

## ¿Qué es el teorema CAP y cómo elegir la base de datos para un proyecto?.

Cuando hablamos de almacenar y gestionar información hay varios conceptos que se van haciendo cada vez mas relevantes y que es indispensable que tengas en cuenta a la hora de seleccionar el tipo de [base de datos](#) que vas a usar, estos conceptos son parte de lo que conocemos como Requisitos de Calidad o requerimientos no funcionales.

Los requerimientos no funcionales son aquellos atributos inherentes a la operación del sistema y de su comportamiento. Este tipo de atributos es importante definirlos desde el comienzo del desarrollo del proyecto porque es a partir de ellos que se toman las decisiones mas costosas del proyecto.

Por ejemplo: ¿vas a desarrollar un proyecto en el que la tasa de crecimiento de usuarios es del 50% mensual o del 5% anual?, necesitas que tu aplicación soporte transaccionalidad o es solo un sitio de consulta?, este tipo de preguntas definen el rumbo de la arquitectura de tu proyecto.

Pero, ¿qué tiene esto que ver con la selección de la base de datos que vas a usar en tu proyecto? Tiene mucho que ver, y para esto vamos a hablar de tres atributos de calidad que servirán de base para la selección de tu base de datos:

## *El Teorema CAP:*

---

En [Ciencias de la computación](#), el teorema CAP, también llamado (Conjetura de Brewer), enuncia que es imposible para un [sistema de cómputo distribuido](#) garantizar simultáneamente:

La [consistencia](#) (Consistency), es decir, cualquier lectura recibe como respuesta la escritura más reciente o un error.

La [disponibilidad](#) (Availability), es decir, cualquier petición recibe una respuesta no errónea, pero sin la garantía de que contenga la escritura más reciente.

La [tolerancia al particionado](#) (Partition Tolerance), es decir, el sistema sigue funcionando incluso si un número arbitrario de mensajes son descartados (o retrasados) entre nodos de la red.

Según el teorema, un sistema no puede asegurar más de dos de estas tres características simultáneamente.

# Visual Guide to NoSQL Systems



Debemos tener en cuenta las exigencias de nuestro proyecto para saber qué atributos de calidad necesitamos y así elegir el tipo de base de datos que necesitaremos.

CP (Consistencia y Tolerancia al particionamiento): Esto quiere decir que no se garantiza la disponibilidad, hay clientes que requieren que el sistema esté disponible 100% del tiempo o muy cerca, con bases de datos que cumplan con CP no es posible garantizar esto, no quiero decir que no se pueda lograr en cierto nivel, pero el sistema está enfocado en aplicar los cambios de forma consistente aunque se pierda comunicación con algunos nodos.

AP (Disponibilidad y Tolerancia al particionamiento): En este caso no se garantiza que los datos sean iguales en todos los nodos todo el tiempo, en este caso el sistema siempre estará disponible para las peticiones aunque se pierda la comunicación entre los nodos.

CA (Consistencia y disponibilidad): En este caso no se puede permitir el particionado de los datos, porque se garantiza que los datos siempre son iguales y el sistema estará disponible respondiendo todas las peticiones.

Por ejemplo, los sistemas de bases de datos relacionales (SQL de toda la vida) son CA porque todas las escrituras y lecturas se hacen sobre la misma copia de los datos.

Si en tu proyecto lo que requieres es que todos tus clientes tengan acceso a la misma vista de los datos en cualquier momento que sean requeridos (Consistencia) y además desde el comienzo sabes que se va a requerir disponibilidad de los datos casi el 100% del tiempo (Tolerancia al Particionamiento), el tipo de base de datos que deberías usar es una como MongoDB o Redis.

## MongoDB

---

Bases de datos como MongoDB o Redis son llamadas bases de datos NoSQL o Not Only SQL, estas son tipos de bases de datos que se han optimizado para solventar diferentes necesidades en tus

proyectos, por ejemplo MongoDB te proporciona velocidad en el almacenamiento y acceso a la información consistentemente y es realmente fácil, rápido y seguro implementar opciones de sharding y particionamiento, si quieres ver como hacerlo te invito a la [clase en vivo](#) de MongoDB.

Hay muchas otras implementaciones de bases de datos no relacionales, conocerlas y conocer sus capacidades te va a ayudar a elegir la que mejor se ajusta a tus necesidades, si te interesa ver algunas implementaciones reconocidas de bases de datos no relacionales te recomiendo el post de [Implementación de bases de datos no relacionales](#) en el que Yohan Graterol, profesor del [Curso de MongoDB y Redis](#) nos enseña sobre este tema.

En el [curso de MongoDB y Redis](#) veremos qué es NoSQL y porqué deberías usar este tipo de bases de datos en tus proyectos además aprenderás a utilizar MongoDB, uno de los motores de bases de datos NoSQL más importantes de la actualidad.