

REPORTE
CERTAMEN 1
TALLER DE BASES DE DATOS
AVANZADAS

TOMÁS SOLANO

Tabla Comparativa

Para esta tabla comparativa se utilizará tanta información obtenida utilizando funciones de registro en TypeScript, estos registros se encuentran dentro del archivo comprimido (.ZIP) donde se encuentra este reporte, también se utilizará información obtenida de fuentes diferentes estas estarán disponibles en la bibliografía. Para los costos de espacio físico se usa de referencia una máquina EC2 de amazon.

Nombre	MongoDB	DynamoDB	Cassandra
Costo (\$USD) Base de datos	\$USD 0 (Open-Source), por lo cual técnicamente es posible utilizar un computador personal para la integración del sistema.	Depende del uso y el plan utilizado, en demanda se obtiene lo siguiente: Lectura: \$US1.25 por millón de RRUs (unidad de lectura) Lectura: \$US6.25 por millón de WRUs (unidad de escritura) Almacenamiento: \$US0.25 por GB-Mes	\$USD 0 (Open-Source), por lo cual técnicamente es posible utilizar un computador personal para la integración del sistema.
Costo (\$USD) plataforma física (EC2)	T1.MICRO (US\$0 para estudiantes de AWS Academy, precio general: US\$0.02 en demanda)	\$US 0, cobro por uso, vive en la nube	T2.MEDIUM (US \$ 0.0464 por hora en demanda)
Tiempo de respuesta promedio queries (ms)	132 ms	140 ms	133.6 ms
Tiempo de respuesta query de alta complejidad con datos no ordenados (ms)	1409 ms	19318 ms	20084 ms

Facilidad de instalación	Simplificada, versión moderna existente con baja dependencia de librerías existentes, y bajo uso de recursos, las bases de datos y tablas (colecciones son creadas automáticamente), además de tener sistemas automatizados de procesamiento de datos con json (y objetos “raw” de javascript)	Sencilla, DynamoDB es SaaS por lo que solo hay necesidad de crear tablas sea por AWS o código, es más complejo si se desea añadir distintas relaciones entre distintas tablas (GSI), por lo cual se suele mantener una formación de una sola tabla.	Compleja, la documentación para la última versión es baja, mientras que la versión anterior (4.0), utiliza sistemas deprecados, es necesario crear todas las keyspaces (bases de datos/schema), tablas y sus atributos de antemano a su uso, por lo cual requiere planificación para su utilización.
Disponibilidad de documentación	Alto nivel de documentación e integración, no solo se puede encontrar la documentación de mongo, sino que la comunidad basada en mongo es alta, por lo cual se pueden encontrar distintas librerías para simplificar y mejorar el proceso de integración y producción utilizando mongo (ejemplo: mongoose)	Documentación de AWS disponible, contiene introducción para principiantes tanto como ejemplos de nivel avanzado	Documentación de apache baja, solo contiene ideas básicas del sistema, además presenta problemas de disponibilidad por uso de librerías deprecadas (bajo nivel de actualización del sistema)
Impacto en la industria	Utilizado por: eBay, Electronic Arts, Uber, Linkedin, Sega, Cisco y otros	Utilizado por: Netflix, Duolingo entre otros.	Utilizado por: Instagram, IoTs, análisis real y entre otros.
Donde destacan	Videojuegos, una alta cantidad de compañías de desarrollo de videojuegos utiliza mongo	Sistemas estructurados con flexibilidad de información, por ejemplo, un software de Streaming de video podría utilizar DynamoDB	Sistemas estructurados con flexibilidad de presentación, sistemas que son interactivables o intercambiables entre otros que pueden

	especialmente por su flexibilidad con objetos, lo cual es altamente compatible con videojuegos que suelen ser programados utilizando metodologías OOP.	como base debido su estructura sistematizada pero con necesidad de flexibilidad vertical.	utilizar la información de cada componente, por ejemplo, Instagram, el cual añade componentes cada cierto tiempo que utilizan sistemas anteriores para su funcionamiento donde también es muy importante la rapidez de escritura y lectura, la cual cassandra es capaz de entregar en $O(1)$
--	--------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Análisis

MongoDB:

A la hora de trabajar en un sistema de datos compuestos, complejos u objetos el uso de mongo es recomendado, en la comunidad de mongo existe la librería de mongoose, esta librería se encarga de la automatización de distintos procesos siendo posible usar un sistema de nivel alto, sin preocupación de los procesos internos, esto da espacio para preocuparse del proyecto general, teniendo esta información en cuenta los puntos donde mongo destaca son en ellos cuales la información puede ser volátil, con profundas relaciones (embebido profundo) o flexible, esto, fue probado con una query para este certamen, donde se ingresó un objeto con alto nivel de profundidad (ficha médica de un paciente), la cual contiene en su interior en formato objeto las revisiones o historial y dentro de estas los procedimientos donde se pueden encontrar las identificaciones de los distintos médicos, al buscar esta información mongo se demora 1.4s comparados con su competencia que estuvieron cerca de los 20s (se utilizó ALLOW FILTERING en cassandra, con el fin de comparar la obtención de información para un caso parecido, el cual es obtención de información en tiempo real de información sin procesamiento, sea un objeto de un videojuego y sus estadísticas), esto da a conocer las aplicaciones en las cuales mongo sería mejor: Videojuegos, Aplicaciones de negocio y operacional donde es complicado mantener la información volátil (sea un E-Commerce, por ejemplo), o internet de las cosas, las cuales suelen ser programas utilizando metodología OOP.

Cassandra:

Cassandra es un sistema altamente estructurado, especialmente en comparación con su competencia, por lo cual requiere un nivel de planificación alto en su integración y escalabilidad, para este certamen fue imposible estructurar esta base de datos completamente para maximizar la potencia de este sistema, sin embargo, cuando es utilizado de forma correcta la rapidez de lectura y escritura no tiene rival, ya que utiliza tablas para obtener y actualizar la información (por ejemplo, un promedio de tablas se mantendría en una tabla y actualizar cuando sea necesario, por lo cual la lectura y escritura es simplificada), sin embargo, aquí nace su mayor desventaja, todo existe en tablas, si no se encuentra en tablas la obtención de datos no es mejor que otras. Es por esto, que su uso es mejor en sistemas de los cuales se conoce la idea de antemano, los cuales necesitan flexibilidad vertical en su creación, sin embargo la actualización de información y componentes no es alta además de ser, más importante la integración entre componentes que nada, es por eso que sistemas como redes sociales es donde destaca cassandra, en una red social los sistemas y la estructura de esta es pensada de antemano, sin embargo, es necesario que el sistema sea capaz de adherir o desacoplar componentes cuando sea requerido, sin cambiar la estructura principal del sistema.

DynamoDB:

Dynamo destaca en un punto entre medio de Mongo y Cassandra, sistemas que necesitan flexibilidad, pero de igual manera necesitan mantener cierta estructura, esta puede cambiar o puede acoplar otros componentes al sistema, por lo cual en sistemas de Streaming de Videos Online, su uso es recomendado, debido a que el acople y desacople de videos haciendo usos de Sort Keys, sería simplificado además de, mantener una estructura para la filtración y visualización de información. De hecho, Netflix hace uso de DynamoDB, por razones similares a estas, se puede observar el uso de esto en como es visualizada la información, utilizando categorías y otros sistemas de metadatos para la presentación de vídeos, lo cual lleva a otra parte en la cual Dynamo destaca, su uso de metadatos para la organización es mejor que su competencia, ya que simplifica la búsqueda y obtención de datos, además del filtramiento, esto es un punto clave que tiene, debido a que tanto mongo como cassandra no son capaces de utilizar un sistema de metadatos de forma inmediata, por lo cual cuando se esta lidiando con lectura de datos basada en metadatos, dynamo sería la mejor opción a tomar en cuenta, por ejemplo, si se quisiese ingresar toda la información de la librería de Alejandría, la cual crearía una cantidad gigantesca de metadatos, Dynamo sería capaz de filtrar la información para su recuperación con mayor agilidad, además de tener conexión con S3, para guardar la información compleja necesaria.

Para la veterinaria

Para la veterinaria, viendo que es un sistema el cual puede contener bastante información, la cual puede aumentar en cualquier momento, añadir datos no estructurados, o nuevos datos de información de sus pacientes, utilizaría DynamoDB, esta permitiría a la veterinaria a añadir distintos metadatos dependiendo de la consulta dada por un paciente, la cual puede cambiar dependiendo de la consulta, sea añadir síntomas, distintos procedimientos u otros. Esta información, siendo tratadas como metadatos, se podrán visualizar de manera simplificada al hacer queries dependiendo de estos metadatos, también sería fácil añadir información. Por otro lado, usando DynamoDB daría la estructura necesaria para crear un sistema multi-tablas con baja relación del cual obtener y registrar información tendría un alto balance entre velocidad y sencillez de implementación.

En resumen, como la información que recibe la veterinaria es de importante acceso, variable y filtrable, se descarta el uso de cassandra la cual destaca en tener su información dividida y no filtrada, mientras que mongo es mejor para casos de sistemas de interacción, como es necesario, filtrar la información además de tener información que pueda cambiar, las funciones de metadatos de DynamoDB son la mejor opción.

Justificación DataLake

Para la veterinaria se utilizo un sistema de data lake, con posibilidad de data warehouse usando nest js, ya que de esta forma la interacción de información entre las distintas bases de datos se simplifica al tener un objeto normalizado el cual tome toda la información y cree relaciones entre las distintas bases de datos, esto con dos fines en específico, uno, para el procesamiento de datos, tanto subida como bajada de información de tal forma que simplifique la generación de un frontend para la veterinaria, por otra parte, se desea capacidades de escalamiento, de tal manera que al crear objetos es posible, por una parte, ingresar información de cualquier base de dato que la veterinaria llegase a usar solo siendo necesario crear una capa de traducción para cada uno, y también cuando (y pasará) que se necesite la información centralizada se puede enviar esta en un formato normalizado creando así una data warehouse.

Bibliografía

- ❓ Dynomate. (s.f.). *DynamoDB Pricing Calculator*. Dynomate. Recuperado el 17 de julio de 2025, de <https://dynomate.io/blog/dynamodb-pricing-calculator/>
- ❓ NestJS. (s.f.). *MongoDB*. NestJS. Recuperado el 17 de julio de 2025, de <https://docs.nestjs.com/recipes/mongodb>
- ❓ CloudySave. (s.f.). *Use Cases of AWS DynamoDB*. CloudySave. Recuperado el 17 de julio de 2025, de <https://www.cloudysave.com/aws/use-cases-of-aws-dynamodb/>
- ❓ upGrad. (s.f.). *MongoDB Real World Use Cases*. upGrad. Recuperado el 17 de julio de 2025, de <https://www.upgrad.com/blog/mongodb-real-world-use-cases/>