el caso de los módulos de un agente o los agentes de un grupo. También se puede aplicar esto al caso de las políticas de Pandora FMS, las cuales tienen asociados ciertos agentes y ciertos módulos que se consideran asociados a cada agente.

Dicha estructura es especialmente útil para *evaluar los estados de los módulos a simple vista*. Esto se consigue propagando hacia arriba en esta organización los estados, otorgándole así estado a los agentes, grupos y políticas.

### ¿Qué estado tendrá un agente?

Un agente se mostrará con el *estado más crítico* de los estados de sus módulos. A su vez, un grupo tendrá el *estado más crítico* de los estados de los agentes que a él pertenezcan, y lo mismo para las políticas, que tendrán el *estado más crítico* estado de sus agentes asignados.

De este modo, *al ver un grupo con estado crítico*, por ejemplo, se sabrá que al menos uno de sus agentes tiene ese mismo estado. Para localizarlo, se deberá bajar otro nivel, al de los agentes, para estrechar el cerco y encontrar el módulo o módulos causantes de propagar ese estado crítico.

### ¿Cuál es la prioridad de los estados?

Como se propaga el *estado más crítico* de los estados, se debe tener claro cuáles estados son prioritarios frente a los demás:

1. Alertas disparadas.
2. Estado crítico.
3. Estado de advertencia.
4. Estado desconocido.
5. Estado normal.

Cuando un módulo tiene alertas disparadas, su estado tiene prioridad sobre todos lo demás y el agente al que pertenece tendrá ese estado y el grupo al que pertenezca ese agente, a su vez, también.

Por otro lado, para que un grupo, por ejemplo, tenga estado normal, todos sus agentes deben tener dicho estado; lo que significa que a su vez todos los módulos de dichos grupos tendrán estado normal.

## Código de colores

● Estado de Alertas disparadas

● Estado Crítico

● Estado de Advertencia

● Estado Desconocido

● Estado Normal

## Gráficas de Pandora FMS

Las **gráficas** son unas de las implementaciones más complejas de Pandora FMS, extraen datos en tiempo real desde la base de datos y no se utiliza ningún sistema externo (RRDtool o similar).

Existen varios comportamientos de las gráficas en función del tipo de datos origen:

- Módulos asíncronos. Se asume que no hay compactación de datos. Los datos que hay en la base de datos son todas las muestras reales del dato, no hay compactación. Produce gráficas mucho más “exactas” y sin posibilidad de mala interpretación
- Módulos de tipo cadena de texto. Muestran gráficas con la tasa de datos a lo largo del tiempo.
- Módulos de datos numéricos. La mayoría de los módulos reportan este tipo de datos.
- Módulos de datos booleanos. Corresponden a datos numéricos en monitores, *PROC*: por ejemplo, chequeos Ping, estado de interfaces, etc. El valor 0 se corresponde con el estado Crítico, y el valor 1 con el estado “Normal”.

[Volver al índice de documentación de Pandora FMS](#)