

PRÁCTICA 3

SGBD



4 ENERO 2024

PRÁCTICA 3 - EMP

Mario Ventura Burgos 43223476-J

Grado en Ingeniería Informática (GIN 3)

1. ¿QUÉ ES DB2?

IBM Db2, comúnmente conocido como DB2, **es un sistema de gestión de bases de datos** de tipo relacional desarrollado por IBM. DB2 ofrece a los gestores/administradores de bases de datos, arquitectos empresariales y desarrolladores una única plataforma para garantizar el funcionamiento continuo de sus aplicaciones críticas. También permite almacenar y recuperar la información necesaria y agiliza notablemente los procesos de toma de decisiones en su organización.

Poniéndonos en la piel de una organización, Db2 podría ayudarnos a modernizar nuestras apps de misión crítica, implementar IA's y aprendizaje automático a escala, diseñar apps en la nube, proteger y gobernar datos, impulsar el análisis en tiempo real y simplificar escenarios de datos.

2. PROCESOS PARA DB2

Antes de listar y describir los procesos de db2, para poder comprenderlos correctamente es necesario mencionar el modelo de procesos, ya que nos va a facilitar en cierta medida la posterior comprensión de estos. El **modelo de procesos de IBM Db2** es el siguiente:

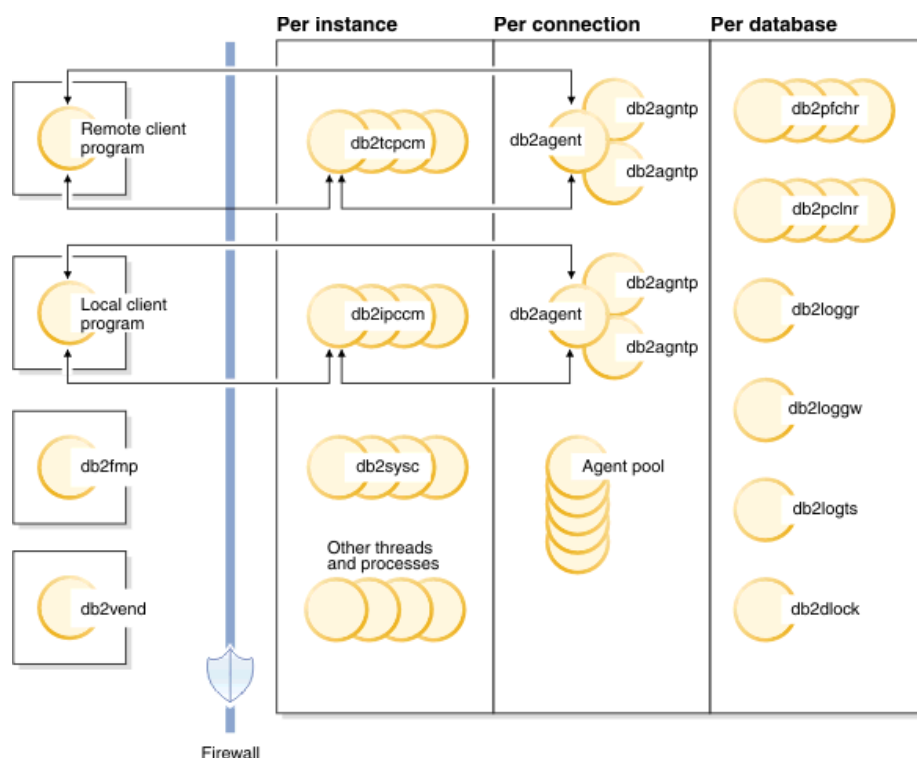


Figura 1, que muestra el modelo de proceso para sistemas de base de datos Db2.

Tal y como vemos en la *Figura 1*, el modelo se divide en dos partes principales: la parte del cliente y la parte del gestor de base de datos de Db2. Estas dos partes están separadas por un firewall.

En la parte del cliente encontramos dos procesos esenciales, que son los que proporcionan una capa de seguridad para la comunicación cliente-gestor, asegurando que esta se lleve a cabo de forma segura, confiable y sin interrupciones. Estos dos procesos son db2fmp y db2vend.

2.1 db2fmp & db2vend

Los **programas cliente** en IBM Db2 son aplicaciones o componentes de software que se ejecutan en las máquinas cliente y que interactúan con un servidor de bases de datos Db2. Estos programas permiten a los usuarios y aplicaciones enviar comandos y consultas al servidor de bases de datos, así como recibir y procesar los resultados. Son componentes esenciales que facilitan la interacción entre los usuarios o las aplicaciones y el servidor de bases de datos. Proporcionan la interfaz para enviar consultas, recibir resultados y gestionar la conectividad y las operaciones en la base de datos.

Los procesos de la parte del cliente, tal y como hemos mencionado previamente, son db2fmp y db2vend. Descubramos qué hace cada uno de ellos.

• db2fmp

Es un proceso de modalidad delimitada que es responsable de la ejecución de procedimientos almacenados delimitados y funciones definidas por el usuario del firewall. Se refiere a la **ejecución de rutinas externas** tales como procedimientos o funciones definidas por el usuario.

Es importante el hecho de que estas rutinas sean externas ya que esto asegura que estas rutinas estén fuera de las operaciones comunes SQL como consultas, y esto asegura que se ejecutan en un entorno separado y de forma segura. Esto es de gran utilidad ya que sirve para prevenir y evitar que rutinas creadas con intenciones maliciosas tengan acceso a recursos importantes como archivos o memoria compartida almacenados en el servidor.

Se usa cuando se crean funciones de usuario, se compilan en una biblioteca compartida (shared library) que se carga en el proceso de Db2 cuando es necesario. En estos casos, "db2fmp" facilita la ejecución de estas funciones de usuario en un entorno seguro y controlado.

El nombre db2fmp viene de concatenar a "db2" el acrónimo "fmp", que significa *Fenced Mode Process*. Esto hace referencia a que las funciones/rutinas mencionadas se ejecutan en modo fenced, asegurando que se cumple con todo lo mencionado.

• dbvend

Es un proceso para ejecutar código de proveedor externo. Es decir, se refiere a la ejecución de aplicaciones o programas que **no han sido producidos por IBM** pero que pueden interactuar con IBM Db2.

Es importante comprender que son componentes diferentes y cumplen funciones distintas, pero tanto db2fmp como db2vend son elementos esenciales en el ecosistema de Db2 para permitir la extensibilidad de funciones y garantizar el cumplimiento de las licencias y acuerdos asociados con productos adicionales.

Por otro lado, los procesos del otro lado del firewall (lado del gestor) se pueden dividir, separar o clasificar según su finalidad u objetivo. Basta con mirar la *Figura 1* para entender que, en el contexto de IBM Db2, los procesos a menudo se categorizan en función de su asociación con la instancia, la base de datos y la conexión.

A continuación, una breve explicación de cada una de estas 3 categorías de procesos:

1. **Para instancia** → Procesos que dan servicio al cliente y están asociados con la instancia de IBM Db2.
2. **Para conexión** → Procesos asociados con conexiones específicas a la base de datos que gestionan las operaciones relacionadas con una conexión de aplicación.
3. **Para la base de datos** → Procesos asociados con una base de datos específica que gestionan operaciones a nivel de la base de datos.

2.2 PROCESOS A NIVEL DE INSTANCIA

Como ya hemos dicho, los procesos a nivel de instancia dan servicio al cliente. Hay varios procesos que son esenciales para poner en marcha del gestor de Db2, como, por ejemplo, los listeners.

Los listeners son procesos cuya función es, tal y como dice su nombre, escuchar. Se encargan de escuchar en un puerto específico con el fin de recibir solicitudes de conexión por parte de los clientes, para direccionarlas posteriormente a los procesos del servidor.

Dado que hay varios protocolos de comunicación, hay un listener para cada uno de ellos configurado y un listener de comunicaciones entre procesos para programas locales. Los listeners incluyen:

- **db2ipccm** → Se encarga de establecer las comunicaciones entre el cliente local y el gestor con la ayuda de un agente llamado db2agent. El agente recibe la conexión y se encarga de coordinar las peticiones del cliente y comunicarse con él, con el fin de establecer la conexión entre la aplicación cliente (que se conecta a la base de datos a través del agente) y el propio coordinador.
- **db2tcpcm** → Se encarga de establecer comunicaciones TCP/IP entre una aplicación cliente y el gestor. También hace uso del agente coordinador db2agent para llevar a cabo su tarea, de forma que db2agent se comunica con el cliente y, al igual que en el caso anterior, lo conecta a la base de datos

Además de estos listeners, a nivel de instancia, encontramos otro proceso que es indispensable para el correcto funcionamiento de Db2 y para que el servidor de base de datos funcione. Estamos hablando del proceso **db2sysc**, el controlador principal del sistema.

En primer lugar, debemos saber que el controlador del sistema es un componente vital en la arquitectura de IBM Db2 y actúa como el punto de inicio y coordinación para el servidor de bases de datos. Su presencia es esencial para que el servidor funcione correctamente, y su ausencia o mal funcionamiento puede afectar negativamente la capacidad del servidor para atender solicitudes de bases de datos y ejecutar operaciones.

En Windows, el controlador del sistema es **db2sysc.exe**; y en UNIX es **db2sysc**. Este controlador debe existir necesariamente si el servidor de base de datos va a funcionar ya que db2sysc se encarga de gestionar los eventos críticos de Db2.

Este proceso juega un rol vital en cuanto a la gestión y supervisión, realizando tareas como el inicio y la detención de la instancia, la gestión de recursos del sistema (asignaciones y liberaciones de memoria, así como disponibilidad de recursos del sistema) y conexiones con clientes. También monitoriza la actividad en la instancia generando información de gran utilidad de las operaciones realizadas.

2.3 PROCESOS A NIVEL DE CONEXIONES

En cuanto a procesos asociados al nivel de conexiones, estos son procesos que realizan tareas de conexiones específicas a la base de datos, que gestionan las operaciones relacionadas con una conexión de aplicación. En esta “categoría” encontramos a los agentes: procesos encargados del control de los usuarios y aplicaciones que se conectan a la base de datos de forma simultánea generando cargas de trabajo concurrente.

La figura de los agentes es de una relevancia capital en IBM Db2. En este contexto, IBM habla de los agentes mencionando lo siguiente: “todas las solicitudes de conexión de programas cliente locales o remotos (apps) se les asigna un agente llamado **db2agent**. Este agente es un agente coordinador que, cuando se crea, realiza las solicitudes de base de datos en nombre de la app”.

Existen diferentes tipos de agentes en IBM Db2 , donde cada uno cumple con roles específicos:

1. **AGENTE COORDINADOR (db2agent)** → Los agentes coordinadores se encargan de la gestión y distribución de solicitudes de conexión de los clientes con el objetivo de que estos puedan conectarse a la base de datos. Los agentes coordinadores distribuyen estas solicitudes a los subagentes (explicados en el siguiente punto).

Se asemejan al concepto de proceso en un sistema operativo, puesto que tienen su propia memoria privada pero los recursos globales se comparten entre otros agentes.

2. **SUBAGENTES (db2agntp, db2agntps & db2agnta)** → Los subagentes reciben las solicitudes de la base de datos que los agentes coordinadores les han delegado al hacer gestión y distribución de solicitudes, mencionadas anteriormente. Los subagentes realizan el trabajo que se les ha delegado y devuelven el resultado a los agentes coordinadores.

Una vez han hecho su trabajo pasan a estar inactivos, aunque siguen estando asociados a una app. Es entonces cuando quedan alojados en una “*agent pool*” a la espera de que se les asigne un nuevo trabajo (*db2agnta*).

3. **AGENTES INACTIVOS (idle)** → Son los agentes que no están realizando ninguna operación y se encuentran alojados en una *agent pool* a la espera de que se les asigne un nuevo trabajo. No tienen conexión saliente o conexión con una base de datos puesto que no realizan ninguna operación activa en ese momento.

Tener una agent pool es de gran utilidad en contextos de grandes cargas de trabajo ya que tener procesos a la espera de recibir trabajo implica que estos pueden ser usados en el momento que se les necesite, y esto resulta en una mejora de rendimiento ya que no se necesita crear agentes nuevos para cada nueva conexión.

En resumen, el agente coordinador distribuye solicitudes a subagentes (**db2agntp** y **db2agnts**) que realizan el trabajo que se les ha delegado. Al acabar su trabajo, pasan a estar inactivos a la espera de nuevos trabajos. Estos subagentes desocupados que están asociados a una aplicación se llaman *db2agnta*. Los agentes coordinadores pueden estar conectados a una base de datos con un alias o a una instancia.

3.4 PROCESOS A NIVEL DE BASE DE DATOS

Como se comentó en su momento, los procesos a nivel de base de datos son procesos asociados con una base de datos específica que gestionan operaciones a nivel de la base de datos. En una base de datos se pueden hacer operaciones de muchos tipos, como, conexiones a la base de datos, recuperación de datos en caso de fallos, aseguramiento de la integridad de la información almacenada, etc.

Vamos a detallar con qué procesos cuenta IBM Db2 para cada una de estas operaciones:

1. Para Buffer Pool
2. Para logging
3. Db2lock
4. Para Backup & Restore
5. Procesos opcionales

Un **buffer pool** en el contexto de IBM Db2 se refiere a una memoria caché donde se almacenan de forma temporal una serie de datos modificados en memoria antes de escribirlos en el disco. Estos datos reciben el nombre de **dirty** y los bloques en memoria son conocidos como **páginas**. Esto sirve para mejorar el rendimiento ya que permite reducir el acceso al disco.

Para la buffer pool, Db2 cuenta con los siguientes procesos (aparecen en la Figura 1):

- **db2pfchr** → Este proceso se encarga de cargar los datos en una *buffer pool* de forma anticipada antes de que se necesiten. Esta operación recibe el nombre de **prefetching** y es la que da nombre al proceso. Contribuye a la mayor eficiencia en el uso de la memoria ya que reduce notablemente la cantidad de operaciones de Entrada/Salida y realiza la captura previa de páginas en almacenamientos intermedios.

Permite que las operaciones y consultas que necesitan el acceso a datos almacenados en la base de datos puedan llevarse a cabo de forma más eficiente.

- **db2pclnr** → Recibe su nombre debido a que es el **pool cleaner**, es decir, se encarga de la limpieza de la *buffer pool*. Su tarea es la de limpiar el espacio del buffer pool para asegurarse de que el sistema tiene espacio suficiente para almacenar nuevas páginas cuando lo necesita. En pocas palabras, es un limpiador de páginas de agrupación de almacenamientos intermedios que contribuye a la gestión eficiente de la memoria y al rendimiento del sistema.

Para llevar a cabo esta limpieza, debe identificar el contenido del buffer pool para determinar que páginas han sido modificadas desde su carga en memoria, escribir estas páginas identificadas en el disco y liberar el espacio que estas páginas ocupaban en el buffer pool una vez se ha asegurado que estas páginas modificadas han sido escritas en disco.

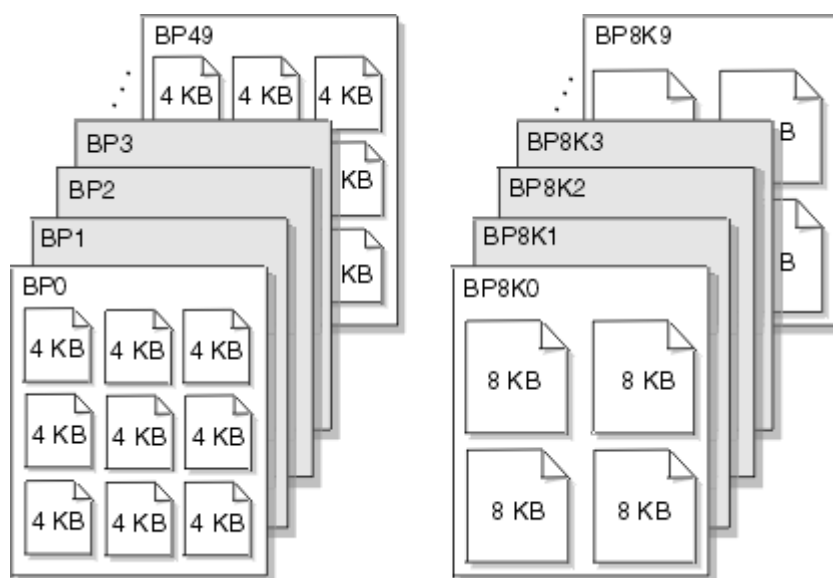


Figura 2 que ilustra buffer pools con páginas de 4KB y 8KB

Por otro lado, tenemos los procesos **para logging**, operación que consiste en el registro sistemático de todas las transacciones que modifican la información almacenada en la base de datos. Esto es especialmente importante para la recuperación de datos en caso de fallos en el sistema, además de asegurar la consistencia de estos datos. Dentro de la categoría de “procesos para logging” encontramos:

- **db2loggr** → La tarea principal de este proceso es crear los archivos de *log*, pero también se encarga de renombrarlos si se necesitase espacio adicional, caso en el cual db2loggr busca y renombra los archivos log que puedan ser renombrados en el *log path* activo y hayan sido copiados en el archivado. Es un proceso utilizado para manejar archivos de nuestro registro y gestionar el proceso y la recuperación de transacciones, siendo fundamental para la gestión de los archivos de registro, lo que incluye la escritura, manipulación y recuperación de transacciones en la base de datos.
- **db2loggw** → Es el proceso *log writer*, encargado de la escritura de los registros del buffer de logs (*log buffer*) en los archivos de log. Es el proceso responsable de escribir registros de anotaciones en los archivos de registro, lo que proporciona información adicional y metadatos de gran utilidad para el diagnóstico y análisis de problemas, además de garantizar que este proceso de escritura de información se haga de forma segura en los archivos de log.
- **db2logmgr** → Es el proceso log manager, encargado de hacer de gestor de archivos de log. La tarea de gestión de archivos de log incluye mover archivos de log activos a espacios de memoria/almacenamientos seguros, manejar archivos de log para una base de datos recuperable, administrar de manera integral los archivos de log y recuperarlos si es necesario.
- **db2lfr** → Es el proceso log file reader, encargado de la lectura de archivos de log. Este proceso lee los archivos de log del log path activo e incluye los datos de estos archivos de log en la imagen de backup.

Respecto a **db2dlock**, este es un proceso encargado de la monitorización y análisis de posibles interbloqueos (deadlocks) en la base de datos. Es importante porque ayuda a identificar, gestionar y afrontar los problemas relacionados con los interbloqueos y la concurrencia.

IBM Db2 cuenta también con procesos destinados a hacer tareas relacionados con las copias de seguridad (**backup**) y la recuperación de información (**restore**). Una backup es una copia de seguridad de una base de datos que conserva el estado de esta base de datos, los datos almacenados y la estructura de la base de datos en el momento en el que se hizo esta copia. Al tener una copia, se puede recuperar y restaurar toda esta información posteriormente si así se desea.

Se suelen hacer backups antes de hacer operaciones críticas en una base de datos. Por ello, Db2 cuenta con procesos que dan soporte a estas copias de seguridad.

- **db2bm** → Es el *buffer manipulator*. Su función es leer información del contenedor de datos de la tablespace y volcarlos en buffers de memoria vacíos para que, posteriormente, el proceso de controlador de medios **db2med** escriba los datos de estos buffers en la imagen del backup. Este es el proceso que se sigue para hacer una copia de seguridad de la base de datos.
- **db2med** → Como hemos mencionado, es el proceso “*media controller*”. Son usados para el restore, para leer información de los archivos de imagen del backup (hay un db2med por cada path/file) y volcarla en buffers de memoria vacíos. Posteriormente, los procesos **db2bm** son usados para escribir el contenido de estos buffers de memoria (ahora llenos) en los contenedores del tablespace. De esta forma se restaura (restore) la información de la copia de seguridad previamente realizada.

Por último, existen una serie de **procesos de carácter “opcional”**. Estos procesos pueden ser usados al iniciar la base de datos en Db2 si así se desea, aunque su uso no es estrictamente necesario. A continuación, se detallarán algunos de ellos:

- **db2resync** → Es el proceso encargado de resincronización, que escanea y explora la lista de resincronización global. Opera en entornos donde la replicación de datos está activa y se encarga de sincronizar información crítica entre nodos o componentes. Es de gran utilidad para corregir fallos de sincronización causados por fallos o interrupciones.
- **db2gds** → Es el proceso *global daemon spawner*. Su función es inicializar nuevos procesos en entornos basados en UNIX.
- **db2wdog** → En entornos basados en UNIX, el proceso db2wdog es el proceso que actúa como perro guardián (*watchdog*, que da el nombre ‘*wdog*’ al proceso). Es el proceso que actúa como vigilancia, manejando terminaciones anómalas en los procesos que supervisa en sistemas operativos UNIX y Linux. En caso de detectar anomalías también toma medidas correctivas.
- **db2fcmdm** → Es el proceso *fast communications manager daemon*. Su función es la de el manejo de comunicaciones internas entre nodos.
- **db2pdbc** → Es el proceso controlado del sistema paralelo. Facilita la ejecución eficiente de operaciones paralelas mediante el manejo de solicitudes paralelas en contextos de entornos donde se realizan múltiples operaciones en paralelo.
- **db2panic** → Proceso que actúa como “agente de pánico”. Interviene en situaciones de emergencia respondiendo a condiciones críticas, encargándose de manejar las solicitudes urgentes de pánico producidas cuando sucede algún tipo de error para el cual no se conoce una forma de actuar o gestionarlo. Toma medidas para mitigar situaciones urgentes y evitar posibles daños o fallos mayores.

Por último, detallaremos algunos procesos relacionados con **High Availability Disaster Recovery** (HADR), o “recuperación de desastres de alta disponibilidad” en español. HADR proporciona una solución de alta disponibilidad para fallos completos o parciales.

- **db2hadrp** → Proceso encargado de la recuperación tras un desastre en entornos de alta disponibilidad y recuperación ante desastres (HADR). Proporciona soluciones a fallos totales o parciales mediante la replicación de modificaciones de datos en una base de datos fuente a una base de datos destino, llamadas *primary database* y *standby database* respectivamente. Garantiza la consistencia y disponibilidad de la base de datos.
- **db2redom** → Es el proceso “*redo master*”. Juega un rol vital en la recuperación de la base de datos en entornos de alta disponibilidad, ya que su función es procesar los registros del *redo log* y coordinar el trabajo y asignar estos registros a los trabajadores de redo (*redo workers*), para ser procesados.
- **db2redow** → Es el proceso de “*redo worker*”, y como se ha mencionado anteriormente, trabaja en conjunto con el *redo master*, procesando los registros de *redo log* que el redo master le ha asignado en el proceso de recuperación.

4. BIBLIOGRAFÍA

Las fuentes de información usadas para responder a las cuestiones planteadas son las siguientes:

- **¿Qué es Db2?**

<https://www.ibm.com/es-es/products/db2>

<https://www.precisely.com/glossary/db2>

https://en.wikipedia.org/wiki/IBM_Db2

<https://www.techtarget.com/searchdatamanagement/definition/Db2>

- **Información sobre los procesos necesarios y opcionales para Db2**

https://www.ibm.com/docs/es/db2/11.1?topic=architecture-db2-process-model#c0008930_title_6

<https://www.ibm.com/docs/es/db2-for-zos/11?topic=objects-application-processes-transactions>

<https://www.ibm.com/docs/es/db2-for-zos/11?topic=procedure-stored-processing>

<https://www.ibm.com/docs/es/db2-for-zos/11?topic=concepts-application-processes-concurrency-recovery>

<https://www.ibm.com/docs/es/db2/11.1?topic=architecture-db2-process-overview>

<https://www.ibm.com/docs/en/db2-for-zos/11?topic=system-role-buffer-pools-in-caching-data>

<http://thinkingdb2.blogspot.com/2015/03/understanding-and-tuning-db2luw-backup.html>

<https://www.ibm.com/support/pages/db2-transaction-logging-related-edus>

<https://www.ibm.com/docs/en/db2/11.5?topic=server-high-availability-disaster-recovery-hadr>

<https://www.ibm.com/docs/en/db2/11.5?topic=architecture-database-agents>