

PRÁCTICA 3

Sistemas de Gestión de Bases de Datos



26 NOVIEMBRE

PRÁCTICA 3 - EMP

Mario Ventura Burgos 43223476-J

Grado en Ingeniería Informática (GIN 3)

CURSO 2023-2024

ÍNDICE

- 1. INTRODUCCIÓN3
- 2. ¿QUÉ ES DB2?.....4
- 3. PROCESOS NECESARIOS Y OPCIONALES PARA BD26
 - 3.1 ARQUITECTURA6
 - 3.2 CATEGORÍAS DE PROCESOS7
 - 3.3 db2fmp & db2vend.....8
 - 3.4 PROGRAMAS CLIENTE.....9
 - 3.5 LISTENERS.....9
 - 3.6 HEBRAS Y PROCESOS DEL SERVIDOR DE BASES DE DATOS.....10
 - 3.7 AGENTES13
 - 3.8 EDUs DE BASE DE DATOS.....14
- 4. BIBLIOGRAFÍA19

1. INTRODUCCIÓN

En esta práctica se nos pide lo siguiente:

“Llista, descriu i explica tots els processos que sempre es necessiten per a tenir una base de dades DB2 en marxa. A més, també llista, descriu i explica tots els processos opcionals.”

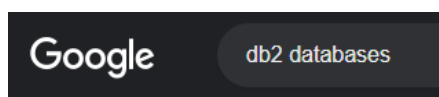
Para contestar a esto, debemos comprender antes varios conceptos:

1. ¿Qué es DB2?
2. Procesos necesarios y opcionales existen para tener DB2 en marcha
3. ¿Qué categorías de procesos hay?

De esta forma, se espera comprender qué es DB2, en qué contextos se usa, ventajas o desventajas presenta con otros tipos de bases de datos, qué procesos obligatorios/necesarios (y opcionales) se necesitan para tenerlo en marcha y de qué categoría o tipo son estos procesos.

2. ¿QUÉ ES DB2?

En este primer apartado trataremos de descubrir qué es db2, qué características presenta, en qué contextos se usa, etc. El primer paso que casi todos hacemos cuando queremos saber qué es algo es, probablemente, buscarlo en Google. Si buscamos en Google lo primero que nos encontramos es lo siguiente:



Lo primero que nos sale es un enlace que nos lleva directos a la página web de IBM. Si indagamos un poco, resulta que db2 es (citado textualmente de Google): “una marca comercial, propiedad de IBM, bajo la cual se comercializa un sistema de gestión de base de datos.”

Tras un poco de investigación en este asunto, podemos finalmente presentar la siguiente respuesta:

IBM Db2, comúnmente conocido como DB2, **es un sistema de gestión de bases de datos** de tipo relacional desarrollado por IBM. DB2 es, por tanto, un producto que IBM pone a nuestra disposición, y que ha sido clave en la cartera de productos de bases de datos de la empresa durante años (el lanzamiento inicial fue en 1993). Está descrito por ellos mismos como la base de datos nativa de la nube diseñada para impulsar las transacciones de baja latencia y el análisis a escala en tiempo real.

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.

Los productos de db2 incluyen bases de datos operativas, almacenes de datos, data lakes y fast data (data lake es una solución de análisis centrada en desbloquear el valor de conjuntos de datos no estructurados). Se usa en una amplia variedad de contextos y entornos gracias a su versatilidad y capacidad de adaptarse a distintas necesidades empresariales. Algunos de los contextos en los que se usa son empresas y corporaciones que quieran implementarlo, sector de la salud (gestión de grandes volúmenes de datos), el gobierno (gestión de datos críticos como registros civiles), telecomunicaciones, instituciones financieras, investigación, etc.

IBM ofrece 3 ediciones:

- **IBM Db2 Community Edition:** Descarga gratuita, limitado a 4 núcleos de CPU, 16GB de RAM y sin soporte de la compañía.
- **IBM Db2 Standard Edition:** Pago bajo licencia, limitado a 16 núcleos de CPU, 128GB de RAM y soporte de la compañía.

-
- **IBM Db2 Advanced Edition:** Solo disponible como componente del IBM Hybrid Data Management Platform (HDMP). Como tal, está disponible mediante licencia perpetua o suscripción mensual para uso sin restricciones en producción y con soporte de IBM Premium.

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.

Por último, algunas de las ventajas que presenta IBM Db2 son:

1. **Escalabilidad:** Db2 es capaz de manejar grandes volúmenes de datos y escalar para adaptarse a las necesidades de las aplicaciones empresariales, desde entornos pequeños hasta sistemas empresariales de gran envergadura.
2. **Rendimiento:** Ofrece un rendimiento eficiente, lo que es crucial para aplicaciones que requieren respuestas rápidas y procesamiento de datos ágil y en tiempo real.
3. **Soporte Multiplataforma:** Db2 es compatible con diversas plataformas, incluyendo mainframes, servidores Unix, Linux y Windows, lo que brinda flexibilidad a las organizaciones con infraestructuras heterogéneas.
4. **Integración con Tecnologías Emergentes:** Db2 se ha adaptado para integrarse con tecnologías emergentes, como la inteligencia artificial (IA) y el aprendizaje automático (ML), lo que permite a las organizaciones aprovechar las ventajas de estas capacidades para el análisis de datos.
5. **Seguridad:** Db2 ofrece robustas medidas de seguridad para proteger la integridad y la confidencialidad de los datos almacenados. Esto incluye funciones de control de acceso, cifrado y auditoría.
6. **Soporte para Big Data:** Db2 ha evolucionado para manejar grandes volúmenes de datos y entornos de big data, lo que permite a las organizaciones gestionar y analizar datos a gran escala.
7. **Flexibilidad de Implementación:** Ofrece diversas ediciones, desde versiones gratuitas para pequeñas empresas hasta ediciones empresariales para grandes corporaciones, lo que permite a las organizaciones elegir la opción que mejor se adapte a sus necesidades y presupuesto.

Estas ventajas hacen que Db2 sea una opción atractiva para aquellas organizaciones que buscan un sistema de gestión de bases de datos fiable y escalable.

3. PROCESOS NECESARIOS Y OPCIONALES PARA DB2

Antes de listar y describir los procesos de db2, hablemos un poco acerca de su arquitectura, ya que nos va a facilitar en cierta medida la comprensión de estos procesos:

3.1 ARQUITECTURA

Como bien sabemos, para poner en marcha un sistema gestor de bases de datos se necesita un conjunto de procesos que dan servicio a los clientes (listener, dispatcher...). En el caso de db2 , si buscamos en la documentación ofrecida por IBM para las versiones más recientes de Db2 (11 y 11.1), en el apartado de “arquitectura del sistema”, podemos ver que hay una sección llamada “El modelo de proceso de db2”. En esta sección de la documentación, se habla de que db2 debe realizar muchas tareas diferentes, como procesar solicitudes de aplicación de base de datos, asegurarse de que los registros de anotaciones cronológicas se graban en el disco, etc. Normalmente, cada una de estas tareas la realiza una unidad independiente llamada **EDU**.

Así pues, db2 tiene una **arquitectura multihebra**, donde una nueva hebra requiere menos memoria y menos recursos del sistema operativo que un proceso, porque hay ciertos recursos del sistema operativo que pueden ser compartidos por distintas hebras del mismo proceso. Para cada base de datos que se accede, se inician distintas EDU, independientes entre sí, que se ocupan de tareas de la base de datos como la captación previa, comunicación o registro.

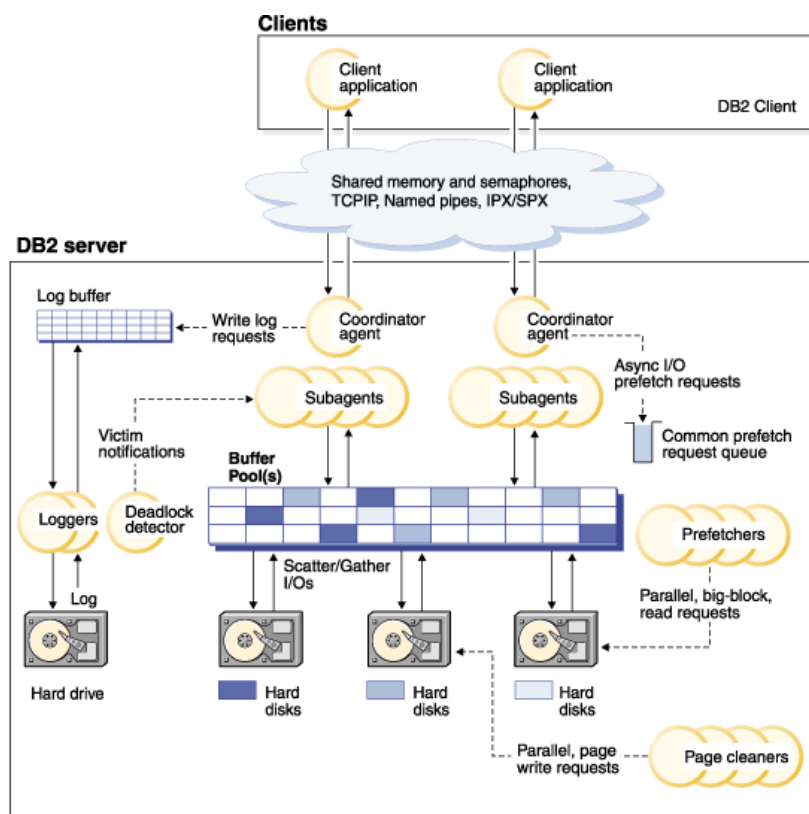


Figura 1, que muestra una visión general de la arquitectura y procesos de Db2.
Las EDU se muestran como círculos o grupos de círculos

Las EDU se implementan como hebras en todas las plataformas. Los agentes de Db2 son el tipo más común de EDU. Estos agentes realizan la mayor parte del proceso de SQL y XQuery en nombre de las aplicaciones (captadores previos y limpiadores de páginas son otras EDU comunes).

Ahora que conocemos que tipo de arquitectura usa db2, vamos a profundizar en los procesos:

3.2 CATEGORÍAS DE PROCESOS

Los procesos se pueden dividir, separar o clasificar según su finalidad u objetivo. 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. Tenemos 3 categorías de procesos en función de cuál es su objetivo:

1. **Para conexión** → Procesos asociados con conexiones específicas a la base de datos que gestionan las operaciones relacionadas con una conexión de aplicación.
2. **Para instancia** → Procesos asociados con la instancia de Db2 que son responsables de gestionar los recursos a nivel del sistema.
3. **Para la base de datos** → Procesos asociados con una base de datos específica que gestionan operaciones a nivel de la base de datos.

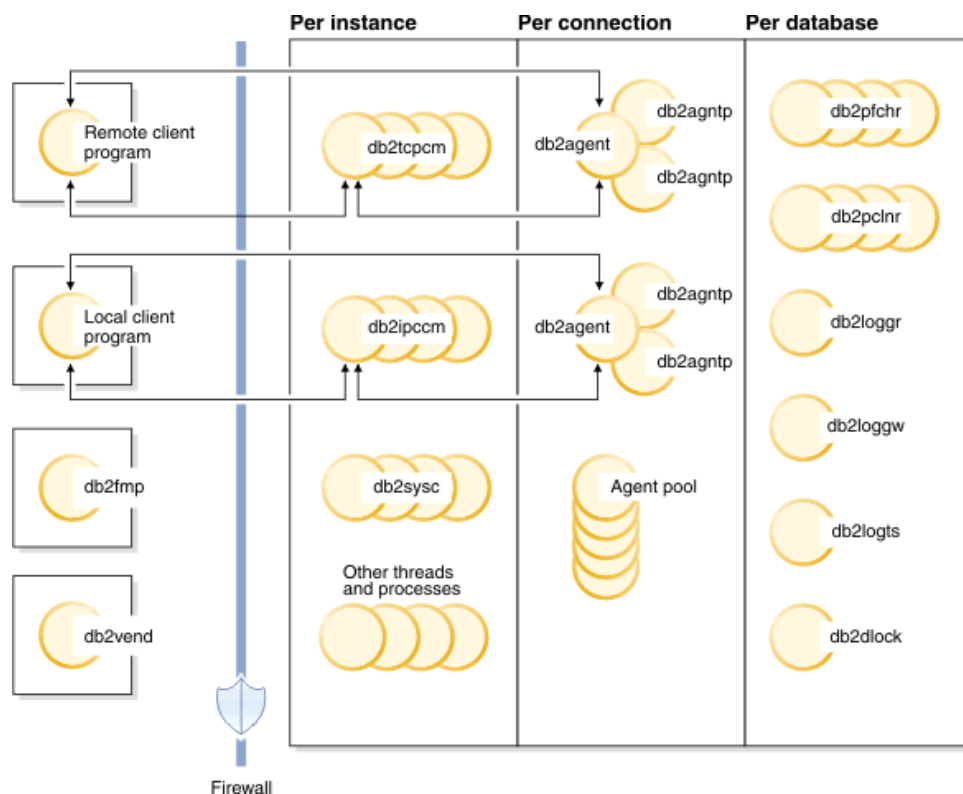


Figura 2, que muestra el modelo de proceso para sistemas de base de datos Db2.

En la documentación de IBM sobre Db2, no se muestran los procesos separados por categoría. Se clasifican en función de lo que hace cada uno, pero no se menciona la categoría a la que pertenecen (por ejemplo, listeners y agentes son de la misma categoría, pero se explican en apartados diferentes). Aun así, puesto que sabemos qué hace cada uno, se puede llegar a la conclusión de que los procesos detallados en la documentación son de la siguiente categoría o tipo:

- De conexión → Son los procesos detallados en los apartados de “listeners” y “agentes”, detallados en la primera parte de la documentación
- De base de datos → Son los procesos detallados en el apartado “Database EDUs” de la documentación
- De instancia → Son los procesos detallados en el último apartado de la documentación, que recibe el nombre de “Database server threads and processes”.

Se podría añadir, además, una categoría para los procesos que, en la imagen anterior, se puede apreciar que están al otro lado del firewall. Estos son los **programas cliente** y los procesos **db2fmp** y **db2vend**

A continuación, mostraremos una lista de estos procesos de IBM Db2, hablando detalladamente sobre qué hace cada uno. Comenzaremos hablando de los procesos que están “al otro lado” (a la izquierda en la *figura 2*) del firewall, y posteriormente hablaremos de los procesos de cada una de las categorías mencionadas.

3.3 db2fmp & db2vend

• 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 funcionalidad de funciones de usuario en el entorno de IBM Db2. Estas funciones son módulos de código que los usuarios pueden escribir en lenguajes como C o Java y ejecutar dentro del proceso del servidor de bases de datos para realizar operaciones específicas. Siempre es independiente, pero puede ser multihebra, en función de los tipos de rutinas que ejecute.

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.

• dbvend

Es un proceso para ejecutar código de proveedor en nombre de una EDU, como, por ejemplo, ejecutar un programa de salida de usuario para el archivado de registros en (solo UNIX). Es decir, se refiere a un directorio dentro de la instalación de IBM Db2 que contiene información relacionada con proveedores de software y licencias asociadas. En este directorio, se almacenan archivos y configuraciones que son relevantes para la administración de licencias y acuerdos con proveedores de terceros.

Se utiliza para gestionar la licencia de productos asociados o integrados con IBM Db2. Puede incluir archivos que definen restricciones de licencia, acuerdos de uso y otra información contractual.

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.

3.4 PROGRAMAS CLIENTE

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.

Pueden ser **remotos** (entorno externo) o **locales** (misma máquina que el servidor de base de datos). Es por eso que, en la *figura 2* aparecen 2 representaciones de programas locales llamadas “local client program” y “remote client program”, en representación de estos dos tipos.

3.5 LISTENERS

Los programas cliente pueden ser remotos o locales, es decir, pueden ejecutarse en la misma máquina que el servidor de base de datos. Los programas cliente hacen el primer contacto con una base de datos a través de una escucha (Listener) de comunicación que se inicia cuando se inicia el servidor de base de datos Db2. 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**, para conexiones de cliente local
- **db2tcpcm**, para conexiones TCP/IP
- **db2tcpdm**, para solicitudes de la herramienta de descubrimiento TCP/IP

Nuevamente, podemos categorizar estos procesos si miramos la *figura 2*, donde podemos ver que tanto db2ipccm como db2tcpcm aparecen en la categoría de procesos para instancia.

Los listeners en un sistema de gestión de bases de datos como IBM Db2 son necesarios para permitir que el servidor de base de datos acepte conexiones entrantes desde clientes. La presencia de listeners es esencial para la comunicación entre clientes y el servidor de base de datos y, por tanto, son procesos necesarios, no opcionales.

3.6 HEBRAS Y PROCESOS DEL SERVIDOR DE BASES DE DATOS

Tal y como se ha comentado, la documentación oficial ofrecida por IBM no muestra los procesos ordenados o agrupados por categoría. Sin embargo, con el objetivo de facilitar la comprensión de la vista general de todos los procesos, vamos a dar un salto hasta el último apartado de esta documentación para hablar sobre las hebras y procesos del servidor de bases de datos, ya que estos son procesos para instancia al igual que los Listeners (mencionados en el apartado anterior).

En esta sección que se describen varios procesos y hebras del servidor de bases de datos que son esenciales para su funcionamiento. Aquí hay una explicación de cada uno de ellos:

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. Es una EDU imprescindible puesto que su función principal es el manejo y gestión de eventos críticos o situaciones urgentes en un servidor Db2.

Ahora sí, hablemos de las hebras y procesos que llevan a cabo diversas tareas del servidor de base de datos:

- **Db2acd** → Es un daemon autónomo, anteriormente conocido como db2hmon, que aloja el supervisor de salud, programas de utilidad de mantenimiento automático y el planificador de tareas administrativas. Gestiona tareas relacionadas con la salud del sistema, programas de mantenimiento automático y tareas administrativas programadas.
- **Db2aiiothr** → Gestiona solicitudes de E/S asíncronas para una partición de base de datos (disponible solo en sistemas UNIX). Se usa para mejorar el rendimiento gestionando operaciones de E/S de manera asíncrona
- **Db2alarm** → Como bien dice su nombre, actúa como alarma, notificando a las EDUs cuando caduca el temporizador solicitado (disponible solo en sistemas UNIX).
- **Db2disp** → Es un Asignador del concentrador de conexiones de cliente. Se encarga de gestionar y asignar conexiones de clientes a las Unidades de Despacho de Eventos (EDUs) correspondientes para manejar las solicitudes de manera eficiente.
- **Db2fcms** → Es un daemon emisor del gestor de comunicaciones rápidas.
- **Db2fcmr** → Es el daemon receptor del gestor de comunicaciones rápidas.
- **Db2fmd** → Este daemon supervisa y gestiona errores en el sistema. Está diseñado para detectar y responder a condiciones de error en el entorno de Db2.

-
- **Db2i**periodic → Hebra del planificador de tareas. Esta hebra está asociada con el planificador de tareas y ejecuta tareas programadas periódicas. Además, las hebras denominadas *db2i*periodicWatch<numero_tarea> y *db2i*periodicExec<numero_tarea> ejecutan tareas específicas.
 - **Db2licc** → Es un gestor de las db2 instaladas que gestiona las licencias instaladas en el entorno de Db2. Garantiza que la instalación cumpla con las restricciones y términos de licencia.
 - **Db2panic** → Agente de pánico. Maneja solicitudes urgentes después de alcanzar límites críticos del agente. Interviene en situaciones de emergencia respondiendo a condiciones críticas, tomando medidas para mitigar situaciones urgentes y evitar posibles fallas o daños mayores.
 - **Db2pdbc** → Es el controlador del sistema paralelo. El encargado de manejar solicitudes paralelas de particiones de base de datos remotas en entornos de bases de datos particionadas y en entornos Db2 pureScale. De esta forma, facilita la ejecución eficiente de operaciones paralelas en particiones de bases de datos remotas
 - **Db2resync** → Agente de resincronización que explora la lista de resincronización global. Se encarga de sincronizar información crítica entre nodos o miembros del clúster.
 - **Db2rocm & db2rocme** → Son supervisores del estado de clúster. En entornos Db2 pureScale, supervisan el estado operativo de los miembros de Db2 y los recursos de almacenamiento en memoria caché (CF) de clúster en cada host. Distribuyen información de estado de clúster a todos los miembros y CF de Db2 de la instancia. Mantienen la visibilidad y la consistencia en el estado operativo del clúster en un entorno Db2 pureScale, facilitando la coordinación y la toma de decisiones.
 - **Db2sysc** → Es la EDU del controlador del sistema principal, que maneja sucesos críticos del servidor Db2 y gestiona eventos críticos del servidor, asegurando la capacidad de respuesta y la acción adecuada frente a condiciones excepcionales.
 - **Db2sysc desocupado** → Procesos desocupados de Db2 que permiten la luz de reinicio de un miembro invitado en un host de forma rápida y sin competir con el miembro residente para los recursos.
 - **Db2syscque** → Es la escucha de colas de mensajes que proporciona colas de solicitudes y distribuye servicios al controlador del sistema en un entorno Db2 pureScale y facilita la comunicación y la distribución eficiente de solicitudes en un entorno Db2 pureScale utilizando colas de mensajes.
 - **Db2thcln** → Recicla recursos cuando termina una Unidad de Despacho de Eventos (EDU) (disponible solo en sistemas UNIX). Se usa para liberar recursos utilizados por una EDU después de que ha completado su tarea, contribuyendo a la gestión eficiente de los recursos del sistema.

-
- **Db2wdog** → Es el proceso de vigilancia que maneja terminaciones anómalas en sistemas operativos UNIX y Linux. Actúa como un proceso de vigilancia.
 - **Db2wlm** → Es la hebra de planificación del asignador de WLM. Forma parte del asignador de Workload Management (WLM) y se encarga de la planificación de recursos, gestionando la asignación y la planificación de recursos para las operaciones del sistema, contribuyendo a un uso eficiente de los recursos disponibles.
 - **Db2wlmtm** → La hebra del temporizador del asignador de WLM. Esta Hebra del temporizador asociada con el asignador de Workload Management (WLM) gestiona tareas relacionadas con el tiempo, como la ejecución de operaciones programadas o la actualización periódica de la planificación de recursos.

3.7 AGENTES

La figura de los agentes también es relevante en IBM Db2. En este contexto IBM define los agentes explicando 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”.

En otras palabras, un agente db2agent es una **entidad asociada a un hilo de ejecución o proceso** que trabaja en nombre de una aplicación o usuario para realizar operaciones en la base de datos. Juegan un papel crucial, por tanto, en la gestión de las conexiones y ejecución de consultas en el entorno de Db2.

En entornos de bases de datos particionadas o con habilitación de paralelismo intraquery, los agentes en IBM Db2 cumplen roles específicos. El agente coordinador distribuye solicitudes a subagentes (**db2agntp** y **db2agnts**), mientras que los subagentes desocupados 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.

Sobre agentes, en la documentación ofrecida por IBM se distinguen estos tipos de agentes coordinadores:

- Conectado a la BD con un alias tal como db2agent (DATA1), que estaría conectado al alias de la base de datos DATA1.
- Conectado a una instancia como db2agent (user1), que está conectado a la instancia user1.

El servidor de bases de datos crea instancias de agentes para funciones particulares, como **db2agnti** para supervisores de sucesos y **db2agnsc** para paralelizar reinicios de base de datos. Los agentes de pasarela (**db2agentg**) permiten la conectividad indirecta a bases de datos remotas.

Los agentes desocupados residen en una agrupación y están disponibles para solicitudes de agentes coordinadores o subagentes. Mantener un tamaño adecuado de la agrupación mejora el rendimiento, ya que los agentes desocupados pueden utilizarse cuando sea necesario, evitando asignar nuevos agentes para cada conexión de aplicación.

Los agentes agrupados pueden estar asociados a bases de datos remotas (**db2agntgp**) o locales (**db2agntdp**), ofreciendo flexibilidad en la gestión de conexiones y recursos.

En términos generales, los agentes en Db2 a menudo se asocian con la gestión de conexiones y ejecución de operaciones en nombre de aplicaciones o usuarios. Bajo la perspectiva de que los agentes de aplicación están vinculados a conexiones y gestionan la interacción entre una aplicación y la base de datos, en este sentido, los agentes de aplicación pueden ser vistos como procesos de conexión. De hecho, así aparecen en la *figura 2*, donde varios de los procesos que se aprecian son agentes como los mencionados en el transcurso de este apartado.

3.8 EDUs DE BASE DE DATOS

Lo siguiente que nos encontramos en la documentación ofrecida por IBM es una lista de las EDU importantes que utiliza cada base de datos (es por eso que estamos hablando de procesos que entrarían en la categoría de base de datos). Recordemos que la arquitectura multihebra dividida en EDUs de Db2 hace que cada una de estas EDU se encargue de una tarea específica, y esto implica la ejecución de uno o varios procesos en cada EDU.

A continuación, la lista de algunas de las EDU importantes que utiliza cada base de datos:

- **db2castructevent** → Maneja el proceso de entrada y llegada de CF en un cluster pureScale. Se encarga de coordinar y gestionar los eventos relacionados con cambios en la estructura del clúster y se usa cuando se añade o elimina un miembro del clúster pureScale, momento en el cual db2castructevent se activa para gestionar la reconfiguración y asegurar que la estructura del clúster se actualice correctamente.
- **db2CFConnPoolMgr** → Vuelve a equilibrar las conexiones de CF (Coordinator Flow) entre los enlaces de miembro-CF en un clúster pureScale. Se encarga de gestionar y optimizar la distribución de conexiones para mejorar el rendimiento y la eficiencia del clúster.
- **db2cfmt** → Se encarga del ajuste de memoria CF dinámica en un clúster pureScale. Gestiona la asignación y liberación de recursos de memoria para adaptarse a las necesidades cambiantes del clúster. Se suele usar cuando la carga de trabajo en el clúster pureScale varía, momento en el cual db2cfmt ajusta dinámicamente la asignación de memoria para optimizar el rendimiento.
- **db2clstrRscMon** → Supervisa los cambios en los estados de los adaptadores de red en un clúster pureScale, es decir, se asegura de que el sistema esté informado sobre la disponibilidad y cambios en los adaptadores de red. Si hay cambios en la conectividad de red o se produce una falla en un adaptador, db2clstrRscMon detectará la situación y tomará medidas.
- **db2dlock y db2glock** → db2dlock se usa para la detección de puntos muertos. En entornos de base de datos particionada, se utiliza una hebra adicional (db2glock) para coordinar la información recopilada por db2dlock en cada partición del clúster pureScale.
- **db2fw y db2fwx** → db2fw es el grabador rápido del supervisor de sucesos, utilizado para la grabación en paralelo de grandes volúmenes de datos del supervisor de sucesos en tablas, archivos o conductos. db2fwx es una hebra específica de db2fw, de grabado rápido de supervisor de sucesos, donde la X indica el número de hebra. El número de hebras db2fwx es igual al número de CPU lógicas del sistema (para CPU de varios núcleos, cada núcleo cuenta como una CPU lógica).
- **db2hadrp** → Esta hebra se encarga del servidor primario de recuperación tras un desastre en entornos de Alta Disponibilidad y Recuperación ante Desastres (HADR). Se usa cuando ocurre un desastre y la base de datos se conmuta a un servidor en espera, que es cuando db2hadrp gestiona la recuperación del servidor primario, garantizando la consistencia y disponibilidad de la base de datos.

- **db2hadrs** → Esta hebra se refiere al servidor en espera en entornos HADR. En este caso, el uso tiene lugar mientras el servidor primario está activo; entonces, db2hadrs permanece en espera, listo para asumir el rol principal en caso de que el servidor primario experimente un fallo o se desconecte.
- **db2lfr** → Se usa para lectores de archivos de registro que procesan archivos de registro individuales. Es responsable de leer y procesar archivos de registro individuales, lo que es fundamental para la recuperación y reproducción de transacciones.
- **db2loggp** → Es utilizado en el contexto de trabajo de registro periódico como. Realiza tareas periódicas, como la gestión del espacio de registro y la determinación de la ventana de recuperación.
- **db2loggr** → Hebra utilizada para manejar archivos de nuestro registro y gestionar el proceso y la recuperación de transacciones. Fundamental para la gestión de los archivos de registro, lo que incluye la escritura, manipulación y recuperación de transacciones en la base de datos.
- **db2loggrq** → Proceso utilizado para la lógica de captura remota asistida utilizada por HADR en el entorno pureScale y que es vital en entornos pureScale con HADR.
- **db2loggw** → Proceso utilizado para escribir registros de anotaciones en los archivos de registro. db2loggw es responsable de escribir registros de anotaciones en los archivos de registro, lo que proporciona información adicional y metadatos que es útil para el diagnóstico y análisis de problemas.
- **db2logmgr** → Proceso utilizado como gestor de registros, encargado de manejar archivos de registro para una base de datos recuperable y de administrar de manera integral los archivos de registro.
- **db2logts** → Realiza el seguimiento de qué espacios de tabla tienen registros de anotaciones en qué archivos de registro. La información se registra en el archivo DB2TSCHG.HIS en el directorio de la base de datos. Su correcto funcionamiento es crucial para el seguimiento y registro de espacios de tabla y registros de anotaciones, proporcionando información valiosa para el diagnóstico y análisis del sistema.
- **db2lused** → Se utiliza para actualizar el uso de objetos, es decir, esta hebra es responsable de mantener actualizado el uso de objetos en el sistema.
- **db2periodic** → Ejecuta una tarea de base de datos interna que se debe ejecutar periódicamente. Algunos ejemplos de casos de uso en los que se realizan tareas periódicas podrían ser: recopilación de estadísticas o la ejecución de operaciones de mantenimiento rutinarias.
- **db2pclnr** → Se utiliza para limpiadores de páginas de agrupación de almacenamientos intermedios, contribuyendo a la gestión eficiente de la memoria y al rendimiento del sistema.

- **db2pcsd** → Se utiliza para la limpieza autónoma de la memoria caché de paquetes, mejorando la eficiencia y el rendimiento del sistema.
- **db2pfchr** → Se utiliza para captadores previos de agrupación de almacenamiento intermedio. Realiza la captura previa de páginas en almacenamientos intermedios, contribuyendo a la mayor eficiencia en el uso de la memoria
- **db2pkgrb** → Se utiliza para la revinculación autónoma de paquetes no válidos, es decir, realiza intentos de volver a vincular cada paquete no válido durante el inicio del nodo de catálogo de la base de datos.
- **db2redom** → Se utiliza para el “redo master”. Procesa registros de rehacer y asigna registros para los trabajadores de rehacer durante la recuperación. Juega un papel vital, por tanto, en la recuperación, ya que su trabajo consiste en ir procesando registros de rehacer y coordinando el trabajo de los trabajadores de rehacer para garantizar la consistencia del sistema después de un evento de recuperación.
- **db2redow** → También relacionado con el proceso de recuperación, ya que trabaja en conjunto con el *redo master* para procesar registros de rehacer durante la recuperación, asegurando la integridad y consistencia de la base de datos.
- **db2shred** → Usado para procesar registros de anotaciones individuales dentro de las páginas de registro. Cuando se ejecutan transacciones o se realizan cambios en la base de datos, los detalles se registran como registros de anotaciones en las páginas de registro. Entonces, db2shred se encarga de procesar y analizar estos registros.
- **db2stmm** → Controla la característica de gestión de memoria de autoajuste (STMM = Self-Tuning Memory Manager). El proceso db2stmm supervisa continuamente la carga de trabajo y ajusta automáticamente los parámetros de configuración de la memoria para optimizar el rendimiento.
- **db2taskd** → Se encarga de la distribución de tareas de base de datos en segundo plano. Las tareas son ejecutadas por las hebras db2taskp. El proceso db2taskd coordina la distribución de estas tareas entre las hebras db2taskp para garantizar una ejecución eficiente y en paralelo.
- **db2wlmd** → Realiza la recopilación automática de estadísticas de gestión de carga de trabajo. En un entorno donde la carga de trabajo de la base de datos varía, el proceso db2wlmd puede recopilar automáticamente estadísticas sobre el rendimiento y la utilización de recursos.
- **db2dbctrlid** → Se utiliza para la recopilación de estadísticas de gestión de carga de trabajo de forma automática
- **db2Xlnot** → Maneja la invalidación de Xi en un clúster pureScale.

A continuación, mostraremos la estructuración de las hebras del supervisor de sucesos. Estas hebras realizan tareas como diversas funciones de supervisión y gestión de eventos en la base de datos. Algunos ejemplos podrían ser: supervisado de eventos globales del sistema, eventos locales no coordinados, etc.

Estas hebras se identifican de la forma siguiente:

- **db2evm%1%2(%3)**

donde %1 puede ser:

- 1- **g** → Supervisor de sucesos de archivo global
- 2- **gp** → Supervisor de sucesos de canalizado global
- 3- **l** → Supervisor de sucesos de archivo local
- 4- **lp** → ... canalizado local
- 5- **t** → ... de tabla

%2 puede ser:

- 1- **i** → coordinador
- 2- **p** → no coordinador

y %3 es el nombre del supervisor de procesos

Estas hebras podrían estar monitoreando y respondiendo a eventos específicos, como cambios en la configuración o alertas de rendimiento.

Por otro lado, las hebras de copia de seguridad y restauración proporcionan la infraestructura necesaria para llevar a cabo estas tareas críticas. Estas hebras realizan funciones específicas durante los procesos de copia de seguridad y restauración, y su identificación sigue un patrón específico. El patrón es el siguiente:

- **db2bm.%1.%2** → Es el manipulador de almacenamiento intermedio de copia de seguridad y restauración
- **db2med.%1.%2** → Es el controlador de copia de seguridad y restauración

En este caso, %1 y %2 son:

%1 → identificador de EDU del agente que controla la sesión de copia de seguridad o restauración

%2 → valor secuencial que se utiliza para distinguir entre hebras que pertenecen a una sesión de copia de seguridad o restauración determinada

Estas hebras son esenciales para la integridad y la disponibilidad de los datos en un entorno de base de datos y trabajan en conjunto para garantizar que las operaciones de copia de seguridad y restauración se realicen de manera eficiente y confiable.

Por último, mencionaremos algunas EDU que son usadas para bloquear un entorno Db2 pureScale. Estas son opcionales ya que, si no se tiene un entorno de este tipo, es evidente que no son procesos estrictamente necesarios.

-
- **Db2LLMn** → Maneja las notificaciones de bloqueo desde las estructuras GLM/GCL del proceso CF (Coordinador de Fila) en un clúster pureScale. Cuando se realiza un bloqueo local en un miembro del clúster, este proceso asegura que las notificaciones de bloqueo sean procesadas y gestionadas adecuadamente.
 - **Db2LLMn1** → Procesa la información enviada por el gestor de bloqueo global. Hay dos de estas EDUs en cada miembro, una para el CF primario y otra para el CF secundario. Coordina y procesa la información sobre bloqueos globales, asegurando la coherencia y sincronización entre los nodos primarios y secundarios del clúster.
 - **Db2LLMn2** → Procesa la información enviada por el gestor de bloqueo global para un tipo especial de bloqueo utilizado durante el proceso de activación y desactivación de la base de datos. También hay dos de estas EDUs en cada miembro, una para el primario y otra para el secundario.
Maneja de manera especializada los bloqueos durante operaciones críticas como la activación y desactivación de la base de datos.
 - **Db2LLMng** → Garantiza que los bloqueos retenidos por este miembro se liberen de manera puntual cuando otros miembros están a la espera de estos bloqueos. Controla la liberación oportuna de bloqueos locales retenidos por un miembro.
 - **Db2LLMrI** → Usado en el procesamiento de la liberación de bloqueos en el gestor de bloqueos global. Se encarga de gestionar la liberación de bloqueos globales, asegurando la consistencia y la coordinación en todo el clúster.
 - **Db2LLMrc** → Se usa para el proceso que ocurre durante las operaciones de recuperación de la base de datos y durante la recuperación de CF (Coordinador de Fila). Se asegura de gestionar la lógica de bloqueo durante operaciones críticas de recuperación.
 - **Db2LLMrf** → Maneja la lógica de resincronización de notificación de bloqueo con el proceso CF en caso de que se descarte y restablezca la conexión en un clúster pureScale. La finalidad es asegurar la coordinación de la resincronización de notificaciones de bloqueo en situaciones donde la conexión se ha descartado y restablecido.

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>