

Tema 1

Funciones y tareas de la Administración de Datos y de BD

Administración de Bases de Datos



Daniel López Fernández
Pedro P. Alarcón Cavero

Contenido

1. Introducción
2. Administración de datos
3. Administración de BD
4. Estrategia de datos
5. Tipos de datos
6. Modelos de datos
7. Gestores y motores de BD



1. Introducción

Sistema Información

Conjunto de procedimientos y datos, herramientas y equipos, contruidos, operados y mantenidos para recoger, registrar, procesar, almacenar y recuperar información

Sistema Base Datos

Sistema para registrar y mantener información coherente
Componentes: datos, hardware, software, usuarios

Sistema Gestor Base Datos

Software específico para la gestión y administración de datos

Base Datos

Conjunto de datos, útil para una organización o persona, almacenado digitalmente y accesible en tiempo útil, sobre el que se pueden realizar consultas y actualizaciones

Objetivos de los SGBD

Independencia lógica y física de datos

Capacidad de modificar el esquema conceptual y/o el esquema físico de la BD sin tener que modificar necesariamente los programas

Control de redundancia

Evitar inconsistencias

Imponer restricciones de integridad

Asegurar la validez de los datos

Control de autorizaciones

Evitar accesos indebidos a la BD

Proporcionar interfaces de usuario

Para diferentes tipos de usuarios

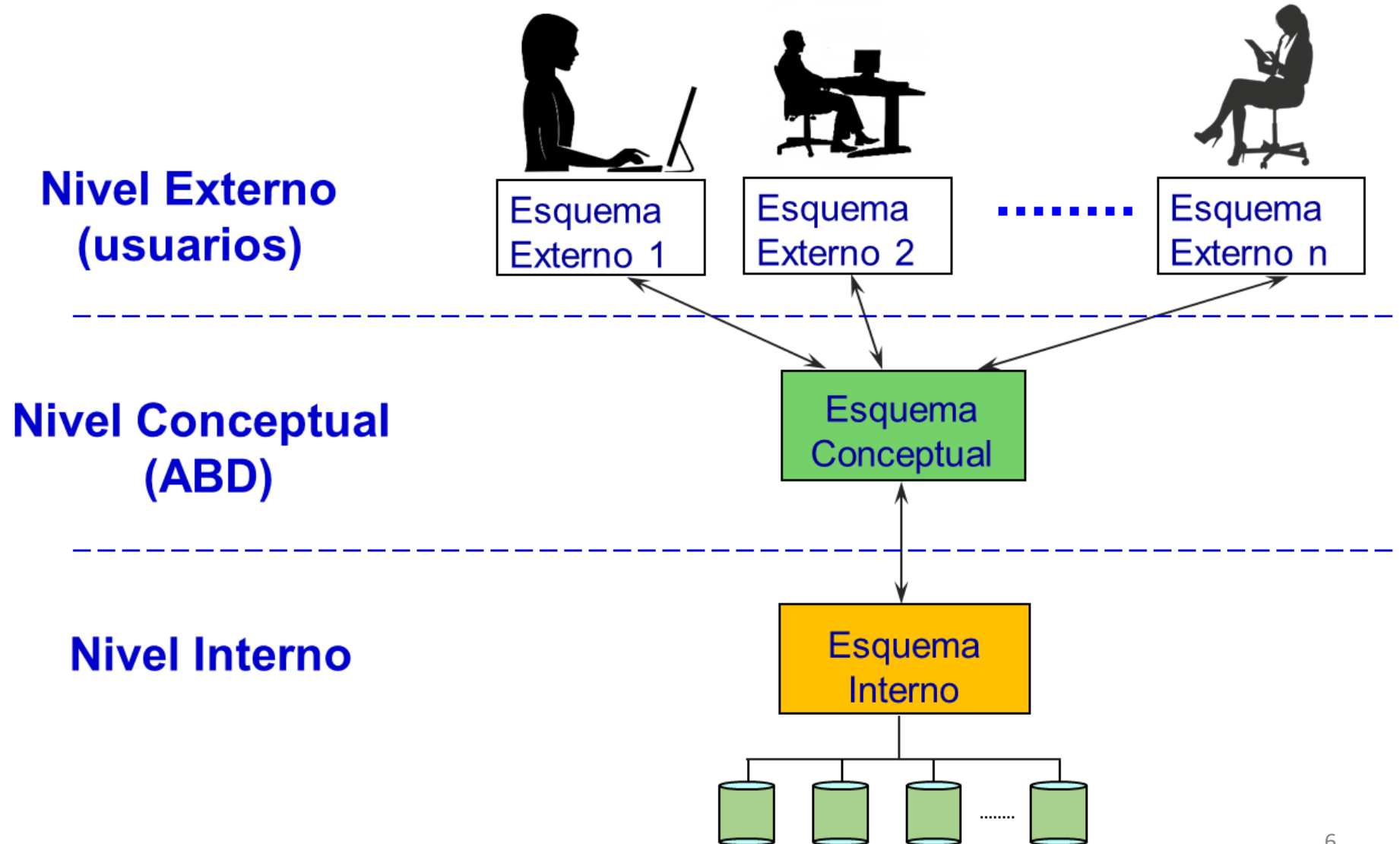
Control de copias de seguridad y recuperación

Prevenir la pérdida de datos frente a fallos

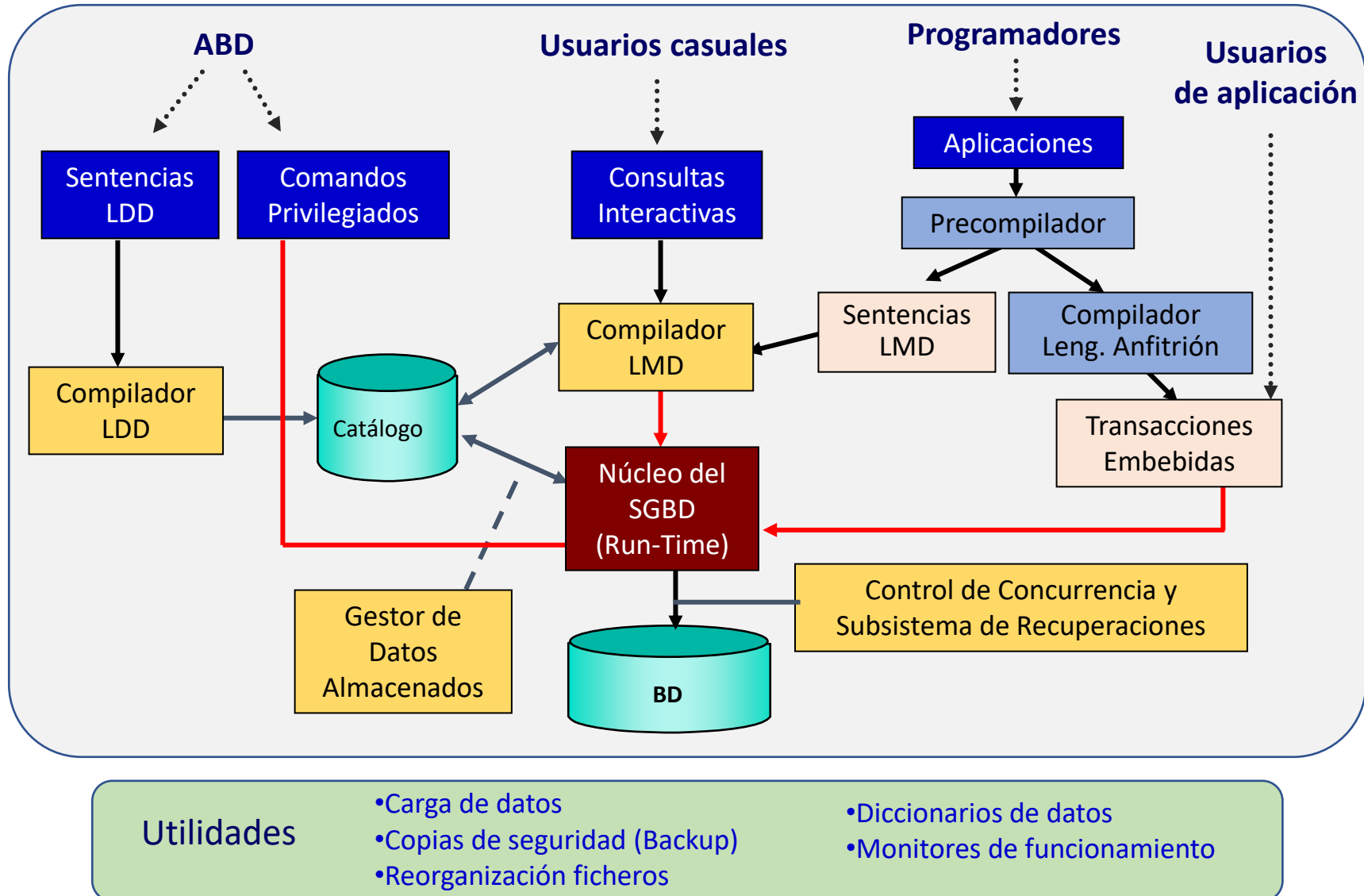
Proceso de transacciones

Permitir el acceso simultáneo y múltiple a la BD

Arquitectura ANSI/X3/Sparc



Componentes de un SGBD



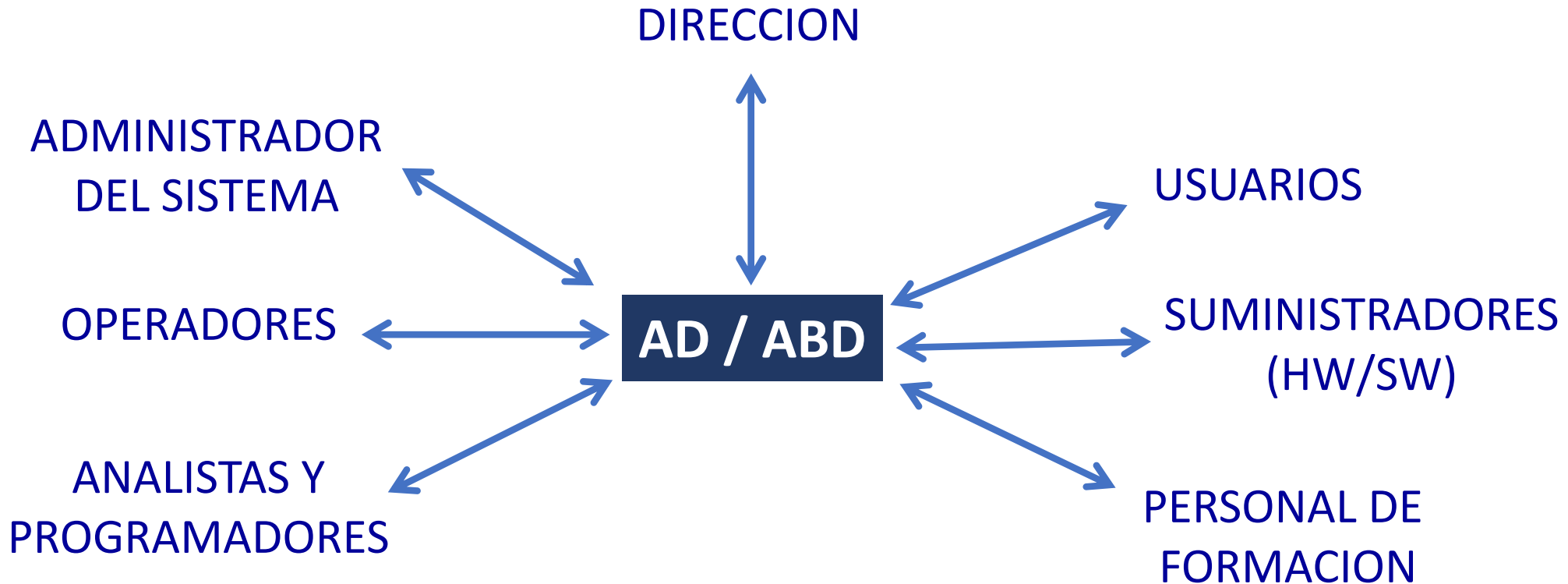
Ciclo de vida de una BD

1. Planificación estratégica: esquema (inicial)
2. Análisis del sistema: modelo de datos (conceptual)
3. Diseño del sistema: esquemas y transacciones
4. Implementación del sistema: base de datos creada
5. Carga o conversión (*ETL: Extract – Transform - Load*)
6. Validación y Transferencia: BD y documentación
7. Operación: servicios
8. Mantenimiento y Evolución

***Nuevos
Requisitos***



Personajes o *Stakeholders* alrededor de una BD



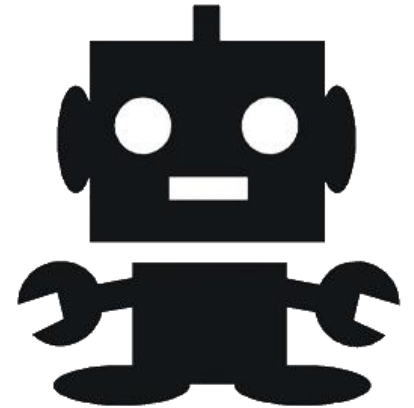
Administradores alrededor de una BD

| | | | |
|---|--------------------------------|---------------------------------------|--|
| Infraestructura TI | Administrador del Sistema (AS) | ABD (si no hay AS) | |
| Política de metadatos y datos | Administrador de Datos (AD) | ABD (si no hay AD) | |
| Análisis | | | |
| Diseño | | | |
| Desarrollo | | Administrador de Bases de Datos (ABD) | |
| Testing | | | |
| Implementación (Bases de datos, aplicaciones) | | | |
| Mantenimiento y Tuning | | | |

Figura adaptada de "Database Administration: The Complete Guide to Practices and Procedures"

Administrador del sistema

- Tecnólogo puro cercano al hardware
- Ingeniero de computadores
- Llamado a veces programador de sistemas
- Existe en algunas organizaciones
- Funciones
 - Instalación y configuración de los recursos TIC
 - Soporte de infraestructura
 - Responsable de la instalación y soporte del SGBD
 - No es el responsable del diseño de las bases de datos



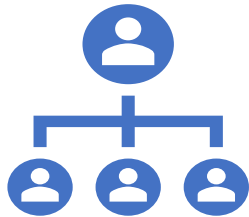


2. Administración de datos

Administrador de datos - AD (Chief Data Officer - CDO)



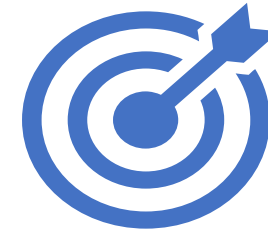
**Responsable de los datos como recurso
para una organización**



Suele ser directivo

No está definido en todas las organizaciones

Lo definen las organizaciones preocupadas por la calidad, integridad y reutilización de datos



Objetivos del AD

Establecimiento de políticas y procedimientos

Gestión de los datos como recurso corporativo

Responsable principal de la calidad de los datos

Administrador de datos (CDO: Chief Data Officer)



 movistar /  O₂ /  vivo / SON MARCAS TELEFÓNICA

 [Acerca de Telefónica](#) [Sala de prensa](#) [Accionistas e inversores](#) [Negocio Responsable](#) [Empleo](#) [Políticas Públicas](#) [Innovación](#)

Usted está en: [Telefónica](#) > [Acerca de Telefónica](#) > [Equipo directivo](#) > [Estructura organizativa](#) > **Chema Alonso**

Quiénes somos

Dónde estamos

Nuestra misión

Principios de Negocio Responsable

Centro de Privacidad y Seguridad

Estructura organizativa

Consejo de administración

Equipo directivo

Estructura societaria

EQUIPO DIRECTIVO



descargar

Chema Alonso

Chief Data Officer (CDO), Telefónica S.A.

Chema Alonso es actualmente CDO –Chief Data Officer- de Telefónica. En este cargo lidera la estrategia de Big Data, Publicidad y Cuarta Plataforma de Telefónica. Es asimismo responsable de los productos y servicios de ciberseguridad global de la unidad ElevenPaths y de de BigData de la unidad LUCA que Telefónica proporciona a sus clientes. Así mismo, lidera la construcción y el lanzamiento de AURA, la Cognitive Intelligence de Telefónica, para ofrecer una nueva forma de relación a los clientes con Telefónica.

Es Doctor en Seguridad Informática por la Universidad Rey Juan Carlos e Ingeniero Técnico de Sistemas Informáticos por la Universidad Politécnica de Madrid. Esta última institución le nombró Embajador Honorífico de la Escuela Universitaria de Informática en el año 2012.

https://www.telefonica.com/es/web/about_telefonica/estructura_organizativa/equipo-directivo

Administrador de Datos: Funciones

- ❖ Coordinar la fase de definición de estrategia
- ❖ Producir un modelo de datos de la empresa que incorpora los datos utilizados por todos los procesos de negocio
- ❖ Diseñar conceptualmente la BD: Diccionario de datos
- ❖ Proporcionar y permitir la compartición de datos
- ❖ Impulsar auditorías y monitorizaciones de la BD
- ❖ Participar en la planificación y desarrollo de nuevas aplicaciones y gestionar el impacto a nivel de datos
- ❖ Conocer las necesidades de los usuarios desde el punto de vista de información/datos
- ❖ Coordinar al ABD y al Administrador del Sistema
- ❖ Establecer estándares sobre políticas de datos
- ❖ Establecer a nivel funcional los requisitos de seguridad y privacidad
- ❖ Proporcionar plan de formación



3. Administración de BD

Administrador de Bases de Datos - ABD (*Database Administrator - DBA*)



Responsable técnico del sistema de bases de datos

- Perfil técnico
- Implementa políticas definidas por el AD
- Trabaja también con el administrador del sistema
- Responsable efectivo de la calidad de los datos
- Asegura la funcionalidad operativa y la eficiencia de las BD
- Asegura los mecanismos de recuperación de las BD

Administrador de Bases de Datos: Funciones

- ❖ Diseñar la base de datos (diseño lógico y físico)
- ❖ Asistir a las negociaciones para adquirir el SW y HW para el SBD
- ❖ Poner en marcha el sistema nuevo en paralelo con el antiguo
- ❖ Monitorizar el funcionamiento de la BD: auditorías de seguridad, rendimiento, ocupación de espacios físicos, etc.
- ❖ Afinamiento o *tunning*
- ❖ Reorganizaciones físicas y lógicas de la BD
- ❖ Proponer y controlar refactorizaciones de la BD
- ❖ En ocasiones, realizar también funciones de administración del sistema
- ❖ Trabajar en el día a día con el usuario
- ❖ Controlar aspectos relativos a la seguridad, confidencialidad e integridad de los datos
- ❖ Definir procedimientos de recuperación y respaldo de la BD
- ❖ Conocer nuevos productos relacionados con los SBD
- ❖ Documentar los procedimientos de operación del SBD (sus actividades)

3.1 Tipos de DBA

En grandes organizaciones suelen diferenciarse varios tipos de DBA

- System DBA
- Database Architect
- Database Analyst
- Data Modeler
- Application DBA
- Task-Oriented DBA
- Data Warehouse Administrator

System DBA

- Se centra en cuestiones técnicas más que del negocio, principalmente en el área de administración de sistemas
- Existe cuando la organización no tiene departamento oficial de administración del sistema o de programación de sistemas
- Tareas:
 - Instalación física y rendimiento del software de SGBDs
 - No suele involucrarse en la implementación de bases de datos y aplicaciones



Database Architect

- Diseña nuevas BD, para aplicaciones nuevas o existentes
- Tareas:
 - Creación del modelo de datos lógico
 - Diseño físico de la BD a partir del modelo lógico de datos
 - Implementa BD eficientes, incluyendo aspectos físicos: espacios e índices
 - Análisis del acceso a los datos para asegurar tanto consultas eficientes como un diseño óptimo
 - Creación de estrategias de *backup* y recuperación para las nuevas BD



Database Analyst

- No hay una definición clara para el trabajo de analista de BD
- Puede entenderse como
 - DBA junior
 - Arquitecto de bases de datos
 - Administrador de datos o analista de datos
 - O simplemente otra denominación utilizada para ABD



Data Modeler

- Este rol se define a veces cuando no existe la figura de AD
- Un *Data Modeler* realiza un subconjunto de tareas responsabilidad de un AD
- Tareas
 - Definición de requisitos de datos para el desarrollo de un proyecto
 - Análisis de los requisitos de datos
 - Diseño del modelo de datos conceptual y lógico de los diferentes proyectos
 - Creación y actualización del modelo de datos corporativo
 - Trabajar con los DBA para que tengan un conocimiento sólido de los modelos de datos



Application DBA

- Se centra en el diseño y administración de la BD de un proyecto concreto
- Experto en sentencias complejas y en el acceso a la BD desde aplicaciones

Ventajas

- Al centrarse en una aplicación concreta proporciona mejor servicio a los desarrolladores de la aplicación
- Forma parte del equipo de desarrollo por lo que conoce los planes de nuevos desarrollos o de cambios futuros
- Mejor comprensión de cómo la aplicación repercute en la empresa en general

Desventajas

- Al centrarse en una sola aplicación puede perder de vista necesidades globales de datos de la empresa
- Por el mismo motivo, puede no estar al tanto de nuevas características y funcionalidades de los SGBD de la empresa
- Puede quedarse aislado, perdiendo la comunicación con el grupo de DBA



Task-Oriented DBAs

- DBAs especializados en un subconjunto de las tareas de administración de BD
- Pueden existir solo en grandes organizaciones
- Ejemplos:
 - Backup & Recovery DBAs
 - Diseñador de BD
 - Analista de rendimiento BD



Data Warehouse Administrator

- DBA especializado en entornos de Data Warehouses
- Con experiencia en proyectos de Business Intelligence y analítica de datos
- Tareas:
 - Diseño de Data Warehouses
 - Procesos ETL: extracción, transformación y carga de datos
 - Procesos de calidad de datos
 - Aplicación de técnicas OLAP (análisis multidimensional)
 - Consultas y reporting para analítica de datos
 - Minería de datos



4. Estrategia de datos

Importancia de los datos como recurso corporativo

Ausencia de una estrategia bien definida

- Datos sucios
- Datos redundantes
- Datos inconsistentes
- Dificultad para integrar
- Rendimiento pobre
- Disponibilidad complicada
- Falta de responsabilidades
- Usuarios insatisfechos
- Sensación general de estar fuera de control el SBD

Recomendable contemplar

- A. Integración de datos
- B. Calidad de datos
- C. Metadatos
- D. Modelado de datos
- E. Roles y responsabilidades organizacionales
- F. Rendimiento y métricas
- G. Seguridad y privacidad
- H. Selección de SGBD
- I. Inteligencia del negocio
(*Business Intelligence*)

A. Integración de Datos



Situación

Existen volúmenes masivos de datos

Dificultad de integración (diferentes SGBD durante décadas)

Datos heredados (*legacy* data) requeridos por nuevas aplicaciones de BI

Alto grado de redundancia



Procesos

Integrar y unificar en lo posible las diversas fuentes de datos

Actualizar *legacy* (apps y datos) progresivamente



Beneficios

Minimizar redundancia de datos

Disponer de una vista comprensible de todos los datos de la empresa

Minimizar esfuerzos y errores entre las diferentes aplicaciones que manejan datos de diferentes fuentes

B. Calidad de Datos



Situación

Diversas fuentes de datos, más o menos correctas que otras

Tipos de datos usados incorrectamente, campos usados para múltiples propósitos, valores por defecto inapropiados, valores inexactos, etc.



Procesos

Limpieza y actualización de datos (validez de los datos)

Validación del proceso ETL (Extracción, Transformación y Carga)



Beneficios

Menor tiempo en corrección manual de problemas

Mayor confianza por parte de los usuarios

Menor tiempo de dedicación del personal directivo

C. Metadatos



Situación

No existe información del conjunto de datos de la organización (datos sobre los datos) o está desactualizado



Procesos

Producir o actualizar el modelo conceptual de los datos de la organización



Beneficios

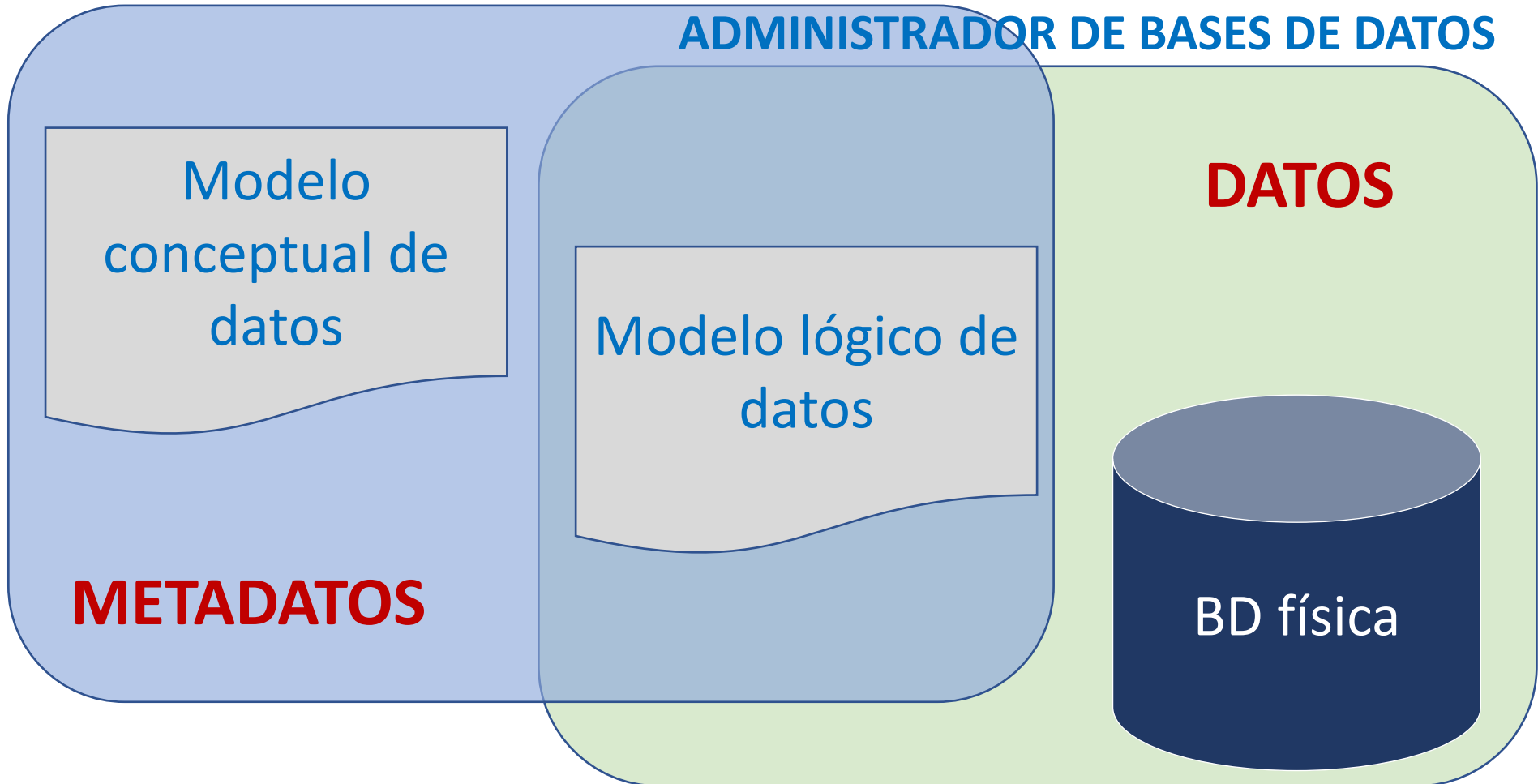
Reducción del tiempo en mantenimiento de programas

Reducción del tiempo de análisis y desarrollo de nuevas aplicaciones

Mejor disponibilidad de los datos (menor redundancia)

C. Metadatos

ADMINISTRADOR DE DATOS



D. Modelado de datos



Situación

La representación y estructura de los datos ha sufrido múltiples añadidos a lo largo del tiempo



Procesos

Realizar un nuevo modelado (conceptual, lógico y físico) y actualizar si procede

Seleccionar el modelo de datos más adecuado (ver sección 6 “Modelos de datos”)



Beneficios

Aumento de eficiencia

Mejor comprensión de los datos

Minimiza datos redundantes

Punto de partida para capturar nuevos metadatos

E. Roles y responsabilidades organizacionales



Situación

Existen tareas propias de AD y ABD (ver secciones 2 y 3) no cubiertas

Existen problemas de rendimiento, seguridad, integridad, etc.



Procesos

Definir las figuras y funciones de AD y ABD

Asegurar el cumplimiento de las tareas clave:
definición de políticas de datos, afinamiento y “securización” del sistema, etc.



Beneficios

Correcta administración de los datos de la organización

F. Rendimiento y métricas



Situación

Existe un gran número de procesos/usuarios y un gran volumen de datos

No se realiza un afinamiento periódico del sistema

Los tiempos de respuesta son altos y/o mejorables



Procesos

Optimizar las estrategias de almacenamiento y ocupación de espacios

Mejorar tiempos de respuesta mediante el afinamiento del sistema (por ejemplo, índices o reagrupamientos)



Beneficios

Identificar problemas

Adelantarse al surgimiento de problemas mayores

Conocer índices de utilización por parte de los usuarios

Priorizar medidas para mejorar el rendimiento

G. Seguridad y privacidad



Situación

Sistema global accedido a través de la red

Existencia de datos sensibles (sanitarios, financieros, etc.)

Se han identificado brechas de seguridad y/o se han producido ataques



Procesos

Implementar mecanismos de seguridad (por ejemplo, medidas para evitar SQL-Injection)

Monitorizar y controlar el acceso a los datos



Beneficios

Cumplir con normativas y regulaciones legales

Conseguir la confianza de los usuarios/clientes

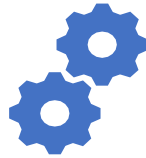
H. Selección de SGBD



Situación

No existe una política orientada a la estandarización

Existen múltiples SGBD y/o ERP (*Enterprise Resource Planning*)



Procesos

Reducir diversidad SGBD
Reducir interfaces entre SGBDs

Aumentar conocimiento del DBA en SGBDs

Seleccionar el/los SGBDs más adecuados (ver sección 7 “Gestores y motores de BD”)



Beneficios

Mejora la capacidad de integración de datos y aplicaciones

Economía de escala

Menores costes de formación

I. Business Intelligence / Analytics



Situación

Se necesitan mecanismos de soporte a la toma de decisiones



Procesos

Empleo de Data Warehouse (DW) y DataMarts (DM)

Incorporación de nuevas infraestructura, métodos, y herramientas.



Beneficios

Mejora notable en la integración y análisis de datos

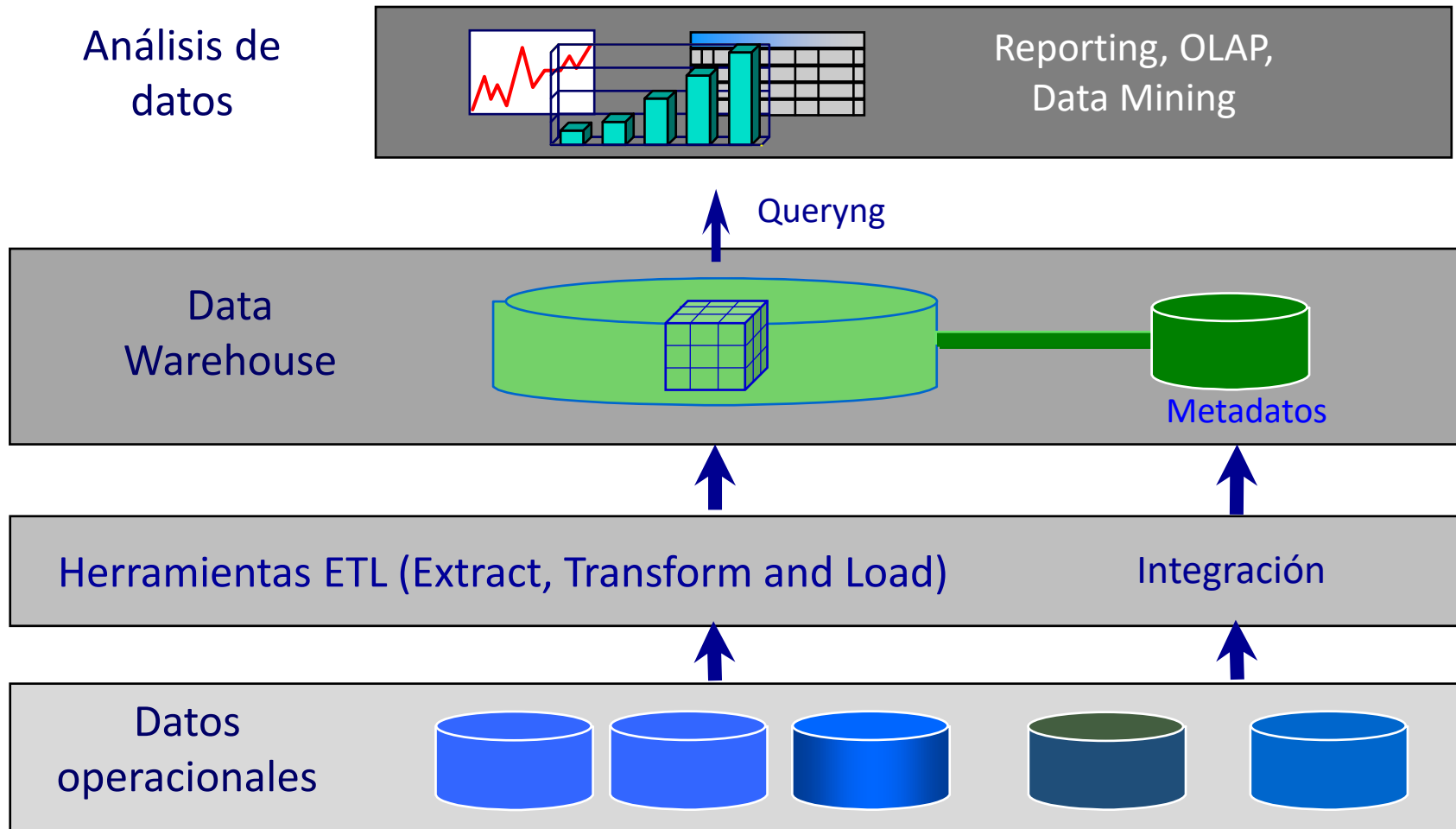
Aumento de ingresos

Contención del gasto

Reducción de fraudes

Fidelización de clientes

I. Business Intelligence / Analytics



Basado en: FORWISS 2001

I. Business Intelligence / Analytics:

DataWarehouse

“Colección de datos orientados a un tema, integrados, no volátil, que varía con el tiempo y que sirve de soporte a consultas de ayuda a la toma de decisiones”

Inmon

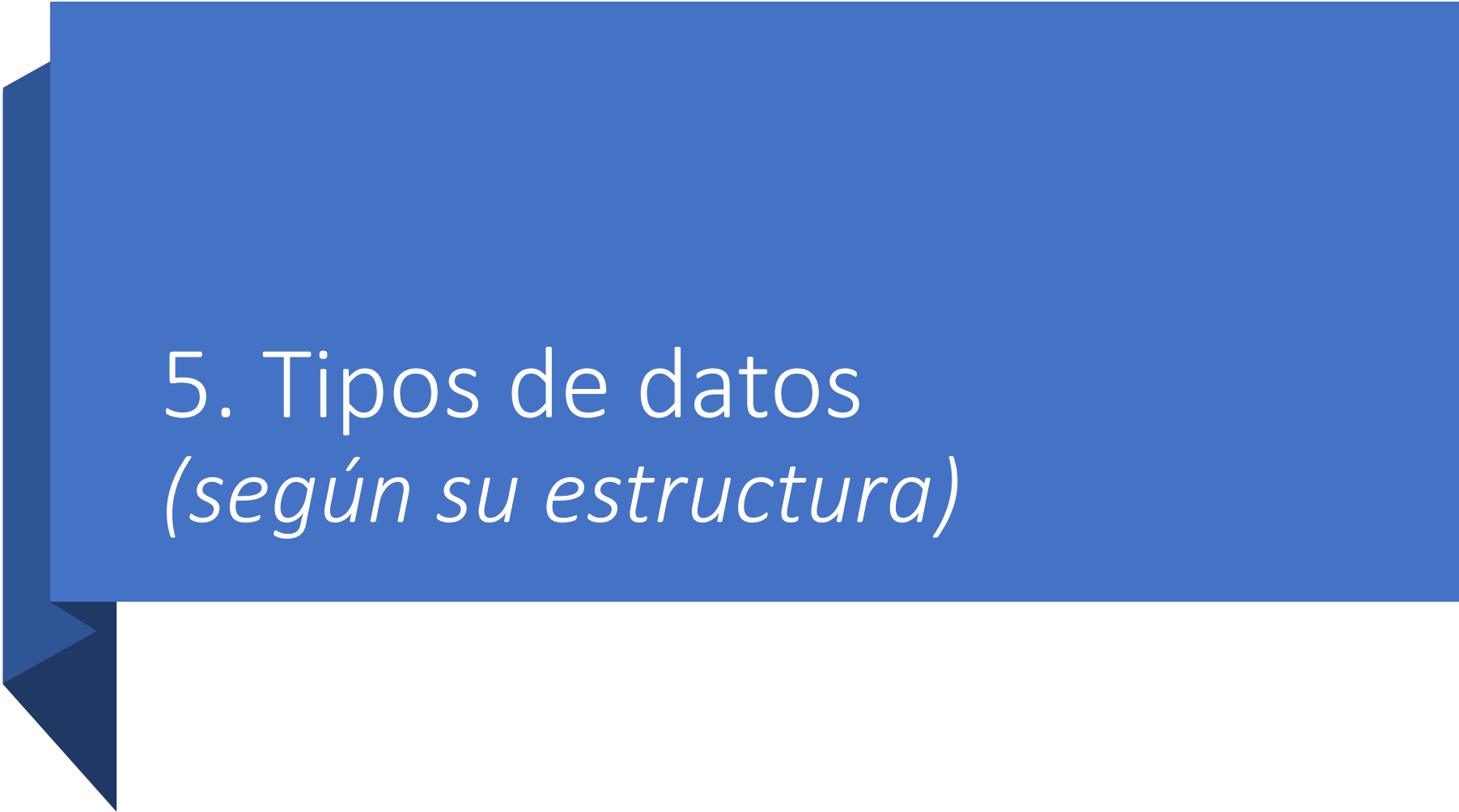
- DataWarehouse
 - Repositorio estructurado de datos históricos
 - Se construye mediante la integración de fuentes de datos múltiples, y heterogéneas
- DataMart
 - Subconjunto del DW para un “departamento” de la empresa

I. Business Intelligence / Analytics:

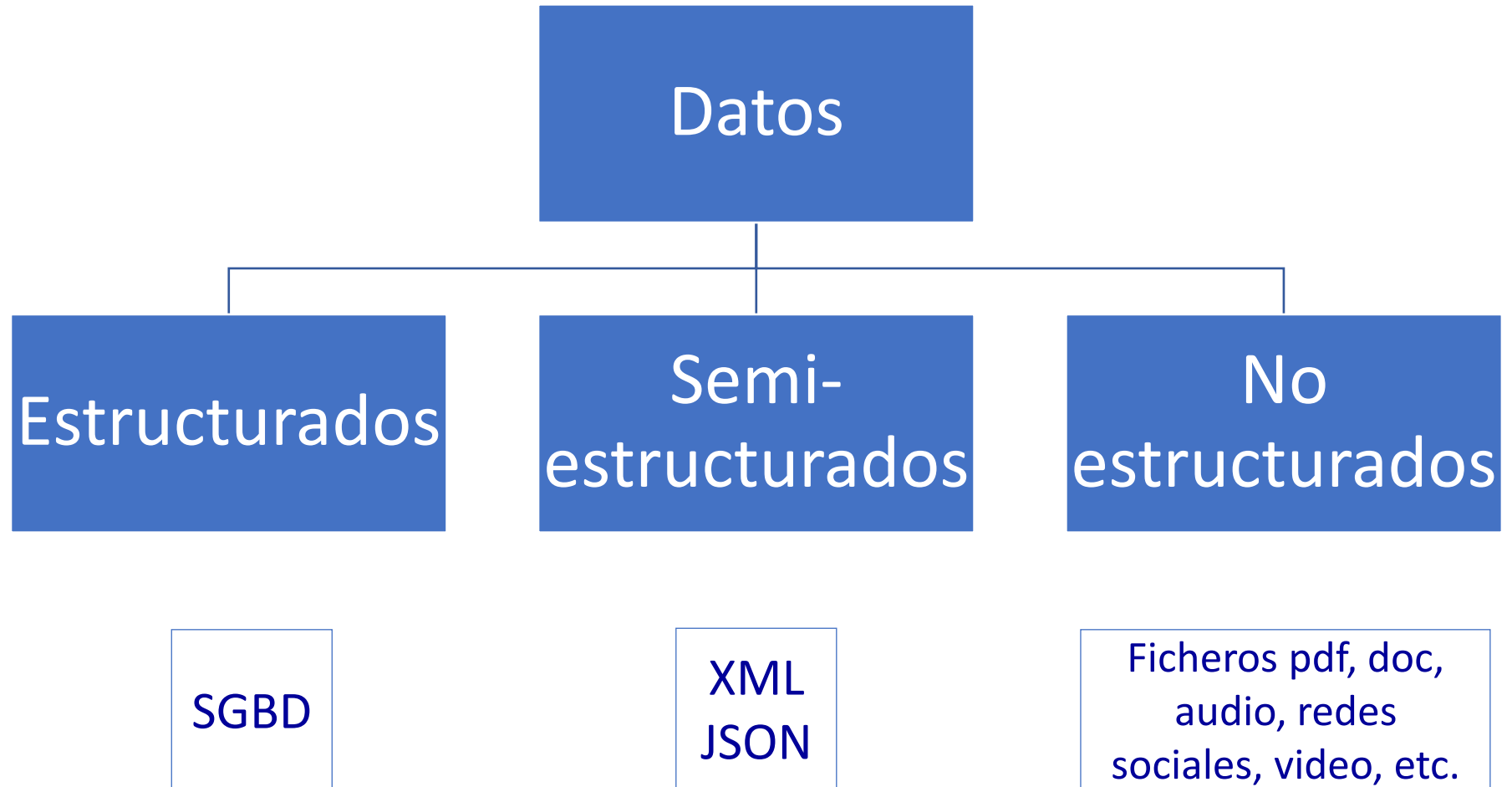
Tipos de procesamiento

- OLTP: On-Line Transaction Processing (Bases de Datos)
- OLAP: On-Line Analytical Processing (Datawarehouse)
 - ROLAP: Relational OLAP
 - MOLAP: Multi-dimensional OLAP

| Característica | OLTP | OLAP |
|-----------------|-------------|-------------------|
| Tamaño BD | Gigabytes | Giga a TeraBytes |
| Origen de datos | Interno | Interno y Externo |
| Actualización | On-Line | Batch |
| Periodo | Actual | Históricos |
| Consultas | Predecibles | Ad hoc |
| Actividad | Operacional | Analítica |



5. Tipos de datos *(según su estructura)*



Datos estructurados

- Los datos se estructuran por medio de tablas, objetos o ficheros de registros
- Cada tupla/registro/objeto es conforme con un esquema
- El esquema facilita el acceso a los datos e incorpora significado semántico
- ¿Qué sucede si perdemos el esquema?
 - Los datos de una BD no son independientes, dependen de un software concreto
- Modelos: relacional, jerárquico, objeto-relacional, OO
- El modelo más ampliamente utilizado es el modelo relacional. Los SGBD Relacionales están muy consolidados



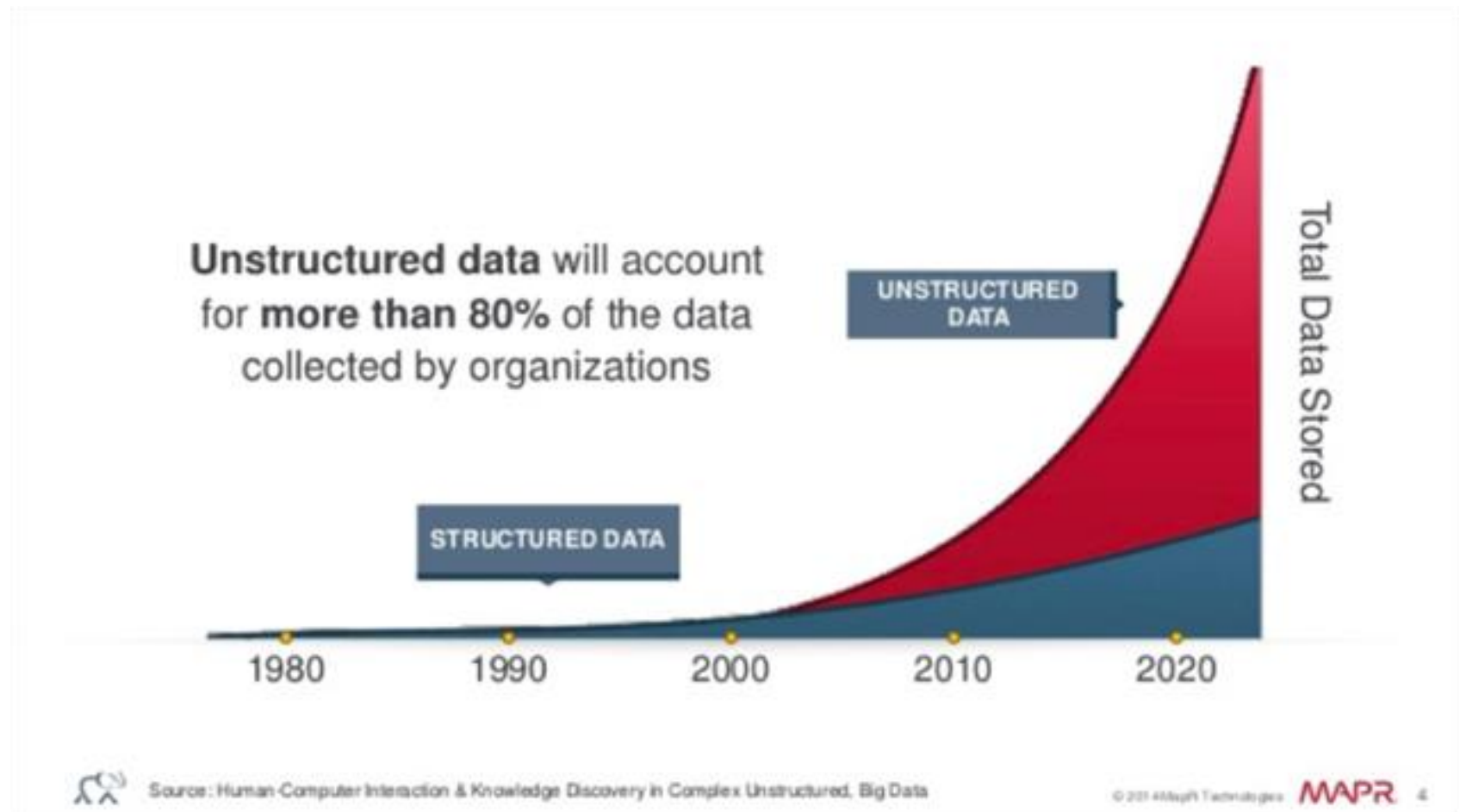
Datos semi-estructurados

- Algunos datos tienen cierta estructura (Texto: frases, párrafos, secciones, capítulos, etc. / Web: HTML)
- Fuerzan a que los ficheros estén bien formados (tienen estructura)
- Permiten acceso y manipulación de datos
- Pueden forzar a que los ficheros sean válidos (conformes a un esquema determinado)
- Facilitan la interpretación de los datos, incluyendo etiquetas
- Altamente portables, aunque incluyen redundancia (por ejemplo, XML)
- Se pueden incorporar en BD
- Independencia de un software concreto



Datos No Estructurados

- No tienen un modelo/estructura bien definido para extraer/acceder a la información que contienen
- Consisten en texto, audio, imagen, vídeo, etc. (Ejemplos: PDF, email, foto, canción, película, etc.)
- Pueden aportar un valor importante a empresas que disponen de grandes cantidades de datos no estructurados
- Dificultad para extraer información útil, se necesitan herramientas específicas para extraer información:
 - Text mining
 - Audio mining
 - Video mining



Martyn Holman - Thursday, February 18, 2016, <http://www.oxcp.com/blog/import-io-investing-in-the-future-of-data>



6. Modelos de datos

6.1 Modelos de Datos Conceptuales

- Modelo Entidad Relación
- Diagrama de clases UML

6.2 Modelos de Datos Lógicos

- Relacional
- Con orientación a objetos
- Jerárquico
- Linked data
- NoSQL



Modelo de datos con OO

- Incorporan al modelo relacional aspectos de la Orientación a Objetos
- Soportados por algunos SGBD
- Variantes :
 - Objeto-Relacional
 - Orientado a objetos

Modelo de datos objeto-relacional

Conservar el modelo compatible con el modelo relacional

- No perder 40 años de investigación en el modelo relacional, con un fundamento teórico muy sólido

Extender el modelo relacional para incorporar las características deseables de la orientación a objetos

- Tipos de datos definidos por el usuario
- Incorporación de objetos en el modelo relacional
 - Definición de tablas con tipos
 - Jerarquías de generalización/especialización entre tablas
- Extensión de SQL
 - Para consultar modelos de datos objeto-relacional
 - Incorporar otras extensiones no O-R, como la recursividad
- Soportado por algunos SGBD (Oracle, SQL Server,...)

Modelo de datos objeto-relacional: Ejemplo

```
CREATE TYPE punto AS (  
  x FLOAT, y FLOAT)
```

```
CREATE TYPE linea AS (  
  inicio punto,  
  fin punto)  
INSTANCE METHOD longitud()  
RETURNS FLOAT
```

```
CREATE TYPE poligono AS (  
  forma STRING,  
  coordenadasP punto ARRAY)  
INSTANCE METHOD area()  
RETURNS FLOAT
```

```
CREATE TABLE tramo_carretera (  
  id INTEGER,  
  propietario VARCHAR(50),  
  estado VARCHAR(10),  
  trazado linea,  
  geometria poligono)
```


```
SELECT sum(trazado.longitud),  
       sum(geometria.area)  
FROM tramo_carretera  
WHERE estado = 'Regular'
```

Modelo de datos orientado a objetos

Los objetos (estado y comportamiento) se almacenan en la BD

Se utilizan las mismas estructuras y relaciones que en la POO





Modelo de datos jerárquico

- Los datos se organizan mediante una estructura previamente definida
- Muy utilizado para intercambiar datos entre plataformas
- Se pueden manejar manualmente y mediante aplicaciones/SGBD
- Formatos más extendidos:
 - XML
 - JSON

Modelo de datos jerárquico: XML

- Formato de intercambio de datos estructurados entre plataformas
- Altamente usado en sistemas web
- Muchos SGBD almacenan y gestionan datos XML
 - Como texto en SGBDs convencionales
 - En columnas BLOB (Binary Large Object)
 - Limitación en búsquedas (contenido e indexación)
 - Como datos XML en SGBDs convencionales
 - En columnas específicas XML Type
 - Permiten consulta a los datos XML con Xpath y Xquery (SQLX)
 - Limitación en control de integridad (datos XML y relacionales)
 - Problema: Mapeo Jerárquico (XML) => Relacional
 - Como datos XML en SGBDs nativos
 - Almacenan elementos XML como datos, manteniendo su estructura intacta
 - Los esquemas permiten reglas de almacenamiento e indexación
 - Permiten la gestión de los datos como XML

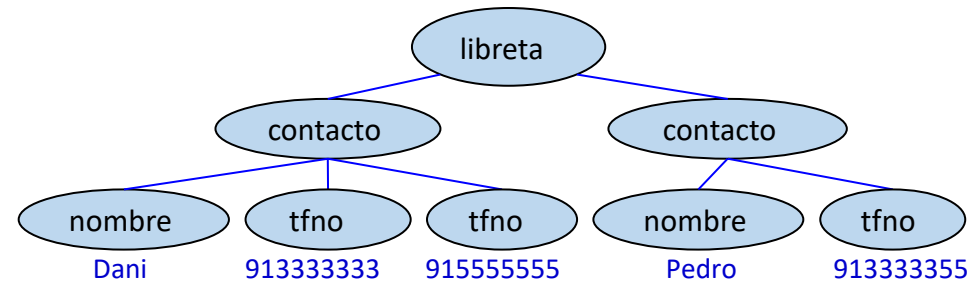
Modelo de datos jerárquico: XML (Ejemplo)

Documento xml-schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="libreta">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element ref="contacto"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="contacto">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="nombre"/>
        <xs:element ref="tfno" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="nombre" type="xs:string"/>
  <xs:element name="tfno" type="xs:string"/>
</xs:schema>
```

Documento xml

```
<?xml version="1.0" encoding="UTF-8"?>
<libreta>
  <!-- Primera persona de la libreta -->
  <contacto>
    <nombre>Dani</nombre>
    <tfno>913333333</tfno>
    <tfno>915555555</tfno>
  </contacto>
  <!-- Segunda persona de la libreta -->
  <contacto>
    <nombre>Pedro</nombre>
    <tfno>913333355</tfno>
  </contacto>
</libreta>
```



Modelo de datos jerárquico: JSON

JSON (JavaScript Object Notation)

- Formato ligero de intercambio de datos
- Auto-descriptivo y fácil de entender por humanos
- Fácil de analizar y generar por las máquinas
- Basado en un subconjunto de JavaScript (standard ECMA-262)
- Nació como alternativa a XML
- Dos estructuras:
 - Colecciones de parejas nombre/valor
 - Lista ordenada de valores

BSON: Binary JSON

- Representación de JSON en formato binario
- Diseñado para ganar eficiencia (almacenamiento y velocidad)
- Extiende JSON proporcionando tipos de datos adicionales, campos ordenados, y mayor eficiencia en la codificación/decodificación entre diferentes lenguajes

Modelo de datos jerárquico: XML vs JSON

Documento XML (**231** caracteres)

```
<guerreros>
  <guerrero>
    <nombre>Son</nombre>
    <apellidos>Goku</apellidos>
  </guerrero>
  <guerrero >
    <nombre>Son</nombre>
    <apellidos>Gohanda</apellidos>
  </guerrero >
  <guerrero>
    <nombre>Son</nombre>
    <apellidos>Goten</apellidos>
  </guerrero>
</guerreros>
```

Documento JSON (**127** caracteres)

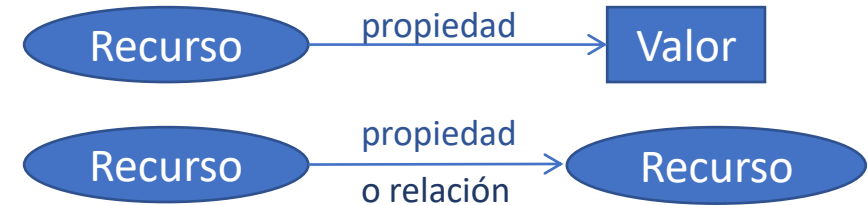
```
{“guerreros”: [
  {“nombre”:”Son”,
   “apellidos”:”Goku”},
  {“nombre”:”Son”,
   “apellidos”:”Gohanda”},
  {“nombre”:”Son”,
   “apellidos”:”Goten”}
]}
```

Linked Data

- Asociado a la web de datos enlazados
 - Enlaza “cosas” (o descripciones de cosas), no documentos
 - Orientado a máquinas, no personas
- Infraestructura basada en metadatos (ontologías)
 - Permite *razonar* en la Web
- Información mejor definida
 - Mayor simplicidad y rapidez en encontrar información
- Se apoya en lenguajes definidos por el W3C
 - RDF (Resource Description Framework)
 - Información descriptiva (metadatos) sobre los recursos en web
 - OWL (Ontology Web Language)
 - Lenguaje para definir ontologías estructuradas
 - SPARQL
 - Lenguaje de consulta sobre RDF y OWL

Modelo de datos RDF (Resource Description Framework)

- Basado en XML
- Recurso: representado por tripletas
- Tripletas: sujeto, predicado, objeto



```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:prof="http://etsisi.upm.es/abd/profes/">
  <rdf:Description rdf:about="www.eui.upm.es/MI/profes/1">
    <prof:nombre>Dani</profesor:nombre>
    <profemail rdf:resource="mailto:daniel.lopez@etsisi.upm.es"/>
    <prof:titulo>Dr.</prof:titulo>
  </rdf:Description>
</rdf:RDF>
```

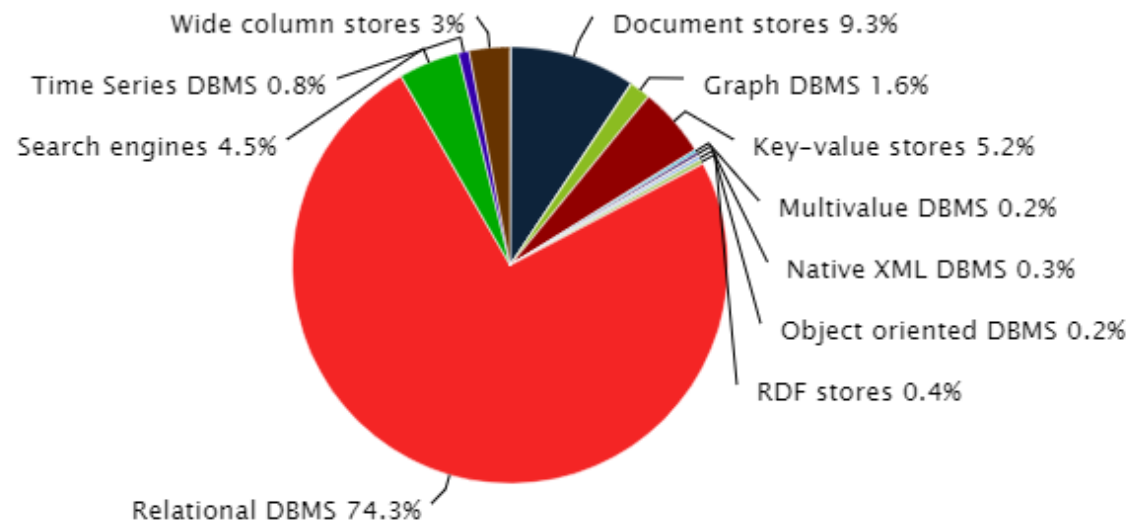


Modelo NoSQL o NoRelacional

Manejo eficiente de grandes cantidades de datos

- MongoDB
 - Orientado a documento
 - Basado en Bson (Json)
- Apache Hadoop
 - Pareja clave-valor
 - Una clave única para cada elemento de información
 - Equivalente a una sola tabla muy grande, convenientemente indexada

6.3 Popularidad* de modelos de datos



© 2020, DB-Engines.com

<https://db-engines.com/>

** La popularidad se calcula mediante el número de apariciones en buscadores (google), foros especializados (stackoverflow.com), ofertas de empleo, etc.*



7. Gestores y motores de BD

7.1 Sistema gestor de BD

Software que permite la manipulación de una base de datos mediante operaciones CRUD (Create, Read, Update, Delete)

ORACLE®



Popularidad de SGBD

| Rank | | | DBMS | Database Model | Score | | |
|----------|----------|----------|------------------------|------------------------------|----------|----------|----------|
| Sep 2019 | Aug 2019 | Sep 2018 | | | Sep 2019 | Aug 2019 | Sep 2018 |
| 1. | 1. | 1. | Oracle + | Relational, Multi-model i | 1346.66 | +7.18 | +37.54 |
| 2. | 2. | 2. | MySQL + | Relational, Multi-model i | 1279.07 | +25.39 | +98.60 |
| 3. | 3. | 3. | Microsoft SQL Server + | Relational, Multi-model i | 1085.06 | -8.12 | +33.78 |
| 4. | 4. | 4. | PostgreSQL + | Relational, Multi-model i | 482.25 | +0.91 | +75.82 |
| 5. | 5. | 5. | MongoDB + | Document | 410.06 | +5.50 | +51.27 |
| 6. | 6. | 6. | IBM Db2 + | Relational, Multi-model i | 171.56 | -1.39 | -9.50 |
| 7. | 7. | 7. | Elasticsearch + | Search engine, Multi-model i | 149.27 | +0.19 | +6.67 |
| 8. | 8. | 8. | Redis + | Key-value, Multi-model i | 141.90 | -2.18 | +0.96 |
| 9. | 9. | 9. | Microsoft Access | Relational | 132.71 | -2.63 | -0.69 |
| 10. | 10. | 10. | Cassandra + | Wide column | 123.40 | -1.81 | +3.85 |

<https://db-engines.com/>

7.2 Motor de BD

- Componente clave de un SGBD responsable de almacenar, procesar y proteger los datos
- Acepta peticiones lógicas de otros subsistemas, que convierten a sus equivalentes físicos y procesan accediendo a los ficheros y al diccionario de datos correspondientes.
- Los SGBD con licencia GPL (e.g. MySQL) permiten conocer los detalles técnicos de los motores que tiene disponibles (e.g. InnoDB, MyISAM, etc.)
- El ABD debe seleccionar el SGBD y, en algunas ocasiones, también el motor que va a utilizar.

InnoDB

Motor por defecto en MySQL

- Permite:
 - Bloqueo a nivel de registro
 - Restricciones de clave foránea
 - Transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad)
 - Recuperación automática en caso de caída
 - Almacena físicamente los registros en el orden de la clave primaria



Soporte de transacciones (conurrencia)
Claves foráneas (integridad referencial)
Alta eficiencia y confiabilidad



Ineficiente en BD orientadas únicamente a la lectura de datos

MyISAM

Motor por defecto en MySQL hasta v5.5

Permite:

- Bloqueo a nivel de tabla
- Compresión elevada de tablas
- Alta velocidad de lectura (SELECT “simples”)
- Almacena físicamente los registros en el orden en el que se insertan



Menor consumo de memoria y mayor velocidad para recuperar datos



No soporta transacciones, ni claves foráneas

Bajo rendimiento en operaciones de escritura

Mala recuperación ante caídas

Referencias

- *Adelman S. Et al. "Data Strategy". Ed Addison Wesley 2005.*
- *Mullins C.S. "Database Administration. The Complete Guide to DBA Practices and Procedures". Second Edition. Ed Addison Wesley 2013.*
- *Elmasri, R.A.; Navathe, S.B. "Fundamentos de Sistemas de Bases de Datos (5ª edición)". Ed Addison-Wesley 2007.*
- *Elmasri, R.A.; Navathe, S.B. "Fundamentals of Database Systems (7h edition)". Ed Pearson 2015.*