

Introducción a las bases de datos

Cuidar las base de datos es cosa de todos,
filosofía DevOps

Por: Nicola Strappazzon C

Bio

Soy Nicola Strappazzon Carotenuto, Licenciado en ciencias de la computación en Venezuela desde el 2006, Siempre he desarrollado, administrado servidores y bases de datos, ésta última ha sido siempre mi pasión. Trabajo con base de datos desde los últimos 10 años en empresas cómo Rakuten, Letgo, Binlogic, Zinio, The Hotel Network y muchas más.

nstrappazzonc@gmail.com

@debeando

<https://debeando.com>

Agenda

- Advertencia
- ¿Qué es y porqué usar un gestor de base de datos?
- ¿Qué limitaciones tiene un gestor de base de datos?
- ¿Cómo funciona un gestor de base de datos?
- ¿Que gestores de base de datos existen?
- ¿Cómo escoger el gestor de base de datos?
- ¿Cual es el lenguaje de un gestor de base de datos?
- ¿Cuales son los componentes de un gestor de base de datos?
- ¿Porqué hay que especificar el tipo de dato?
- ¿Cómo escalan las bases de datos?

Advertencia

- Esta sección del curso trata de cubrir de forma genérica y conceptual el funcionamiento general de muchos gestores de base de datos como; MySQL, MariaDB, PostgreSQL, Cassandra, Kafka, Redis, y MongoDB.
- El uso de ciertos términos puede ser propio de un gestor de base de datos y en otros casos común para la mayoría.
- Toda tecnología usada es Open Source.

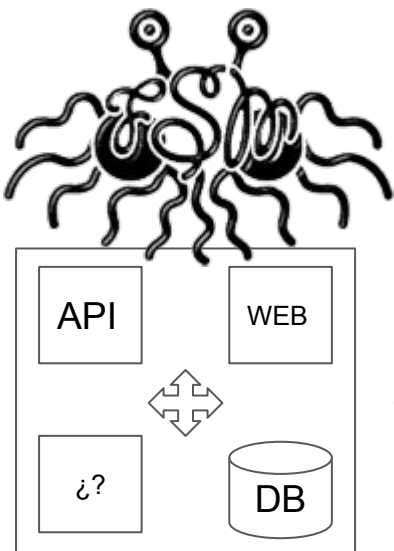
Objetivo

Entender los términos básicos, las arquitecturas, la replicación, la normalización de los datos, el hardware, las tecnologías, y cómo todo está relacionado.

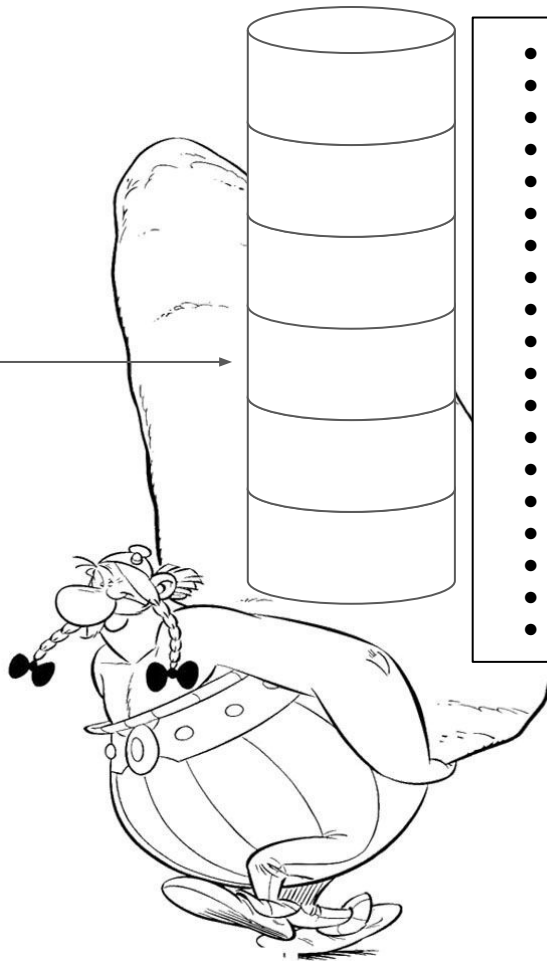


¿Qué es un gestor de base de datos?

- Es un software que permite tener un control los datos.
- Gestiona la concurrencia y el aislamiento para tener consistencia de los datos.
- Garantiza la calidad de guardar (durabilidad) los datos y poder recuperar antes un incidente.
- Tiene un lenguaje para manipular datos.
- Ofrece niveles de seguridad para el acceso y manipulación.
- Gestiona la replicación y/o distribución de los datos en un cluster.



Menhir / Monolito



- 300 tablas
- 50 columnas
- 6 Tb de datos
- 4 Tb de índices
- Búsquedas lentas
- Consultas de 500 líneas
- Tablas al gusto
- Un cambio, una semana
- Aquí no hay quien viva
- Imposible particionar
- Tablas por años
- 8 horas para calcular estadísticas
- Caídas constantes
- Todo inoperativo
- Producto no puede innovar
- El de sistemas no duerme
- El negocio pierde dinero
- Cuesta más que un equipo de IT
- Reingeniería para que sea usable

¿Porqué usar un gestor de base de datos?

Resguardar los datos en un único punto.

Se asegura que los datos están correctamente grabados.

Sabe gestionar la concurrencia de los datos.

Gestión de cambios atómicos.

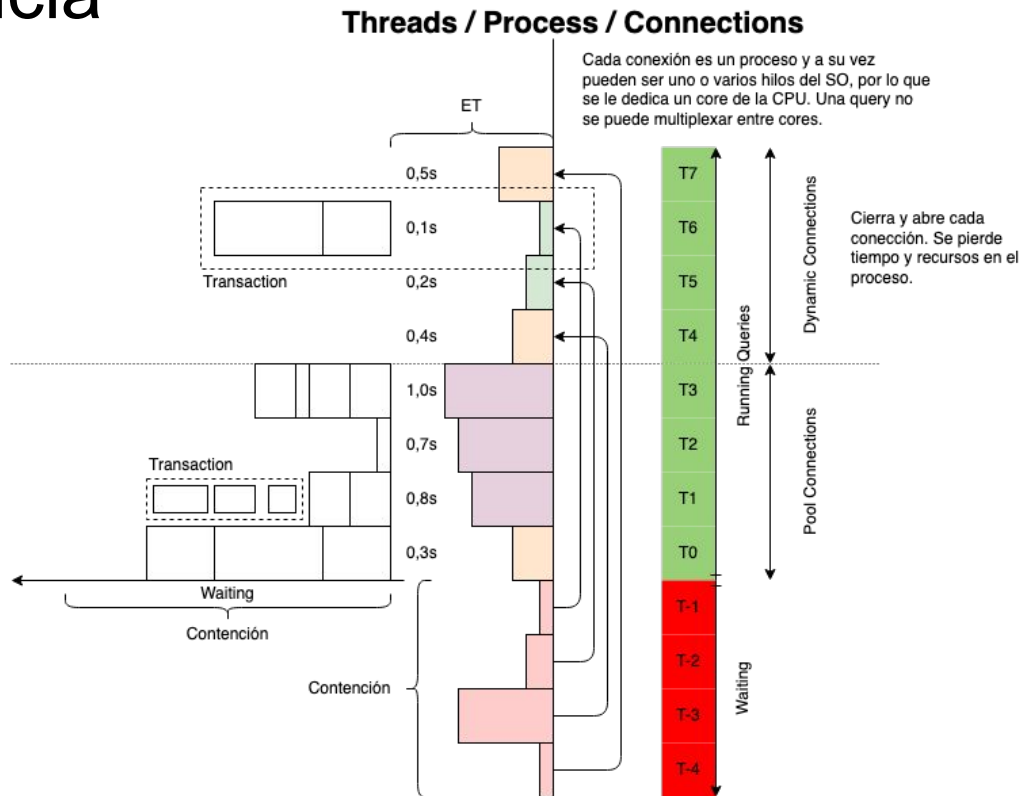
Puede ofrecer mecanismos de recuperación, manipulación y cálculo de datos.

Algoritmos diseñados para un propósito específico en alto rendimiento.

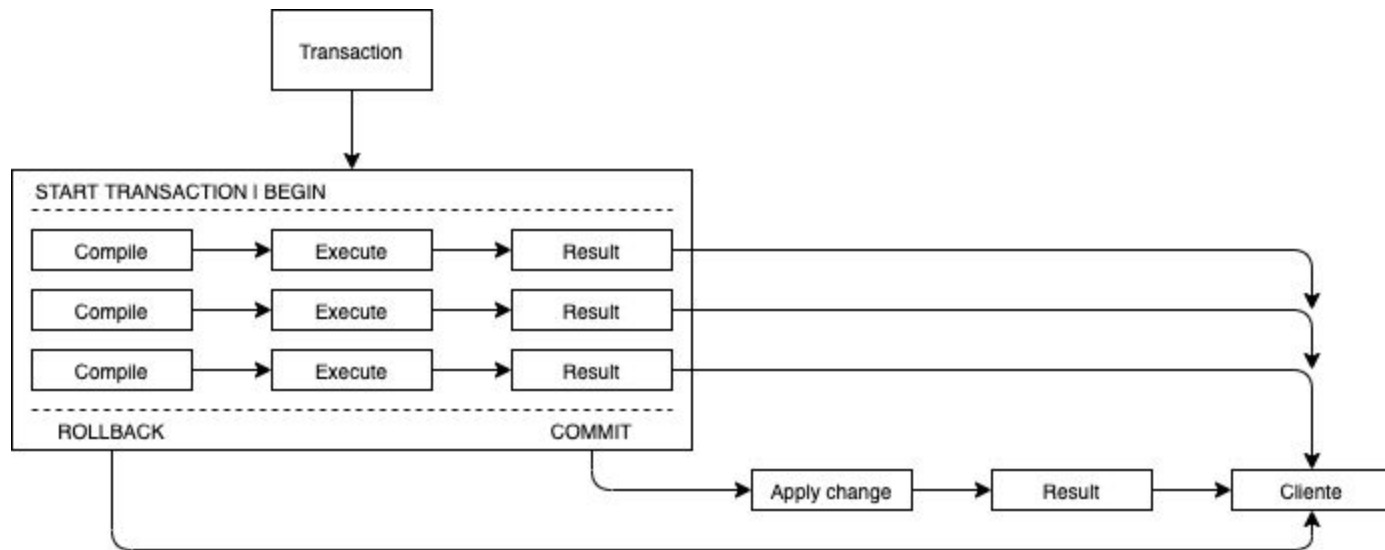
ACID es la clave

- Atomicidad: Todos los cambios o nada.
- Consistencia: Último grabado, disponible par leer.
- Aislamiento (Isolation): Una transacción no se pisa con otra.
- Durabilidad: Asegura lo grabado.

Concurrencia



Transacción

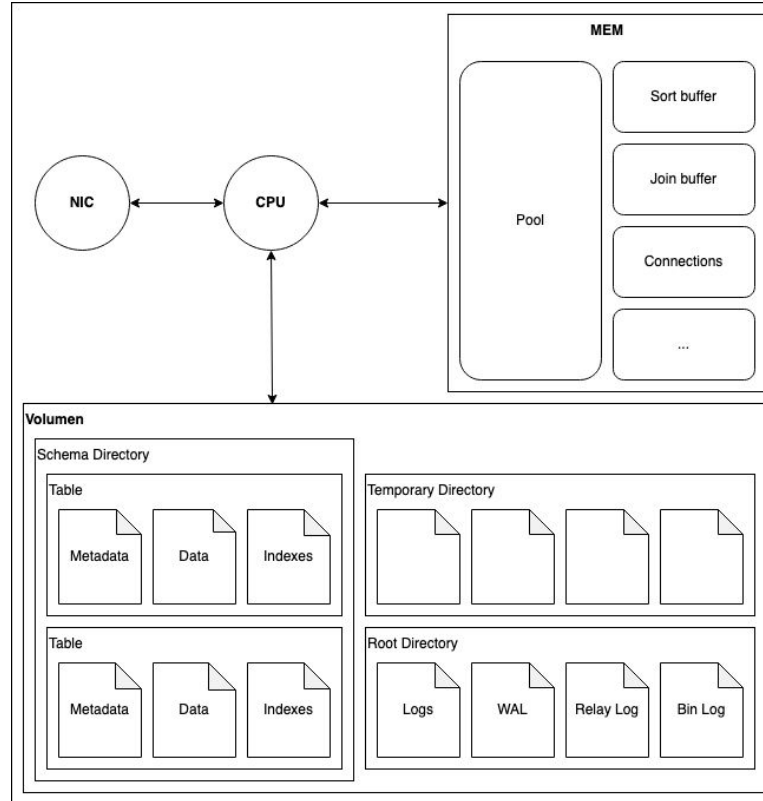


¿Qué limitaciones tiene un gestor de base de datos?

Existen los límites físicos y tecnológicos: cómo la memoria, procesador y volumen entre otros.

Los límites tecnológicos están dados por los diferentes gestores, en el mercado existen muchos y para un propósito.

¿Cómo funciona un gestor de base de datos?



¿Que gestores de base de datos existen?

Existen estos tipos de tecnologías para gestionar los datos y un gestor de base de datos puede hacer uso de una o de varias de estas:

Tipo	Engine	Casos	Ejemplos
Relacional	MySQL, PostgreSQL	Datos delicados	users,payments,
Key Value	Redis,Kafka	Colas, Tokens, Eventos	auth, jobs
Documentales	MongoDB	Información compleja y cambiante.	Items,Details,Bio,Wiki
Wide Column Stores	ClickHouse	Analizar datos	Visits, Stats
Search	ElasticSearch,Solar	Analisis idiomático del texto	Logs,
Stream	Kafka	Cola de eventos a procesar	Logs, Events
Graph	Neo4j, TerminusDB		
Time Series	InfluxDB	Graph	Logs, Visits, Stats
Engine	Spark	Analizar datos	Visits, Stats

Paradigma de las SQL y NoSQL

- NoSQL es la forma de describir todo aquello que no es “relacional”, pero es una palabra con mucho marketing sin valor científico.
- Por más que se haga referencia a una No Relacional, algunas de ellas tienen componentes relacionales y/o ACID cómo MongoDB entre otros.
- Las NoSQL agrupa todo aquello que guarda los datos como clave valor; MongoDB, Redis, ElasticSearch.

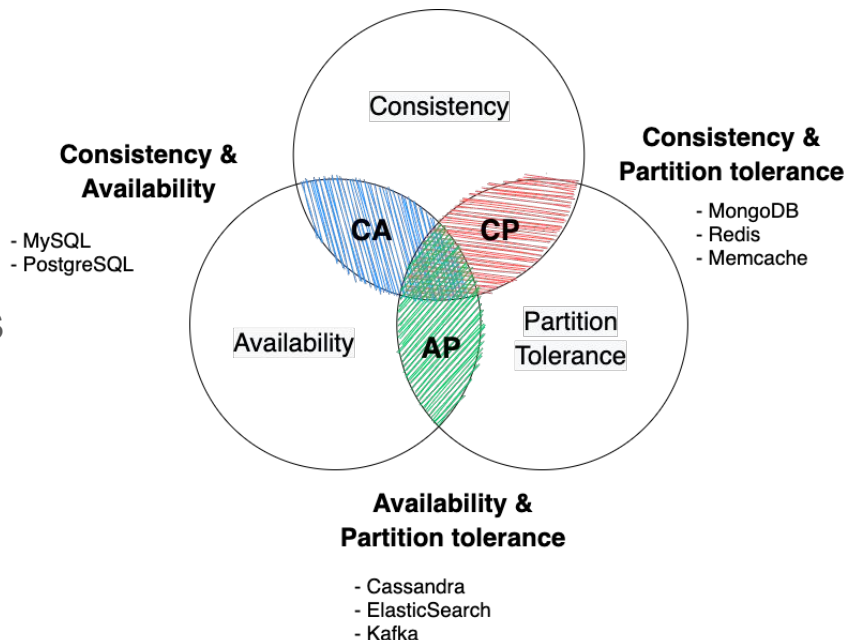
¿Cómo escoger el gestor de base de datos?

Teorema de CAP permite orientar cual es el mejor gestor según las necesidades reales, creando un equilibrio entre las deficiencias y bondades de cada grupo.

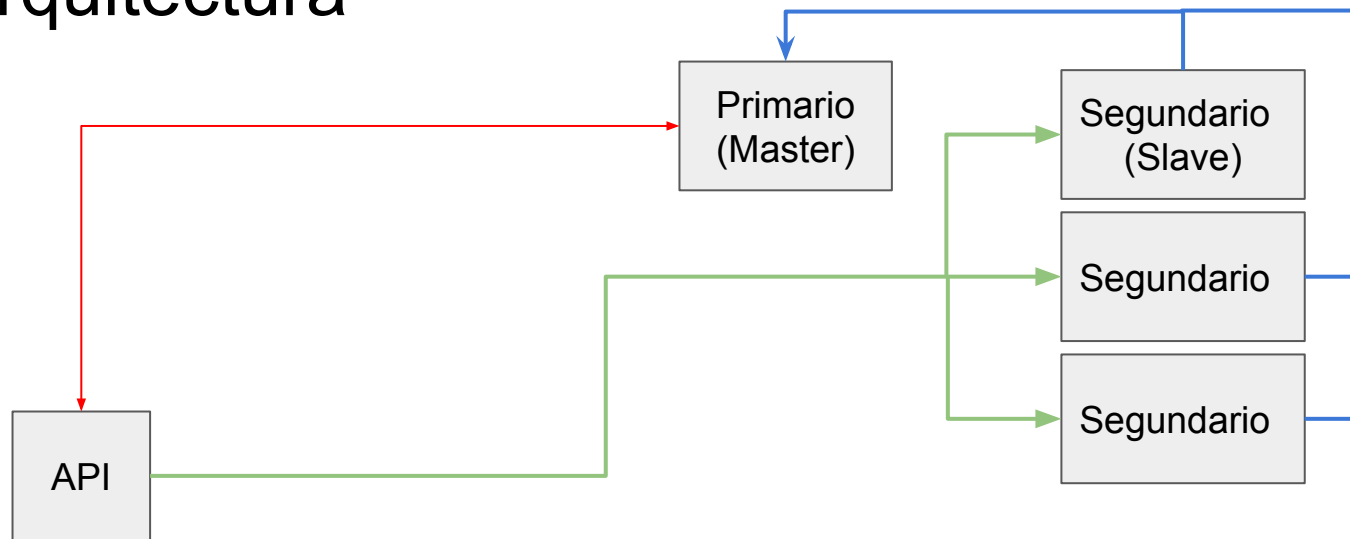
Consistencia: Todas las réplicas tienen la escritura más reciente.

Availability: Cada petición recibe una respuesta, sin garantizar la escritura más reciente.

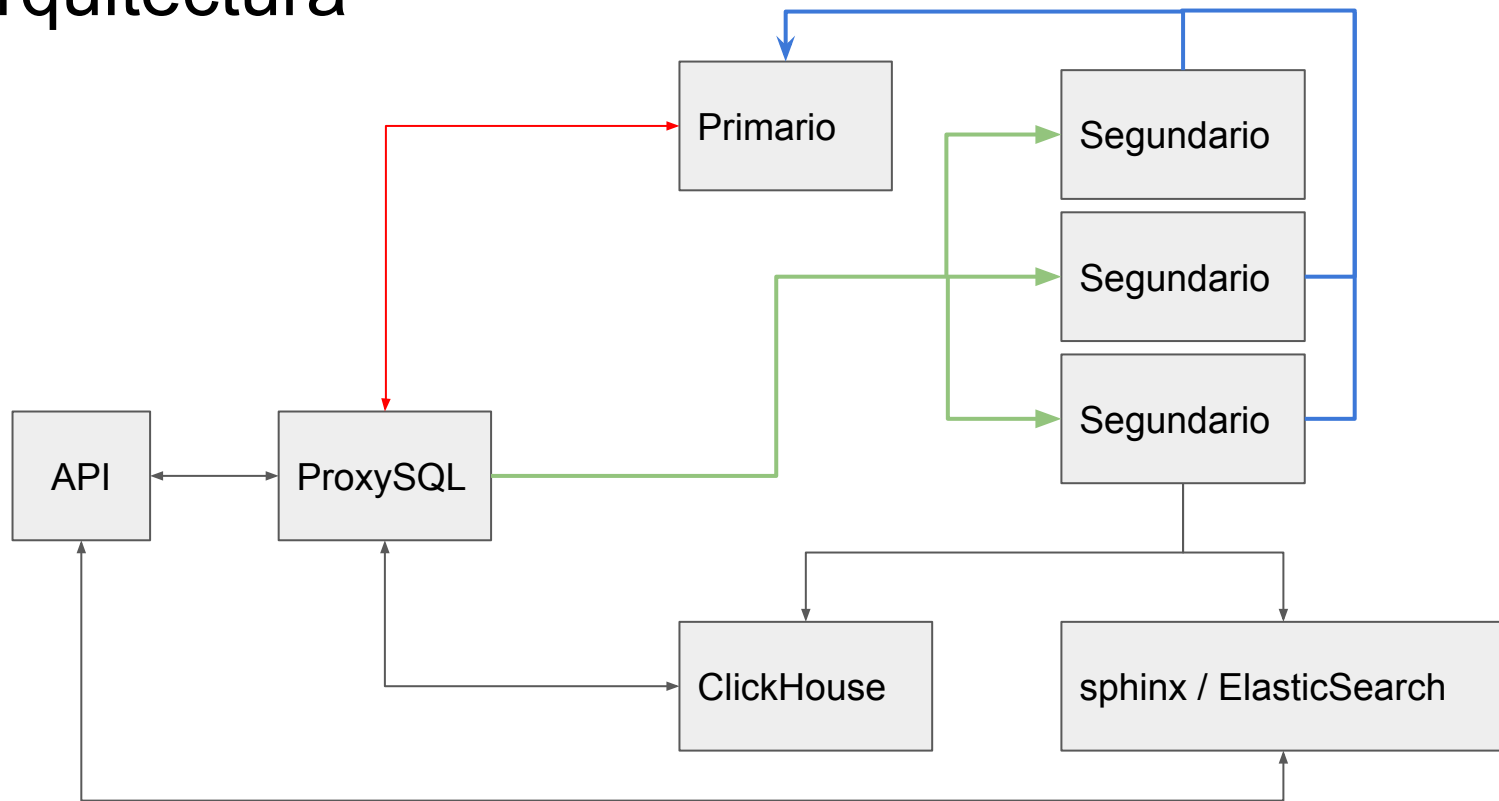
Partition tolerance: El sistema debe operar sin importar los problemas de red.



Arquitectura



Arquitectura

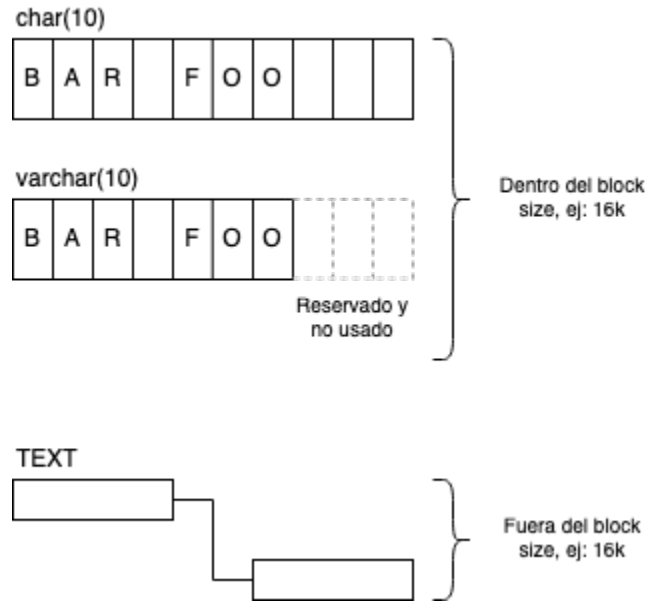


¿Cómo influye el modelo de datos?

1. Mientras mayor es la normalización, mayor es el consumo de recursos.
2. El modelo debe expresarse sin documentación cómo está relacionado una tabla con otra haciendo un simple seguimiento intuitivo con nombres predictivos y usar un estándar igual cuando se programa.
3. Debe haber un equilibrio entre rendimiento y consistencia.
4. Divide y vencerás; será más fácil identificar los cuellos de botella, escalar y la distribución de recursos como de incidencias.

¿Por qué los tipos de datos?

Para optimizar el uso de la memoria, de la CPU, del volumen y del bus.



Términos

Listar los términos importantes para recordar a lo largo de la charla:

1. Gestor de base de datos / Engine
2. Aislamiento
3. Concurrencia
4. ACID
5. Atomicidad
6. Transacción
7. Primario / Secundario (Master / Slave).
8. Teorema de CAP
9. SQL

Material de apoyo

1. Documentación oficial.
2. <https://es.slideshare.net/nicola51980>
3. <https://www.mysqltutorial.org>
4. <https://sqltest.net>
5. <https://debeando.com>