



FACULTAD DE CIENCIAS Y TECNOLOGÍA

Ingeniería en Ciencias de la Computación  
Ingeniería en Tecnologías de la Información y  
Seguridad

## Capítulo II

# MODELO DE DATOS



Base de Datos

Resultado			
PRODUCT ID	PRICE LIST	QTY	DATE

*Ing. Edgar T. Espinoza R.*



¿Por qué son tan importantes las RN en el diseño de bases de datos?

- Ayudan a definir los elementos de un modelo de datos: Entidades, restricciones y relaciones entre entidades.
- Ayudan a estandarizar la visión de la organización con respecto a la administración de sus datos.
- Ayudan al diseñador a entender la naturaleza de los datos.
- Permiten la creación de modelos de datos adecuados.

## Evolución de los modelos de datos



La siguiente tabla muestra un resumen de los tipos de modelos de datos existentes a lo largo del tiempo.

Tiempo	Nombre
50s < t < 70s	Sistemas de archivos
50s < t < 80s	Modelo Jerárquico
60s < t < 90s	Modelo de Red
70s < t	Modelo Relacional => <b>(RDBMS)</b>
75 < t	Modelo Entidad - Relación
80s < t	Modelo orientado a objetos => <b>(OODBMS)</b>
90s < t	<ul style="list-style-type: none"><li>Modelo orientado a objeto / Relacional (<b>ORDBMS</b>)<ul style="list-style-type: none"><li>ORMs (Object Relational Mapping)</li><li>Soporte para procesamiento de documentos XML.</li><li>Soporte para procesamiento de documentos JSON</li><li>Bases de datos Multidimensionales</li></ul></li></ul>
2000<t	<ul style="list-style-type: none"><li>Modelos NoSQL</li></ul>

## Sistema de Archivos



### Almacenamiento manual:

Históricamente no existían sistemas, dichas tareas se realizaban de forma manual. Los datos de las empresas se escribían en papel y se almacenaban en folder o carpetas almacenadas en grandes estantes. Esta estrategia se vuelve problemática conforme la empresa crece y por lo tanto la cantidad de datos a mantener



### Almacenamiento empleando sistemas computacionales:



Surge entonces un nuevo rol llamado “Especialista de procesamiento de datos” (DP) encargado de crear un sistema basado en computadora encargado de almacenar los archivos y generar los reportes.

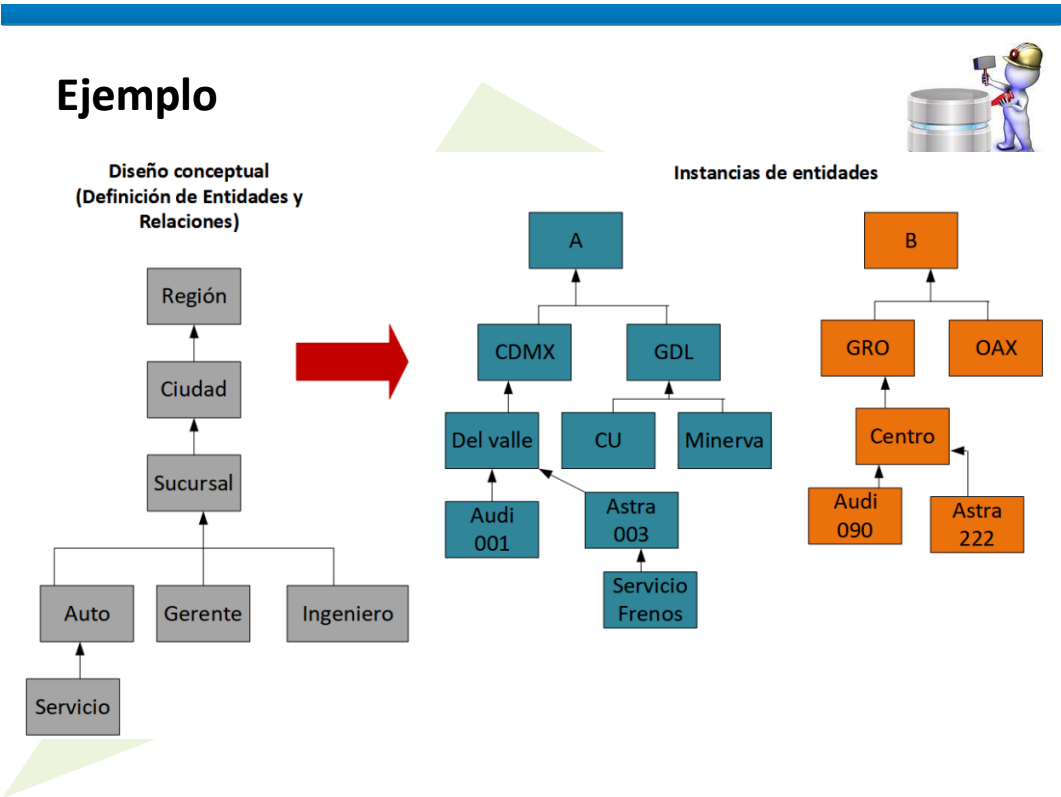
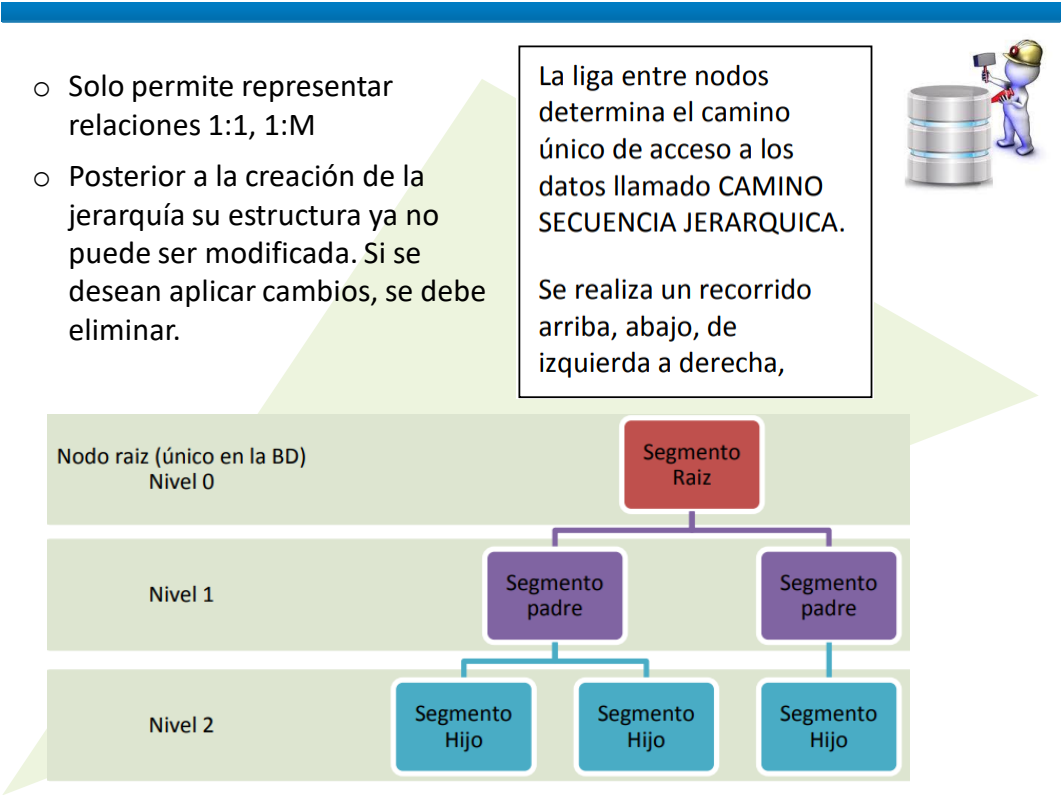
C_NAME	C_PHONE	C_ADDRESS	C_ZIP	A_NAME	A_PHONE	TP	AMT	REN
Alfred A. Ramas	615-844-2573	218 Fork Rd., Babs, TN	36123	Leah F. Hahn	615-882-1244	T1	100.00	05-Apr-2014
Leona K. Dunne	713-894-1238	Box 12A, Fox, KY	25246	Alex B. Alby	713-228-1249	T1	250.00	16-Jun-2014
Kathy W. Smith	615-894-2285	125 Oak Ln, Babs, TN	36123	Leah F. Hahn	615-882-2144	S2	150.00	29-Jan-2015
Paul F. Olowski	615-894-2180	217 Lee Ln., Babs, TN	36123	Leah F. Hahn	615-882-1244	S1	300.00	14-Oct-2014
Myron Orlando	615-222-1672	Box 111, New, TN	36155	Alex B. Alby	713-228-1249	T1	100.00	28-Dec-2014
Amy B. O'Brian	713-442-3381	387 Troll Dr., Fox, KY	25246	John T. Okon	615-123-5589	T2	850.00	22-Sep-2014
James G. Brown	615-297-1228	21 Tye Rd., Nash, TN	37118	Leah F. Hahn	615-882-1244	S1	120.00	25-Mar-2015
George Williams	615-290-2556	155 Maple, Nash, TN	37119	John T. Okon	615-123-5589	S1	250.00	17-Jul-2014
Anne G. Farriss	713-382-7185	2119 Elm, Crew, KY	25432	Alex B. Alby	713-228-1249	T2	100.00	03-Dec-2014
Olette K. Smith	615-297-3809	2782 Main, Nash, TN	37118	John T. Okon	615-123-5589	S2	500.00	14-Mar-2015

### Modelo de datos Jerárquico:



Desarrollado en la década de los 60s para administrar grandes cantidades de datos para proyectos complejos como fue el proyecto del Cohete Apollo Rocket que aterrizó en la luna en 1969.

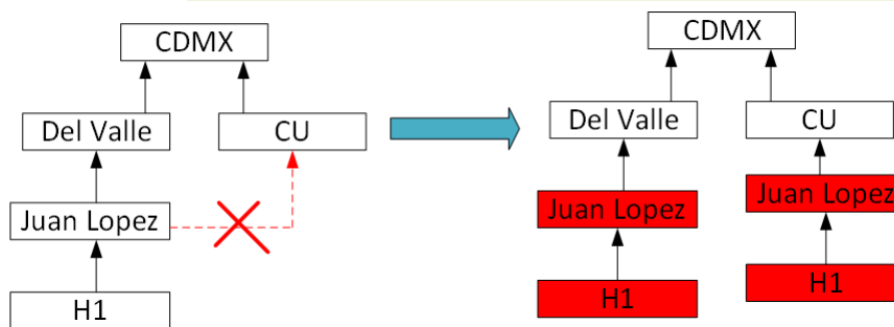
- Representa el primer modelo de datos empleado en un DBMS comercial formado por una estructura de árbol.
- No existen estándares ni formalizaciones del modelo jerárquico como en el caso del relacional.
- Sistema principal que implementa a este modelo es IMS (Information Management System) de IBM el cual emplea un lenguaje de datos DL/I



## Problemas del modelo jerárquico



Redundancia al no permitir que un nodo tenga más de un padre, por ejemplo, representación de relaciones M:N Una agencia tiene varios agentes de ventas , un agente puede trabajar en varias agencias:



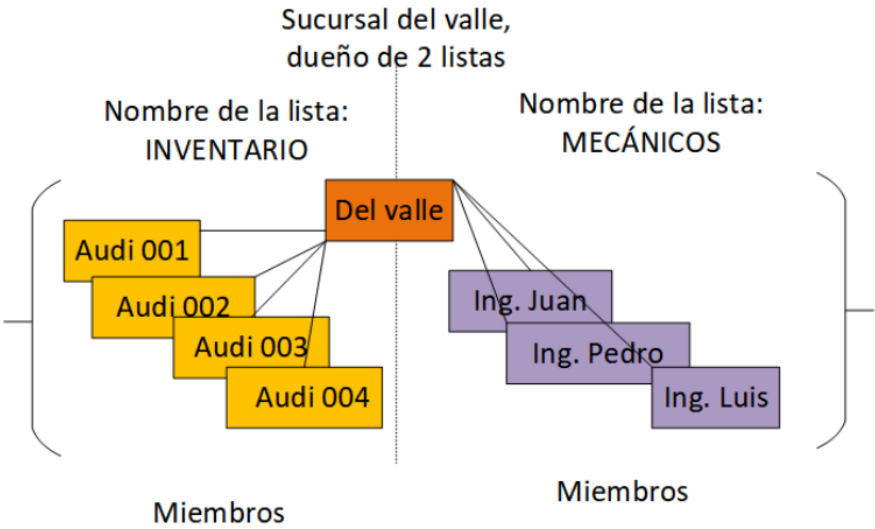
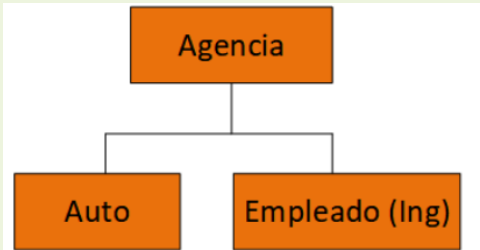
## Modelo de Red:



- Creado para representar relaciones complejas de una forma más efectiva con respecto al modelo jerárquico principalmente para mejorar desempeño.
- Similar al jerárquico, las entidades se representan como nodos y sus relaciones son las líneas que los unen.
- Ejemplos de este modelo: CODASYL. Fue desarrollado en 1971 por un grupo conocido como CODASYL (Conference on Data System Languages, Data Base Task Group)



Las relaciones entre instancias de una entidad deben ser descompuestas en un conjunto de listas.



En la figura anterior se tiene que una instancia de Agencia llamada **Dueño** (Owner) se asocie con un conjunto de Autos llamados **Miembros**. Lo mismo ocurre para los Ing. Mecánicos.

---

## Modelo Relacional



- Propuesto por E. F. Codd en 1970.
- El modelo no hace uso de punteros como en el caso del modelo jerárquico y de red.
- Las entidades se pueden acceder de forma directa sin utilizar jerarquías
- Hace uso de conceptos de lógica, teoría de conjuntos, conceptos matemáticos

---

## Modelo Entidad Relacion



- Propuesto por Peter P. Chen en 1976
- Empleado para construir una vista unificada de los datos empleando un enfoque natural fácil de entender para un usuario final e independiente al tipo de modelo de datos.
- En 1988 ANSI lo seleccionó como modelo estándar para los sistemas de diccionarios de recursos de información.
- Comúnmente empleado para realizar el diseño conceptual de una base de datos

# Bases de Datos Orientadas a Objetos (OODBMS)



- En estos manejadores ya no se requiere realizar un mapeo entre el modelo de objetos y el modelo relacional.
- Los objetos de una aplicación se guardan como tal en la base de datos.
- No existen tablas, ni SQL relacional.
- Existe un solo modelo, que en este caso es el modelo de objetos.

Ejemplos de algunas Implementaciones:

- Versant Object Database (antes DB4O)
- ObjectDatabase++ (ODBPP)
- ObjectDB
- Objectivity/DB

## Ejemplos:

Operaciones básicas empleando OODBMS



### Guardando objetos:

```
Piloto piloto;  
piloto = new Piloto("Schumacher", 100);  
db.store(piloto);  
System.out.println("Stored " + piloto);
```

### Recuperando objetos:

```
Piloto pilotoEjemplo;  
pilotoEjemplo = new Piloto("Juan", 2);  
ObjectSet result;  
result = db.queryByExample(pilotoEjemplo);  
listResult(result);
```

Observar lo sencillo que resulta realizar operaciones básicas a la base de datos gracias a la eliminación del modelo relacional.



## Modelos Objeto – Relacional (ORDBMS).



Los ORDBMS son RDBMS agregando funcionalidades asociadas con conceptos de la programación orientada a objetos (POO).

Entre sus principales características se mencionan las siguientes:

- Tipos de datos personalizados

```
create type Persona (nombre varchar(20), dirección varchar(20))
```

- Herencia entre tablas (no todos los manejadores la implementan)

```
create type Estudiante under Persona (curso varchar(20), departamento varchar(20))
```

- Funciones definidas por el usuario.

```
create function myFunction(p1,P2) BEGIN .... END;
```



En un ORDBMS todos los elementos de la base de datos son considerados como objetos con sus 2 elementos similar a un objeto en POO

- Características -> atributos.
- Comportamiento -> métodos.

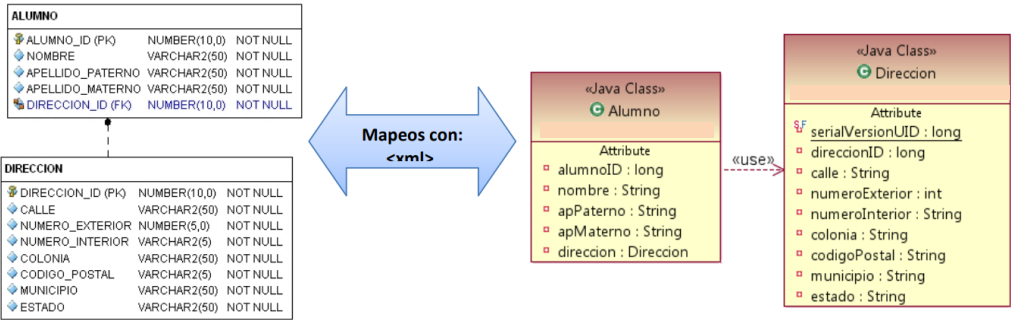
Una tabla, registro, columna, vista, índice, usuario, todos son considerados como objetos.

## Mapeos Objeto – Relacional ORMs



- Para poder sincronizar el estado de un conjunto de **objetos** en memoria (valores de sus atributos) con los valores de las columnas de un conjunto de **tablas** en el modelo relacional se requiere programar y ejecutar sentencias SQL que permitan comunicar a ambos modelos: **POO y RDBMS**.
- Un ORM permite realizar esta actividad sin la necesidad de realizar programación de sentencias SQL.
- El ORM las genera a partir de mapeos especificados ya sea en XML o por anotaciones.

Ejemplos de implementación de ORMs:  
Hibernate,  
JPA.



## Mapeo con XML y anotaciones



- ❖ Mapeos con XML
- ❖ Mapeos con anotaciones

```
<hibernate-mapping package=" bo.edu.usfx.fcyt.entidades ">
  <class name="Alumno" table="ALUMNO" dynamic-update="true" >
    <id name="alumnoID" column="alumno_id">
      <generator class="sequence" name="ALUMNO_SEQ"/>
    </id>
    <property name="nombre" type="string" column="nombre" />
    <property name="apPaterno" column="ap_paterno" />
    <property name="apMaterno" column="ap_materno" />
    <many-to-one name="direccion" class="Direccion"
      column="direccion_id" cascade="save-update"/>
  </class>
</hibernate-mapping>
```

## Mapeo con XML y anotaciones



```
@Entity
@Table(name="ALUMNO")
@SequenceGenerator(name="seq" sequenceName="ALUMNO_SEQ")
public class Alumno {

    @Id
    @Column(name="ALUMNO_ID")
    @GeneratedValue(strategy=SEQUENCE,generator="seq")
    private long alumnoID;

    @Column(name="NOMBRE")
    private String nombre;

    @Column(name="APELLIDO_PATERNO")
    private String apPaterno;

    @Column(name="APELLIDO_MATERNO")
    private String apMaterno;

    @ManyToOne(fetch=FetchType.LAZY)
    @JoinColumn(name="direccion_id")
    private Direccion direccion;
```

## Soporte de los ORDBMS para procesar documentos XML



- ❖ Este soporte está representado por extensiones implementadas como parte de las características de un ORDBMS para almacenar documentos XML en columnas con tipo de dato XML.
- ❖ La gran mayoría de los manejadores relacionales actuales soportan tipos de datos XML.
- ❖ Para realizar operaciones sobre los datos que contiene el documento XML se emplea el estándar **SQL/XML** y herramientas como XPath, XQuery.

## Ejemplo



- ❖ Creación de una tabla con una columna tipo XML que permite almacenar documentos XML

```
create table empleado (  
  empleado_id  number(3),  
  datos_empleado xmltype  
);
```



- ❖ Código empleado para insertar un registro en el que el valor de la columna datos\_empleado es un documento XML.

```
insert into empleado (empleado_id, datos_empleado)
values(
  1, xmltype('
    <empleado>
      <nombre>daniel</nombre>
      <fecha_nacimiento>12/1/1951</fecha_nacimiento>
      <email>damorgan@u.washington.edu</email>
    </empleado>')
);
```



- ❖ Código empleado para realizar consultas. Observar que es posible hacer referencia a los atributos y elementos del documento XML. El RDBMS analiza el documento, extrae los datos y los presenta en formato tabular.

```
select empleado_id, xmlquery(
  'for $i in /empleado
  where $i /nombre= "daniel"
  order by $i/nombre
  return $i/nombre'
  passing by value datos_empleado
  returning content
) xmldata
from datos_empleado;
```

```
<DEPARTMENT deptid="15" deptname="Sales">
  <EMPLOYEE>
    <EMPNO>10</EMPNO>
    <FIRSTNAME>CHRISTINE</FIRSTNAME>
    <LASTNAME>SMITH</LASTNAME>
    <PHONE>408-463-4963</PHONE>
    <SALARY>52750.00</SALARY>
  </EMPLOYEE>
  <EMPLOYEE>
    <EMPNO>27</EMPNO>
    <FIRSTNAME>MICHAEL</FIRSTNAME>
    <LASTNAME>THOMPSON</LASTNAME>
    <PHONE>406-463-1234</PHONE>
    <SALARY>41250.00</SALARY>
  </EMPLOYEE>
</DEPARTMENT>
```

DEPTID	DEPTNAME
15	Sales

DEPTID	EMPNO	FIRSTNAME	LASTNAME	PHONE	SALARY
15	27	MICHAEL	THOMPSON	406-463-1234	41250
15	10	CHRISTINE	SMITH	408-463-4963	52750

## Soporte de los ORDBMS para procesar documentos JSON



De forma muy similar a documentos XML, la mayoría de los ORDBMS actuales implementan el estándar SQL/JSON que permite almacenar documentos JSON.

Ejemplo:

```
create table orden_detalle(  
  id varchar2 (32) not null primary key,  
  fecha_orden timestamp (6) with time zone,  
  productos varchar2 (32767)  
  constraint ensure_json check (productos is json)  
);
```

```
insert into detalle_orden(id,fecha_orden,productos) values (  
  1,to_date('30-dec-2014'),  
  '{ "numorden" : 1600,  
    "referencia" : "abull-20140421",  
    "solicitante" : "alexis bull",  
    "usuario" : "abull",  
    "centrocostos" : "a50",  
    "instruccionesentrega" : {...},  
    "instruccionesespeciales" : null,  
    "habilitaentregasparciales" : true,  
    "productos" : [...]  
  }'  
);
```

Código empleado para realizar una consulta SQL para obtener datos del documento JSON.

```
select do.productos.referencia from detalle_orden do;
```

## Bases de datos multidimensionales



- Se emplean principalmente para aplicaciones OLAP como parte de las herramientas empleadas en el área de **Business Intelligence**.
- Las dimensiones del BD multidimensional se implementan a través del concepto de Cubo OLAP.
- A partir de la construcción del cubo se generan tablas (generalmente cantidades mayores de columnas) con las que se realiza el análisis y consultas de forma rápida y eficiente.
- Los datos de la tabla resultante ya no se pueden modificar. Se debe regenerar o rediseñar el cubo

Dimensión:

- (Por Periodo, Por Producto)

Jerarquías:

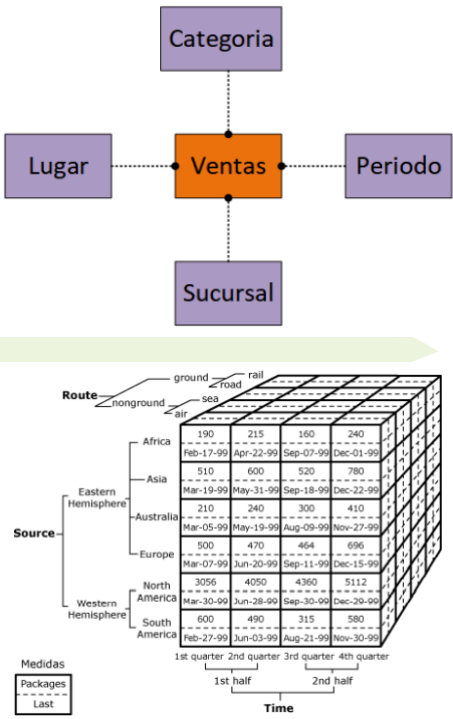
- (Año->Semestre->Mes->Semana),  
(Categoría->Línea->Marca)

Hechos:

- (Ventas, Inventario, Defectos, Devoluciones)

Métricas:

- (PD:=Devoluciones/Ventas, %Defectos)



## Big Data

La necesidad para administrar y ser capaces de atender las actuales demandas y tendencias en cuanto a la rapidez de crecimiento de los datos, desempeño, escalabilidad, y bajos costos ha disparado un fenómeno conocido como Big Data.




- Imaginar almacenar y analizar cada clic y actividad que realiza un cliente en un sitio web altamente concurrido para presentarle ofertas con base a sus gustos.
- Imaginar los requerimientos que implementa Facebook, YouTube, Twitter, o cualquier otra red social para almacenar el contenido producido en un día...
- Imaginar la capacidad de análisis de Amazon para ofrecer sugerencias a sus clientes en diferentes sitios web ...
- Imaginar los requisitos para que ideas como "Internet de las cosas" puedan ser una realidad...



- Incremento considerable del volumen de información a almacenar: Terabytes por día o por horas.
- Facilidad para consultar grandes cantidades de datos de una forma simple y eficiente.
- Necesidad de distribuir el procesamiento a través de clusters para mejorar el desempeño (escalamiento horizontal).
- Posibilidad de almacenar, analizar y procesar datos que no cuentan con una estructura definida (datos no estructurados), por ejemplo, bitácoras, datos que provienen de dispositivos, imágenes, videos, etc.
- Al hablar de este término, lo primero que se viene a la mente es pensar en volumen de datos.









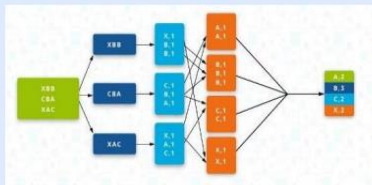


- Volúmenes grandes de datos no es la única característica de BigData.
- Existen 4 principales características que definen el término:
  - Volumen
  - Velocidad
  - Variedad o diversidad
  - Valor.

A nivel general, Big Data se refiere a un movimiento que surge para encontrar nuevas y mejores formas para administrar grandes cantidades de datos y al mismo tiempo ofrecer un alto desempeño, escalabilidad a costos razonables.



Las siguientes tecnologías han permitido a las empresas procesar masivas cantidades de datos en múltiples formatos a costos aceptables.

Tecnología	Ejemplos.
Incorporación de modelos y bases de datos NoSQL	Basados en documentos, basadas en clave llave valor, etc.
Frameworks para procesar cantidades masivas de datos distribuidos en clusters	Apache Hadoop, Spark 
Sistemas de archivos distribuidos que permitan almacenar cantidades masivas de datos	Hadoop Distributed File System (HDFS) 
Frameworks que permitan comunicar sistemas distribuidos a través de colas de mensajes	Apache Kafka 
Nuevas estrategias y técnicas de programación que permitan el procesamiento de cantidades masivas de datos de forma distribuida.	MapReduce 

## Bases de datos y modelos NoSQL



Surgen como respuesta para implementar nuevos requerimientos de una forma más eficiente y menos costosa de lo que un RDBMS puede normalmente realizar.

- Sistemas que no requieren un modelo con una estructura fija y rígida como lo es el relacional.
- Sistemas que no requieren un estricto control de concurrencia, por ejemplo, uso de las propiedades ACID de las transacciones.
- Sistemas que cuentan con una estructura de datos sencilla y a la medida de cada aplicación (El modelo relacional es un modelo genérico, no específico).
- La cantidad de datos a procesar tiende a ser exponencial al transcurrir el tiempo.
- Sistemas que requieren distribución de datos a través de un esquema de escalamiento horizontal, uso de clusters con múltiples servidores a bajo costo.
- La posibilidad de almacenar gran cantidad de registros con estructuras diferentes y variables (datos no estructurados)

## Empresas con desarrollos NoSQL



### Ejemplos de implementaciones NoSQL

Orientadas al manejo de documentos:


- CouchDB (de Apache).
- MongoDB (de 10gen).
- SimpleDB (de Amazon).
- IBM Lotus Domino
- Terrastore

Orientadas al manejo de clave/valor

- Cassandra (de Apache).
- BigTable (de Google)
- Dynamo (de Amazon)
- Project Voldemort (de Linkedin)

Orientadas al manejo de Grafos:

- Neo4J
- Allegro
- Virtuoso



COLUMNA	
CLAVE	VALOR
Nombre:	JUAN
Dirección	datos
Forma de pago	datos

FAMILIA DE COLUMNAS  
PERFIL DEL CLIENTE

ORD-001	Datos
ORD-002	Datos
ORD-003	Datos
ORD-004:	Datos
ORD-005	Datos


FAMILIA DE COLUMNAS  
ORDENES DE COMPRA

10923

ROW KEY

Con este solo valor se consultan todos los datos del lado derecho.

### Ejemplo de una arquitectura de solución:



**Adquisición de Datos**

**Organización de Datos**

**Análisis de Datos**

**NoSQL**  
Flexible  
Especializado

**SQL**  
Estructurado  
Seguro  
Administrado

**Data acquisition**

**Data processing**

**Data Analytics & visualization**

**Analytical Database**

**Data Warehouse**

**ETL**

**RDBMS (OLTP)**

**Spark**

**Real Time**

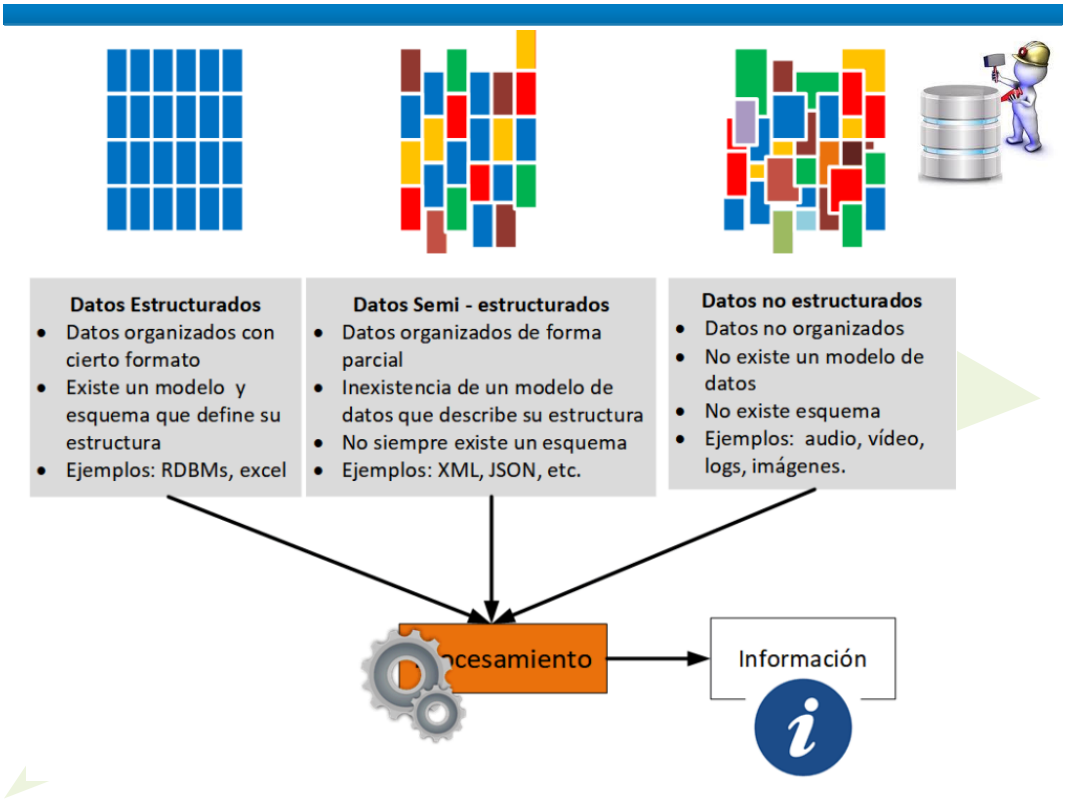
**Map Reduce**

**Hadoop Cluster**

Ciencia de datos.



- La ciencia de los datos es un campo interdisciplinario que abarca el procesamiento y los sistemas para extraer conocimiento de datos que se pueden encontrar en varios formatos:
  - Datos estructurados y
  - Datos no estructurados.
- En especial, los datos no estructurados representan la continuación de áreas de análisis de datos (Business Intelligence) como
  - Estadística,
  - Minería de datos,
  - Análisis predictivo,
  - Descubrimiento de conocimiento en bases de datos (Knowledge Discovery in Databases - KDD).



- Principales áreas de conocimiento que participan en Ciencia de datos:



- Ingeniería de Software
- Expertos en el dominio del problema (dominio de las reglas de negocio y operación asociadas con los datos).
- Investigación de operaciones
- Matemáticas
- Estadística.

- Las 2 principales categorías de un científico de datos:

- Analistas / Expertos en estadísticas
- Ingenieros.

**PREGUNTAS????**



SIS 306 BASE DE DATOS III

Ing. Edgar T. Espinoza R.