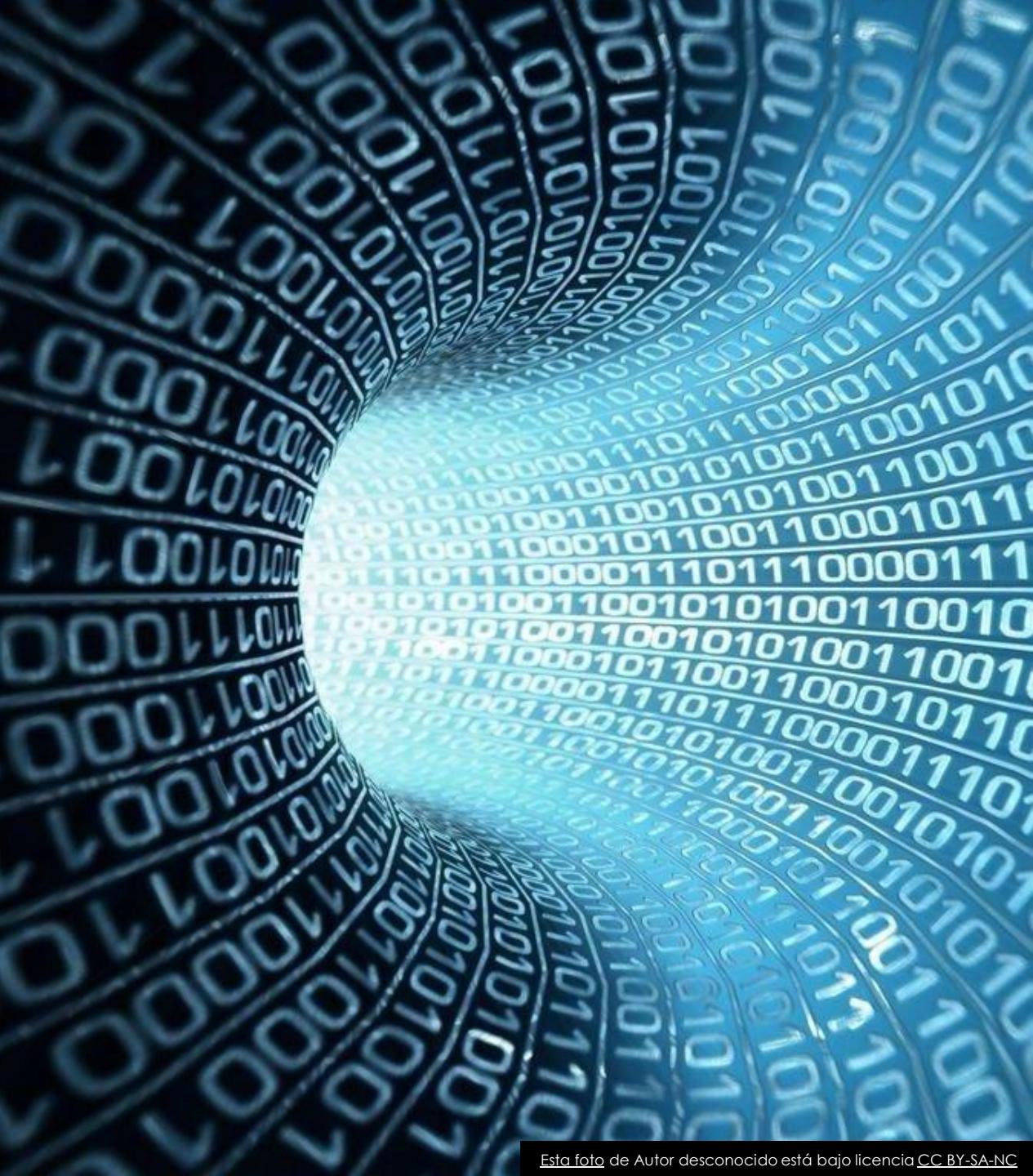


Bases de datos 2

Clase 2

Kevin Morán

bases2.fiusac@gmail.com



ARQUITECTURA DE

ORACLE®

DATABASE

Comunicación con los servidores Oracle Database

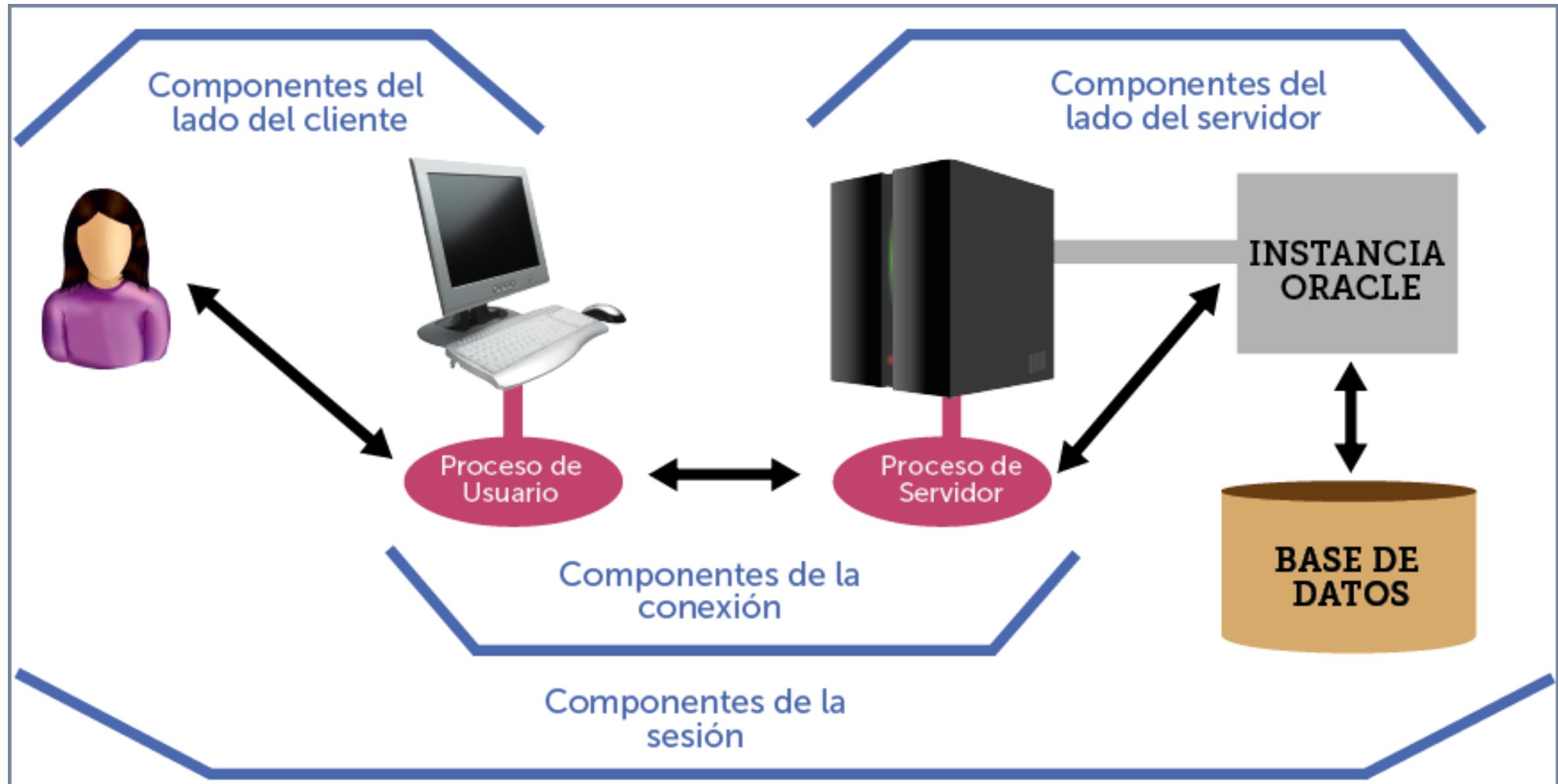


Elementos de comunicación con un servidor Oracle

La comunicación entre el cliente y el servidor se realiza a través de dos procesos:

- **Proceso de usuario.** Software que se ejecuta en el lado del cliente y se encarga de recoger las instrucciones lanzadas por el usuario y enviarlas al servidor.
- **Proceso servidor.** Software que se ejecuta en el servidor de bases de datos y que se encarga de procesar el código lanzado por el usuario.

Normalmente, hay un proceso servidor para cada usuario que conecte con la base de datos. Es decir, si hay diez conexiones, habrá diez procesos de usuario y diez procesos servidores.



Sesión y conexión

Elementos

Conexión. Mecanismo de comunicación entre el lado del cliente y el lado del servidor. A los extremos de esa comunicación se encuentran los procesos de usuario y de servidor.

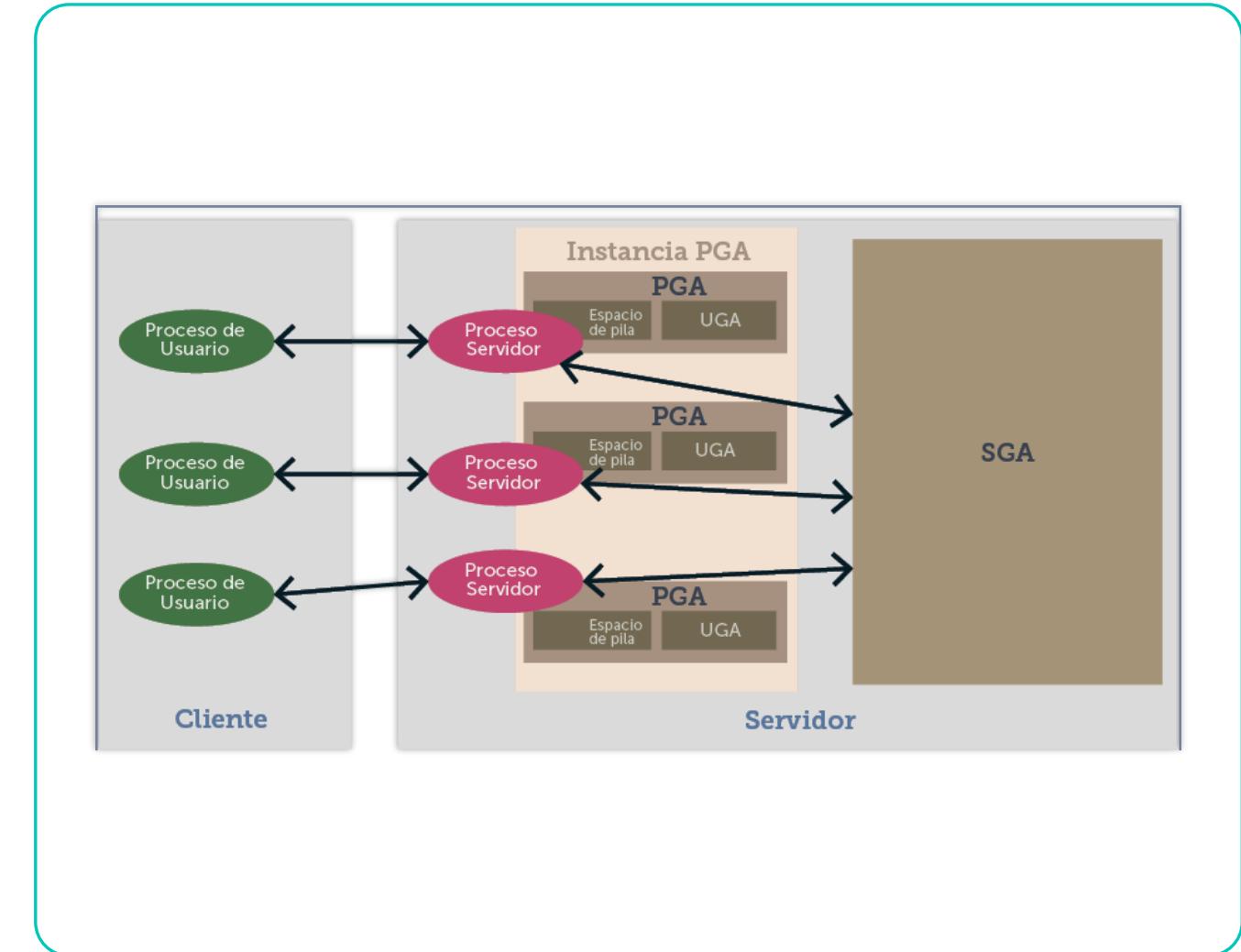
Sesión. Abarca la comunicación desde el mismo cliente hasta llegar a la base de datos. Una sesión requiere autentificar al usuario y otorgarle los recursos necesarios para una correcta comunicación. La sesión finaliza cuando el usuario abandona la aplicación de usuario o bien cuando desconecta. Un mismo usuario puede establecer varias sesiones (normalmente).

Modos de servidor

Dedicado y compartido

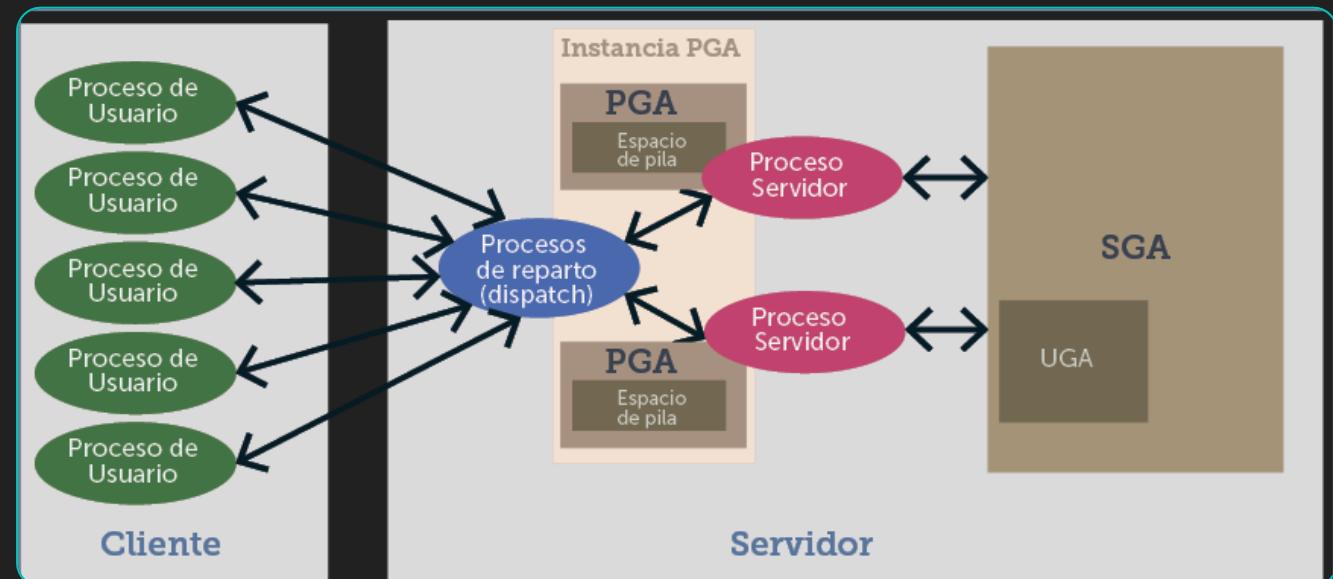
Servidor dedicado

- En principio, la forma de trabajar de Oracle Database es la que se conoce como modo de **servidor dedicado**. En ella por cada proceso servidor atiende a un único proceso de usuario. Dicho de otro modo, hay tantos procesos servidores como procesos de usuario.



Servidor compartido

- Existe la posibilidad de trabajar en modo de **servidor compartido**. En este caso cada proceso servidor atiende a varios procesos de usuario. Uno, o más, procesos, llamados **dispatchers** (repartidores), se encargan de asignar a cada proceso de usuario el proceso servidor adecuado.
- En este modo se ahorra memoria, ya que la memoria de usuarios se almacena en la zona global compartida (llamada SGA).



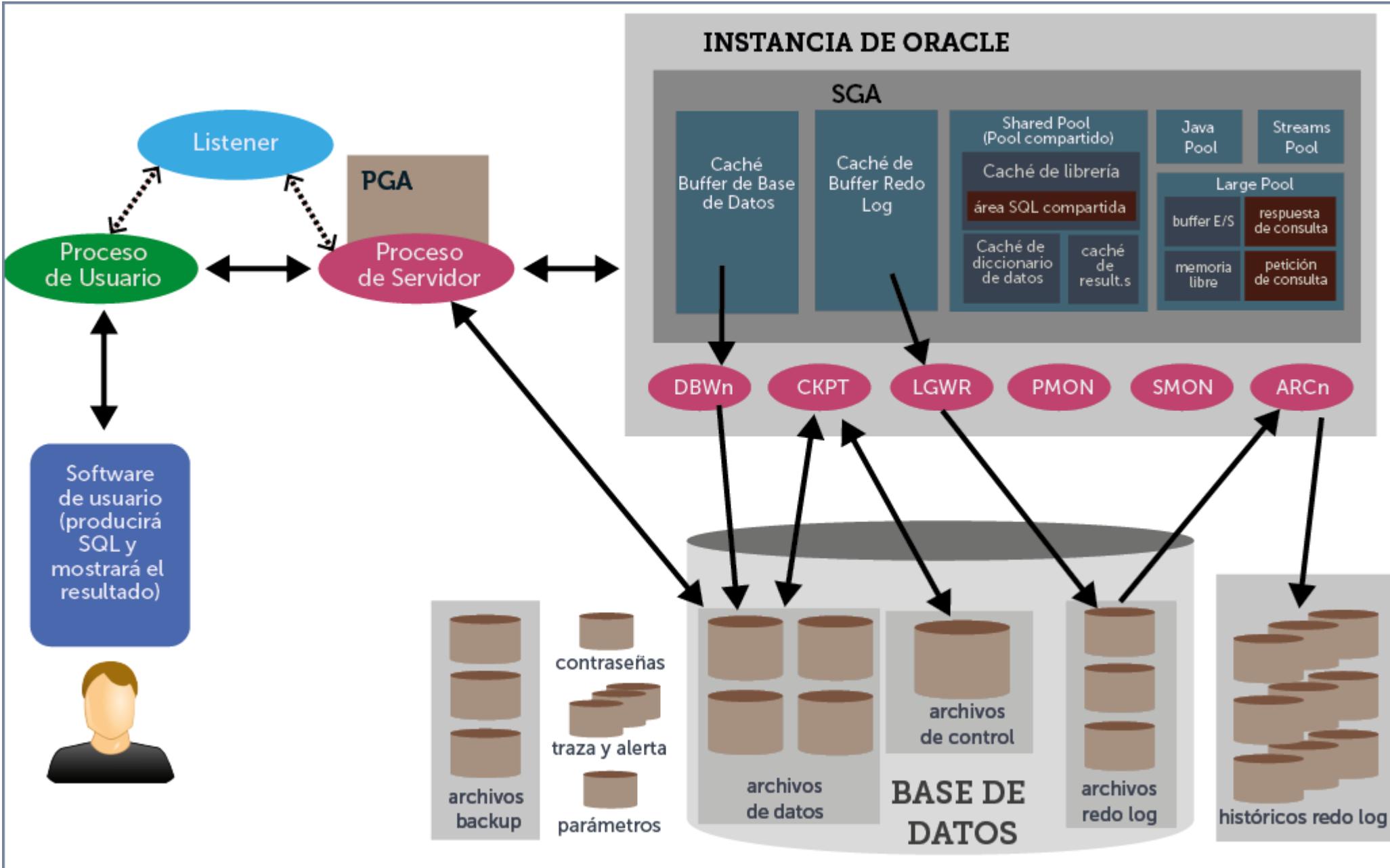
Establecimiento de la conexión

- La conexión típica a Oracle comienza con una petición de acceso desde el lado del cliente.
- Un proceso conocido como **Listener**, consigue “escuchar” dicha petición. El Listener es uno de los elementos fundamentales de Oracle Database. Su labor es gestionar el tráfico de las peticiones del cliente.
- Una vez el Listener detecta la nueva petición, se establece conexión y comenzarán a comunicarse el proceso de usuario con su proceso servidor correspondiente.
- El Listener se mantiene escuchando la comunicación.

Funcionamiento de la instancia de Oracle



Arquitectura general de Oracle Database

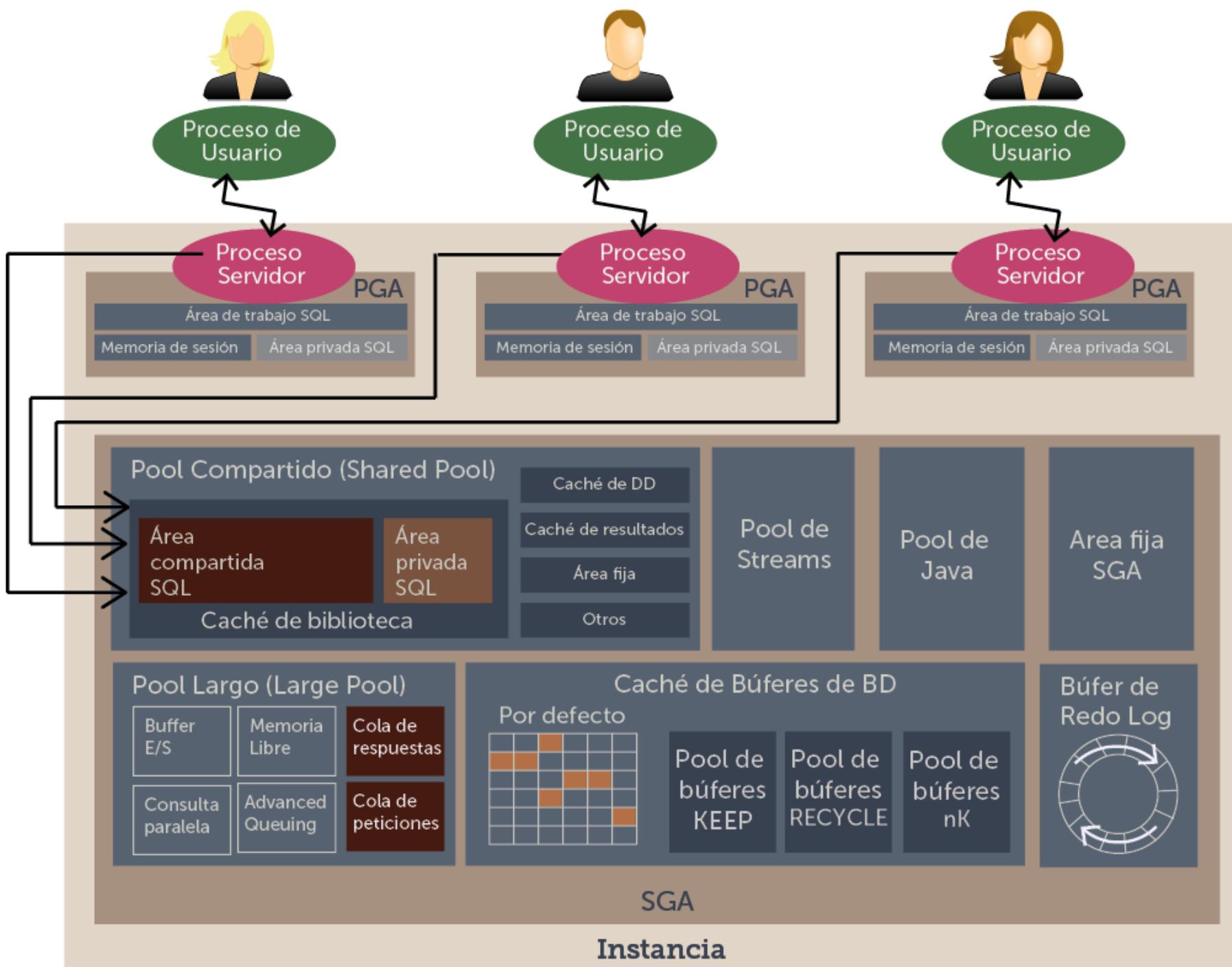


Un servidor Oracle Database es el conjunto formado por estos dos elementos:

- **La instancia de Oracle.** Formada por el conjunto de procesos y las estructuras de datos en memoria que requiere el servidor cuando está en funcionamiento.
- **Archivos de la base de datos.** Los archivos en disco que almacenan de forma permanente la información de la base de datos. La base de datos en sí, la forman los **archivos de datos**, los de **control** y los **Redo Log**.

Estructuras en memoria de la instancia de Oracle

Elementos de la memoria

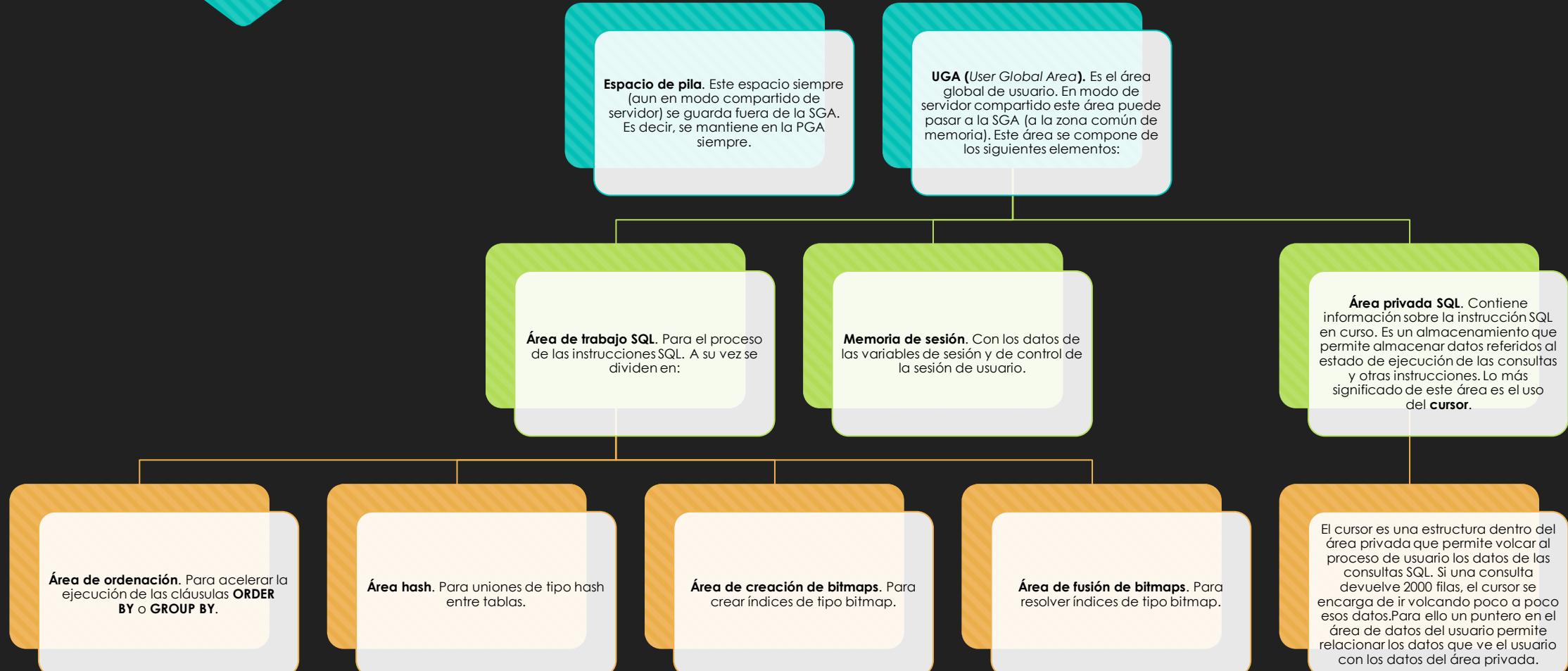


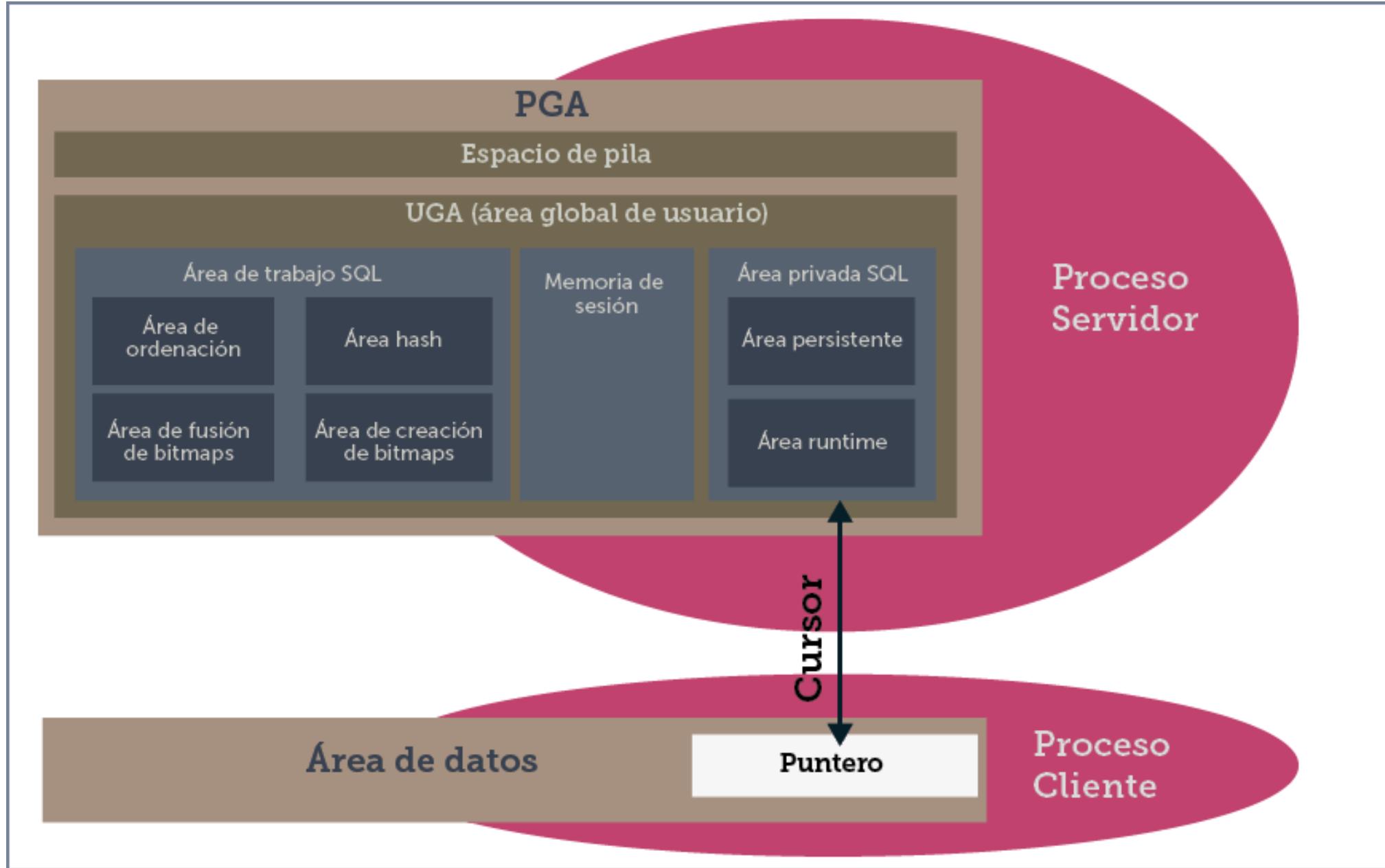
Principales estructuras de almacenamiento

SGA (Server Global Area). Zona de la memoria en la que se guardan los datos globales de la instancia. Esos datos son los que comparten todos los procesos servidores, por lo que la mayoría de sus componentes son memorias de tipo caché. Muchas de sus áreas llevan el nombre de Pool, término inglés que, en este contexto, puede traducirse como fondo. En el sentido de un espacio en el que se reservan activos.

PGA (Program Global Area). Zona de la memoria en la que se guardan los datos referentes a un proceso servidor concreto. Si el modo de trabajo es dedicado, si hay 5 conexiones habrá 5 procesos servidores y, por lo tanto, 5 PGAs. Al conjunto de todas las PGAs en uso, en un momento dado, se le llama **instancia PGA**. El tamaño de la instancia PGA se puede calibrar dentro de las opciones de configuración.

Componentes de la PGA





Componentes de la SGA

Pool Compartido (Shared Pool)

Se trata de una zona de memoria utilizada para acelerar la ejecución de las instrucciones SQL y PL/SQL.

- **Caché de biblioteca** (*Library Cache*). Almacena código ejecutable de SQL y PL/SQL. Siempre que se ejecuta una nueva instrucción (SQL o PL/SQL), se comprueba si la ejecución de la misma está disponible en este área. Dentro de este caché tenemos:
 - **Área SQL compartida** (*Shared SQL area*). Contiene el plan de ejecución de la instrucción y el árbol de análisis de la misma. De esta forma se ahorra memoria cuando se repiten las instrucciones. También los elementos PL/SQL son cacheados de esta forma.
 - **Área SQL privada**. Normalmente los datos privados de los usuarios, referidos a información sobre su sesión, necesarios para que una instrucción se ejecute correctamente, se almacenan en la PGA (como se puede comprobar en el apartado anterior). Pero en el modo compartido de servidor, se almacena en la SGA, concretamente en la caché de biblioteca en una zona separada del área de SQL compartida.
- **Caché de Diccionario de Datos** (*Data Dictionary Cache*). Se la conoce también como **caché de fila** ya que sus datos se almacenan en forma de fila (en el resto de cachés se almacenan en forma de bloques). Contiene información para acelerar el acceso a los metadatos que utilizan las instrucciones (también en la Caché de Biblioteca se guarda información sobre metadatos).
- **Caché de resultados** (*Results cache*). Tanto para SQL como para PL/SQL, almacena fragmentos de consultas SQL y PL/SQL para aprovecharlos en consultas posteriores. El parámetro de sistema **RESULT_CACHE_MODE** permite establecer si todas las consultas almacenarán fragmentos en caché o sólo las marcadas de forma específica.
- **Área fija**. Es una zona pequeña donde se almacenan los datos necesarios para la carga inicial de la SGA en memoria.

Pool Compartido (*Shared Pool*)

Caché de biblioteca (*Library Cache*)

Área Compartida SQL (*Shared SQL Area*)

Planes de ejecución y árbol de procesamiento de instrucciones para SQL y PL/SQL

Área privada SQL

Solo funciona en modo compartido de servidor. Contiene las estructuras privadas de cada sesión

Caché de Diccionario de Datos (*Data Dictionary Cache*)

Datos del Diccionario almacenados en filas

Caché de resultados (*Server Result Cache*)

Caché de resultados de SQL

Caché de resultados de PL/SQL

Área fija (*Reserved Pool*)

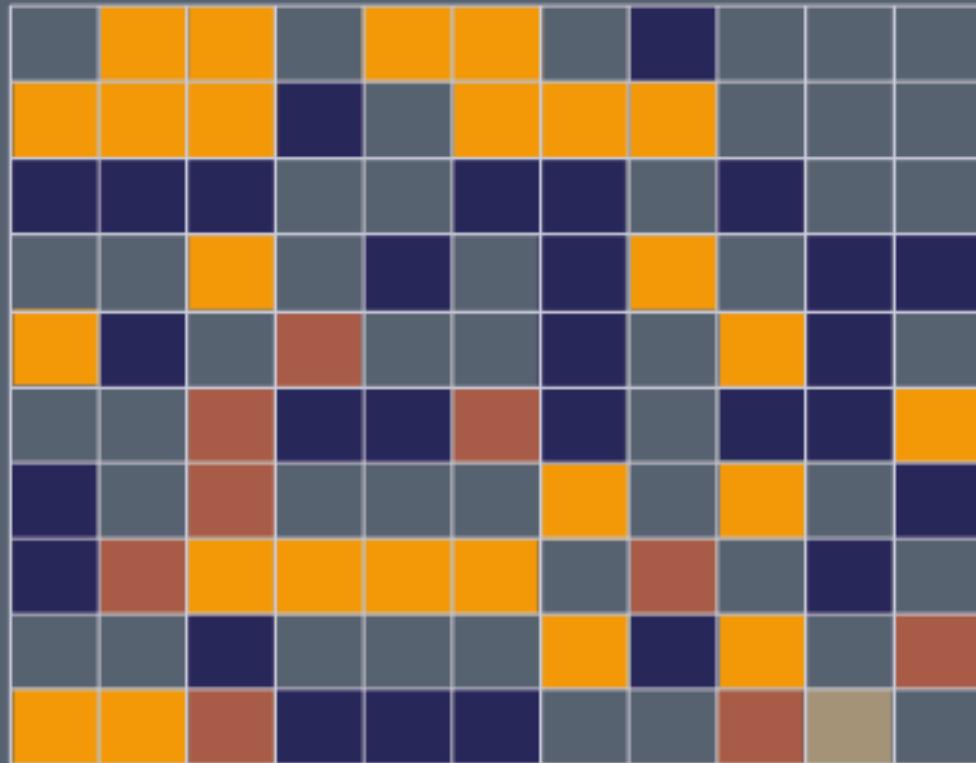
Otros

Caché de búferes de datos (Database Buffer Cache)

- Está dividida en bloques (más adelante se explica lo que es un bloque de base de datos) y es la estructura (normalmente) que más ocupa en la SGA. Oracle Database intenta que la modificación de datos en el disco tarde lo menos posible. La estrategia principal es escribir lo menos posible en el disco.
- Por ello cuando una instrucción provoca modificar datos, inicialmente esos datos se graban en esta caché. La grabación ocurre tras la confirmación de una transacción. Después, cada cierto tiempo, se grabarán todos de golpe en los ficheros de datos (concretamente cuando ocurra un *checkpoint*).

Caché de Búferes de la Base de Datos (Database Buffer Cache)

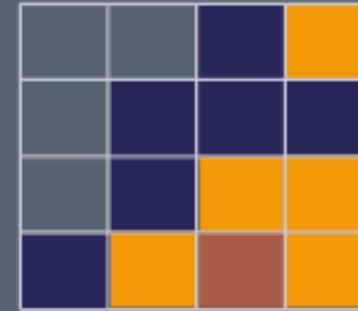
Pool por defecto



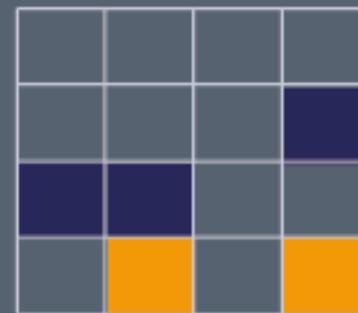
Pool KEEP



Pool RECYCLE



Pool 2K



Pool 4K



- Pinned
- Sucio
- Limpio
- Sin uso

Los datos, aunque no estén grabados realmente en disco, serán ya permanentes y aparecerán en las consultas que se realicen sobre ellos, además esas consultas serán más rápidas al no acceder al disco. Esto último es la segunda idea, que los datos a los que se accede más a menudo, estén en memoria y no se necesite leerlos del disco.

Los búferes de la caché se asignan con un complejo algoritmo basado en **LRU** (*Last Recently Used*) que da prioridad a los búferes que se han utilizado más recientemente.

Cada búfer puede estar en uno de estos estados

- **Sin uso** (*Unused*). Son bloques que no se están utilizando actualmente, por lo que están libres para cualquier proceso que requiere almacenar datos en ellos.
- **Limpios** (*Clean*). Son búferes que se han utilizado, pero cuyos datos ya están grabados en disco. El siguiente *checkpoint* no necesitaría grabar estos datos, por lo que están disponibles si se requiere su reutilización.
- **Sucios** (*Dirty*). Contienen datos que no se han grabado en disco. Se deben de grabar en el disco en cuanto ocurra un *checkpoint*, de otro modo podríamos perder información.

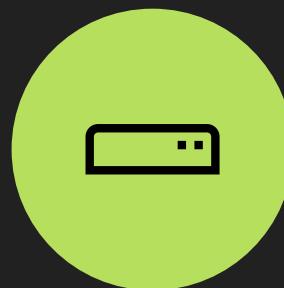
Además, con respecto al acceso, los búferes pueden tener estas dos situaciones:

- **Libres** (*Free* o *Unpinned*). Ningún proceso le está utilizando.
- **Pinned**. Están siendo utilizados por un proceso, por lo que otra sesión no puede acceder a este búfer.

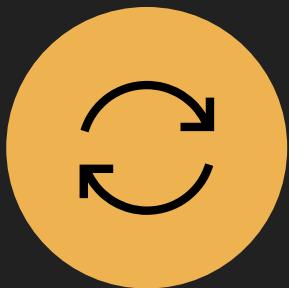
La lectura de búferes sigue este proceso:



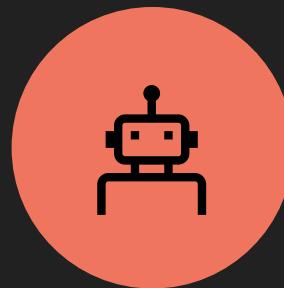
[1] Si necesitamos un dato, se le busca en esta caché. Si se encuentra se entrega, siempre y cuando no esté ocupado por otro proceso (*pinned*).



[2] Si el dato requerido no se encuentra en el búfer, ocurre un **fallo de caché**. Este fallo implica leer los datos del disco y pasarlo a memoria (los búferes necesarios se marcarán como búferes limpios).



[3] En ambos casos, si la instrucción implica modificar los datos de ese búfer, se modifica y pasa a ser un búfer sucio.



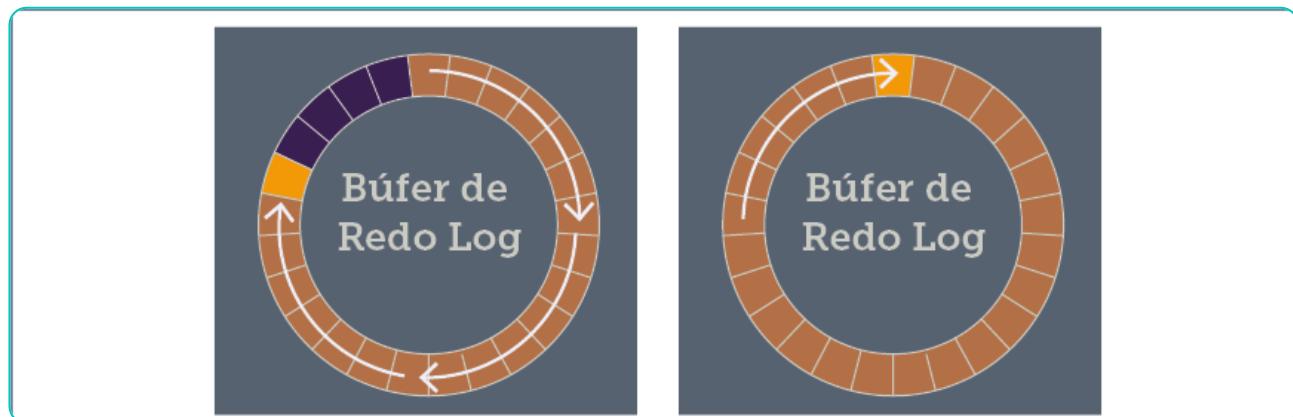
[4] Cuando ocurre un checkpoint, el proceso **DBWn**, graba búferes sucios a disco (lo mismo si se detecta que quedan pocos búferes limpios o sin uso) y se marcan como búferes limpios.

Fondos de almacenamiento (Pools)

- **Pool por defecto.** Es el de uso normal. Normalmente es el único, salvo que se especifique el uso de otros.
- **Pool KEEP.** Se utiliza para bloques de uso muy frecuente, pero que dentro del fondo por defecto podrían ser reutilizados continuamente si no disponemos de espacio suficiente.
- **Pool RECYCLE.** Se utiliza para bloques de uso muy poco frecuente. El uso de este pool previene de usar estos bloques en el pool por defecto.
- **Pools nK.** El tamaño de bloque es una cuestión importante en una base de datos (más adelante se especificará lo que es un bloque). Aunque la base de datos tiene un tamaño de bloque por defecto, en algunos tablespaces (elemento en el que se guardan los objetos, se describe más adelante), podríamos variar ese tamaño. Por ello podríamos tener otros cachés orientados a estos elementos de tamaño especial.

Búfer de Redo Log

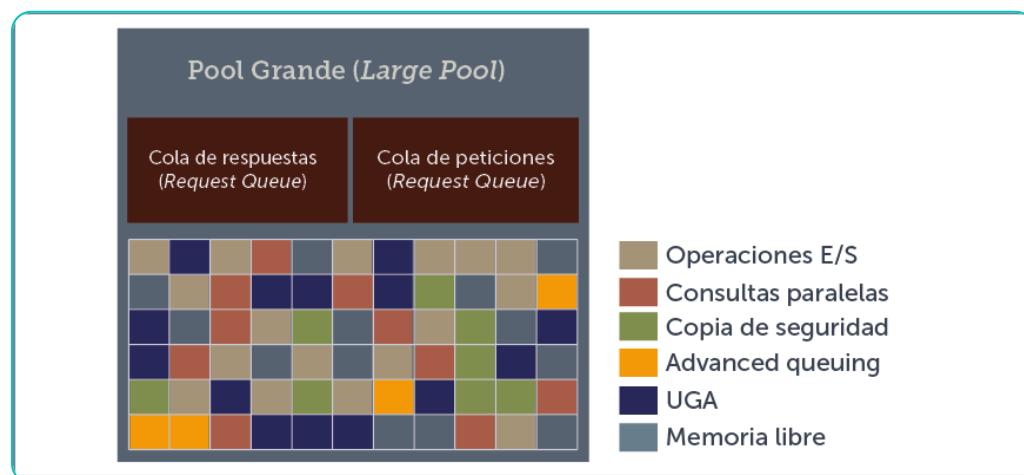
- Se trata de un búfer circular. Se van utilizando los bloques y, cuando se han usado todos, se vuelven a reutilizar los primeros en caso necesario y así continuamente.
- Su función es almacenar la información acerca de los últimos cambios (DML confirmados y DDL) realizados sobre la base de datos. Esta información se vuela continuamente, mediante el proceso **LGWR**, a los archivos de datos.
- Los Redo Log son necesarios para recuperar los datos que no han podido ser grabados definitivamente en disco (normalmente por ocurrir una situación de excepción antes de un CHECKPOINT).



Pool Grande (Large Pool)

Área opcional de la SGA que proporciona espacio para los datos necesarios para realizar operaciones que impliquen muchos datos:

- **Backup y restauración**, para copias de seguridad.
- **Procesos de Entrada/Salida del servidor**.
- **Memoria libre**. Para aliviar el trabajo de la instancia.
- **Consultas en paralelo**.
- **Almacenamiento de la UGA**. Normalmente la UGA se almacena en la PGA, pero en modo de servidor compartido se podría almacenar en este área.
- **Cola de peticiones**. Especialmente importante cuando el servidor recibe numerosas peticiones. No se almacenan bloques de datos en este caso.
- **Cola de respuestas**. Tampoco utiliza bloques, sino estructuras más apropiadas para las colas.



Procesos de memoria de Oracle

Tipos de procesos

- **Proceso de usuario.**
- **Procesos de bases de datos.** Compuestos por:
 - **Proceso servidor.**
 - **Procesos en segundo plano.**
- **Procesos de la aplicación.**
Son **demonios** (*daemons*), servicios residentes que se ejecutan de forma automática. El más conocido es el *listener* de red.

Tareas de los procesos servidores

Los procesos servidores atienden a los procesos de usuario. Sus labores fundamentales son:

- Analizar el código SQL o PL/SQL lanzado por el proceso de usuario
- Establecer el plan de ejecución de la consulta SQL lanzada por el usuario
- Leer los datos requeridos por ese código en la memoria SGA o en el disco (si no están en la SGA).
- Devolver los resultados al proceso de usuario o indicarle, si es el caso, el error producido.

Procesos en segundo plano

Los procesos en segundo plano se ejecutan al lanzar la instancia de Oracle y quedan residentes en memoria realizando diversas labores en el servidor. La vista **V\$DBPROCESS** permite obtener información de los procesos en memoria.

DBWn

- El **proceso de escritura de base de datos** (*DataBase Writer Process*) Escribe los bloques modificados del buffer de cache de la base de datos (*dirty buffers*) de la caché de búferes de datos de la SGA a los archivo de datos en disco. Eso no ocurre en todo momento, sino cuando se produce un evento de tipo **checkpoint**.
- El hecho de que esto se haga solo cada cierto tiempo (el tiempo establecido para el checkpoint) se debe a que, de otro modo, el funcionamiento del servidor sería muy lento si se accediera más a menudo al disco.
- El parámetro de sistema DB_WRITER_PROCESSES se encarga de definir el número de procesos DBWn.
- Existe checkpoint incremental y total

El DBW actúa si:

- **No hay búferes libres** en el búfer de datos de la SGA para seguir almacenando información; es decir, la caché de datos está a tope de ocupación. Si esto ocurre, Oracle escribe unos cuantos bloques sucios en los archivos de datos para poder liberar la ocupación de la memoria y disponer de más búferes libres.
- **Hay demasiados bloques sucios.** En ese caso se escriben unos cuantos bloques sucios en disco para liberar espacio. Es un caso previo al anterior que se basa en una medida de ocupación preventiva para no esperar a que no haya búferes libres, que sería más dramático.
- **Se cumplió el tiempo de tres segundos de espera.** En una base de datos ocupada, este evento no se cumple jamás. Ocurre cuando el sistema está sin hacer nada durante tres segundos. En ese caso se escriben unos cuantos bloques sucios. Si la base de datos sigue desocupada los siguientes tres segundos, se escriben otros tantos bloques. Así poco a poco (mientras la base de datos siga desocupada) acabará volcando todos los búferes en disco.
- La aparición de un evento de **checkpoint** (lo que hemos denominado un checkpoint total). Esto provoca escribir absolutamente todos los bloques sucios. Es el momento de máximo trabajo en disco y, por lo tanto, el momento más crítico porque en un checkpoint se pueden tener que grabar miles de datos. Por ello, Oracle inicialmente pone el parámetro que controla el tiempo de aparición de un checkpoint a cero (que es lo mismo que tiempo infinito).

LGWR

Proceso de escritura de logs (*LoG WriteR Process*). Proceso encargado de escribir en los archivos redo log. Escribe los datos del búfer de Redo Log (en la SGA) a los archivos Redo Log en disco. Lo hace cuando:

- **Se confirma una transacción.** En ese caso LGWR incluye una entrada de confirmación en el búfer Redo Log. Así quedará marcado que, en ese momento, los archivos Redo Log están al día. Hasta no grabar esos datos, no se retorna al usuario la confirmación de su transacción. A la transacción se le asigna un número de cambio de sistema (llamado **SCN**). Este SCN se graba en los archivos Redo Log. Este número es fundamental para recuperar los datos en caso de desastre.
- **Si se llena un tercio del buffer.** En este caso se graban datos sin confirmar. Solo se hace por temor a que se llene el búfer y perdamos información.
- **Justo antes de que actúe el proceso DBWn.** Antes de que el proceso DBW escriba búferes sucios en disco indica a LGWR que escriba los registros Redo Log en disco. Cuando confirma la escritura, entonces DBW actuará escribiendo en los archivos de datos..
- **Cada tres segundos.**

CKPT

- Proceso encargado de registrar la llegada de un *checkpoint*, momento en el que se graban búferes de datos en los archivos de datos. CKPT graba en los archivos de control la posición de checkpoint y el SCN correspondiente en los Red Log.
- También graba en los archivos de datos (en la cabecera) para actualizarla con la información del punto de control y la información sobre el último SCN que se ha grabado.
- Además comunica a DBW la necesidad de grabar los búferes sucios.

SMON

System Monitor. Proceso encargado de monitorizar el sistema. Sus principales labores son:

- Recuperar la instancia al iniciar.
- Limpiar segmentos temporales que ya no se usan.
- Recuperar los datos de las transacciones existentes en los Redo Log que no están grabadas en los archivos de datos durante el arranque de la instancia.

SMON puede ser invocado por otros procesos.

PMON

Process Monitor. Se encarga de gestionar el fallo en un proceso de usuario. Sus labores son:

- Comprobar el estado de los procesos servidores y distribuidores (*dispatchers*).
- Reiniciar procesos servidores y distribuidores que hayan fallado.
- Si falla un proceso de usuario
 - Limpia la caché de búferes de datos
 - Libera los recursos que se utilizaron para atender a ese proceso de usuario

RECO

- Se usa solo en bases de datos distribuidas. Resuelve los fallos ocurridos en transacciones distribuidas. Cuando detecta fallos en una transacción que tiene diferentes datos en los distintos servidores, se encarga de resolver la situación.

MMON y MMNL

- MMON es el proceso monitor de manejabilidad (*manageability monitor process*), encargado de realizar tareas relacionadas con el **AWR**, área de volcado de estadísticas de los servidores Oracle.
- MMNL es el proceso ligero de monitorización de manejabilidad (*manageability monitor lite process*), encargado de escribir estadísticas desde el histórico de sesiones activas (**ASH**) en la SGA de Oracle a el disco.
- Ambos son procesos necesarios para que la información estadística sobre la ejecución de Oracle esté al día.

ARCn

- Procesos de archivado (*archiver processes*), encargado de escribir los archivos redo log históricos. Estos archivos son copias de los archivos Red Log. Se usan para recuperar información o para devolver la base de datos a un estado anterior.
- Solo funcionan en modo **ARCHIVELOG** de la base de datos.
- La **n** indica que pueden ser varios procesos (ARC0, ARC1, etc.).

CJQ0 y Jnnn

- Es el gestor de colas de trabajo (*job queue processes*). Los trabajos son tareas programadas por los usuarios que se pueden ejecutar varias veces.
- Cada trabajo se asocia a un proceso **Jnnn** (por ejemplo *J001*).
- El programador de tareas de Oracle puede invocar a CJQn cuando se necesitan ejecutar trabajos. Entonces CJQ0 lanza los procesos Jnnn de forma apropiada. Cuando finalizan los trabajos, CJQ0 se queda en estado de espera, hasta que se le vuelva a necesitar.

FBDA

- El *FlashBack Data Archiver Process*, es el proceso encargado de grabar la información del área de Flashback. Esta área se usa para el caso de necesitar que la base de datos regrese a un estado anterior.

Estructuras de almacenamiento en disco



Una base de datos Oracle necesita los siguientes archivos para grabar la información de la misma:

- Archivos de datos
- Archivos de control
- Archivos Red Log

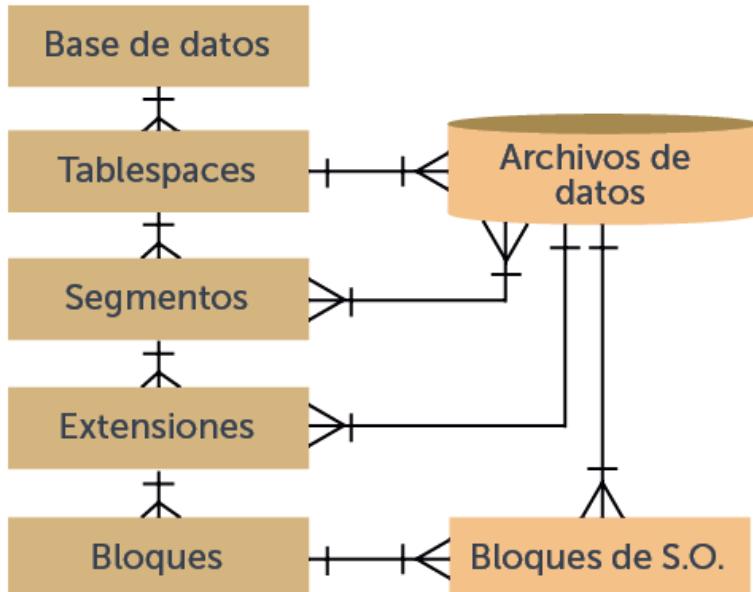
Además posee (o puede poseer) estos otros archivos para que la ejecución sea correcta:

- Archivos de parámetros
- Archivo de contraseña
- Archivos de traza y alerta
- Archivos históricos (o archivados) Red Log
- Archivos de copia de seguridad

Archivos de datos

Son los archivos que almacenan los datos en sí de la base de datos. Graban la información de las tablas. Para optimizar su funcionamiento y su gestión, Oracle utiliza una estructura que relaciona la lógica de la base de datos, con la parte física.

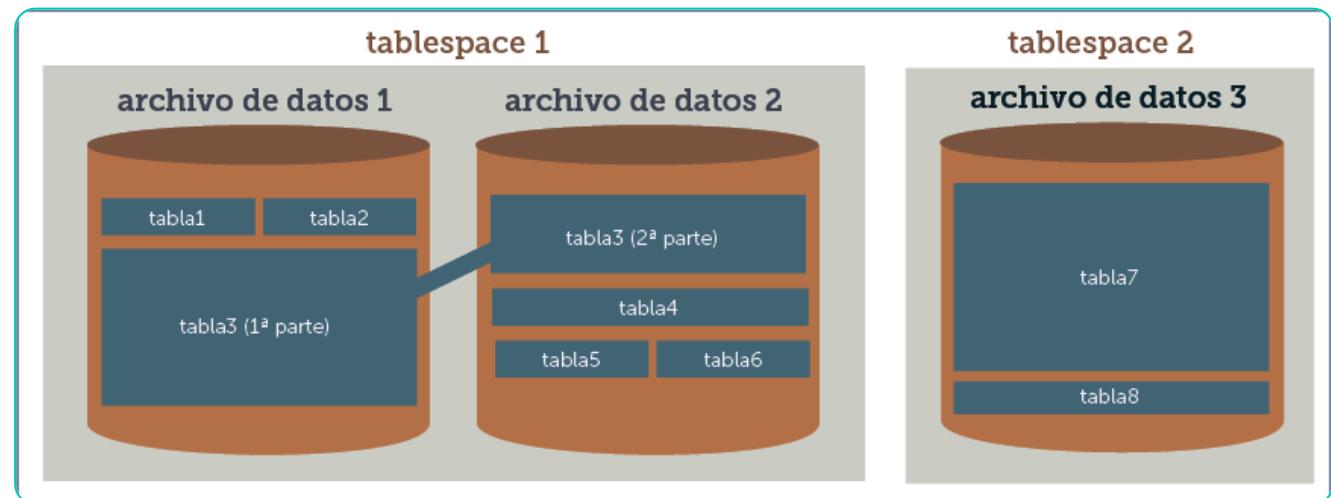
Estructuras lógicas de almacenamiento de Oracle



- Oracle llama a estas estructuras *Logical Storage Structures*, por lo que en español se suele traducir como estructuras lógicas de almacenamiento.

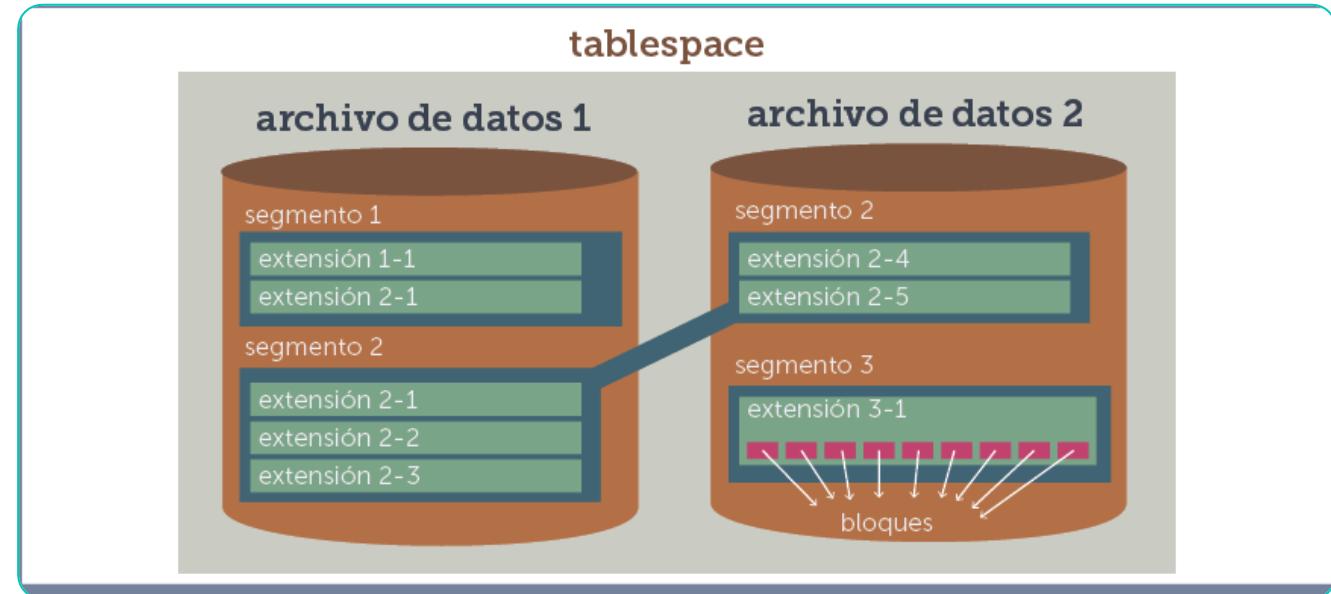
Tablespaces

- En la base de datos, se manejan objetos a nivel lógico (tablas, columnas, filas, vistas, índices,...). La información de esos objetos se tiene que almacenar en archivos de datos. Oracle crea los **tablespaces** como un elemento intermedio entre el nivel lógico y el nivel físico de la base de datos. Relaciona ambas ópticas para optimizar el funcionamiento del sistema.



Por defecto Oracle proporciona los siguientes espacios de tabla:

- **USERS**. Almacén por defecto en el que los diferentes usuarios de la base de datos almacenan sus objetos.
- **SYSTEM**. Para los objetos del sistema como el **diccionario de datos**
- **SYSAUX**. Para componentes adicionales de la base de datos como por ejemplo el repositorio del **Enterprise Manager**.



Segmentos

En cada tablespace existen segmentos, que están relacionados directamente con un objeto de la base de datos (una tabla, un índice,...). Hay tres tipos de segmentos:

- **Segmentos de usuario.** Almacenan los objetos de base de datos creados por los usuarios. Por ejemplo:
 - **Segmentos de datos.** Almacenan los datos de las tablas (incluidas las particionadas y las tablas en cluster, que son tipos de tablas avanzadas).
 - **Segmentos de índice.** Almacenan los datos necesarios para la creación de índices.
 - **Segmentos de tipo LOB.** Para almacenar los datos de tipos grandes de Oracle: CLOB y BLOB.
- **Segmentos de anulación (undo).** Almacena la información necesaria para revertir cambios realizados en los datos. El uso típico es la ejecución de la instrucción **ROLLBACK** (aunque hay muchos más).
- **Segmentos temporales.** Almacenan datos intermedios que Oracle necesita para completar consultas o instrucciones complejas que relacionan gran cantidad de datos (**CREATE INDEX**, **SELECT DISTINCT**, uniones, intersecciones, **GROUP BY**, etc.). Estos segmentos les crea y elimina Oracle automáticamente.

Consulta:

```
CREATE TABLE obras(titulo VARCHAR2(30));
```

Segmentos:

- Segmento para almacenar los datos de la tabla.

Consulta:

```
CREATE TABLE obras(  
    id_obra NUMBER PRIMARY KEY,  
    titulo VARCHAR2(30) UNIQUE,  
    texto CLOB  
);
```

Segmentos:

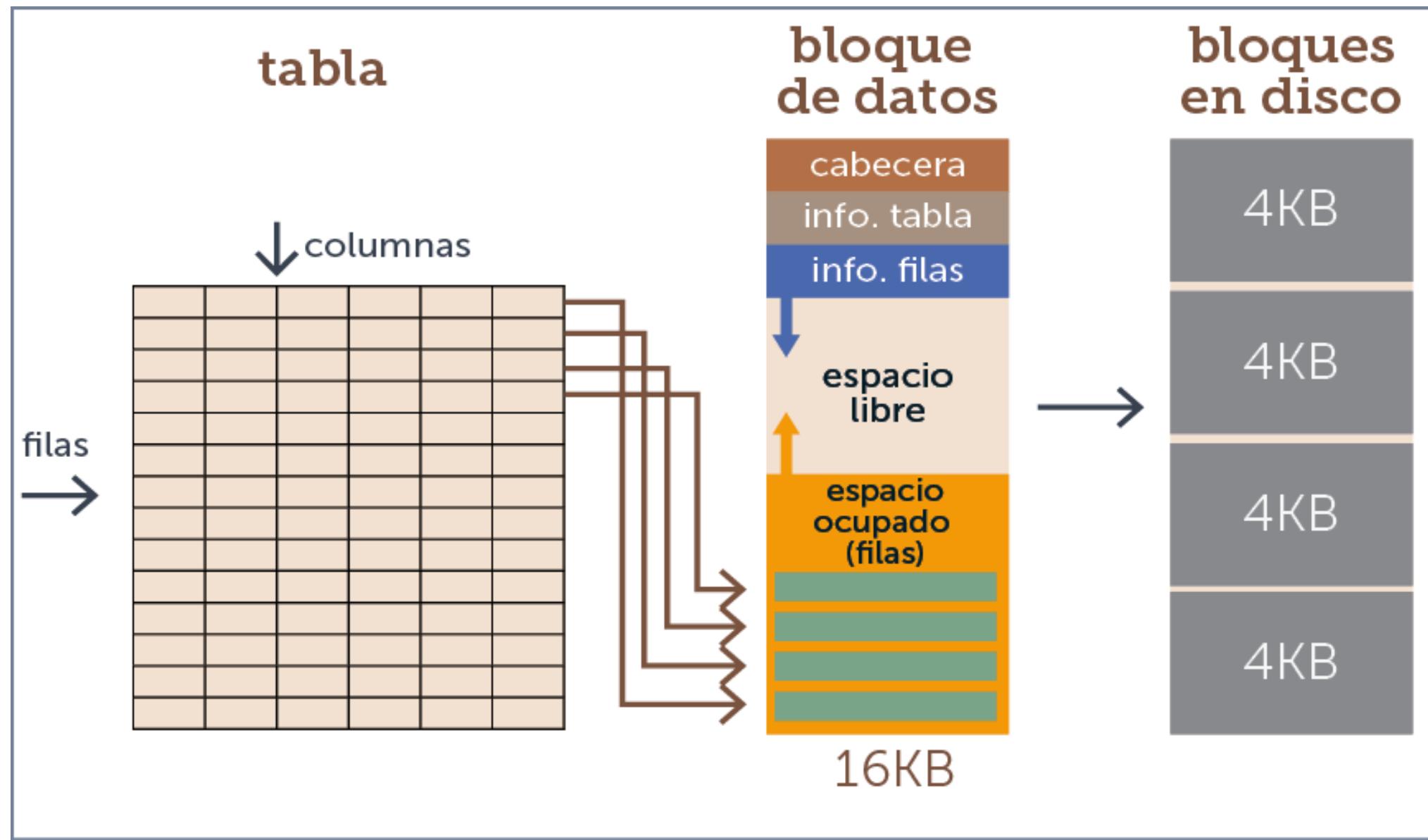
- Un segmento para los datos de la tabla *obras*.
- Otro para el índice de la clave principal.
- Otro para el índice de la restricción **UNIQUE** sobre la columna *título*.
- Otro para los datos CLOB.
- Otro, de tipo índice, para relacionar los datos CLOB con las filas de la tabla.

Extensiones

- Las extensiones son divisiones dentro del segmento que permiten asegurar que el sistema tiene reservado un conjunto de bloques contiguos en el disco. Es decir las extensiones evitan fragmentar en exceso los discos.
- El funcionamiento es el siguiente: cuando hemos llenado un segmento, entonces se añade una extensión (el tamaño de las extensiones se puede calibrar) y eso significa que se amplia el segmento en el tamaño de dicha extensión. Es decir, se reserva ese espacio.
- De esa forma los bloques siguientes (hasta llenar la extensión) que se añadan, tendremos la seguridad que irán contiguos en disco, disminuyendo su fragmentación.
- Como es lógico, una extensión está asociada a sólo un archivo.

Bloques de datos

- Es el elemento de datos más pequeño distinguible por Oracle. Cada extensión consta de una serie de bloques. El tamaño de bloque se puede configurar por parte del DBA para optimizar el rendimiento.
- El tamaño del bloque de datos debe cumplir que sea múltiplo del tamaño de bloque del disco del Sistema Operativo. Es decir si en un disco concreto, el Sistema Operativo tiene un tamaño de bloque de 16KB, sólo podremos asignar tamaños de bloque de 16 KB, 32 KB, 48 KB, 64 KB etc.
- En el bloque de datos de Oracle, los datos se organizan en filas. Así se asegura que cada fila se almacena junta en el bloque de datos. Además en cada bloque se graba una cabecera con información general del bloque, de la tabla a la que pertenece y de las filas que almacena.
- El espacio libre del bloque está en el interior, de modo que la cabecera y la información de las tablas crecen hacia el interior del bloque.

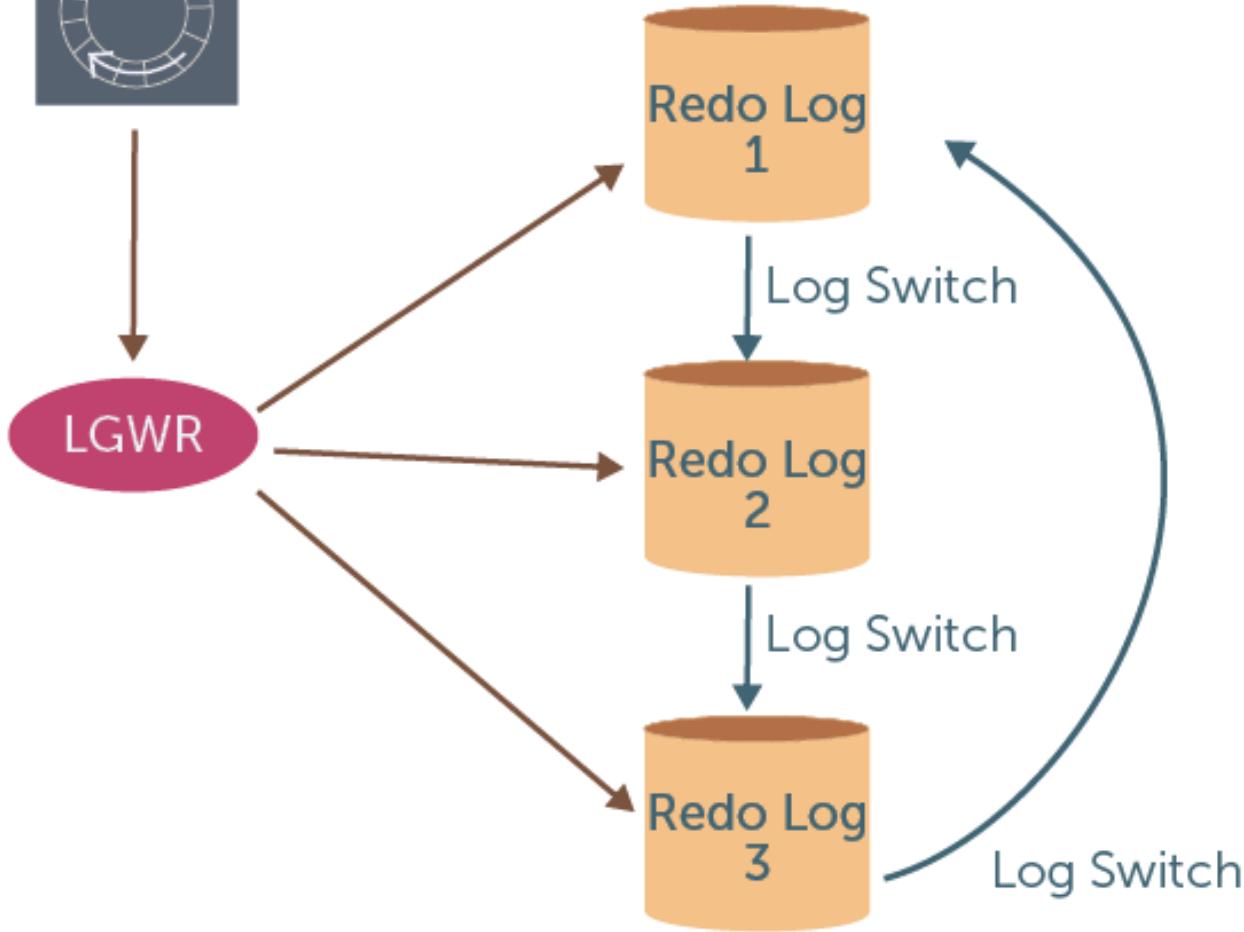


Posibilidades de gestión física de los archivos de datos

- **Como archivos normales del Sistema Operativo.** Es el caso más fácil y habitual, la gestión de los archivos se delega al sistema operativo. Lo que implica que se puedan copiar, borrar o examinar como cualquier otro archivo del sistema.
- **Como particiones RAW.** En ese caso todos los archivos de datos se integran en una estructura que, desde el sistema, parecería un archivo binario normal. La única finalidad es ocultar los archivos contenidos; en la práctica no se usa mucho.
- **Gestionadas mediante el Automatic Storage Management (**ASM**).** Se trata de un sistema de archivos propietario de Oracle que permite que la gestión de los archivos se realice desde una capa propia, en lugar de delegarla al Sistema Operativo.

Archivos Redo Log

- Los Redo Log son dos o más archivos que sirven para almacenar las instrucciones DML que se van confirmando en la base de datos. No graban datos, sino la información necesaria para que esa instrucción se lleve a cabo.
- Sirven para recuperar los datos en caso de desastre. Por ejemplo, si la base de datos ha confirmado una transacción correspondiente al número de secuencia de cambio número 9550, significará que esa secuencia estará grabada en los Redo Log. Sin embargo, puede que los archivos de datos estén en la 9540. Lo cual significa que hay 10 secuencias en los Redo Log que no se han grabado en los ficheros de datos. Si ocurre un desastre en este instante y el sistema se cierra, dependemos de los Redo Log para recuperar esas secuencias.
- Los redo log graban las instrucciones de transacciones confirmadas y los datos necesarios para volver a realizarlas. La información en los redo log se graba casi instantáneamente, por lo que siempre están más al día que los archivos de datos. Además, sirven como histórico de los cambios realizados en la base de datos. Lo que puede permitir retornar a la base de datos a un punto concreto.



números de secuencia

1, 4, 7, 10,...

2, 5, 8, 11,...

3, 6, 9, 12,...

Funcionamiento

- [1] Cuando el usuario realiza instrucciones DML o DDL, el proceso servidor graba en el Búfer Redo Log en la SGA, la información necesaria para que esa instrucción se vuelva a ejecutar, si fuera necesario.
- [2] El proceso **LGWR**, cuando las instrucciones son definitivas (bien transacciones aceptadas o instrucciones DDL por ejemplo), copia los datos del búfer a los archivos redo log.
- [3] Cuando llenamos el archivo redo log actual, se produce el evento **log switch**; entonces los siguientes datos pasan al siguiente archivo redo log. Con éste ocurrirá lo mismo y así sucesivamente con cada archivo del que dispongamos.
- [4] En cada cambio de archivo, se genera un **número de secuencia** que va anotando la secuencia de redo log (lo que es lo mismo, el número de eventos **log switch** que han ocurrido), así podemos tener ahora el número de secuencia 12 indicando que hemos hecho 12 cambios de archivo (o sea, han ocurrido 12 eventos log switch).
- [5] Cuando ya hemos ocupado todos los archivos, al terminar de llenar el último se produce el consiguiente log switch y como no hay más archivos disponibles, grabará los siguientes datos en el primer archivo sobrescribiendo los que ya existieran (y perdiendo esos datos). El número de secuencia seguirá incrementándose, no volverá a empezar de nuevo.
- [6] Todo este proceso sigue continuamente

Archivos de control

Se trata de archivos binarios y de tamaño pequeño que contienen la estructura de la base de datos, es decir metadatos. Este archivo también se puede multiplexar para aumentar la seguridad, con lo que puede haber varios. Lo normal es tener al menos dos, ya que es un recurso crítico.

Los archivos de control contienen:

- Nombre de la base de datos
- Fecha y hora de la creación de la base de datos
- Información sobre **checkpoints** y **redo logs** (secuencia actual de redo log, nº de checkpoint,...)
- Modo de archivado de la base de datos.
- Número de secuencia del redo log actual.
- Metadatos para labores de backup y recuperación de datos.
- Ubicación de los archivos de datos, redo log e históricos redo log
- Información sobre los tablespaces actuales