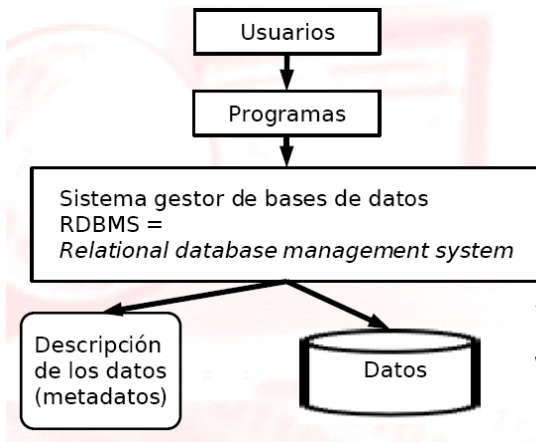




Universidad Nacional de la Patagonia San Juan Bosco
Facultad de Ingeniería

Cátedra: **Bases de datos**

Fundamentos de las Bases de Datos



DNI-Empleado	Apellido	Nombre	Domicilio	Departamento	Sueldo	Año-ingreso

SELECT Apellido, Nombre, Departamento
FROM EMPLEADO
WHERE Sueldo = 2000

Apellido	Nombre	Departamento

- Base de Datos (BD): Conjunto de datos que:
 - Se relacionan entre sí (por su significado).
 - Modelan el mundo real con un propósito específico (aplicaciones necesarias, tipo de usuarios)
- Sistema de gestión de bases de datos (SGBD, DBMS)
 - Conjunto de programas que permite crear y mantener una base de datos:
 - ✓ **Definir**: tipos de datos, estructuras y restricciones de esos datos: DDL
 - ✓ **Construir**: almacenar los datos y sus estructuras en un soporte físico.
 - ✓ **Manipular**: Consultas y modificaciones de los datos: DML
 - **Objetivo primordial de un SGBD**: Proporcionar un entorno que sea práctico y eficiente para ser utilizado con el fin de extraer y almacenar información en la BD.

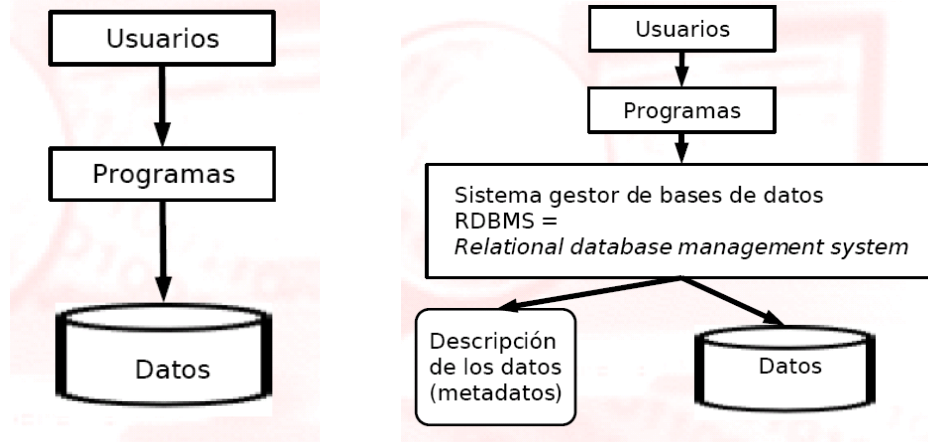
DNI-Empleado	Apellido	Nombre	Domicilio	Departamento	Sueldo	Año-ingreso

```
SELECT Apellido, Nombre, Departamento
FROM EMPLEADO
WHERE Sueldo = 2000
```

Apellido	Nombre	Departamento

- Enfoque Tradicional vs. SGBD

- ✓ Enfoque tradicional: cada aplicación definía y mantenía sus propios datos → redundancia e inconsistencia, dificultad para acceder a los datos, problemas de seguridad, etc.
- ✓ SGBD: los datos se definen y administran de forma integrada → restricciones de integridad



- ¿Qué es un SGBD?

- Su finalidad: almacenar información y permitir a los usuarios recuperar y actualizar esa información en base a peticiones.
- Comprende cuatro componentes principales:
 - ✓ **Datos:** Integrados y Compartidos
 - ✓ **Hardware:** alm. secundario, disp. E/S, memoria, procesadores
 - ✓ **Software:** software SGBD (server)
 - ✓ **Usuarios:** programadores de aplicaciones, usuarios finales y administrador

- ¿Porqué usar un SGBD?
 - Software **sofisticado** para manejo de grandes volúmenes de datos vs. Desarrollar programas usando lenguajes de propósito general → Más trabajo para manipular datos. Productividad, Costos, Calidad.
 - ¿Qué es una BD?
 - **Conjunto compartido de datos** (junto con una descripción) que están diseñados para satisfacer las necesidades de información de una organización.
 - **Datos Persistentes**: los datos almacenados en una BD difieren de otros llamados transitorios como son los de un programa de aplicación: variables, datos de E/S, etc. Los datos de una BD persisten (aunque podrían estar poco tiempo) debido a que, una vez incorporados en la BD, sólo pueden ser removidos por alguna solicitud explícita al SGBD.
- ⇒ La BD es un conjunto de datos persistentes que es utilizado por los sistemas de aplicación de alguna empresa u organismo.

- Ventajas de la implementación de una BD:
 - ✓ **Compactación:** no hay necesidad de archivos en papel.
 - ✓ **Velocidad:** la PC puede recuperar y actualizar datos más rápido que las personas.
 - ✓ **Menos trabajo laborioso:** las tareas mecánicas siempre las realizan mejor las PC.
 - ✓ **Actualidad:** se tiene a disposición información precisa y actualizada.
 - ✓ Los datos pueden **compartirse**.
 - ✓ Es posible disminuir la **redundancia**.
 - ✓ **Se evita la inconsistencia:** Una BD inconsistente puede proporcionar información incorrecta o contradictoria.
 - ✓ Es posible brindar un buen **manejo de transacciones**.
 - ✓ **Conservar la integridad de los datos:** 40 hs. trabajadas x semana, no 400.
 - ✓ Hacer cumplir la **seguridad**.

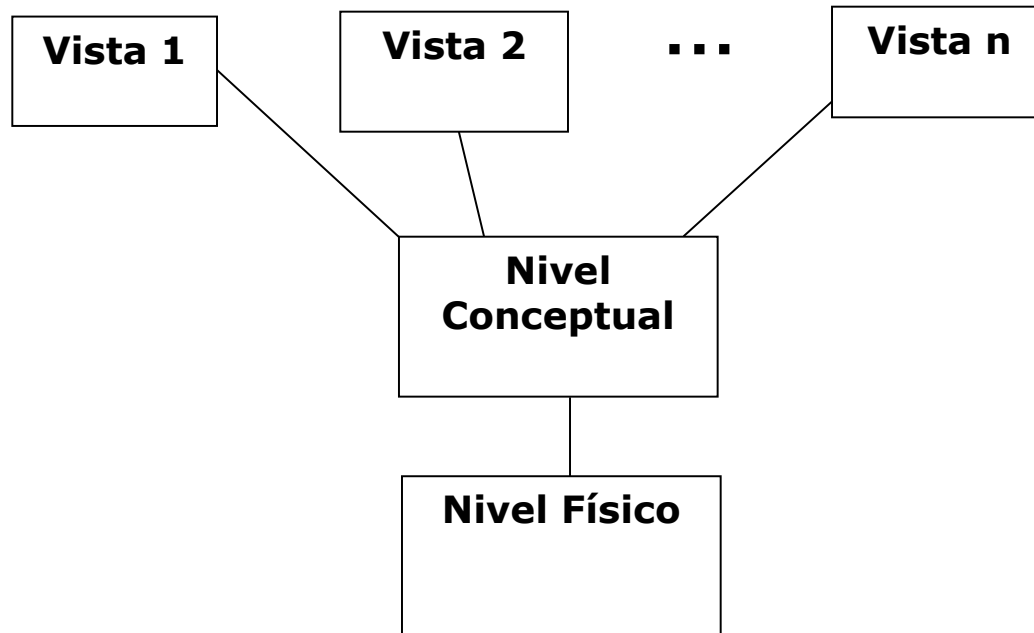
- ACID Test:

- Un buen SGBD debe observar las siguientes propiedades en lo que respecta al manejo de transacciones:

- ✓ **Atomicidad:** los resultados de una transacción deben ser todos completados (commit), o bien pasan a ser todos descartados (rollback). Es decir, o todos los cambios incluidos en una transacción tienen efecto o ninguno lo tiene.
 - ✓ **Consistencia:** las BD se transforman de estados íntegros a estados íntegros, es decir, entre estados válidos. Una transacción sólo se puede completar si el estado final es íntegro.
 - ✓ **Isolation (aislamiento):** los resultados de una transacción son invisibles para el resto de transacciones de otros procesos hasta que la transacción se ha completado.
 - ✓ **Durabilidad:** una vez que una transacción ha sido completada, los resultados (cambios) de la transacción se hacen permanentes, incluso frente a fallos del sistema y de medios de almacenamiento.

✚ Sólo si estas propiedades de las transacciones se cumplen, podemos considerar que un SGBD cumple las funciones de **persistencia, seguridad, integridad y control de acceso concurrente**.

- Los tres niveles de la Arquitectura
 - No todos los SGBD coinciden enteramente con esta arquitectura.
 - Es la arquitectura propuesta por el grupo de estudio en sistemas de administración de BD **ANSI/SPARC** American National Standards Institute / Standards Planning And Requirements Committee



- Los tres niveles de la Arquitectura (Cont.)
 - **Nivel Físico:** describe los detalles de cómo el SGBD utiliza el disco duro, la memoria, etc. Es decir cómo se almacenan realmente los datos. **El sistema debe ser rápido en responder y eficiente en el uso de espacio.**
 - **Nivel Conceptual:** describe qué datos son almacenados en la BD y las relaciones que existen entre ellos. Se describe la BD completa en términos de un número pequeño de estructuras relativamente sencillas (tablas, columnas, etc.) que pueden implicar estructuras complejas del nivel físico. El usuario del nivel conceptual no necesita darse cuenta de esto. Este nivel lo usan los administradores de BD, que deciden qué información se va a guardar en la BD. **Debe reflejar adecuadamente el problema a modelar.**
 - **Nivel de Visión:** Es el nivel más próximo a los usuarios. El nivel de más alta abstracción, describe parte de la BD. A pesar del uso de estructuras sencillas en el nivel conceptual, permanece algo de complejidad debido al gran tamaño de la BD. Los usuarios de la BD no se interesan por toda esta información, sólo necesitan una parte de la BD (Vistas sobre las tablas). **Cada vista debe reflejar adecuadamente la parte de los datos que interesa a cada usuario.**

- Ejemplo de los tres niveles de la Arquitectura

Visión 1 Número Empleado Salario	Visión 2 Número Empleado Departamento
Conceptual EMPLEADO: Num_Empleado CHARACTER (6) Num_Dto CHARACTER (4) Salario NUMERIC (5)	
Físico EMP_ALMACENADO BYTES = 20 EMP# Bytes (6) INDEX= EMPX DEPT# Bytes (4) Sueldo Bytes (6)	

Un usuario puede ver el número de Empleado y sueldo, y otro los depts. en los que trabaja cada empleado

Entidad denominada EMPLEADO con:
 Num_Empleado de 6 caracteres, Num_Dto de 4 caracteres y Salario de 5 dígitos.

Registro denominado EMP_ALMACENADO de 20 bytes de longitud, tiene tres campos de datos y está indexado sobre el campo EMP# por medio de un índice llamado EMPX

- Modelos de Datos

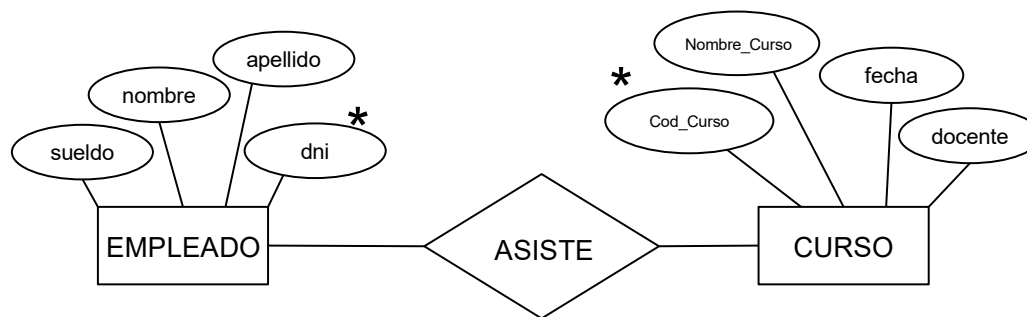
- Es una colección de herramientas conceptuales para describir datos, relaciones entre ellos, semántica asociada a los datos y restricciones de consistencia (aspectos que deben cumplir los datos).
- Un Modelo de Datos permite expresar:
 - ✓ **Estructuras:** CURSO(nro_curso, nombre, horas)
 - ✓ **Restricciones:** Reglas que deben cumplir los datos (nro_curso debe ser único, horas < 120)
 - ✓ **Operaciones:** Insertar, borrar y consultar la BD

Insert into CURSOS (303,"BD",90)

- ✎ Los diversos modelos de datos que se han propuesto se dividen en tres grupos: **modelos conceptuales, modelos lógicos y modelos físicos de datos.**

- Modelos Conceptuales

- Representan la realidad independientemente de cualquier implementación de BD. Se usa en la **etapa de Análisis**.
- Se usan para describir datos en los **niveles conceptual y de visión**.
- Se caracterizan por que proporcionan capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Hay muchos modelos diferentes, y es probable que aparezcan más.
- Algunos de los más extensamente conocidos son:
 - ✓ El modelo entidad-relación
 - ✓ El modelo orientado a objetos
 - ✓ El modelo semántico de datos
 - ✓ El modelo funcional de datos

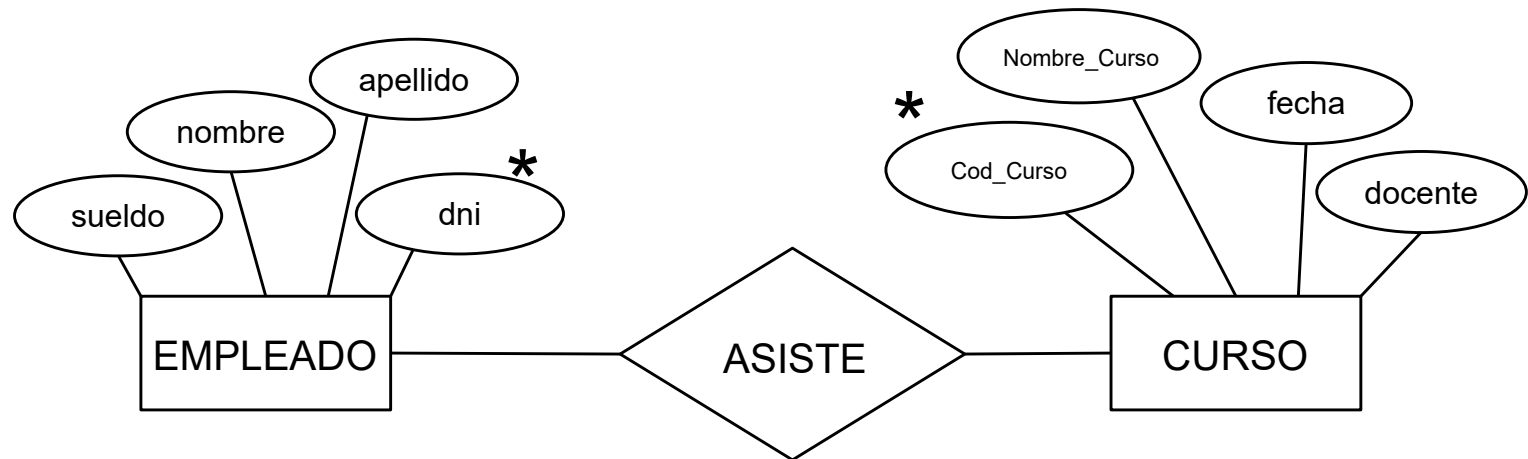


Empleado	: Empleado
Nombre Apellido D.N.I sueldo año_ingr	Nombre= Juan Apellido: Soler D.N.I= 23.345.456 sueldo: 3000 fecha_ingr: 1/3/1990
leer_nombre() calcular_antig()	

- Modelos Conceptuales – Modelo E-R
 - Se basa en una percepción del mundo real que consiste en una colección de objetos básicos llamados entidades y relaciones entre estos objetos.
 - Una **entidad** es un objeto o cosa que es distinguible de otros objetos por medio de un conjunto específico de **atributos**. Por ejemplo, el atributo *dni* describe un *empleado* particular de una *empresa*.
 - Una **relación** es una asociación entre varias entidades. Por ejemplo, una relación *asiste* asocia a un *empleado* con cada uno de los *cursos* a los que asiste.
 - El conjunto de todas las entidades del mismo tipo y relaciones del mismo tipo se denomina conjunto de entidades y conjunto de relaciones, respectivamente.
 - Además de entidades y relaciones, el modelo E-R representa ciertas restricciones a las que deben ajustarse los contenidos de una base de datos. Una restricción importante es la de **cardinalidad** de asignación, que expresa el número de entidades a las que puede asociarse otra entidad mediante un conjunto de relación.

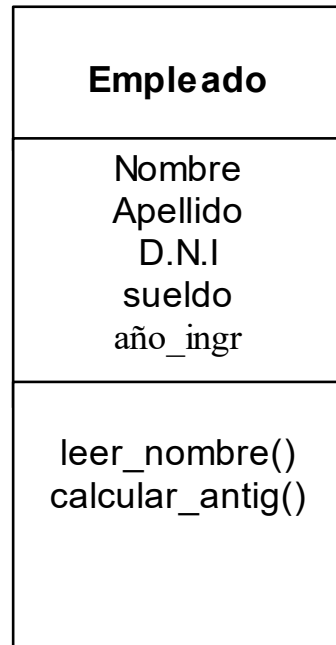
- Modelos Conceptuales – Modelo E-R (Cont.)

- La estructura lógica global de una base de datos puede expresarse gráficamente por medio de un **diagrama E-R**, que consta de los siguientes componentes:

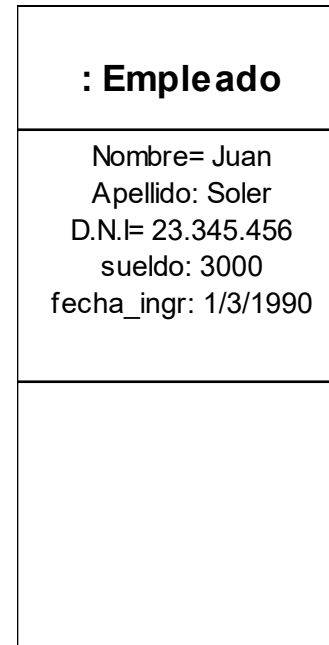


- Modelos Conceptuales – Modelo orientado a objetos (OO)
 - Cada **objeto** encapsula **atributos** con valores, con un id externo (físico), y además podemos asociar a cada uno de esos objetos con un conjunto de **métodos** para manejar los valores. Un objeto puede tener los mismos valores que otro, pero id diferente (diferencia E/R). → no es difícil de asimilar y comprender, concepto intuitivo.
 - Los objetos que contienen los mismos tipos de valores y los mismos métodos se agrupan en **clases**. Una clase puede ser vista como una definición de tipo (plantilla) para objetos. Esta combinación de datos y código en una definición de tipo es parecida al concepto de tipos de datos abstractos en lenguajes de programación.
 - Un objeto no restringe los valores de los atributos a un conjunto de tipos de datos como entero, float, char etc. Los valores pueden ser otros objetos. P. e. uno de los atributos de empleado puede ser el curso o los cursos a los que asiste, y el valor de este atributo puede ser un objeto curso, y viceversa.
 - Este modelo mantiene las relaciones por medio de una “contención”. En el ejemplo empleado-curso, se encontraría al empleado que realiza un curso, “dentro” del curso, como valor de uno de los atributos de CURSO.
 - **Diagrama de Clases** (Estructura Estática) nos muestra una vista en un determinado momento. Las clases son la plantilla de los objetos, y los vemos representados con sus atributos y sus métodos, así como la relación entre ellas.

- Modelos Conceptuales – Modelo orientado a objetos (OO)



Ejemplo de Clase



Ejemplo de Objeto

- A diferencia de las entidades en el modelo E-R, cada objeto tiene su propia identidad única independiente de los valores que contiene.
- Así, dos objetos que contienen los mismos valores son, sin embargo, distintos.
- La distinción entre objetos individuales se mantiene en el nivel físico por medio de identificadores de objeto.

- **Modelo Lógico**

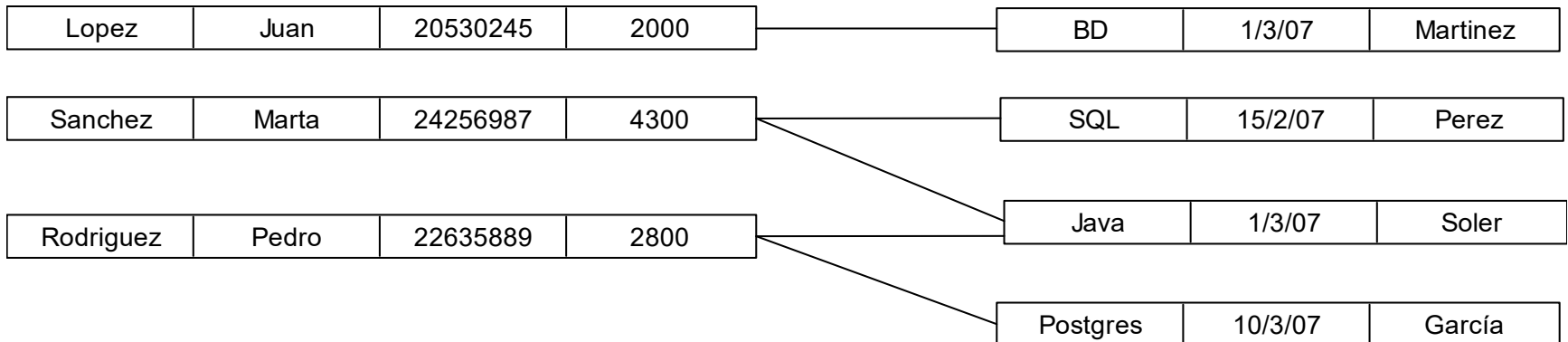
- **Objetivo:** convertir el esquema conceptual de datos en un esquema lógico que se ajuste al gestor de la BD. Puede ser el modelo relacional, de red, jerárquico o el modelo orientado a objetos.
- Una vez establecido el modelo conceptual del problema, el diseño lógico de los datos permite que éstos se puedan representar usando de manera eficiente posibles recursos para estructurar datos y modelar restricciones disponibles en el modelo lógico. Se usa en etapas de **diseño e implementación**.
- Conforme se va desarrollando el esquema lógico, éste se va probando y validando con los requisitos de usuario.
- **Normalización:** técnica que se utiliza para comprobar la validez de los esquemas lógicos basados en el modelo relacional. Asegura que las relaciones (tablas) obtenidas no tienen datos redundantes.
- El esquema lógico es una fuente de **información para el diseño físico**, importante durante la etapa de mantenimiento del sistema, ya que permite que los futuros cambios que se realicen sobre los programas de aplicación o sobre los datos, se representen correctamente en la BD.

- Modelo Lógico – Modelo relacional
 - Representa los datos y las relaciones entre los datos mediante una colección de **tablas**, cada una de las cuales tiene un número de **columnas** con nombres únicos.

Apellido	Nombre	DNI	Sueldo	Nombre_cur
López	Juan	20530245	2000	BD
Sánchez	Marta	24256987	4300	SQL
Rodríguez	Pedro	22635889	2800	Java
Rodríguez	Pedro	22635889	2800	Postgres
Sánchez	Marta	24256987	4300	Java

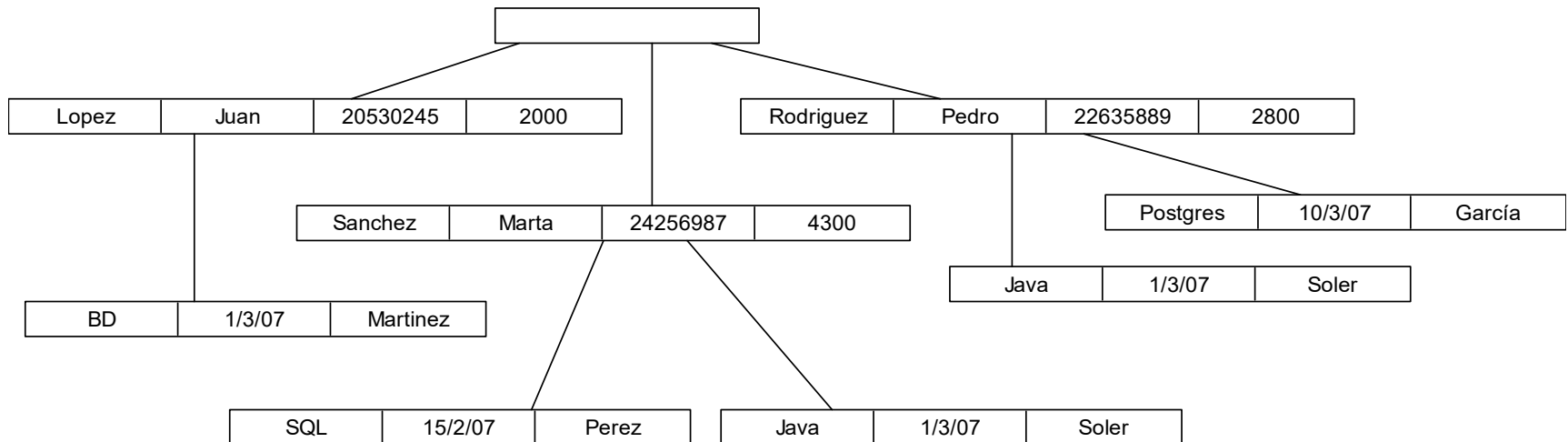
Nombre_cur	Fecha_comienzo	Profesor
BD	1/3/07	Martinez
SQL	15/2/07	Perez
Java	1/3/07	Soler
Postgres	10/3/07	García

- Modelo Lógico – Modelo de red
 - Los datos se representan mediante colecciones de **registros** (en el sentido de la palabra Pascal o PL/I) y las **relaciones** entre los datos se representan mediante enlaces que pueden verse como punteros. Los registros en la BD se organizan como **colecciones de grafos arbitrarios**.

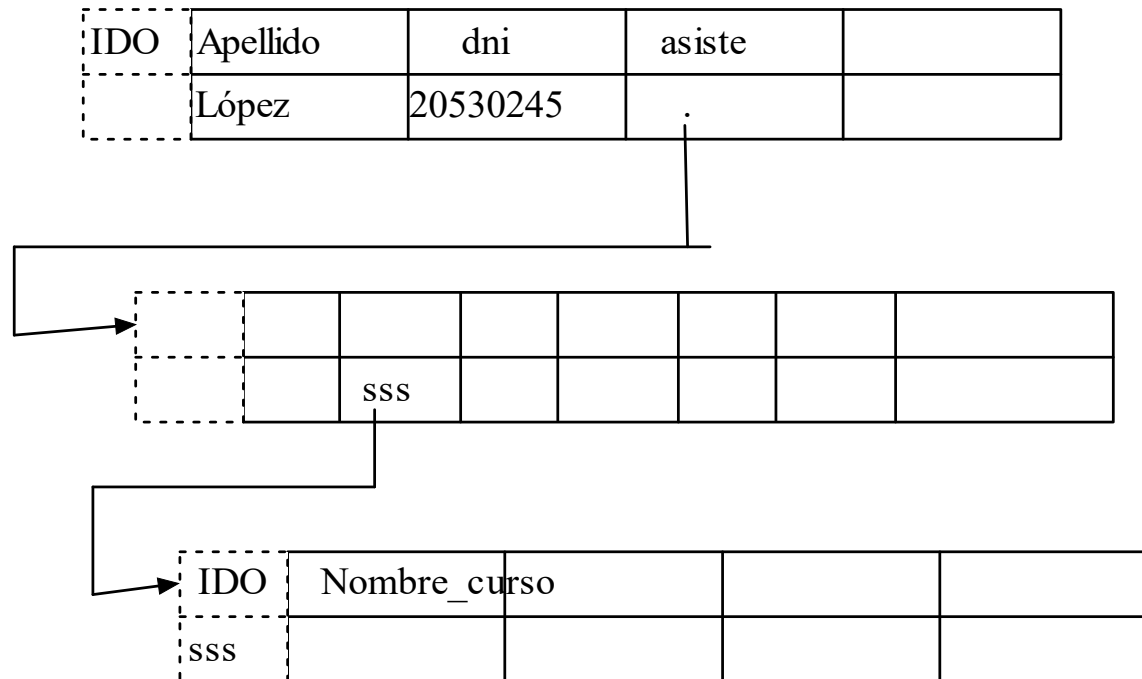


- Modelo Lógico – Modelo jerárquico

- Es similar al modelo de red en el sentido de que los datos y las relaciones se representan mediante **registros** y **enlaces**. Se diferencia en que los registros están organizados como **colecciones de árboles** en vez de grafos arbitrarios.



- Modelo Lógico – Modelo OO
 - Incluye **código ejecutable** como una parte integrante del mismo modelo de datos y se deben escribir métodos que agreguen los **IDO's**.
 - Una BDOO se ve de la siguiente forma:



- Diferencias entre los modelos
 - Los modelos relacionales se diferencian de los modelos de red y jerárquico en que no usan punteros o enlaces. En cambio, el modelo relacional conecta registros mediante los valores que éstos contienen. Esta libertad del uso de punteros permite que se defina una base matemática formal.

- Modelos Físicos De Datos
 - Los modelos físicos de datos se usan para **describir datos en el nivel más bajo** → Es la implementación de las estructuras de datos.
 - A diferencia de los modelos lógicos, hay muy pocos modelos físicos en uso.
 - Instancias y esquemas
 - Las BD cambian a lo largo del tiempo según se añade y se suprime información.
 - La colección de información almacenada en la BD, en un determinado momento en el tiempo, se llama una **instancia**.
 - El diseño global de la BD se llama **esquema**. Se cambian muy raras veces, o nunca.
 - El concepto de esquema de base de datos corresponde a la noción de definición de tipo en el lenguaje de programación.
 - Los sistemas de BD tienen varios esquemas, divididos de acuerdo a los niveles de abstracción. En el nivel más bajo está el esquema físico; en el nivel intermedio, el esquema conceptual; en el nivel más alto, los subesquemas.
- ✚ Es tarea del SGBD mantener la coherencia entre los esquemas de la base de datos y mantener la independencia de los datos.

- Modelos Físicos De Datos – Independencia de Datos
 - Es la capacidad de modificar una definición de un esquema en un nivel sin afectar la definición de un esquema en el nivel superior siguiente:
 - ✓ **Independencia física de datos:** los cambios efectuados en el esquema físico (cambios de versión del SGBD o de SGBD) no afectan al esquema conceptual.
 - ✓ **Independencia lógica de datos:** es la capacidad de modificar el esquema conceptual sin afectar el esquema externo. (p. e. añadir columnas a una tabla).

- Lenguajes De BD – Lenguaje de Definición de Datos (DDL)
 - Un esquema de BD se especifica por medio de un conjunto de definiciones que se expresan mediante un lenguaje especial llamado **Lenguaje de Definición de Datos** (**Data Definition Language**).
 - Compilación de sentencias de DDL → conjunto de tablas. Se almacenan en un archivo especial llamado **Diccionario de Datos** –DD- o directorio.
 - Un directorio de datos es un archivo que contiene metadatos y se lo consulta antes de leer o modificar los datos reales en el SGBD.
 - La estructura de almacenamiento y los métodos de acceso usados por los SGBD se especifican por medio de un conjunto de definiciones en un tipo especial de DDL llamado **lenguaje de almacenamiento y definición de datos**.
 - El resultado de la compilación de estas definiciones es un conjunto de instrucciones que especifican los detalles de implementación de los esquemas de BD que normalmente se esconden a los usuarios. P.e.:
 - ✓ create table
 - ✓ create index
 - ✓ drop table
 - ✓ grant
 - ✓ revoke
 - ✓ modify table

- Lenguajes De BD – Lenguaje de Manipulación de Datos (DML)
 - Manipulación de datos:
 - ✓ Recuperación de información almacenada en la BD
 - ✓ Inserción de información nueva en la BD
 - ✓ Supresión de información en la BD
 - ✓ Modificación de datos almacenados en la BD
 - A nivel físico debemos definir algoritmos que permitan acceso eficiente a los datos. En los niveles de abstracción más altos, se pone énfasis en la facilidad de uso. El objetivo es proporcionar una **interacción eficiente entre las personas y el sistema**.
 - DML es un lenguaje que capacita a los usuarios a acceder o manipular datos según estén organizados por el modelo de datos adecuado. Existen dos tipos:
 - ✓ **Procedimentales:** (bajo nivel) requieren que el usuario especifique qué datos se necesitan y cómo obtenerlos.
 - ✓ **No Procedimentales:** (alto nivel) requieren que se especifique qué datos se necesitan sin especificar cómo obtenerlos.

- Lenguajes De BD – Lenguaje de Manipulación de Datos (Cont.)
 - Los DML no procedimentales son más sencillos de aprender y usar que los procedimentales.
 - Sin embargo pueden generar código que no sea tan eficiente. Esta dificultad puede remediarse a través de varias **técnicas de optimización**, algunas de las cuales se tratan adelante.
 - Una **consulta** es una sentencia que solicita la recuperación de información.
 - El trozo de un DML que implica recuperación de información se llama **lenguaje de consultas**.
 - Aunque técnicamente es incorrecto, suelen utilizarse los términos de lenguaje de consultas y lenguaje de manipulación de datos como sinónimos. P.e.:
 - ✓ select
 - ✓ update
 - ✓ insert
 - ✓ delete

- El Administrador de la BD
 - Implementa las estrategias y políticas con respecto a los datos de la empresa. Una de las razones principales para contar con un SGBD es tener control central de los datos y de los programas que acceden a esos datos → administrador de BD.
 - Sus funciones incluyen:
 - ✓ **Definir el esquema conceptual:** una vez identificadas las entidades de interés (diseño conceptual o lógico de la BD), el administrador escribe un conjunto de definiciones que son traducidas por el compilador de DDL a un conjunto de tablas que son almacenadas permanentemente en el DD.
 - ✓ **Definir el esquema interno:** una vez realizado el diseño físico el administrador define la estructura de almacenamiento y métodos de acceso (esquema interno) que se crean escribiendo un conjunto de definiciones que son traducidas por el compilador del lenguaje de almacenamiento y definición de datos.
 - ✓ **Establecer un enlace con los usuarios:** debe contactarse con los usuarios para asegurar que los datos necesarios estén disponibles y así definir las vistas. También debe asesorar sobre el diseño de aplicaciones, capacitación técnica, etc.

- El Administrador de la BD (Cont.)
 - ✓ **Modificación del esquema y de la organización física:** es el encargado de realizar las modificaciones, tanto al esquema de la BD como a la descripción de la organización física de almacenamiento, aunque relativamente poco comunes.
 - ✓ **Definir las restricciones de seguridad e integridad:** La concesión de diferentes tipos de autorización permite al administrador de la BD regular qué partes van a poder ser accedidas por varios usuarios. Las restricciones de integridad se mantienen en una estructura especial del sistema que consulta el gestor de la BD cada vez que tiene lugar una actualización en el sistema.

- Estructura del Sistema Completo
 - Un SGBD se divide en **módulos** que se encargan de cada una de las **responsabilidades** del sistema completo.
 - Algunas de estas funciones las puede proporcionar el **sistema operativo** de la computadora.
 - En la mayoría de los casos los sistemas operativos proporcionan solo los servicios mas básicos y los SGBD deben construirse sobre esta base. Así, el diseño de un SGBD debe incluir consideraciones de la interfaz entre sí y el sistema operativo.
 - Los componentes funcionales de un SGBD se pueden dividir en:
 - ✓ Componentes de procesamiento de consultas
 - ✓ Componentes de gestión de almacenamiento

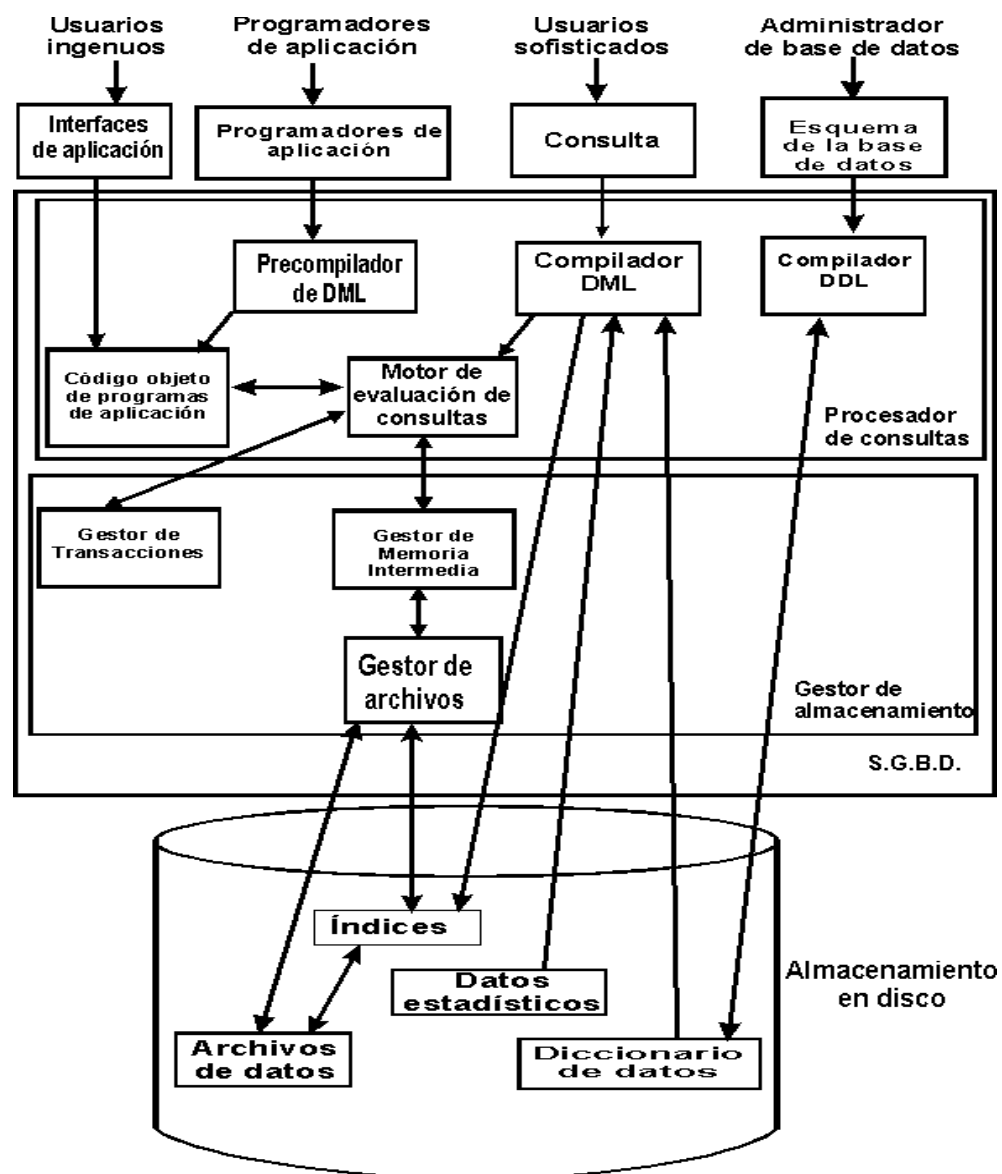


Figura 1.6. Estructura del sistema

- Estructura del Sistema Completo – Componentes de procesamiento de consultas
 - ✓ **Compilador del LMD:** traduce las instrucciones del LMD en lenguaje de consultas a instrucciones a bajo nivel que entiende el motor de evaluación de consultas. Además, el compilador del LMD intenta transformar las peticiones del usuario en otras equivalentes pero mas eficientes.
 - ✓ **Precompilador del LMD:** convierte las instrucciones del LMD incorporadas en un programa de aplicación en llamadas a procesamientos normales en el lenguaje anfitrión. Debe interactuar con el compilador del LMD para generar el código apropiado.
 - ✓ **Interprete del LDD:** interpreta las instrucciones del LDD y las registra en un conjunto de tablas que contiene metadatos.
 - ✓ **Motor de evaluación de consultas:** ejecuta las instrucciones a bajo nivel generadas por el compilador del LMD.

- Estructura del Sistema Completo – Componentes de gestión de almacenamiento
 - Proporcionan la interfaz entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación y envío de consultas al sistema, incluye:
 - ✓ **Gestor de autorización e integridad:** comprueba que se satisfagan las ligaduras de integridad y la autorización de los usuarios para acceder a los datos.
 - ✓ **Gestor de transacciones:** asegura que la BD quede en un estado consistente a pesar de los fallos del sistema, y que las ejecuciones de transacciones concurrentes ocurran sin conflictos.
 - ✓ **Gestor de archivos:** gestiona la reserva de espacio de almacenamiento en disco y las estructuras de datos usadas para representar la información almacenada en disco.
 - ✓ **Gestor de memoria intermedia:** es responsable de traer los datos del disco a memoria principal y decidir qué datos tratar en la memoria caché.

- Estructura del Sistema Completo – Componentes de gestión de almacenamiento (Cont.)
 - Además, se necesitan varias estructuras de datos como parte de la implementación física del sistema:
 - ✓ **Archivos de datos:** almacenan la base de datos en sí.
 - ✓ **Diccionario de datos:** almacena metadatos acerca de la estructura de la BD. El DD se usa mucho, por lo tanto, se debería poner gran énfasis en él.
 - ✓ **Índices:** proporcionan acceso rápido a elementos de datos que tienen valores particulares.
 - ✓ **Datos estadísticos:** almacenan información estadística sobre los datos en la BD. El procesador de consultas usa esta información para seleccionar las formas eficientes para ejecutar una consulta.

↪ **Objetivo de un SGBD:** simplificar y facilitar el acceso de los datos.

- El funcionamiento de un sistema depende de la eficiencia de las estructuras de datos usadas para representar los datos en la BD y de la capacidad de eficiencia de operar sobre esas estructuras.
- Se debe llegar a un compromiso no sólo entre espacio y tiempo sino también entre la eficiencia de un tipo de operación y la de otro.

- **Diccionario de Datos**

- El **Diccionario de Datos (DD)** Es el lugar donde se describirá toda la información sobre el sistema.
- Es una **lista ordenada de todos los elementos de datos** del sistema, con definiciones precisas y rigurosas que permiten que el usuario y el analista tengan una misma comprensión de las entradas, salidas, almacenes y cálculos intermedios (si los hubiera).
- El **DD es dinámico**, ya que se define en la etapa de **especificación de requerimientos** y se irá modificando en el **análisis y diseño**.
- En la mayoría de los sistemas reales con lo que trabaje, los elementos de datos serán lo suficientemente complejos como para que se necesite describirlos en términos de otras cosas.
- Los elementos complejos de datos se definen en términos más sencillos, y los sencillos en términos de los valores y unidades legítimos que pueden asumir.

- Diccionario de Datos (Cont.)
 - La estructura del DD puede variar, se puede optar por:
 - ✓ **Nombre:** Es el nombre principal del elemento de datos o de control, del almacén o de una entidad externa.
 - ✓ **Alias:** Otros nombres usados para el nombre.
 - ✓ **Dónde se usa/cómo se usa:** Un listado de los procesos que usan el elemento de datos o de control y cómo lo usan. P. e. Como entrada al proceso, como salida, como almacén, etc.
 - ✓ **Descripción del contenido:** El contenido representado mediante una notación.
 - ✓ **Información adicional:** Otra información sobre los tipos de datos, los valores implícitos (si se conocen), las restricciones, etc.
 - La notación usada puede ser una de las tres formas fundamentales de construcción:
 - ✓ Como una **secuencia** de elementos de datos.
 - ✓ Como una **selección** entre un conjunto de elementos de datos.
 - ✓ Como una **agrupación repetitiva** de elementos de datos.

- Diccionario de Datos (Cont.)
 - Existen muchos esquemas de notación comunes, el siguiente es de los más utilizados y se compone de varios símbolos sencillos:

Construcción del dato	Notación	Significado
	=	Está compuesto de
Secuencia	+	Y
Selección	[]	Seleccionar una de varias alternativas
		Separa opciones alternativas
Repetición	$m \{ \} n$	Iteración, mínimo m y máximo n repeticiones
	* *	Delimita comentarios

- Modelo de Comportamiento – Datos – DD (Cont.)

- Ejemplo:

- ✓ CLIENTE = {reg_cliente}
 - ✓ reg_cliente = CUIT + nombre_cliente + número_teléfono
 - ✓ CUIT = {dígito} * Identificador – Valor único *
 - ✓ nombre_cliente = título_cortesía + nombre + segundo_nombre + apellido
 - ✓ nombre = {carácter_legal}
 - ✓ segundo_nombre = {carácter_legal}
 - ✓ apellido = {carácter_legal}
 - ✓ carácter_legal = [A-Z | a-z | 0-9 | ` | -]
 - ✓ título_cortesía = ["Sr" | "Sra" | "Srta"]
 - ✓ número_teléfono = [extensión_local | número_exterior]
 - ✓ extensión_local = [212 | 215 | 890]
 - ✓ código_área = [011 | 0297 | 02965]
 - ✓ número_local = número_acceso
 - ✓ número_larga_distancia = código_área + número_local
 - ✓ número_exterior = 9 + [número_local | número_larga_distancia]
 - ✓ número_acceso = {dígito} * cualquier cadena de 7 dígitos *
 - ✓ dígito = * número entero *

- Diccionario de Datos (Cont.)
 - El DD lo crea el **analista** durante el desarrollo del modelo del sistema, pero el **usuario** debe ser capaz de leerlo y entenderlo para poder **verificar el modelo**. Lo que nos lleva a los siguientes planteos:
 - ✓ ¿Podrán los usuarios entender la notación?
 - ✓ ¿Cómo podrían los usuarios verificar que el DD esté completo y correcto?
 - ✓ ¿Cómo se crea el DD?
 - Si bien la notación puede parecer bastante matemática, se utilizan **pocos símbolos** y por lo tanto, no es difícil de aprender.
 - Por otro lado, hay varios detalles de la **corrección del sistema** que el analista puede hacer por su cuenta, como asegurarse que el DD esté completo, sea consistente y no contradictorio.
 - Construir un DD es una de las labores más tediosas y largas del Análisis de Sistemas, pero también es una de las más importantes → sin un DD formal que defina el significado de los términos, no se puede esperar precisión.