

Universidad Don Bosco

Dirección de Educación a Distancia

Diseño y Programación de Software Multiplataforma.



Docente: Ing. Alexander Alberto Sigüenza Campos

Foro de Discusión #1:

GRUPO: G01T

PRESENTADO POR:

NOMBRES	CARNET
Menéndez Zepeda, Josué Erasmo	MZ142582
Vásquez Villalta, Luis Eduardo	VV121782

FECHA DE ENTREGA: 28/04/2024

Índice:

Contenido

Introducción	1
Contenido de la Investigación.....	2
Cloud Firestore:.....	3
Realtime Database:.....	4
Diferencias entre Cloud Firestore y Realtime Database.....	5
Diferencias entre una Base de Datos SQL y una NoSQL.....	7
Conclusiones de investigación.....	8
Ejercicio Práctico.....	9

Introducción

El siguiente documento tiene la finalidad de desarrollar y definir los conceptos claves de una base de datos, en este caso tocaremos como ejemplo la base de datos Firebase, una base de datos que tiene mucho de qué hablar, los encargados detras de su desarrollo es google, una gran empresa de alto impacto en el mundo de las tecnologías y la informática, ademas esta base de datos posee ciertas características que hacen que sea muy robusto y visionaria, ya que va dirigida al manejo actual de los datos, dejando atrás los paradigmas convencionales y adentrándose en el mundo del noSQL.

Al ser una base de datos orientada al desarrollo de aplicaciones móviles se ha vuelto muy popular entre los desarrolladores, ya que permite cierta rapidez al momento de procesar los datos, es flexible y no requiere un servidor como tal para ser desplegado, es por ello tenemos una comparativa para mostrar las ventajas actuales que ofrece la base de datos noSQL sobre los anteriores tipos de base de datos relacionales.

El documento finaliza con una conclusión de parte del equipo de cual se ajusta mejor a su uso para un aplicativo usando la tecnología de React y muestra un ejemplo de cómo usar una base de datos SQL de una forma sencilla y práctica.

Contenido de la Investigación.



Firebase






















Primeramente, antes que todo veremos que es **Firebase** dirigida por **Google**: es una **plataforma en la nube** para el desarrollo de aplicaciones web y móvil. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo.

Sus herramientas son variadas y de fácil uso, considerando que su agrupación simplifica las tareas de gestión a una misma plataforma. Es especialmente interesante para que los desarrolladores no necesiten dedicarle tanto tiempo al backend, tanto en cuestiones de desarrollo como de mantenimiento.

Los distintos módulos que componen Firebase se pueden dividir en tres grandes grupos categorizados según su finalidad:

- **Compilación:** permite iniciar nuestro backend sin administrar servidores y escalar conforme nuestra aplicación va creciendo.
- **Lanzamiento y supervisión:** nos permite gestionar el lanzamiento de nuestra aplicación y supervisar el buen funcionamiento de esta.

- **Participación:** podemos aumentar la participación de los usuarios en nuestra app, comprendiendo cómo estos la utilizan, personalizarla para diferentes segmentos de usuarios, ejecutar experimentos para probar ideas de nuevas funcionalidades...

Compilación	Supervisión y lanzamiento	Participación
Authentication iOS+   C++  	Crashlytics iOS+ 	Analytics iOS+   C++ 
Realtime Database iOS+   C++ 	Performance Monitoring iOS+ 	Remote Config iOS+   C++ 
Cloud Firestore iOS+   C++   	Test Lab iOS+ 	A/B Testing iOS+ 
Cloud Storage iOS+   C++ 	App Distribution iOS+ 	Cloud Messaging iOS+   C++ 
Kit AA iOS+ 		In-App Messaging iOS+ 
Hosting 		Dynamic Links iOS+   C++ 
Cloud Functions iOS+   C++ 		Google AdMob iOS+  C++ 
		Google Ads iOS+ 

Cloud Firestore:

Es la base de datos más reciente de Firebase. Tiene un modelo de datos nuevo y más intuitivo. Permite realizar consultas más complejas y rápidas, y el escalamiento se ajusta a un nivel más alto que Realtime Database.

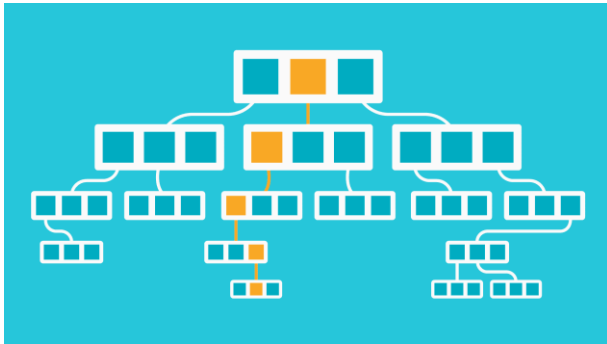
Es una base de datos de documentos NoSQL que permite almacenar, sincronizar y consultar fácilmente datos en tus apps web y para dispositivos móviles a escala global.

Características

- **No requiere servidores:** Esta base de datos, completamente gestionada por Google, no requiere del uso de algún servidor externo, lo que facilita el proceso de escalar los recursos con el objetivo de ajustarse a alguna demanda que tengas como cliente.
- **Actualización y sincronización en tiempo real:** contribuyen en el desarrollo de diferentes tipos de aplicaciones, especialmente las de tipo multiusuario o colaborativa en la web para dispositivos móviles, así como las relacionadas con el internet of things.
- **Asistencia sin conexión:** contiene un modo sin conexión que se encarga de almacenar en el caché aquellos datos que tu aplicación utiliza de manera frecuente y activa. Esto permite que tu app pueda realizar operaciones de lectura y escritura, así como escucha y consulta de datos, incluso en los momentos en el que el dispositivo esté sin conexión. Una vez el equipo vuelve a conectarse, Cloud Firestore se encargará de sincronizar todos tus cambios locales en la plataforma.

- **Flexibilidad:** permite el uso de estructuras de datos flexibles y que mantengan una jerarquía. Esto contribuye a que pueda almacenar tus datos en documentos que se organizan a través de colecciones. Dichos documentos pueden incluir objetos anidados de gran complejidad, así como sub-colecciones con más datos contenidos.

Consulta y estructura datos de la manera que prefieras:



Usa colecciones y documentos para estructurar tus datos con facilidad. Crea jerarquías para almacenar datos relacionados y recuperar los datos que necesitas mediante consultas expresivas de manera sencilla. Todas las consultas se escalan con el tamaño del conjunto de resultados (y no con el del conjunto de datos), por lo que tu aplicación está lista para escalar desde el primer día.

Realtime Database:

Es una base de datos alojada en la nube. Los datos se almacenan en formato **JSON** y se sincronizan en tiempo real con cada cliente conectado. Cuando compilas aplicaciones multiplataforma con nuestros SDK de plataformas de Apple, Android y JavaScript, todos tus clientes comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes.

Realtime Database proporciona un lenguaje flexible de reglas basadas en expresiones, llamado reglas de seguridad de Firebase Realtime Database, para definir cómo se deberían estructurar los datos y en qué momento se pueden leer o escribir. Integrar Firebase Authentication permite que los desarrolladores definan quién tiene acceso a qué datos y cómo acceden a ellos.

Realtime Database es una **base de datos NoSQL** y, como tal, tiene diferentes optimizaciones y funcionalidades en comparación con una base de datos relacional. La API de Realtime Database está diseñada para permitir solo operaciones que se puedan ejecutar rápidamente

Características

- **Tiempo real:** Tal como su nombre indica, Realtime Database ofrece sincronización en tiempo real, lo que permite que los datos se actualicen instantáneamente en todos los dispositivos conectados.

- **Escalabilidad:** Firebase se encarga de ajustar los servidores o la administración de recursos a medida que aumentan los datos, por lo que no necesitamos manejar manualmente esas funcionalidades.
- **Offline:** Firebase también permite almacenar localmente los datos en el dispositivo y se sincronizan automáticamente con la base de datos en la nube ni bien se restablece la conexión.
- **Seguridad:** Proporciona autenticación y reglas de seguridad configurables para proteger tus datos.



Diferencias entre Cloud Firestore y Realtime Database.

Principalmente las diferencias entre estas dos funcionalidad o módulos que nos ofrece Firebase de Google se puede ver como los siguientes puntos importantes:

- Función de la base de datos:
 - ❖ Cloud Firestore: Realizar búsquedas, transacciones y ordenamientos avanzados.
 - ❖ Realtime Database: Sincronizar datos con consultas básicas principalmente.
- Operaciones con datos:
 - ❖ Cloud Firestore: Gran cantidad de datos (desde cientos de GB hasta TB de datos) que se leen con mucha más frecuencia de la que se cambia.
 - ❖ Realtime Database: Una cantidad de datos menor (algunos GB de datos o menos) que cambian con frecuencia.
- Modelo de datos:

- ❖ Cloud Firestore: Documentos organizados en colecciones, Los datos simples son fáciles de almacenar en documentos, que son muy similares a JSON.
- ❖ Realtime Database: Un árbol JSON simple.
- Disponibilidad:
 - ❖ Cloud Firestore: Muy importantes, como puede ser en apps de comercio electrónico. Ofrece compatibilidad multirregional, que garantiza un tiempo de actividad muy alto (99,99%).
 - ❖ Realtime Database: Altas, pero no críticas. Ofrece compatibilidad regional, que garantiza un tiempo de actividad menos alto (99,95%).

Resumiendo, en el siguiente cuadro:

	Cloud Firestore	Realtime Database
Función de la BBDD	Consultas avanzadas	Consultas básicas
Modelo de datos	Datos estructurados	Árbol JSON simple
Eficiencia de las consultas	Elevada	Baja
Actualización en tiempo real	Disponible	Disponible

Otros Aspectos importantes que tomar en cuenta:

- Escalabilidad:
 - ❖ Cloud Firestore: El escalamiento es automático, Escala de forma completamente automática. Los límites de escalamiento son de 1 millón de conexiones simultáneas y 10,000 operaciones de escritura por segund.
 - ❖ Realtime Database: El escalamiento necesita fragmentación, scala hasta alrededor de 200,000 conexiones simultáneas y 1,000 escrituras por segundo en una misma base de datos. Para un escalamiento mayor que ese, se deben fragmentar los datos en distintas bases de datos.
- Seguridad:
 - ❖ Cloud Firestore: Reglas sin formato de cascada que combinan autorización y validación.

- ❖ Realtime Database: Lenguaje de reglas en cascada que separa la autorización de la validación.

➤ Precios:

- ❖ Cloud Firestore: Se cobra principalmente por operaciones ejecutadas en la base de datos (lecturas, escrituras y eliminaciones) y, con una tarifa menor, por ancho de banda y almacenamiento.
- ❖ Realtime Database: Se cobra solo por ancho de banda y almacenamiento, pero con una tarifa mayor.

Diferencias entre una Base de Datos SQL y una NoSQL.

Las diferencias entre una base de datos SQL y una NoSQL son muy notorias, el siguiente cuadro muestra las diferencias más notables entre ellas:

SQL	NoSQL
Modelo relacional, son base de datos con estructuras relacionadas y muy bien normalizadas.	Al no usar un modelo relacional estas ocupan otros métodos de interacción tal es el caso de grafos, documentos o datos con clave-valor
Requiere una estructura fija y modelo bien definido	No requieren un modelo fijo, permiten la creación de datos con estructuras flexibles
Integridad de datos	Diseñadas para ser altamente disponibles y por poseer la cualidad de tolerar fallos
Se destacan por utilizar el lenguaje SQL para el manejo de consultas y la manipulación de datos	Pierden características al momento de generar transacciones ACID
Se emplea para garantizar una consistencia de datos donde se tiene un esquema bien definido	Orientadas a la escalabilidad y flexibilidad.
Visto comúnmente en el área de sistemas de gestión empresarial, sistemas bancarios, sistemas de reservas entre otros.	Visto comúnmente en aplicaciones web, sistemas de análisis con un gran volumen de datos, redes sociales entre otros.
Escalabilidad vertical	Escalabilidad horizontal

Conclusiones de investigación.

¿Cuál de estas bases de datos consideran que sería la mejor opción para implementar en una aplicación desarrollada en React Native?

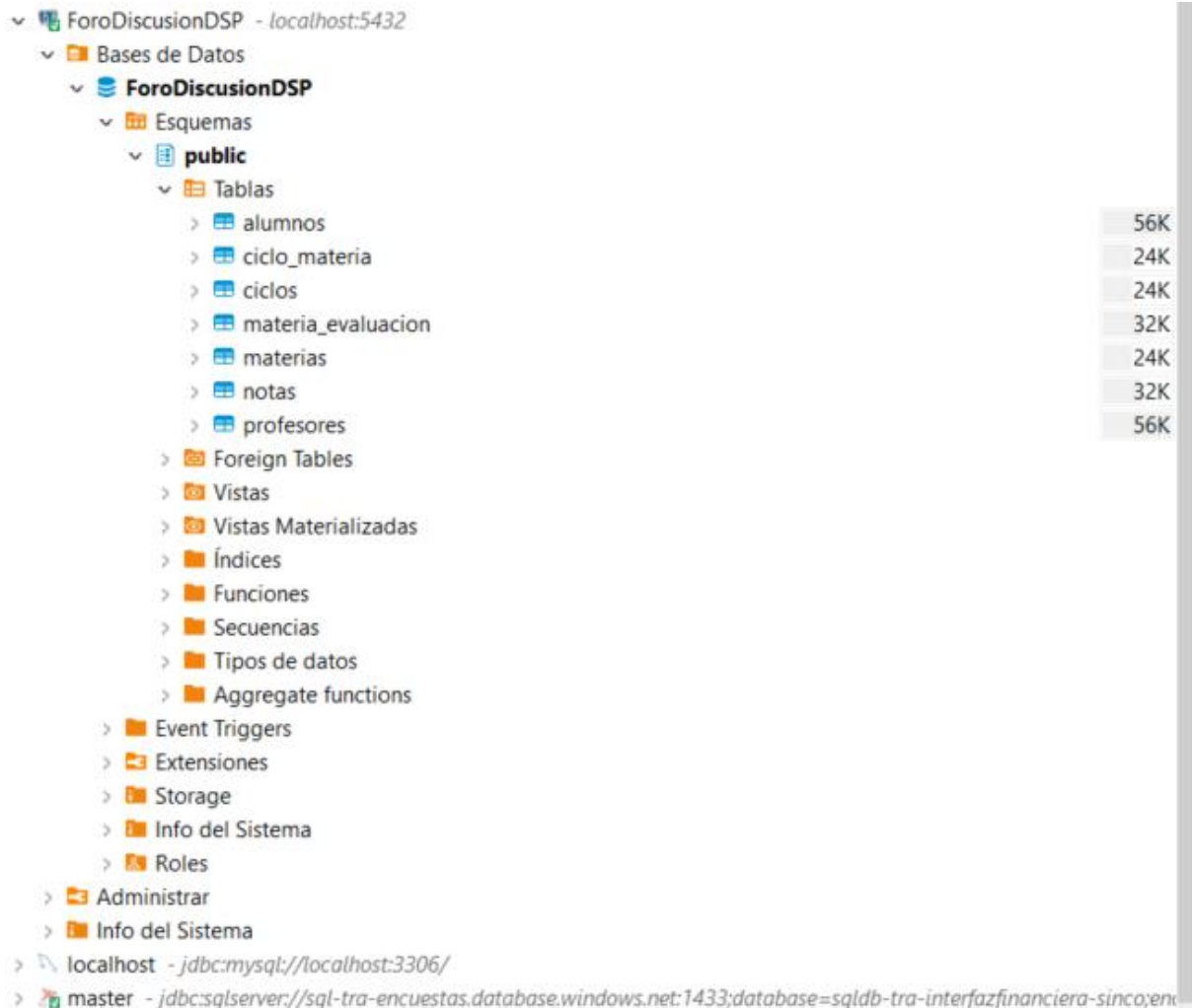
Para el caso concentro de ser una aplicación con React Native y en el cual toda la lógica de negocio se aplique desde el lado del aplicativo móvil lo más sencillo sería incorporar una de las tecnologías investigadas debido que ha se adapta perfectamente a este ambiente en el cual no debemos de estar invirtiendo más tiempos en el desarrollo del BackEnd por lo cual toda la información (data) sería almacenada directamente luego del manejo que se le dé en el aplicativo con React Native.

Caso contrario si fuera recomendable que se construya una arquitectura que contemple una mayor complejidad separando el Front-end y el Back-end aplicando toda la lógica de negocio desde este último punto dando una mayor robustez a la hora de la creación del producto final y dándole simplemente la función de mostrar toda la información almacenada de la base de datos del lado del Front-end.

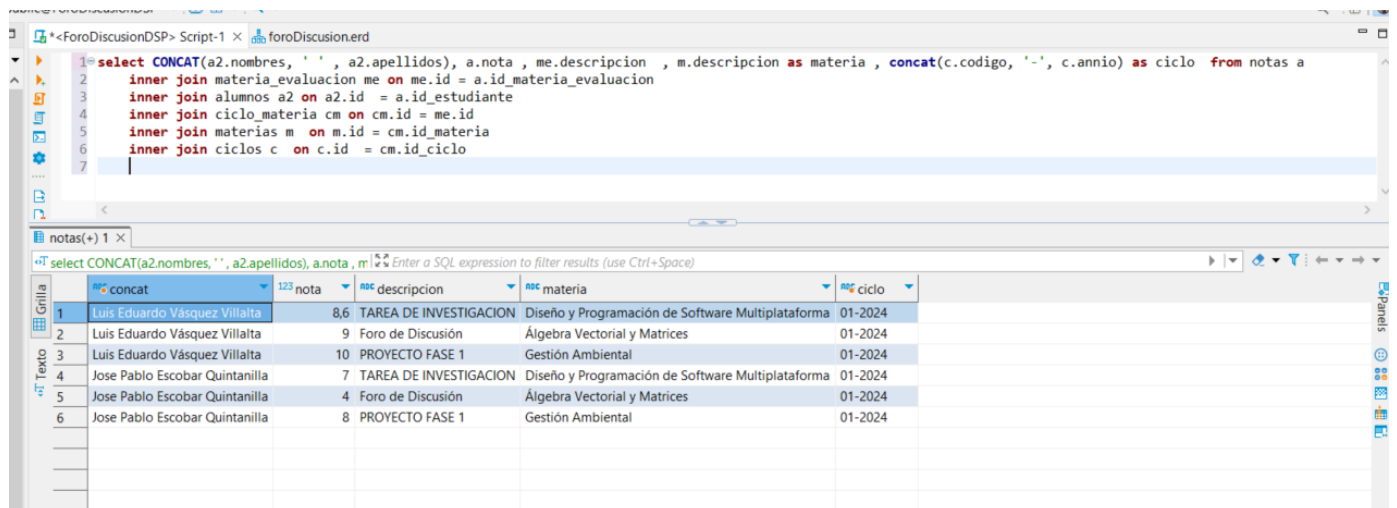
Ejercicio Práctico.

Para el ejercicio de la base de datos Relacional se ocupará una PostgreSQL.

En el script va la creación de las tablas y la inserción de registros. Donde podemos ver que si hay información (data) en las tablas.



Haciendo una consulta para observar los datos serian la siguiente:



Y el diagrama entidad Relación aplicando la normalización sería el siguiente:

