

TRABAJO N°1

MATERIA: Base de datos II  
PROFESOR : NATALIA LUCERO

NOMBRE DEL GRUPO: **THE GODS OF PROGRAMMING**  
ALUMNOS:

* Lopez, Juan Pablo
* Ortigoza Peña, Santiago
* Ramirez Lorca, Jaime Nahuel
* Ramirez Lorca, Mauro Uriel
* Rueda, Juliana

AÑO: 2025

Actividad:

1- Responder cuestionario para la asistencia:

2- Responder las siguientes preguntas en grupo

Repasamos conceptos

A) ¿Qué tarea realiza el administrador de una base de datos?

La tareas que se encargan son el manejo de datos de una empresa, organizando y estructurando, dando paso al manejo fluido de los datos

B) Menciona dos tareas que realiza el administrador de datos

-Crea la base de Datos y planifica su estructura

-Maneja lo datos

C) Mencione 2 tipos de base de datos

Base de datos ditribuidos: estas bases de datos consisten en que los datos o los archivos  
que se están guardando, están ubicados o están dispersos.  
Se almacenan en diversos dispositivos computadoras, servidores que son de la misma ubicación física pero dispersos en diferente red.

Base de datos no sql: esta permite que los datos no estructuradas y semiestructurada se manipule. A diferencia del valor de la escena que define cómo deben comportarse todos los datos insertados en la base de datos.

Este tipo de registro de bases de datos, los cuales se guían por claves y datos y esto se guardan en documentos

Y estos no tienen la relación en sí, sino que solo se guardan de esa manera y se guardan dentro de colecciones. Para consultar datos y se accede a la colección y después de la colección, a los documentos

D)¿Qué tipo de datos existen?

Los datos que existen son:

Numéricos: **INT / INTEGER**:  
 Se usa para guardar números enteros sin decimales, como edades, cantidades, o IDs.  
 Ej: edad = 21, stock = 100.

**SMALLINT**:  
 Similar al INT, pero para números más chicos (menos espacio de memoria).  
 Útil cuando sabés que los valores no serán grandes.  
 Ej: número de hijos, nivel de prioridad (1 a 5).

**BIGINT**:  
 Para enteros muy grandes.  
 Se usa en sistemas que manejan muchos datos o identificadores largos.  
 Ej: identificador de usuario en una red social con millones de registros.

**DECIMAL / NUMERIC**:  
 Guardan decimales con precisión exacta (muy útil para cálculos financieros).  
 Ej: precio = 199.99, saldo = 5230.75.

**FLOAT / REAL / DOUBLE**:  
 También guardan decimales, pero con menor precisión (más rápidos pero menos exactos).  
 Se usan para mediciones científicas, estadísticas o cálculos no tan críticos.  
 Ej: temperatura = 36.7.

Texto:

**CHAR(n)**:  
 Guarda cadenas de texto con longitud fija. Si el texto es más corto, completa con espacios.  
 Ej: códigos postales, abreviaturas ("ARG", "USA").

**VARCHAR(n)**:  
 Guarda texto de longitud variable hasta un límite. Muy usado en la mayoría de los campos de texto.  
 Ej: nombres, correos electrónicos, direcciones.

**TEXT**:  
 Para textos muy largos (sin límite definido).  
 Ej: descripciones, artículos, comentarios.

Fecha y Tiempo:

**DATE**:  
 Guarda solo la fecha (sin hora).  
 Ej: fecha de nacimiento, fecha de registro.

**TIME**:  
 Guarda solo la hora.  
 Ej: hora de apertura, hora de llegada.

**TIMESTAMP**:  
 Guarda fecha y hora juntas.  
 Ej: fecha y hora de creación de un registro, log de acceso.

**INTERVAL**:  
 Representa una duración de tiempo.  
 Ej: "2 días", "3 horas", útil para calcular diferencias entre fechas.

Y también están:

**BLOB (Binary Large Object)**:  
 Guarda archivos binarios (no texto).  
 Ej: imágenes, videos, PDF, audios.

**UUID**:  
 Identificador único universal. Muy usado para identificar registros de forma global y segura.  
 Ej: id de sesión, claves únicas entre sistemas.

**JSON / XML**:  
 Guardan datos estructurados como texto.  
 Útiles cuando se necesita guardar objetos, listas o configuraciones dentro de un solo campo.  
 Ej: preferencias de usuario, historial de compras.

Mongodb: Es un sistema de gestión de bases de datos NoSQL(Not only SQL), orientado a documentos. A diferencia de las bases de datos relacionales como MySQL o PostgreSQL, que usan tablas, filas y columnas, MongoDB almacena los datos en documentos con estructura similar a JSON.

¿Cómo almacena datos?

MongoDB guarda los datos en documentos llamados BSON(Binary JSON), que pueden contener: Texto  
Números

Fechas  
Listas  
Subdocumentos (estructuras anidadas)

Ejemplo de documento en MongoDB:



Sus componentes son:

**Base de datos**: conjunto de colecciones.

**Colección**: conjunto de documentos (equivalente a una tabla).  
**Documento**: unidad básica de datos (equivalente a una fila, pero más flexible).

Características principales:

**Flexible**: No necesita una estructura fija (puede haber documentos distintos dentro de la misma colección).  
**Rápido y escalable**: Ideal para manejar grandes volúmenes de datos.  
**Documentos ricos**: Permite anidar estructuras complejas.  
**Muy usado en apps modernas**: como backend de aplicaciones web, móviles o APIs.

¿Cuándo usar MongoDB?

Cuando los datos no tienen una estructura fija.  
Para aplicaciones que crecen rápido o cambian seguido.  
En proyectos que usan JavaScript o Node.js (porque se integra muy bien).  
Cuando se necesitan consultas rápidas y escalabilidad horizontal.

Comparación: SQL vs MongoDB

| **Característica** | **SQL (MySQL/PostgreSQL)** | **MongoDB** |
| --- | --- | --- |
| Modelo de datos | Tablas (relacional) | Documentos (NoSQL) |
| Lenguaje de consulta | SQL | MongoDB Query Language |
| Estructura | Fija (esquema rígido) | Flexible (sin esquema) |
| Escalabilidad | Vertical | Horizontal |
| Ideal para... | Datos estructurados | Datos semiestructurados |

Sistema gestor de bases de datos:

Un **Sistema Gestor de Bases de Datos (SGBD)** es un **software especializado que permite crear, administrar y utilizar bases de datos** de forma eficiente y segura.

Actúa como **intermediario entre el usuario y la base de datos**, permitiendo consultar, modificar, eliminar o agregar datos sin necesidad de manipular directamente los archivos.

Sus principales funciones:

**Definición de datos**: permite crear la estructura de la base de datos (tablas, campos, tipos de datos).  
**Manipulación de datos**: se encarga de insertar, actualizar, eliminar y consultar información.  
 **Seguridad y control de acceso**: gestiona los permisos y protege los datos sensibles.  
**Manejo de transacciones**: asegura que las operaciones se realicen de forma correcta (todo o nada).  
**Optimización del rendimiento**: mejora la eficiencia de las consultas y el almacenamiento.  
**Respaldo y recuperación**: permite hacer copias de seguridad y restaurar datos en caso de fallas.  
Ejemplos de Sistema gestor de bases de datos:

| **Tipo** | **Ejemplo** | **Descripción breve** |
| --- | --- | --- |
| **Relaciona** | MySQL | Muy usado en aplicaciones web. Gratuito. |
|  | PostgreSQL | Avanzado, open-source, muy potente. |
|  | Oracle | Comercial, muy usado en empresas grandes. |
|  | SQL Server | Microsoft, fuerte en entornos Windows. |
| **NoSQL** | MongoDB | Usa documentos en lugar de tablas. |
|  | Firebase | Ideal para apps móviles y en tiempo real. |

Por qué se usa este sistema de gestor de base de datos:Porque sin él, tendríamos que gestionar toda la información manualmente, lo que sería muy complicado, lento y propenso a errores. El SGBD hace que todo el trabajo con datos sea:

Más rápido   
Más organizado   
Más seguro