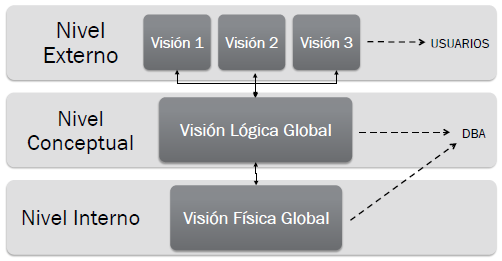
**ARQUITECTURA ANSI/SPARC**: propuesta por el grupo de estudio en sistemas de administracion de bases de datos de NASI/SPARC en el 1975.



* NIVEL EXTERNO: todas las percepciones particulares de la BD por parte de los usuarios -> cada usuario tiene su vision de la BD
* NIVEL CONCEPTUAL: abstraccion global de la BD que integra y aglutina todas las percepciones que los usuarios tiene de ella
* NIVEL INTERNO: representacion mas cercana a la estructura de almacenamiento fisica -> forma en que se implantan las estructuras de datos que organizan los niveles superiores

**LENGUAJES DE APLICACION Y DE DATOS**

* **DSL (DATA SUB-LANGUAGE):** lenguaje especifico orientado a los datos que aporte los mecanismos necesarios para gestiona las teareas de definicion, control y manipulacion de datos
* **DDL (DATA DEFINITION LANGUAGE):** sub-lenguaje de definicion de datos, sub-conjunto de DSL para definir estructuras de datos y esquemas logicos en la BD
* **DML (DATA MANIPULATION LANG.):** sub-lenguaje de manipulacion de datos, subconjunto de DSL prara introducir/modificar/eliminar/consultar datos en los esquemas y para consultar la estructura de los esquemas definidos en la BD
* **DCL (DATA CONTROL LANGUAGE):** sub-lenguaje de control de datos, subconjunto de DSL para gestionar los requisitos de acceso a los datos y otras tareas administrativas

**LENGUAJE ANFITRION O DE APLICACION:** puede ser DEBILMENTE ACOPLADO (lenguajes de proposito general con librerias para acceder a la BD a traves del DSL proporcionado por el SGBD [APIs de acceso a BD]) o FORTEMENTE ACOPLADO (lenguajes de proposito especifico, con DSL como elemento central del lenguaje, para facilitar desarrollo de aplicaciones de gestion de BD [ODBC o JDBC])

**NIVEL EXTERNO**

Se denomina VISTA o VISION EXTERNA cada perspectiva, el conjunto de todas es el niv.ext.

Requiere uso de DDL de nivel externo para elaborar esquemas externos a partir del esquema conceptual (o otros esq externos)

Se valdrà de un DML de nivel externo para manipular sus contenidos

**NIVEL CONCEPTUAL**

Debe ofrecer una unica perspectica o vision global que aglutine y integre todas las vistas que de ella tiene los usuarios en el nivel externo; requiere DDL de nivel conceptual para implantar diferentes restricciones de integridad semantica y para escibir el esquema conceptual a partir de la estructura de almacenamiento expuesto por el nivel interno

El DML de este nivel manipula los contenidos almacenados en el esquema conceptual

El SGBD puede proporcionar un DCL para implantar politicas de seguridad para control de acceso.

**NIVEL INTERNO**

Representacion abstracta de la estructura de almacenamiento proporcionada por el SO que ejecuta el SGBD, disponde de su DDL para describir el esquema interno y de su DML para permitir acceso a datos almacenado en el esquema interno.**ADMINISTRADOR DE BD (DBA)**

* Elabora el esquema conceptual
* Decide la estructura de almacenamiento en el nivel interno
* Conexion con usuarios
* Define las restricciones de integridad
* Define la politica de seguridad
* Define la estrategia de recuperacion frente a fallos
* Optimiza el rendimiento
* Monitoriza el SGBD

**SGBD: COMPONENTES Y FUNCIONES**

Hecho por un paquete de SW para gestionar acceso a BD atendiendo las peticiones de los usuarios (usuario solicita peticion -> SGBD analiza correccion de la operacion (sintaxis) -> SGBD determina la forma mas adecuada para actuar modificaciones o recuperacioens -> desencadena la ejecucion de dicho plan)

* Deficion de datos
* Manipulacion de datos
* Versiones interpretadas y compiladas por los procesadores de DDL/DML
* Optimizacion en la ejecucion de sentencias de datos
* Seguridad
* Integridad
* Gestion concurrencia y consistencia
* Diccionario de datos
* Herramientas de apoyo a la administr.

**ARQUITECTURA DE UN SGBD**

* NIVEL DE SERVIDOR DE DATOS: SGBD con servidores centralizados VS distrib.
* NIVEL DE SERVIDOR DE APLICACIONES: Modelo Cliente/servidor (con aplicaciones mas pesadas) VS clientes ligeros con entornos de ejecucion de aplicaciones
* NIVEL DE CLIENTE: reducir dependencia de HW y SO, configuraciones basadas en estandares abiertos y portables