

# Base de Datos II

# Docentes



Angel Simón - Profesor  
[asimon@docentes.frgp.utn.edu.ar](mailto:asimon@docentes.frgp.utn.edu.ar)



Laura Vélez - Jefe de trabajos prácticos  
[lvelez@docentes.frgp.utn.edu.ar](mailto:lvelez@docentes.frgp.utn.edu.ar)



Abel Faure - Jefe de trabajos prácticos  
[afaure@docentes.frgp.utn.edu.ar](mailto:afaure@docentes.frgp.utn.edu.ar)



Brian Lara Campos - Jefe de trabajo prácticos  
[blara@docentes.frgp.utn.edu.ar](mailto:blara@docentes.frgp.utn.edu.ar)

# Contenidos

## Relacionales (con SQL Server)

- Creación de Tablas y Restricciones.
- Consultas de acción
- Consultas de selección:
  - Where, Order By, Top, Distinct
  - Joins
  - Funciones de Resumen
  - Subconsultas
- Objetos de Base de datos:
  - Funciones de usuario
  - Vistas
  - Procedimientos almacenados
  - Triggers
- Transacciones

## No Relacionales (con MongoDB)

- Modelado de Bases de Datos No Relacionales
- Operaciones CRUD: Inserción, modificación y eliminación de documentos
- Consultas Básicas de Documentos
- Consultas Avanzadas y Optimización de Búsquedas

# Evaluaciones

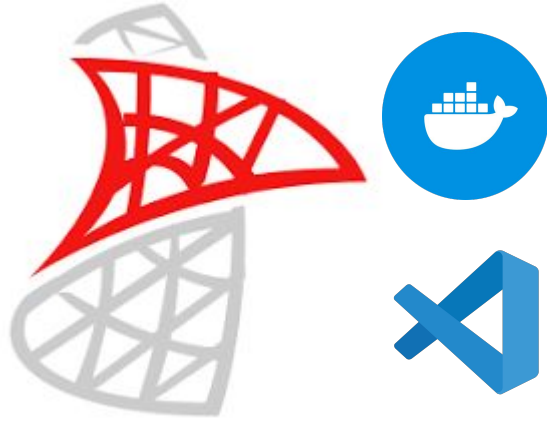
## **Examen 1**

- Examen individual de Bases de Datos Relacionales y Bases de Datos No Relacionales

## **Examen 2 (Trabajo Práctico Integrador)**

- Gestión, presentación y defensa de un proyecto de Base de Datos Relacional

# Software



## GitHub



# Base de Datos Relacionales

# Características



**Representación en Tablas y Columnas**



**Claves Primarias**



**Claves Foráneas**



**SQL (Structured Query Language)**



**Normalización**



**Información consistente y sin redundancia**

# Tipos de datos para representar números

SQL Server	Descripción	C++ (Equivalente)
bit	Valor booleano (0 o 1)	bool
tinyint	Entero sin signo de 8 bits (0-255)	unsigned char o uint8_t
smallint	Entero de 16 bits con signo (-32,768 a 32,767)	short o int16_t
int	Entero de 32 bits con signo (-2,147,483,648 a 2,147,483,647)	int o int32_t
bigint	Entero de 64 bits con signo (-9 trillones a 9 trillones)	long long o int64_t
decimal(p, s)	Número decimal de precisión fija (hasta 38 dígitos)	double o long double (o std::string si se necesita precisión exacta)
smallmoney	Decimal para valores monetarios pequeños (-214,748.3648 a 214,748.3647)	float
money	Decimal para valores monetarios grandes (-922 trillones a 922 trillones)	double

Fuente: <https://learn.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver16>



# Tipos de datos para representar texto

SQL Server	Descripción	C++ (Equivalente)
char(n)	Cadena de texto de longitud fija	char[n] o std::string
varchar(n)	Cadena de texto de longitud variable	std::string
nchar(n)	Cadena Unicode de longitud fija	wchar_t[n] o std::wstring
nvarchar(n)	Cadena Unicode de longitud variable	std::wstring

## varchar vs nvarchar

Característica	VARCHAR(n)	NVARCHAR(n)
Codificación	Usa 1 byte por carácter (ASCII o codificación basada en collation)	Usa 2 bytes por carácter (UTF-16)
Soporte Unicode	❌ No soporta caracteres Unicode	✅ Soporta caracteres Unicode (ej. emojis, caracteres de otros idiomas)

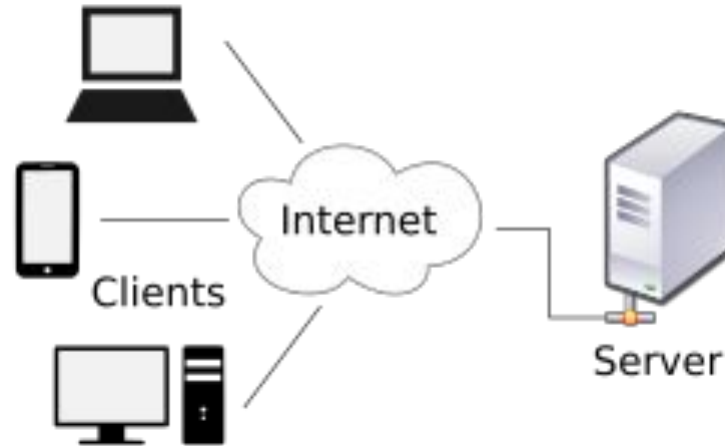
# Tipos de datos para representar tiempo

SQL Server	Descripción	C++ (Equivalente)
date	Fecha sin hora (AAAA-MM-DD)	std::string o clase Fecha propia
datetime	Fecha y hora con segundos (desde 1753 hasta 9999)	std::string o clase FechaHora propia
smalldatetime	Fecha y hora con menos precisión (minutos)	std::string o clase FechaHora propia
time	Solo la hora (hh:mm:ss.nnnnnnn)	std::string o clase Hora propia

## Consejos para manejo de fechas

- Utilizar los tipos de datos apropiados para representar fechas. **Las fechas no se guardan en texto.**
- Un fragmento de fecha u hora se puede guardar en un smallint.
- Un período (Año-Mes) se puede guardar como texto pero utilizar el formato (AAAA-MM) para asegurarse que sea correctamente ordenable. Ej: 2025-03
- Una marca de tiempo es una opción para representar fecha y hora en algunas ocasiones.

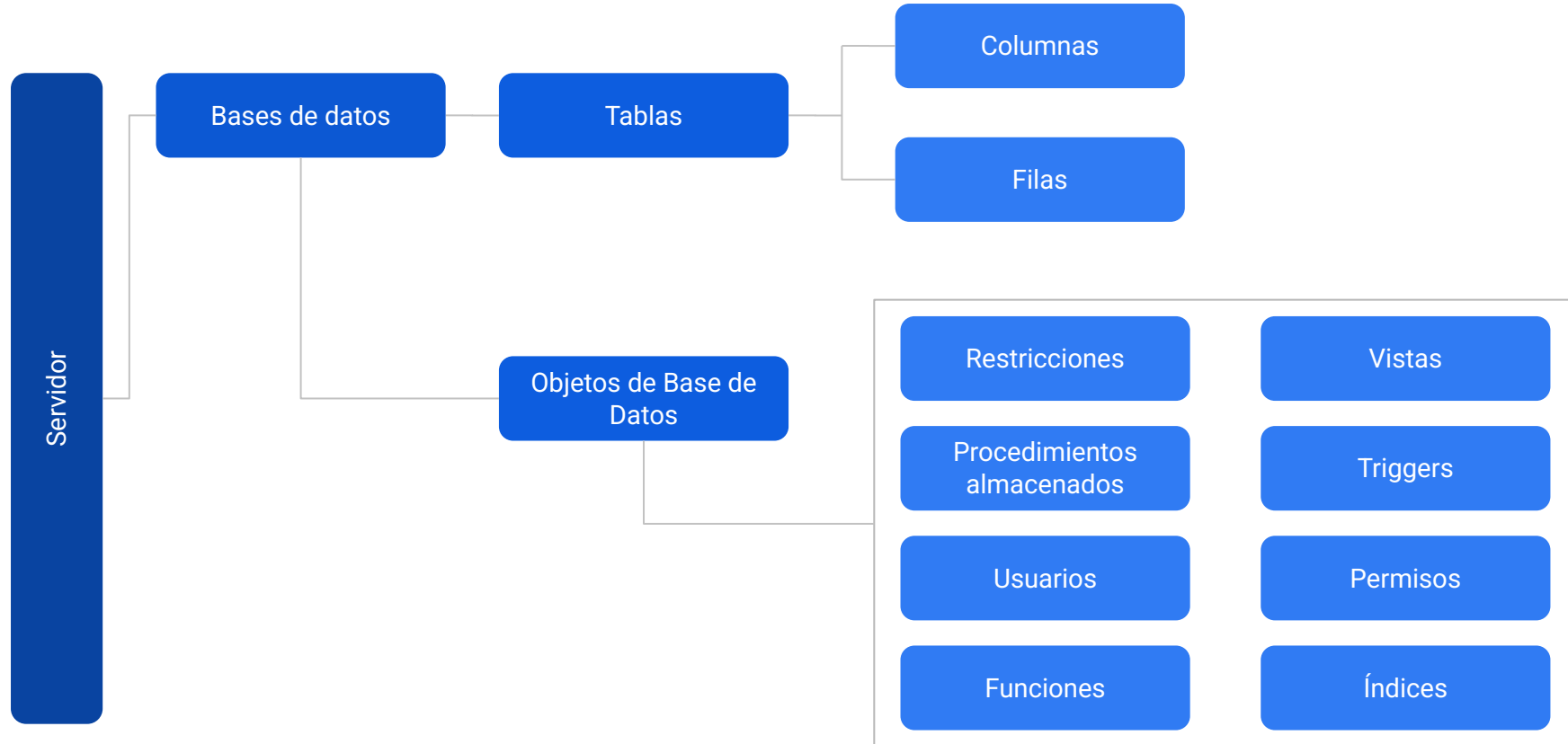
# Bases de datos: Arq. Cliente Servidor



# Ejemplos

Ejemplo	Cliente	Protocolo del Servidor
Correo electrónico	Microsoft Outlook, Mozilla Thunderbird, Gmail App	Servidor SMTP, POP3, IMAP
Servicios web	Navegador web, aplicaciones móviles, etc	Servidor HTTP, HTTPS
Bases de datos	Visual Studio Code, Azure Data Studio, pgAdmin, MySQL Workbench	TCP/IP
Terminales de pago	App de ecommerce, celular, tarjeta de crédito o débito, etc.	NFC, USB, HTTPS, WSS, etc.
Juegos en línea	Steam, Epic, Gog	TCP/IP, UDP
Chats	WhatsApp, Telegram, Facebook Messenger, Slack	HTTP, HTTPS Websockets (WSS)

# Bases de datos: Elementos de bases de datos



# Modelado de una Base de Datos Relacional

#	Descripción
1	Identificar entidades y relaciones
2	Normalizar la base de datos
3	Elegir claves primarias adecuadas
4	Definir claves foráneas correctamente
5	Evitar la redundancia de datos
6	Usar tipos de datos adecuados
7	Mantener la integridad referencial
8	Pensar en la escalabilidad
9	Incluir restricciones y validaciones

# Restricción

**Una restricción es una condición impuesta a una columna o conjunto de columnas para asegurar la integridad de la información.**

 Clave primaria / única

 Clave foránea

 Check

 Not Null

 Default

# Modelo relacional: Clave primaria/única

**Restricción que no permite valores duplicados en la/s columna/s definida como clave.**

Legajo	Apellidos	Nombres
1000	Simón	Angel
2000	Vélez	Laura
<del>2000</del>	Faure	Abel



# Modelo relacional: Clave primaria/única

Restricción que no permite valores duplicados en la/s columna/s definida como clave.

Legajo	IDSucursal	Apellidos	Nombres
1000	100	Simón	Angel
1000	200	Lara	Brian
2000	200	Vélez	Laura
<del>1000</del>	<del>200</del>	Faure	Abel

# Modelo relacional: Clave foránea

**Restricción que no permite valores en la/s columna/s definidas como clave que no existan en la/s columna/s definidas como referentes.**

Legajo	Apellido	IDCarrera
1000	Simón	1
2000	Vélez	1
3000	Lara	2
4000	Faure	<del>10</del>

ID	Nombre	Nivel
1	Técnico Universitario en Programación	Tecnicatura
2	Ingeniería mecánica	Grado
3	Maestría en seguridad informática	Posgrado

# Modelo relacional: Nulidad

**La capacidad de indicar que un atributo no posee un valor.**

Legajo	Apellido y nombres	IDCarrera	Fecha graduación
1000	Simón, Angel	1	10/3/2019
2000	Vélez, Laura	1	11/3/2019
3000	Lara, Brian	2	25/3/2019
4000	Faure, Abel	3	null

# Modelo relacional: Check

**Regla de validación que posee una columna y que debe cumplirse.**

**Ejemplos:**

- Una fecha debe ser menor a la fecha de hoy.
- Un importe debe ser mayor a cero.
- Un número debe estar entre 1 y 10.
- Un carácter sólo puede contener E, C, T
- El largo de una cadena debe ser mayor a 5.

NOTA: Si una columna acepta valores nulos y posee un check. Entonces el valor nulo es válido ante la validación del check.

# Normalización y Codificación de Base de Datos

## Ejercicio de Normalización

<https://docs.google.com/spreadsheets/d/18Bjl1pT3lS5uFtwzmt4GXfB-HJ1OF40lHB-PglSeJZI/edit?usp=sharing>