Bases de Datos Avanzadas

Clase introductoria

UNIVERSIDAD DE LOS ANDES

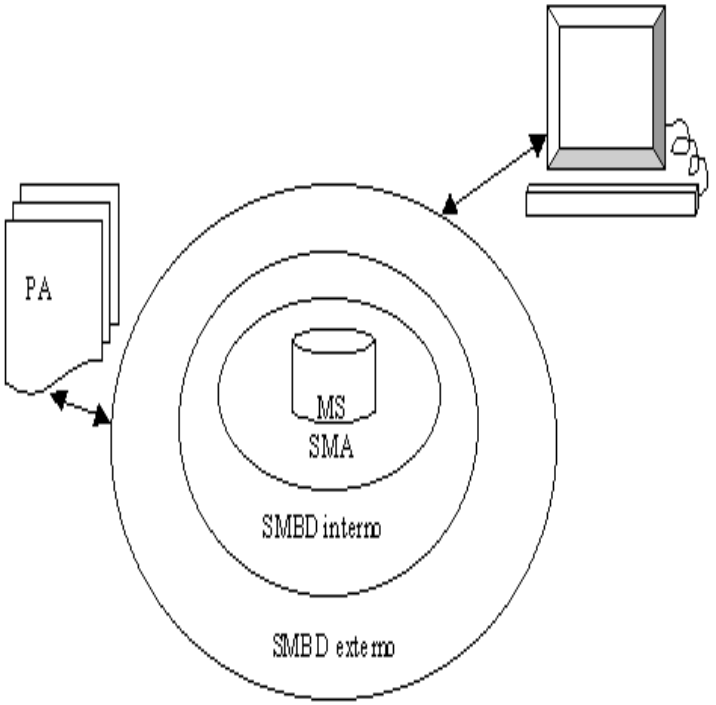
Agenda

- Introducción
- Objetivos
- Principales conceptos
- Tipos y subtipos
- Lenguajes de definición y de manipulación
- Conclusiones

UNIVERSIDAD DE LOS ANDES

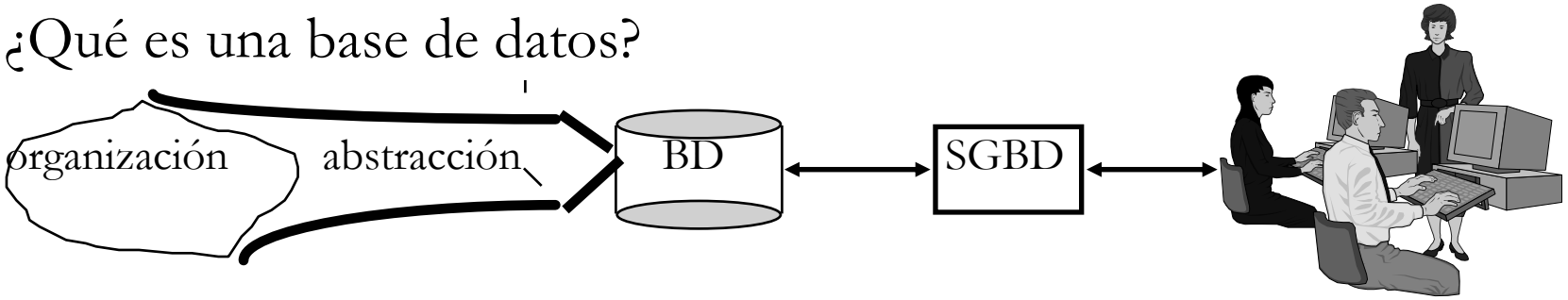
Introducción

- Un **sistema de gestión de bases de datos** (SGBD) es un software que permite insertar, modificar y recuperar eficazmente los datos específicos dentro de una gran masa de información compartida por muchos usuarios.
- Un SGBD asegura la independencia de datos, lo que lo distingue claramente de un **sistema manejador de archivos** (SMA).
- Una **base de datos** es un conjunto de datos gestionados por un SGBD y asociados a una misma aplicación.



Introducción

- ¿Qué es una base de datos?



- ¿Por qué base de datos?

  Porque dicha tecnología permite el mantenimiento y el control de los datos.

- ¿Para qué base de datos?

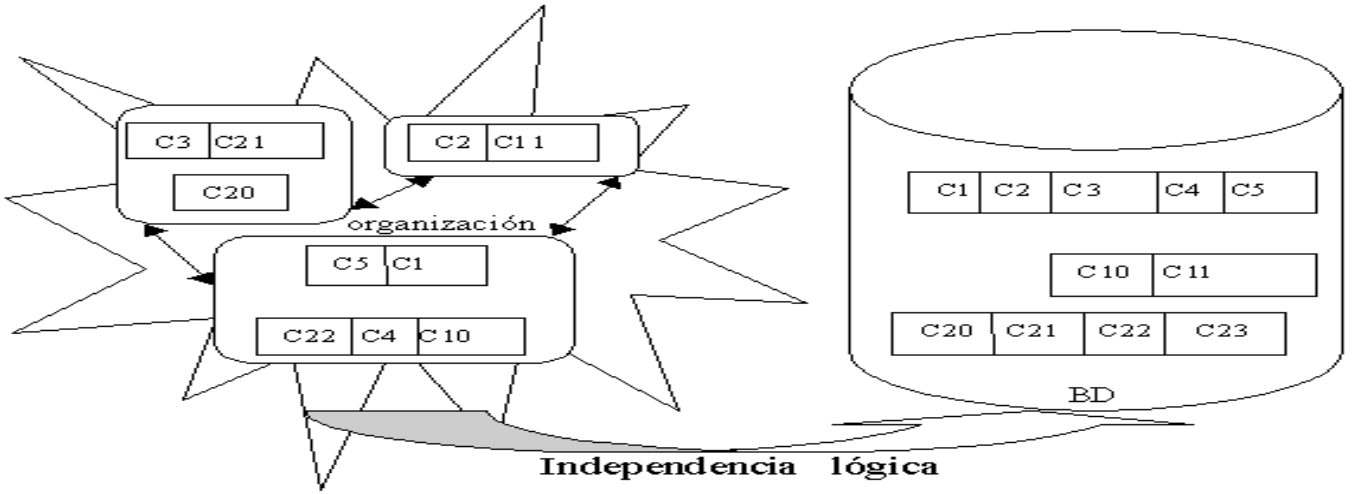
  Para soportar eficientemente algunos sistemas de actividades de una organización

- Tipos:
  - Jerárquica
  - Redes
  - Relacional (BDR)
  - Orientadas por objetos (BDOO)
  - Objeto-relacional (BDOR)

  Subtipos:
  - Distribuida
  - Multimedia
  - Deductivas
  - Temporal
  - Espacio temporal
  - Semi-estructuradas
  - Paralela
  - Activas
  - Espacial

Objetivos

- ✓ ***Independencia física*:** los cambios en la estructura lógica no implican cambios en la de almacenamiento, consideraciones sobre el mejor manejo de los datos quedan a cargo del SGBD y los cambios en la estructura de almacenamiento no implican cambios en los PA.
- ✓ ***Independencia lógica*:** Cierta independencia entre los datos vistos por las aplicaciones y su estructura lógica en la realidad. Ventajas:soporte de la evolución de los datos donde cada grupo de trabajo ve esos datos como lo desea.



UNIVERSIDAD DE LOS ANDES

Objetivos

- ✓ *Manipulación de los datos por personas no especializadas en computación*
- ✓ *Eficacidad en el acceso a los datos*
- ✓ *Administración centralizada de los datos*
- ✓ *Redundancia de datos controlada*
- ✓ *Coherencia de los datos*
- ✓ *Posibilidad de compartir los datos*
- ✓ *Seguridad de los datos*
- ✓ *Habilidad para representar y manipular objetos complejos y compuestos*

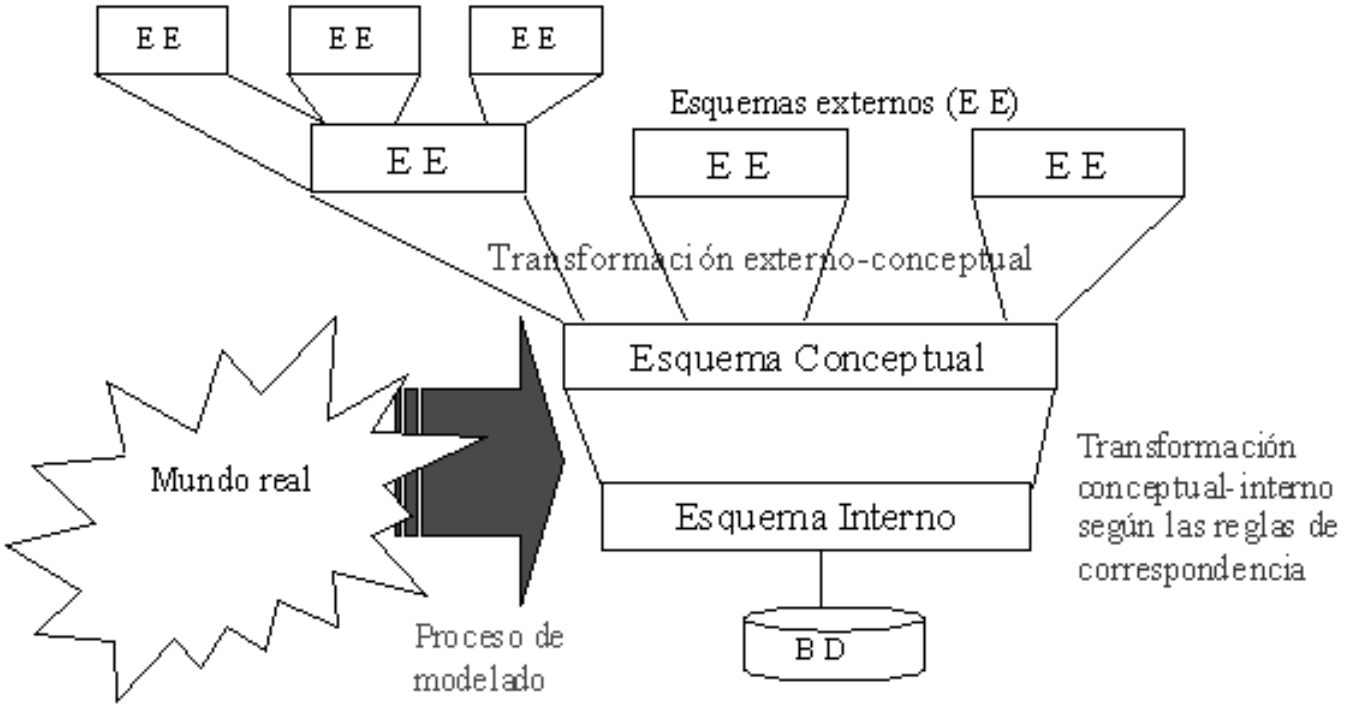
UNIVERSIDAD DE LOS ANDES

Objetivos

- ✓ *Posibilidad de almacenar y recuperar datos arbitrariamente grandes*
- ✓ *Habilidad para definir y manipular cualquier tipo de dato*
- ✓ *Posibilidad de representar y manejar los cambio en la base de datos a través del tiempo*
- ✓ *Representación y manipulación de varios conceptos de modelado semántico, útiles en la aplicaciones*
- ✓ *Posibilidad de especificar reglas y restricciones de integridad para soportar inferencia en las aplicaciones basadas en conocimiento*
- ✓ *Habilidad para manejar transacciones cooperativas de larga duración*

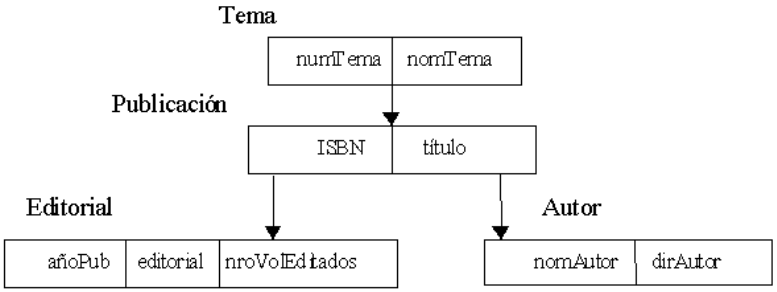
Conceptos básicos

Niveles de descripción



Jerárquica

- Definido (1968) según:
  - ✓ **Campo:** unidad de datos con nombre.
  - ✓ **Segmento:** colección de campos consecutivos con nombre. Unidad de intercambio BD - PA.
  - ✓ **Árbol de segmentos:** colección de segmentos ligados por asociaciones 1:N (padre – hijos), formando una jerarquía, en el ámbito de tipos o de ocurrencias.
  - ✓ **Base de datos jerárquica:** BD compuesta de un bosque de segmentos (árboles de segmentos) cuyos nodos son los segmentos y las aristas las asociaciones 1:N

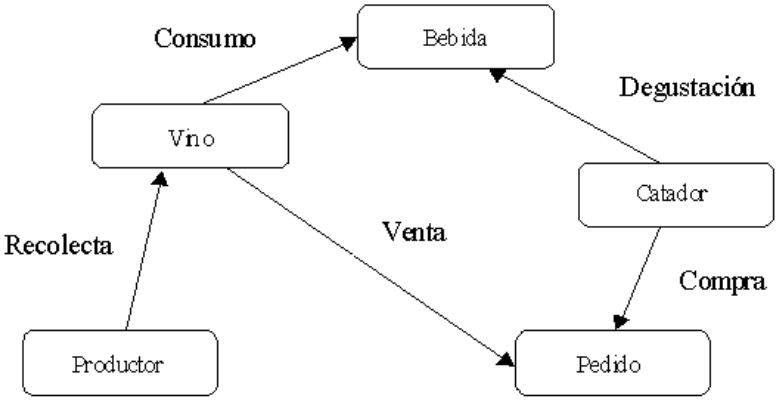



```
DBD NAME = Publica
SEGM NAME = Tema, BYTES = 44
FIELD NAME = (NumTema, SEQ), BYTES = 4, START = 1
FIELD NAME = NomTema, BYTES = 40, START = 5
SEGM NAME = Publicacion, PARENT = Tema, BYTES = 96
FIELD NAME = (ISBN, SEQ), BYTES = 16, START = 1
FIELD NAME = Titulo, BYTES = 80, START = 17
SEGM NAME = Editorial, PARENT = Pub, BYTES = 40
FIELD NAME = (AñoPub, SEQ), BYTES = 4, START = 1
FIELD NAME = Editorial, BYTES = 34, START = 5
FIELD NAME = NroVolEditados, BYTES = 2, START = 39
SEGM NAME = Autor, PARENT = Pub, BYTES = 256
FIELD NAME = (NomAut, SEQ), BYTES = 20, START = 1
FIELD NAME = Direccion, BYTES = 236, START = 21
```

Redes

Definición (1971) según:

- **Átomo o item de dato:** unidad de datos con nombre.
- **Agregado de datos:** colección de átomos consecutivos con nombre. Tipos: vectores (arreglos unidimensionales) y grupos repetitivos.
- **Registro:** colección de agregados y de átomos consecutivos. Unidad de intercambio BD - PA.
- **Conjunto:** asociación entre 1 registro propietario y N registros miembros.
- **Limitaciones:**
  - ✓ un registro o tipo de registro no puede ser propietario y miembro en el mismo conjunto
  - ✓ una ocurrencia de un registro no pueda pertenecer a varias ocurrencias del mismo conjunto.



- **Base de datos en redes:** BD compuesta de registros ligados o asociados entre ellos por los conjuntos.
- **Representación:** grafo de tipos de registros cuyos nodos son los tipos de registros y las aristas son los tipos de conjuntos orientados del propietario hacia los miembros.

Relacional

Propuesto por E. Codd en 1970 y comercialmente disponible 1979

- Basado en la teoría de normalización de las relaciones
- Permite eliminar el comportamiento anormal de las relaciones, luego de actualizaciones, y el control de la redundancia de datos
- **Dominio:** conjunto de valores
- **Relación:** subconjunto del producto cartesiano de una lista de dominios, no necesariamente disjuntos.
- **Atributo:** columna de una relación identificada con un nombre.

R	color	marca
	'azul'	'ford'
	'verde'	'toyota'
	'blanco'	'renault'

Esquema de una relación o de tabla:

Por intensión

R(color, marca)

Por extensión

Base de datos relacional:

base de datos cuyo esquema es un conjunto de esquemas de relación de diferente nombre cada una, y donde sus ocurrencias son las tuplas de esas relaciones

UNIVERSIDAD DE LOS ANDES

Reglas de formación

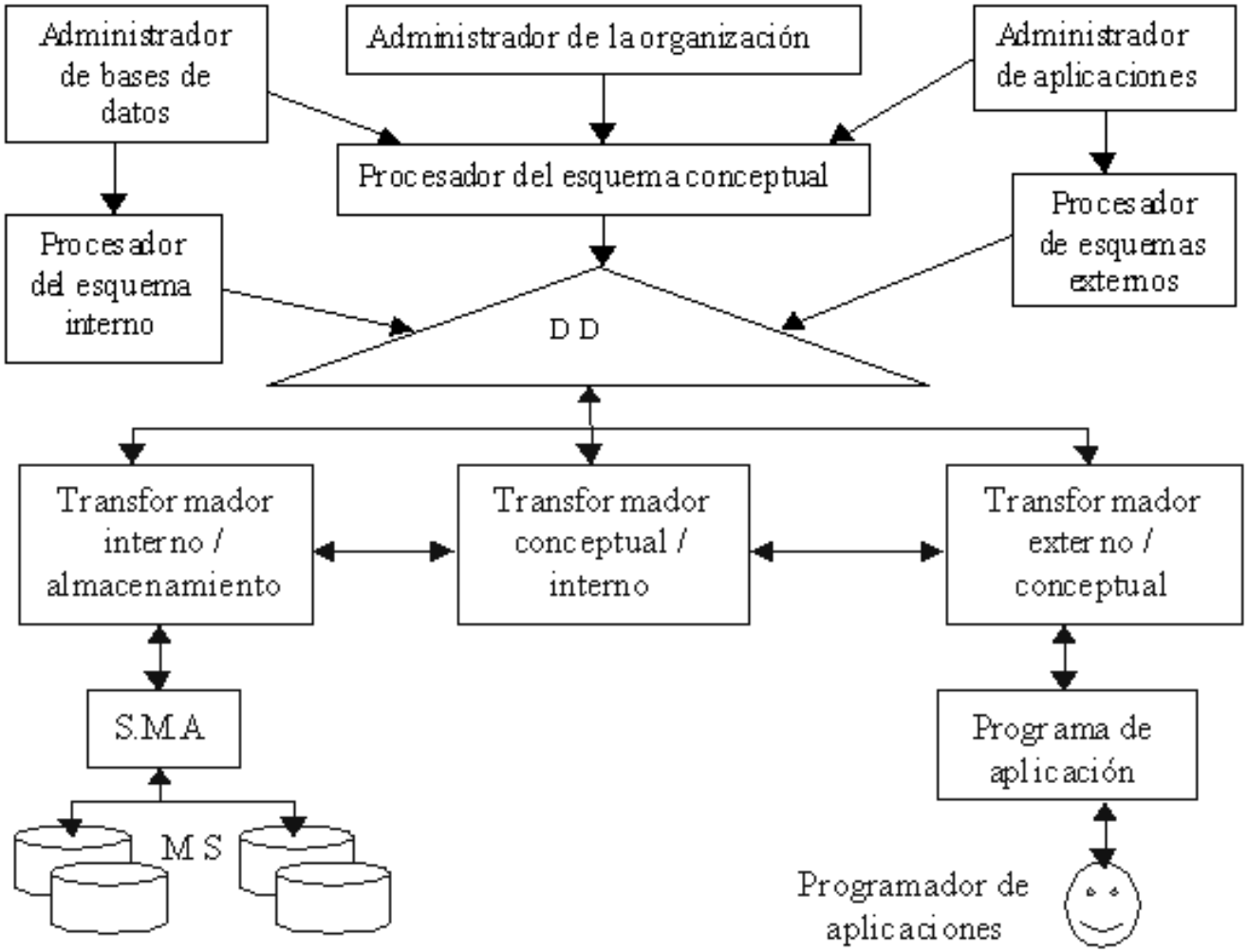
- Cada relación o tabla contiene un solo tipo de fila o tupla.
- Cada tupla tiene un número fijo de atributos o columnas.
- No se permiten atributos compuestos o grupos repetitivos.
- Cada tupla es única y se identifica con su clave primaria.
- Un atributo o grupo de ellos que identifiquen unívoca e inequívocamente cada tupla de la relación es una clave candidata.
- La clave primaria de una relación se selecciona entre las claves candidatas.
- Si un atributo A ∈ R1 es también la clave primaria de R2, entonces A es un atributo foráneo de R1.
- El orden de las tuplas en la relación es **irrelevante**.
- Los valores de los atributos deben pertenecer al dominio de cada atributo definido en ella.
- Un mismo dominio puede ser usado por diferentes atributos.
- A partir de una o más tablas se pueden producir nuevas tablas diferentes mediante el uso de las operaciones del álgebra relacional.

UNIVERSIDAD DE LOS ANDES

Reglas de Integridad

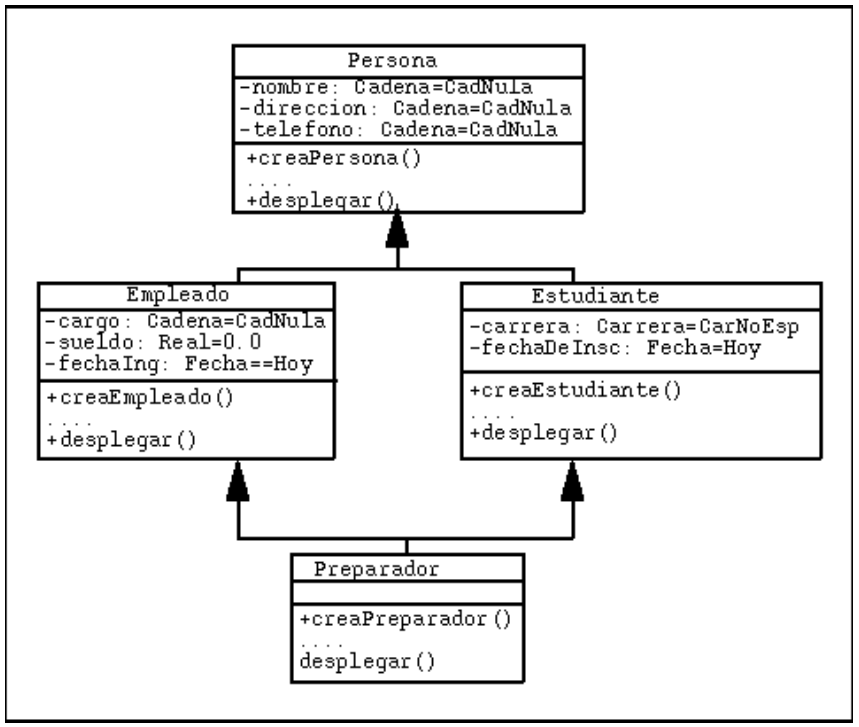
- **De la relación:** ningún componente de un valor de los atributos que conforman la clave primaria puede ser **nulo**.
- **De referencia:** sea A la clave primaria de R1 y también un atributo foráneo de R2, entonces para toda tupla de R2 donde A ≠ nulo debe existir la tupla correspondiente en R1.
- **De los valores de un atributo:** son los predicados definidos por el administrador de bases de datos sobre los valores de los atributos usando el lenguaje de definición de datos.

Arquitectura de referencia



Orientadas por Objetos

- Modelo ODMG versión 1.1 (1995)
- Primitiva: el objeto.
- Los objetos se pueden categorizar en tipos.
- Su estado lo describen los valores de sus propiedades: atributos o relaciones entre él y otros objetos.
- Su comportamiento lo define el conjunto de sus operaciones.



UNIVERSIDAD DE LOS ANDES

BDOO

- **Tipos e instancias:**
  - ✓ **Tipo** define estado y comportamiento de sus instancias.
  - ✓ **Extensión del tipo:** conjunto de todas las instancias del tipo.
  - ✓ **Características del tipo:** Estado y comportamiento.
  - ✓ **Organización de los tipos:** en un grafo de **subtipos** y **supertipos**.
  - ✓ **Subtipo:** hereda todas las características de sus supertipos, pudiendo definir nuevas características para él.
  - ✓ **Tipos abstractos:** no tienen instancias ni implementación.
  - ✓ Los tipos del sistema se organizan como un árbol
  - ✓ Los tipos definidos por el usuario pueden formar un digrafo acíclico (dag).

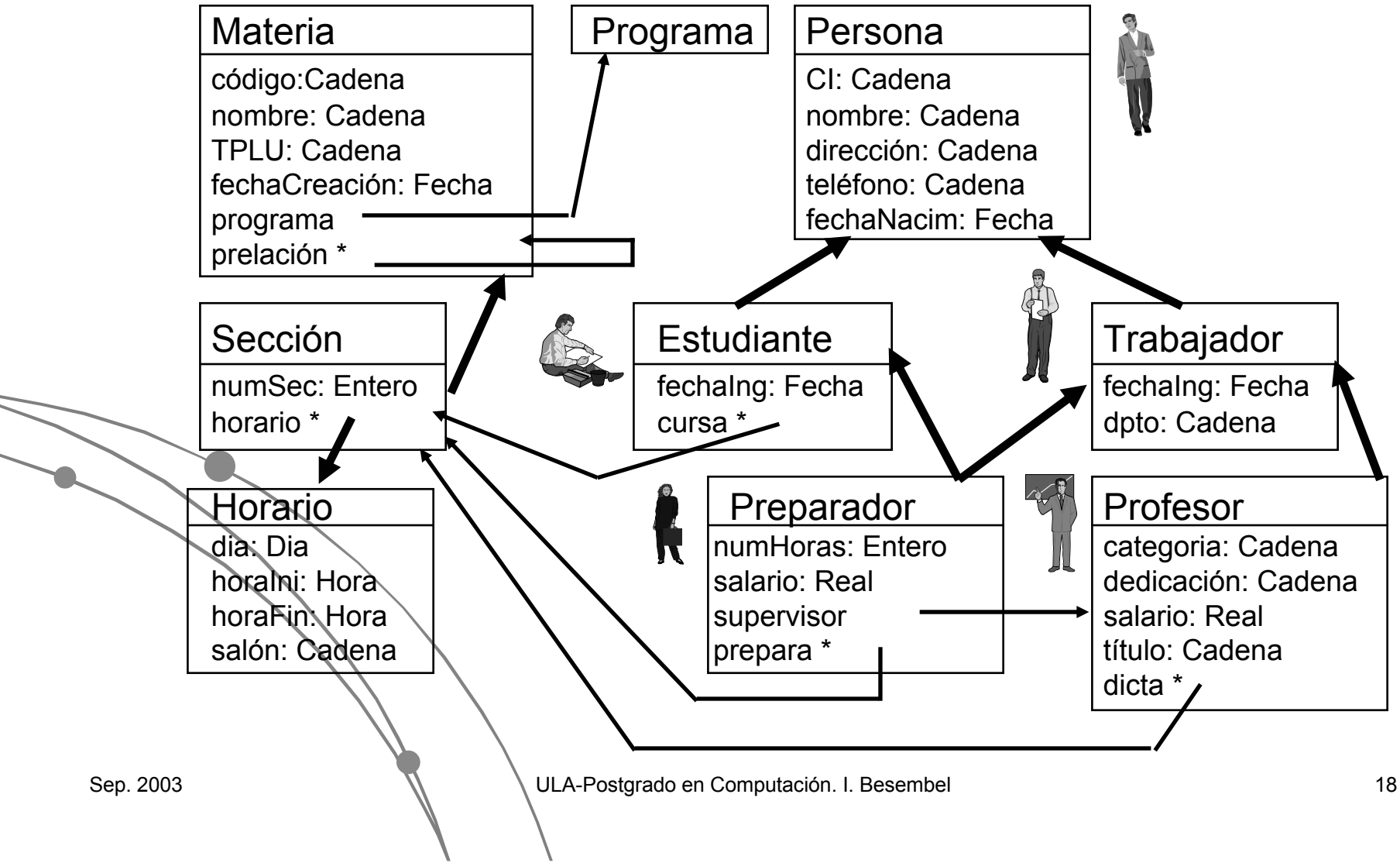
UNIVERSIDAD DE LOS ANDES

BDOO

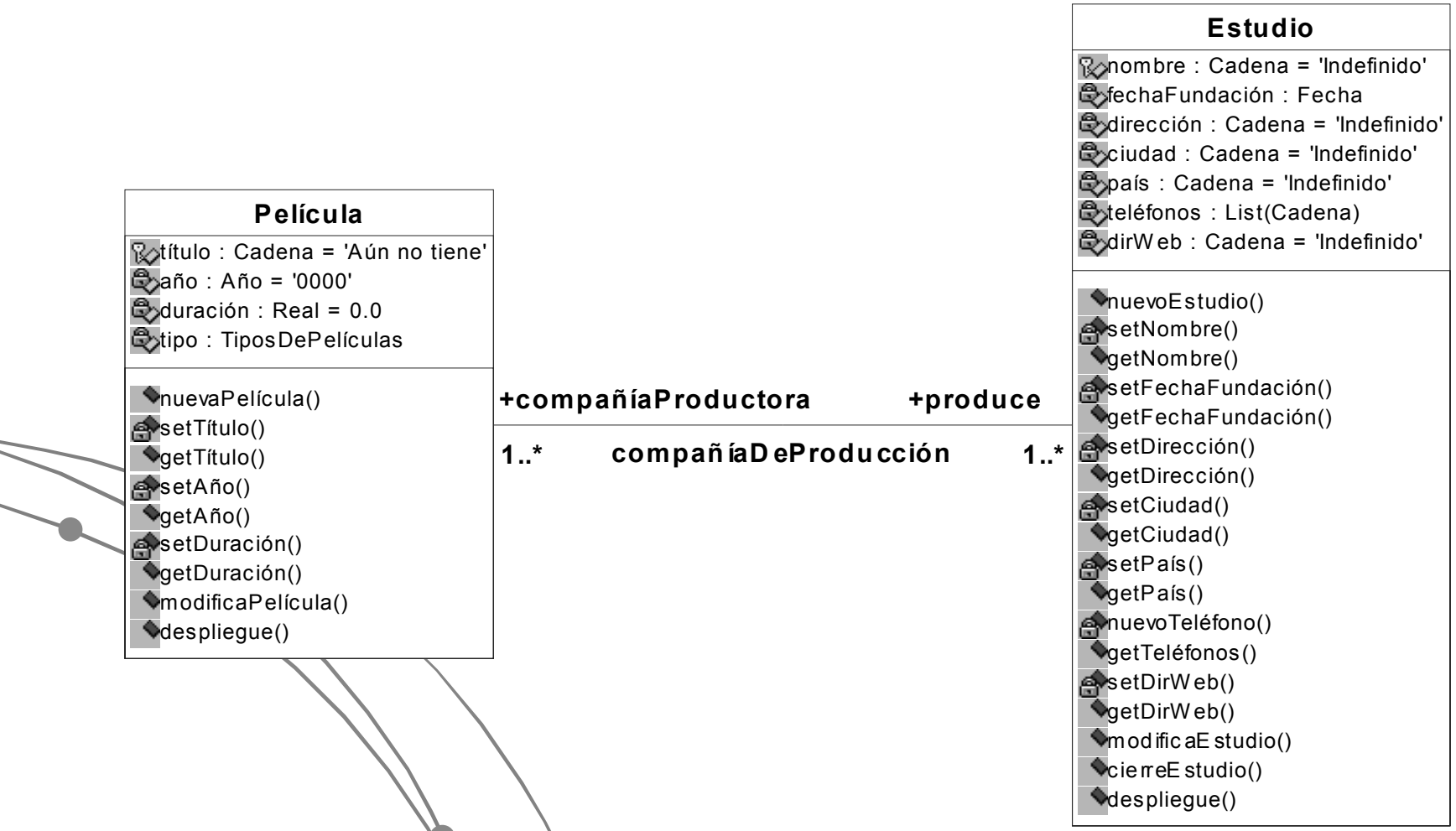
- **Atributos:** definen cada tipo de objeto, su valor es literal y no tiene oid.
- **Relaciones:** solo pueden definirse entre tipos mutables.
  - ✓ Las relaciones son binarias del tipo 1:1, 1:M y N:M.
  - ✓ No tienen nombre pero si sus caminos en cada dirección. Los caminos se declaran dentro de la interfaz del tipo donde la relación se da.
  - ✓ Mantienen la integridad referencial y no tienen oid
- **Comportamiento:** Se representa con un conjunto de operaciones.
  - ✓ Cada **operación** tiene una firma que contiene los nombres y tipos de los argumentos, excepciones y tipos y valores de regreso.
  - ✓ Siempre se aplican sobre un solo objeto.
  - ✓ No hay operaciones independientes de un tipo, ni pueden ser definidas para 2 o más tipos.
  - ✓ Las operaciones son únicas dentro de un tipo, pueden tener el mismo nombre para diferentes tipos
- **Colecciones:** tienen un número variable de elementos del mismo tipo y no tienen ranuras. Se aceptan subtipos del tipo especificado. Set, Bag, List y Array.
- **Estructuras:** tienen un número fijo de ranuras, cada una conteniendo un objeto o un literal.

UNIVERSIDAD DE LOS ANDES

BDOO



Ejemplo


Película
título : Cadena = 'Aún no tiene'
año : Año = '0000'
duración : Real = 0.0
tipo : TiposDePelículas
nuevaPelícula()
setTítulo()
getTítulo()
setAño()
getAño()
setDuración()
getDuración()
modificaPelícula()
despliegue()
+compañíaProductora
+produce
1..*
compañíaDeProducción
1..*
Estudio
nombre : Cadena = 'Indefinido'
fechaFundación : Fecha
dirección : Cadena = 'Indefinido'
ciudad : Cadena = 'Indefinido'
país : Cadena = 'Indefinido'
teléfonos : List(Cadena)
dirWeb : Cadena = 'Indefinido'
nuevoEstudio()
setNombre()
getNombre()
setFechaFundación()
getFechaFundación()
setDirección()
getDirección()
setCiudad()
getCiudad()
setPaís()
getPaís()
nuevoTeléfono()
getTeléfonos()
setDirWeb()
getDirWeb()
modificaEstudio()
cierreEstudio()
despliegue()

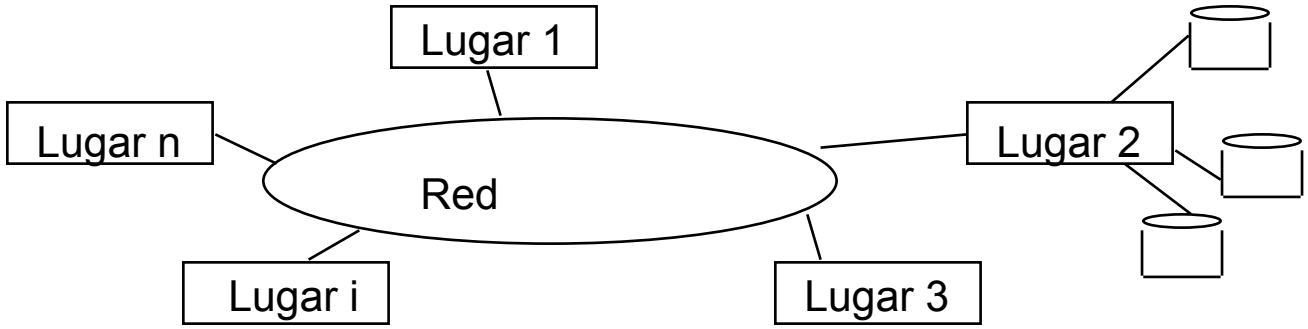
UNIVERSIDAD DE LOS ANDES

Objeto-relacional

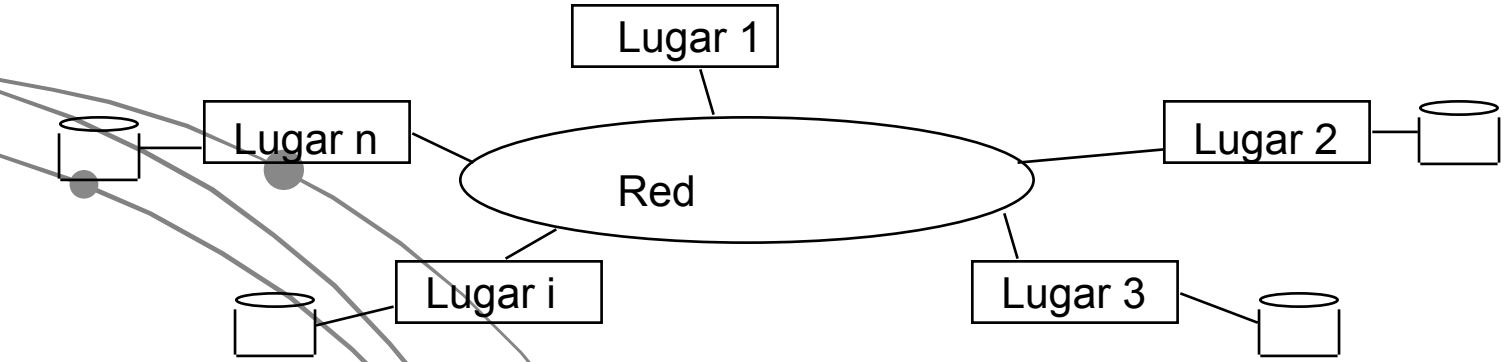
Modelo relacional extendido con facilidades de orientación por objetos:

- ✓ Nuevos tipos de datos:
  - Creación de nuevos tipos compuestos, pero que deben ser soportados por el propietario del tipo, esto es debe definir al menos dos métodos transformadores, uno para convertir el tipo nuevo a ASCII y el otro que convierte de ASCII al nuevo tipo.
  - Soporte de tipos complejos: registros, conjuntos, referencias, listas, pilas, colas y arreglos.
- ✓ Creación de funciones que tengan un código en algún lenguaje de programación, por ejemplo: SQL, Java, C, etc.
- ✓ Creación de operadores: asignándoles un nombre y asociándoselo a una función ya definida o creada con anterioridad.
- ✓ Soporte de encadenamiento dinámico y herencia en los tipos tupla o registro.
- ✓ Posibilidad de incluir el chequeo de las reglas de integridad referencial a través de los gatillos (*triggers)*
- ✓ Soporte adicional para seguridad y activación de la versión cliente-servidor.

BDR Distribuidas



BD central con acceso a la red



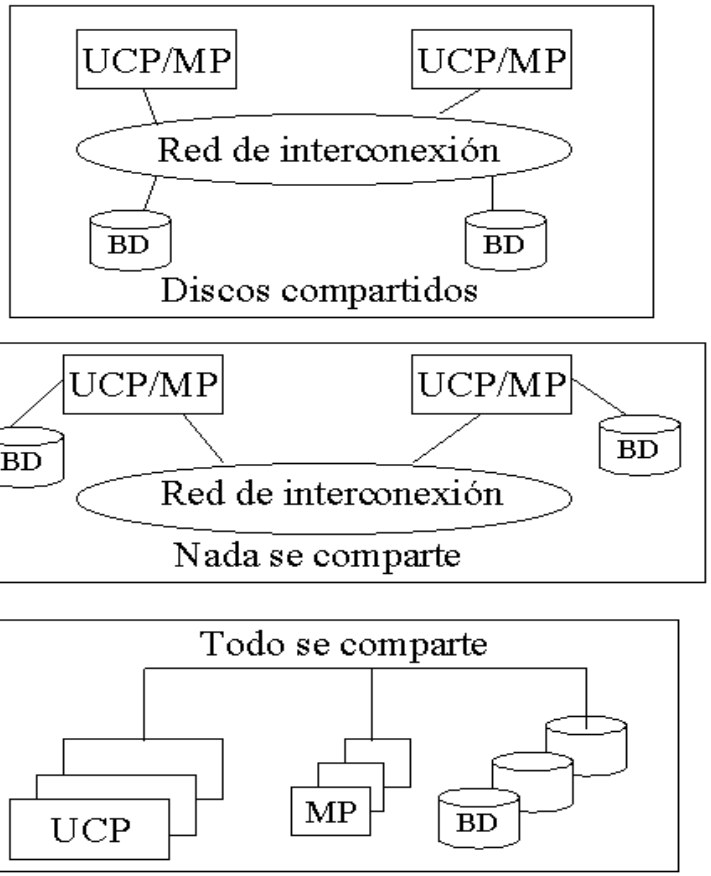
Ambiente de BDD

Problemas claves:

Para responder una consulta, los datos residentes en computadores separados deben ser combinados y movidos.

Como los datos están bajo el control de computadores separados, es necesario coordinar las Ti para preservar la consistencia de datos.

BDR Paralelas



- **Discos compartidos:** Cada procesador tiene su propia MP, pero comparte su MS con todos.
- **Nada se comparte:** Cada procesador tiene su MP y MS, pero se comunican con pase de mensajes.
- **Todo se comparte:** Todos los procesadores acceden una MP común y todas las MS (discos)

UNIVERSIDAD DE LOS ANDES

BDR Multimedia

- Reconocimiento, agrupamiento, tipos y relaciones para capturar y recuperar la información multimedia.
  - ✓ Diferencia entre recuperación en base de datos y en recuperación de información.
- Modelos hipertexto e hipermedia
  - ✓ sugieren un método de navegación para recuperar la información.
- Recuperación basada en contenido
  - ✓ Entendimiento del lenguaje natural, velocidad de procesamiento, visión y modelado de usuarios.

UNIVERSIDAD DE LOS ANDES

BDR Activas

- BD convencionales: **pasivas,** solo ejecutan consultas a petición de un usuario o de un programa de aplicación.
- BD **activa:** tiene la capacidad de monitorear situaciones de interés y cuando estas ocurren disparan una respuesta adecuada.
- El comportamiento deseado se expresa por medio de reglas de producción o reglas evento-condición-acción (ECA), las cuales pueden ser definidas y almacenadas en la BD.

on evento	**Una regla se *dispara* cuando el evento ocurre**
if condición	**se *considera* cuando su condición se evalúa**
then acción	**y se *efectúa* cuando la acción se ejecuta**

- Las reglas serán disparadas por eventos de la BD como: un estado particular de la BD y cambios de estado de la misma, para:
  - ✓ asegurar la integridad de datos,
  - ✓ implementar disparadores y alertadores,
  - ✓ mantener datos derivados,
  - ✓ asegurar las restricciones de acceso,
  - ✓ implementar las políticas de control de versiones,
  - ✓ mantener las estadísticas de acceso para la optimización de consultas, etc.

BDR Deductivas

- Compuesta de una BD intensional y una BD extensional
- **BD extensional:** base de hechos
- **BD intensional:** conjunto de restricciones de integridad y de axiomas deductivos (reglas)

madreDe	madre	hija
	Ana	Berta
	Berta	Juana
	Juana	Mariela
	Berta	Beatriz

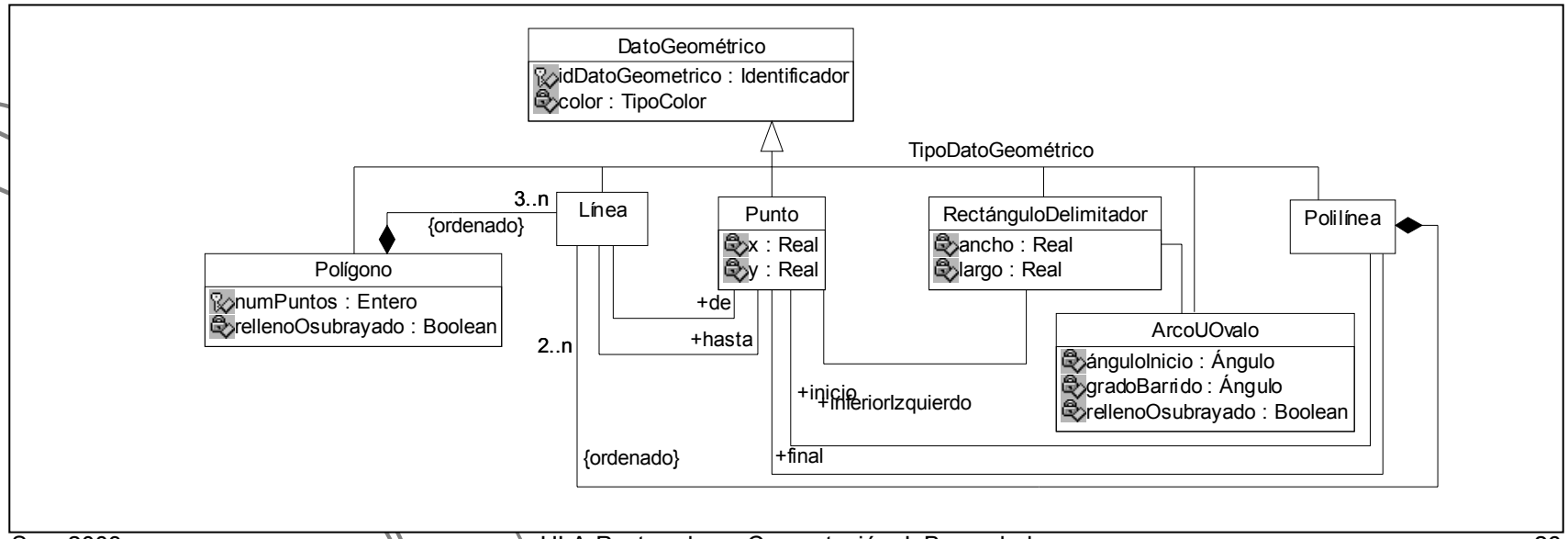
if madreDe(x, y) and madreDe(y, z)
then abuelaDe(x, z)

✓Se basa en el cálculo de predicados y en la teoría de demostraciones
✓Realiza procesamiento de consultas recursivas

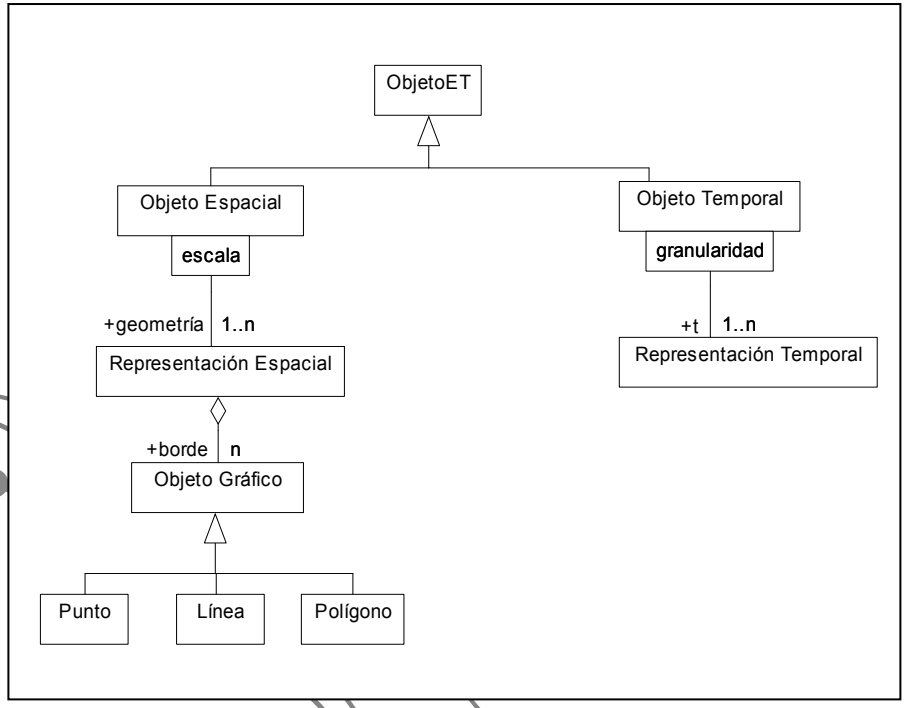
Bases de datos espaciales

Término para describir los datos que pertenecen al espacio ocupado por los objetos en la BD.

- ✓Pueden ser geométricos y variados.
- ✓Se aplica a cualquier dato de un fenómeno distribuido en 2-, 3- o n-dimensiones que son ortogonales y homogéneas (D. Peuquet)
- ✓Consisten del conjunto de objetos espaciales construidos con: puntos, líneas, regiones, rectángulos, superficies, volúmenes, etc.



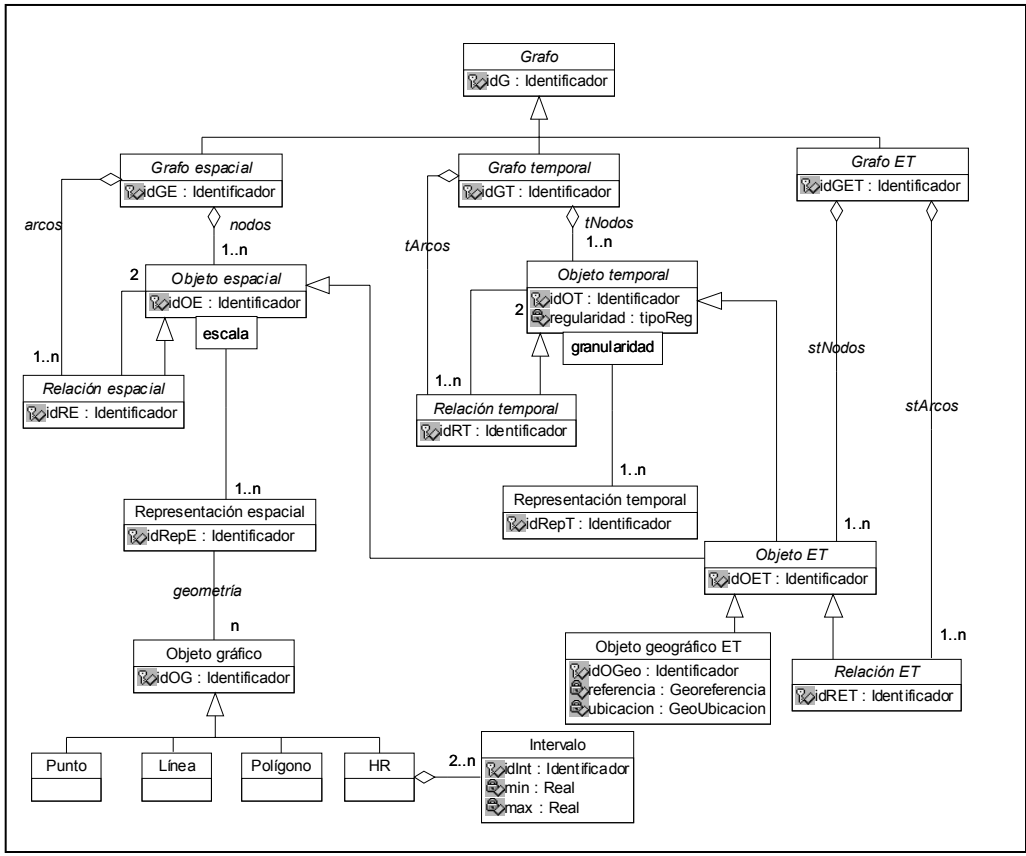
Bases de datos temporales



- **Los objetos y sus relaciones existen en periodos de tiempo definidos, así como los eventos ocurren en puntos específicos del tiempo**
- **Tiempo válido:** tiempo en el cual el hecho se dio o se dará en la realidad independientemente del tiempo en que se registró en la BD.
  - ✓ puede estar o no limitado (normalmente en el pasado)
- **Tiempo transaccional:** tiempo en que el hecho se almacenó en la BD
  - ✓ Es un intervalo que dice cuando se insertó el hecho y cuando se eliminó de la BD.
  - ✓ está siempre limitado en el pasado y en el futuro

Bases de datos espacio temporales

- Tienen al menos tiene una propiedad espacial y una temporal
- Un objeto espacio temporal se puede representar por una 4-upla *(objId, geom, t, att)* donde:
  - ***objId*** es el identificador del objeto,
  - ***geom*** es su localización y forma espacial,
  - ***t*** es la descripción de sus características temporales y
  - ***att*** es un conjunto de otros atributos que describen otras propiedades diferentes de las espaciales y temporales



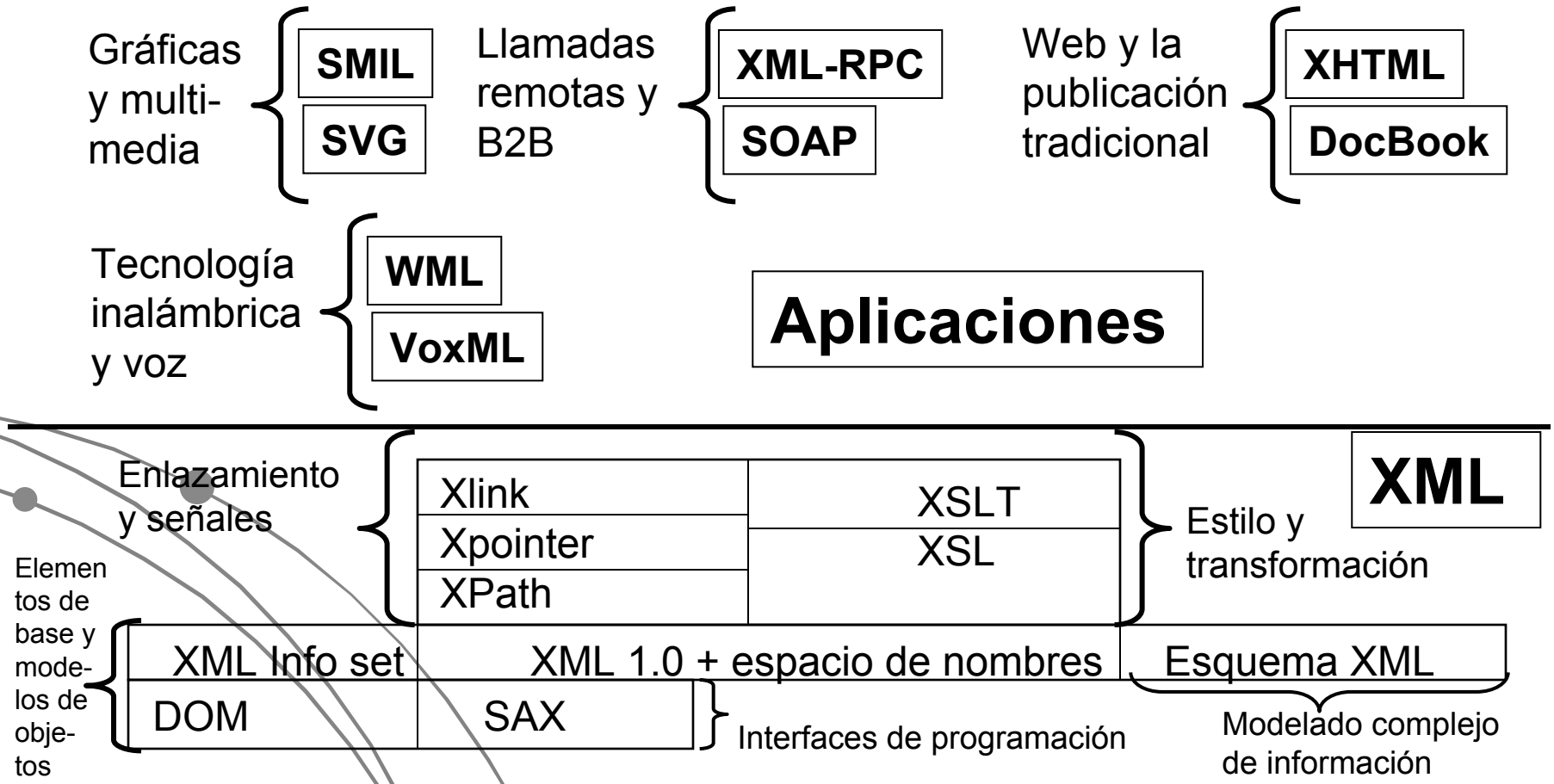
Bases de datos semi-estructuradas

- XML: lenguaje de marcas extensible, diseñado para permitir la implementación e interoperabilidad con SGML y HTML
- Procesador XML: usado para leer documentos XML y permitir el acceso a su contenido (datos) y estructura (esquema)
- Metas:
  - ✓ Usable en Internet, compatible con SGML y soporte de varias aplicaciones
  - ✓ Número mínimo de rasgos adicionales y facilidad de escribir programas que procesen documentos XML (claros y legibles)
  - ✓ Diseño XML formal, conciso y de fácil creación
  - ✓ No importa la minimalidad del conjunto de marcas

```
<INVENTARIO>
    <LIBRO>
        <TITULO>Aprenda XML ya</TITULO>
        <AUTOR>Young, Michael J.</AUTOR>
        <EDITORIAL>McGrawHill-Microsoft</EDITORIAL>
        <ANO>2000</ANO>
        <ISBN>0-7356-1020-7</ISBN>
        <TITULOINGLES>XML Step-by-step</TITULOINGLES>
    </LIBRO>
    <LIBRO>
        <TITULO>Precálculo</TITULO>
        <AUTOR>Faires, J. Douglas y DeFranza, James</AUTOR>
        <EDITORIAL>International Thomson</EDITORIAL>
        <ANO>2001</ANO>
        <ISBN>970-686-032-0</ISBN>
        <TITULOINGLES>Precalculus</TITULOINGLES>
    </LIBRO>
    <LIBRO>
        <TITULO>Multimedia. Manual de referencia</TITULO>
        <AUTOR>Vaughan, Tay</AUTOR>
        <EDITORIAL>MaGraw-Hill Osborne Media</EDITORIAL>
        <ANO>2002</ANO>
        <ISBN>84-481-3626-8</ISBN>
        <TITULOINGLES>Multimedia: Making it
                                    work</TITULOINGLES>
    </LIBRO>
</INVENTARIO>
```

UNIVERSIDAD DE LOS ANDES

XML



UNIVERSIDAD DE LOS ANDES

Lenguajes

SQL (Structured Query Language)

- ✓ Versión comercial del lenguaje SEQUEL creado por IBM para el sistema R (1976) bajo DOS y VM.
- ✓ Lenguaje declarativo, interactivo, de programación, de definición de esquemas y de comunicación con servidores a través de redes.
- ✓ Primera versión: SQL-89 contiene tres partes:
  - Lenguaje de definición de datos (LDD): contiene todas las instrucciones para definir el esquema de una BD, como son: **create, alter y drop**.
  - Lenguaje de manipulación de datos (LMD): tiene las instrucciones de manejo de las tablas como son: **select, insert, delete y update**, y para control de concurrencia como: **commit y rollback**.
  - Lenguaje de control de datos (LCD): para dar y revocar permisos de acceso a los datos de la BD, como son: **grant y revoke**.
- ✓ Las instrucciones pueden ir embebidas en programas escritos en otros lenguajes de programación, como: Cobol, Fortran, Pascal y PL/I

UNIVERSIDAD DE LOS ANDES

SQL

- Segunda versión SQL-92 o **SQL2**, incluye:
  - ✓ Uso de agentes de software,
  - ✓ Nuevos tipos básicos de datos como: **Date, Time, Timestamp, BLOB, Varchar,**
  - ✓ Se establecen conexiones cliente-servidor en sesiones concurrentes,
  - ✓ SQL dinámico,
  - ✓ Mayor granularidad a nivel de transacciones,
  - ✓ Nuevas versiones de la operación producto o conjunción,
  - ✓ Usa un catálogo y maneja códigos de error estandarizados y
  - ✓ Uso de nuevos lenguajes de programación como: C, ada y mumps.
- Última versión **SQL3** amplía o extiende el SQL2 para contener:
  - ✓ Tipos abstractos de datos definidos por el diseñador de la BD,
  - ✓ Manejo de roles de usuarios,
  - ✓ Consultas recursivas,
  - ✓ Gatillos, disparadores o *triggers*,
  - ✓ Procedimientos almacenados y encadenamiento tardío,
  - ✓ Manejo de interoperabilidad a través de un API u ODBC para acceso a bases de datos basado en el estándar SAG(SQL Access Group) y
  - ✓ Manejo de transacciones anidadas.

UNIVERSIDAD DE LOS ANDES

SQL

- LDD
  - **create table** nombreTabla **(** nombreAt tipoAt **default** valorPorOmisión **[null | not null | identity],**... [**constraint** nombreRestricción] **check (** Q **),** [**constraint** nombreRestricción] **primary key (** at1 [, at2, ...]**),** [**constraint** nombreRestricción] **unique (** at1 [, at2, ...]**),** [**constraint** nombreRestricción] **foreign key (** listaAt**) references** nombreTabla **(** listaAt **))**
  - **drop table** nombreTabla
  - **create view** nombreVista [(at1 [, at2 ...]) **as** instrucciónSelect [**with** opciónVerificación]
  - Etc.
- LCD
  - **grant** { **select** | **insert** | **update** | **delete** | **all** } **on** { nombreTabla | nombreVista } **to** listaCódigosUsuario
  - **revoke** { **select** | **insert** | **update** | **delete** | **all** } **on** nombreTabla **from** listaCódigosUsuario

UNIVERSIDAD DE LOS ANDES

SQL

- LMD
  - **select** {listaAt | *}[**into** nombreTabla] **from** listaTablas
    [**where** Q']
    [**group by** listaAt]
    [**having** Q]
    [**order by** listaAt]
  - **insert** [**into**] nombreTabla [**(** listaAt **)**] {**values (** listaValores **)** | instrucciónSelect }
  - **delete** {[**from**] nombreTabla [**where** Q] | **from** listaTablas [**where** Q'] }
  - **update** nombreTabla **set** at1 = {expresión | instrucciónSelect } [, at2 = {expresión | instrucciónSelect}...] [**from** listaTablas] [**where** Q']

UNIVERSIDAD DE LOS ANDES

QUEL

- Basado en el Cálculo Relacional de Tuplas
- Presentado por Zook en 1977
- Lenguaje del Ingres
- LMD
  - **Range of** <listaVariables> **is** <listaTablas>
  - **Retrieve** [[**into**] <nombreTabla>]<listaAtributosVariable>
    [**where** <condición>]
  - **Append** [**to**] <variable> (at = valor, ...) [**where** <condición>]
  - **Replace** <variable> (<listaDeAtributos>) [**where** <condición>]
  - **Delete** <variable> [**where** <condición>]

UNIVERSIDAD DE LOS ANDES

QBE

- Basado en el Cálculo Relacional de Dominios
- Lenguaje del Sistema R, DB2, Informix, Oracle, etc.
- LMD no es computacionalmente completo

Relación	col1	col2	col3	…	coln
	P.		_x		P.AO._y
		_x			>=14
		P.	P.SUM._s		
I.	126	‘5101’	28		230
D.			>30		
	=45		_w		
U.			_w+5		

UNIVERSIDAD DE LOS ANDES

OQL

¿Cuáles son los estudiantes de la sección Nro. 2 de la materia de nombre “Simulación”, cuál es su preparador y su profesor?

- LMD
  - ✓ SELECT DISTINCT STRUCT(Cédula: e.CI, nombre: e.nombre), p.nombre, pr.nombre
    FROM e IN Estudiantes, p IN Preparadores, pr IN Profesor, s IN Secciones
    WHERE s.nombre = ‘Simulación’ AND s.numSec = 2
- LDD
  - **interface Materia (extent Materias keys código, nombre): persistent**

```
{ attribute String código;
    attribute String nombre;
    attribute String tipo;
    attribute String tplu;
    attribute String objetivos;
    attribute String contenidos;
    relationship List< Curso> seOfrecen inverse Curso::esSecciónDe {order_by Curso::sección };
    relationship Set< Materia> prela inverse Materia::esPreladaPor;
    relationship Set< Materia> esPreladaPor inverse Materia::prela;
Boolean nuevaMateria( ) raises (yaEstá);
Boolean cambioDeNombre( ) raises (repetido);
Boolean cambioDeObjetivos( ) raises (repetido);
Boolean cambioDeContenido( ) raises (repetido);
};
```

UNIVERSIDAD DE LOS ANDES

OSQL

¿Cuáles son los estudiantes de la sección Nro. 2 de la materia de nombre “Simulación”, cuál es su preparador y su profesor?

- LMD
  - ✓ SELECT DISTINCT CI(e), nombre(e), nombre(p), nombre(pr)

    FOR EACH Estudiante e, Preparador p, Profesor pr, Sección s

    WHERE nombre(s) = ‘Simulación’ AND numSec(s) = 2

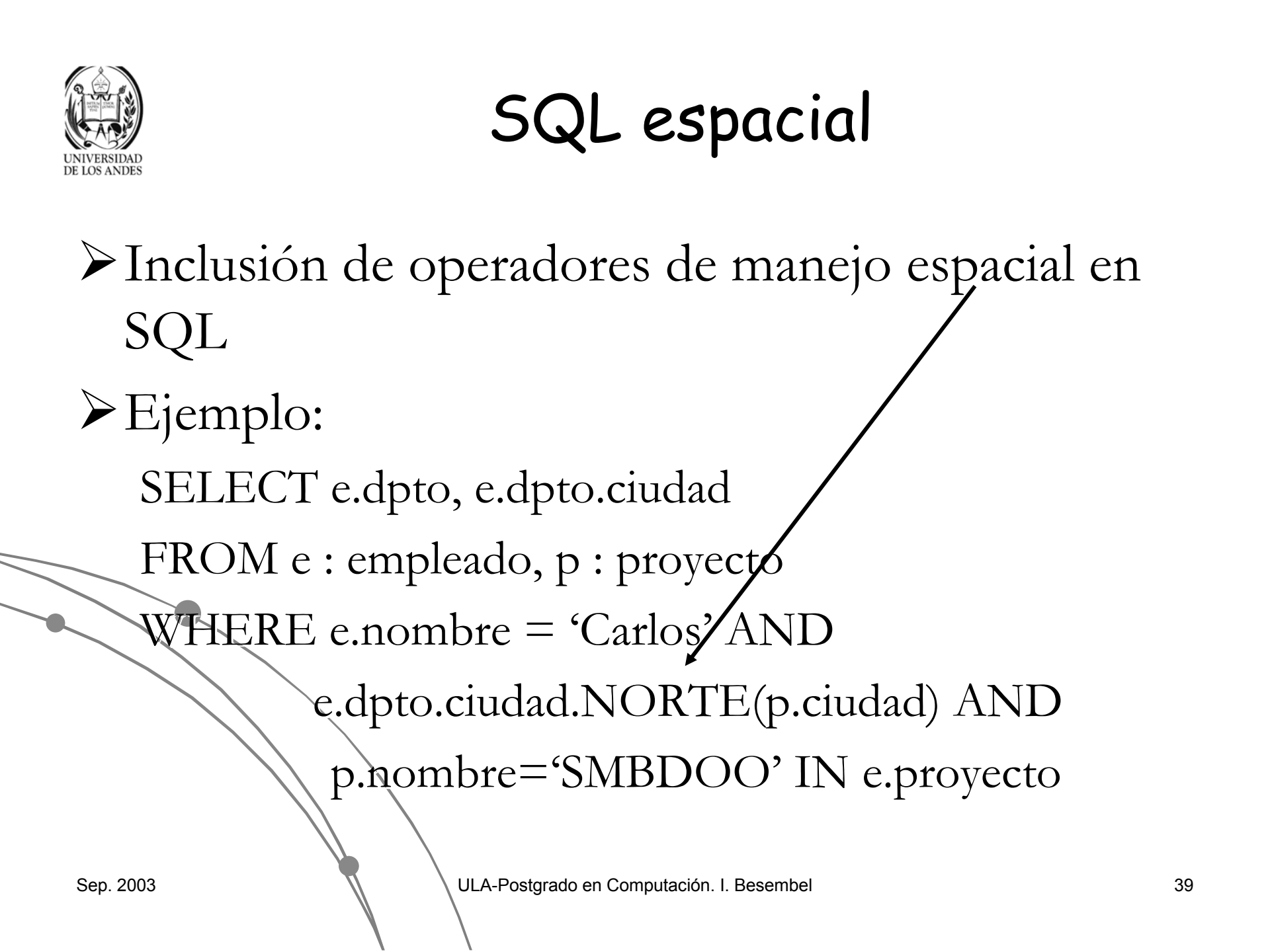
    ORDER BY CI(e);
- LDD
  - create type Estudiante subtype of Persona functions (CI Char (var 10));

UNIVERSIDAD DE LOS ANDES

SQL espacial

- Inclusión de operadores de manejo espacial en SQL
- Ejemplo:

```
SELECT e.dpto, e.dpto.ciudad
FROM e : empleado, p : proyecto
WHERE e.nombre = 'Carlos' AND
        e.dpto.ciudad.NORTE(p.ciudad) AND
        p.nombre='SMBDOO' IN e.proyecto
```



UNIVERSIDAD DE LOS ANDES

SQL temporal

- En el modelo de datos: Algunos lenguajes incluyen cláusulas y/o operadores para manejar el tiempo.
- Ejm: TOSQL, TOOSQL y OOSTSQL.

```
SELECT e.dpto, e.dpto.vt
FROM e : empleado, p : proyecto
WHERE e.nombre = 'Carlos'
WHEN e.dpto.vt.DURING(e.proyectos.vt) AND
        p.nombre='SMBDOO' IN e.proyecto
```

UNIVERSIDAD DE LOS ANDES

Conclusiones

- Tecnología básica para el mantenimiento y control de los datos soportada por los SGBD
- Disponibilidad de uso a través de Web
- Base para la integración de aplicaciones
- Diversidad de tipos y posibilidad de mezclarlos
- Uso extendido actualmente
- Evolución continua