

---

# Bases de Datos

Universidad Nacional de Río Cuarto

---



# Teórico 1

## Introducción

# Integrantes de la Cátedra

Responsable:

Mg. Fabio Zorzan

Auxiliares :

Ing. Guillermo Frascchetti

A.C. Mariana Frutos

Prof. Sandra Angeli

# Horarios

- Teóricos:
  - Martes de 14 a 16 Hs Aula 108 Pab 2.
  - Viernes de 8 a 10 Hs Aula 101 Pab 2.
- Prácticos:
  - Comisión 1:
    - Lunes de 8 a 10 hs Aula 102 Pab 2.
    - Miércoles de 10 a 12 hs Aula 102 Pab 2.
  - Comisión 2:
    - Lunes de 16 a 18 Hs Aula 102 Pab 2.
    - Miércoles de 16 a 18 Hs Aula 102 Pab 2.

# Pagina de la Materia

La cátedra utilizará como herramienta de comunicación y disposición de materiales la plataforma SIAT:

<http://www.siat.unrc.edu.ar>

Los alumnos deben inscribirse en la materia Bases de Datos 2017. Si no están registrados en la plataforma SIAT, deben hacerlo primero.

# Objetivos

- Adquirir y aplicar conocimiento sobre modelización y diseño de bases de datos, especialmente usando el modelo relacional.
- Adquirir conocimiento y experiencia en la implementación de base de datos relacionales sobre motores de bases de datos.
- Adquirir conocimiento sobre la optimización de bases de datos, manejo de transacciones y concurrencia.
- Adquirir conocimientos básicos sobre diferentes tipos de base de datos.

# Evaluación

- **Condiciones de regularidad:**

- Dos exámenes parciales prácticos con sus respectivos recuperatorios. La aprobación requerirá el 50% del examen como mínimo.
- Un proyecto final grupal, el cual consiste del diseño e implementación, sobre un motor de base de datos, de una base de datos relacional a utilizar en un proyecto de desarrollo de software. El proyecto tendrá un grado de interacción con las Materias Análisis y Diseño de sistemas y Algoritmos II

- **Régimen de aprobación alumnos regulares:**

- Examen final teórico-práctico.

- **Régimen de aprobación alumnos libres:**

- Realización de un proyecto final integrador individual, el cual consiste del diseño e implementación, sobre un motor de base de datos, de una base de datos relacional a utilizar en un proyecto de desarrollo de software. Este proyecto debe ser aprobado antes de rendir el examen práctico.
- Examen práctico.
- Examen teórico.

# Bibliografía

- Database System Concepts. Sixth Edition, Edition Silberschatz, Korth, Sudarshan. McGraw Hill Company, 2010.
- Fundamentos de Base de Datos. Quinta Edición, Edition Silberschatz, Korth, Sudarshan. McGraw Hill Company, 2006.
- Fundamentals of Database Systems. Elmasri, Navathe. Addison Wesley, 1997.
- An introduction to Database Systems. Vol 1 C.J. Date, Addison Wesley.
- An Introduction to Database Systems. Vol 2 C.J. Date. Addison Wesley.





# Temas

- Introducción.
- Modelo E/R.
- Modelo relacional
  - Álgebra relacional.
  - Calculo relacional de tuplas.
  - Calculo relacional de dominios.
- SQL
  - DML.
  - DDL.
- Diseño de bases de datos.
- Transacciones.
- Indexación.
- Otras Bases de datos.

# Bases de Datos

- Un sistema de gestión de bases de datos(DBMS siglas en Ingles) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a ellos.
- Otra definición dada por [Navathe 97]:"Un sistema de software de propósito general que facilita el proceso de definir, construir y manipular base de datos para diversas aplicaciones".

# Generalidades

- Casi todos los sistema utilizan bases de datos.
- Salida laboral del Administrador de Base de Datos (DBA)

# Problemas de los sistema de archivos

## Redundancia e inconsistencia de datos

Si una misma información se encuentra repetida en diferentes lugares (archivos), puede causar un problema de sobre utilización de espacio de almacenamiento, además de causar una potencial inconsistencia de datos.

Ejemplo: Supongamos que tenemos los siguientes datos

Personas

DNI	Nombr	
25656325	Juan	
23264524	Manue	
54556544	Carlos	

Vehículos

#Pat	DNI		
BVF 344	256563		
GGF 928	256563		
GFT 893	545565		

# Problemas de los sistema de archivos (sigue)

## Integridad

Los datos almacenados deben cumplir con ciertas condiciones de integridad de datos (ej.: el saldo de una cuenta bancaria no puede ser negativo), estas condiciones pueden cambiar en el transcurso del tiempo, cuando esto sucede deben ser actualizados “todos” los programas que manipulan la información afectada.

# Problemas de los sistema de archivos (sigue)

## Problemas de Atomicidad

Todas las operaciones de una transacción se deben ejecutar o ninguna.

Ej.: Transacción bancaria de una cuenta a otra.

1. **Leer**( $A$ )
2.  $A := A - 500$
3. **Escribir**( $A$ )
4. **Leer**( $B$ )
5.  $B := B + 500$
6. **Escribir**( $B$ )



# Problemas de los sistema de archivos (sigue)

## Acceso concurrente a los datos

Para optimizar el uso de recursos es necesario proveer accesos concurrente a los datos, esto trae aparejado un potencial inconsistencia en los datos.

Ejemplo:

Estado inicial CuentaA = 500;

Programa 1	Programa 2
Leer(CuentaA)	
	Leer(CuentaA)
CuentaA := CuentaA+1000;	
	CuentaA := CuentaA +300;
Escribir(CuentaA);	
	Escribir(CuentaA);

Estado Final CuentaA = 800;



# Problemas de los sistema de archivos (sigue)

## **Seguridad en el acceso a los datos**

No todos los usuarios de un sistema de bases de datos deberían acceder a todos los datos. Utilizando sistemas de archivos esto es muy difícil de garantizar, ya que la gestión de datos no está centralizada.



# Otros problemas...

- Aislamiento.
- Dificultad en el acceso a los Datos.

# Modelo Lógico Basados en Objetos

- Modelo Entidad-Relación.
- Modelo Orientado a Objetos.

# Modelos Lógicos Basados en Registros

- **Modelo jerárquico:** la información se representa con colecciones de registros y las relaciones con punteros. Los registros se organizan como árboles.
- **Modelo de Red:** ídem al anterior pero los registros se pueden organizar como grafos.
- **Modelo Relacional:** Los datos se representan como tablas y las relaciones entre los datos también.

# Tipos de motores de bases de datos SQL

Garantizan Propiedades ACID, Utilizan SQL como lenguaje, pueden ser:

- Cliente-Servidor(ambiente multiusuarios):
  - MySQL(InnoDB, etc).
  - Oracle.
  - Postgres.
- Embebidas (Incluido en las aplicaciones, son útiles para aplicaciones monousuario):
  - Sqlite(No posee chequeo de tipos).
  - MySQL(libreria MySQLd).



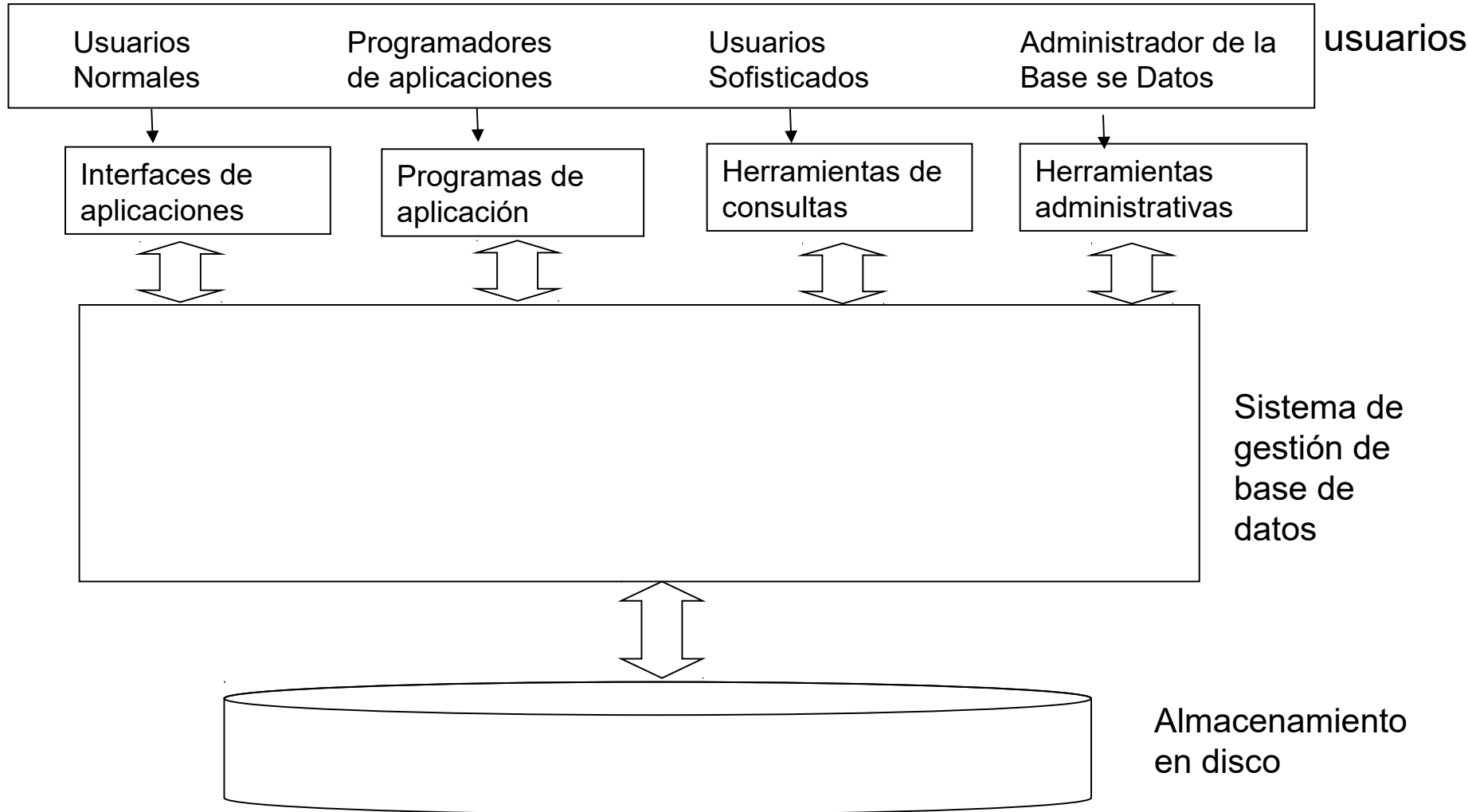
# Bases de datos NOSQL

Estas Bases de datos NO tienen esquema, NO garantizan las propiedades ACID, no permiten JOINS, escalan horizontalmente, entre otras características.

Hay Orientadas a:

- Documentos (MongoDB).
- Columnas(Cassandra).
- Etc..

# Estructura general de un RDBMS



# Estructura de un RDBMS

