



**Diseño y Programación de Software
Multiplataforma (DPS941)**

Foro de Discusión 1

Docente: Alexander Sigüenza.

Ciclo 02 2023

Integrante (s)

Carne

Miranda Ramírez, Rodrigo Alexander

MR181415

Villalta, Rigoberto Alcides

VV00329

Villeda López, Miriam Janeth

VL191443

1. ¿Cuáles son las diferencias fundamentales entre bases de datos SQL y NoSQL?

- **Modelo de Datos**
 - **SQL:** Utiliza un modelo de datos tabular con esquemas predefinidos y estructuras rígidas.
 - **NoSQL:** Ofrece flexibilidad en el modelo de datos, permitiendo almacenar datos de diferentes tipos y estructuras.
- **Escalabilidad**
 - **SQL:** Suele ser más difícil escalar horizontalmente debido a la estructura fija de tablas.
 - **NoSQL:** Proporciona una escalabilidad mas sencilla, ya que puede dividirse en clústeres y escalares horizontalmente.
- **Transacciones ACID.**
 - **SQL:** Ofrece transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) para garantizar la integridad de los datos.
 - **NoSQL:** No garantiza transacciones ACID en todas las operaciones; algunas bases de datos NoSQL priorizan la disponibilidad y la tolerancia a fallos.

2. ¿Cuáles son las diferencias específicas entre Cloud Firestore y Realtime Database?

- **Modelo de Datos:**
 - Cloud Firestore: Utiliza un modelo de documentos y colecciones que permite una estructura jerárquica de datos.
 - Realtime Database: Utiliza un modelo JSON con una estructura plana de datos.
- **Escalabilidad:**
 - Cloud Firestore: Ofrece una escalabilidad más sencilla y automática, ya que está diseñado para manejar grandes cantidades de datos.

- **Realtime Database:** Escalabilidad menos automática y puede requerir una planificación adicional para escalarse a gran escala.
- **Consultas y Búsquedas.**
 - **Cloud Firestore:** Ofrece consultas mas potentes y flexibles, incluyendo consultas compuestas y filtrado avanzado.
 - **Realtime Database:** Las consultas son más limitadas y requieren índices adicionales para consultas complejas.
- **Sincronización en tiempo real.**
 - **Cloud Firestore:** Proporciona sincronización en tiempo real a nivel de documento y colección.
 - **Realtime Database:** Ofrece sincronización en tiempo real a nivel de nodo, lo que puede ser más granular.

3. Basándose en su investigación, ¿cuál de estas bases de datos consideran que sería la mejor opción para implementar en una aplicación desarrollada en React Native?

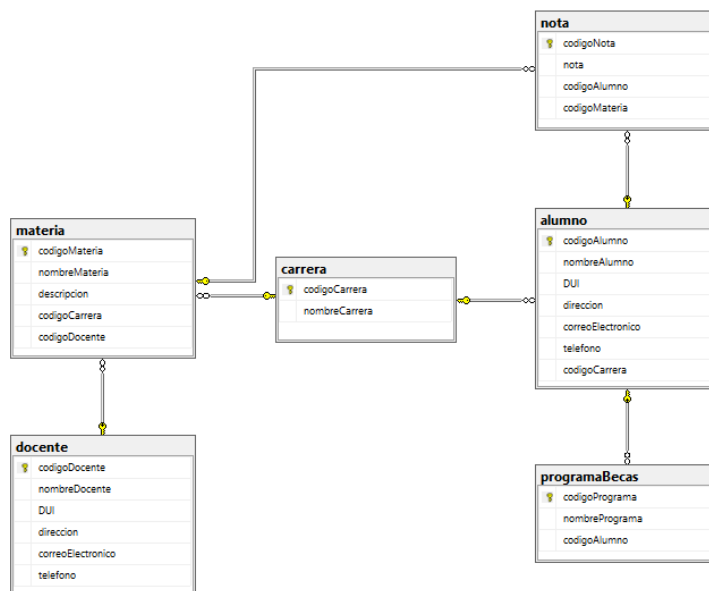
La elección entre Cloud Firestore y Realtime Database para una aplicación en React Native depende de las necesidades específicas de la aplicación.

- Si la aplicación requiere una estructura de datos jerárquica y escalabilidad automática, Cloud Firestore puede ser la mejor opción.
- Si la aplicación necesita sincronización en tiempo real a nivel de nodo y está dispuesta a hacer frente a ciertas limitaciones en las consultas, Realtime Database podría ser la elección adecuada.
- Si la aplicación prioriza la flexibilidad en el modelo de datos y la capacidad de consultar datos de manera más compleja, Cloud Firestore es una mejor opción.
- En última instancia, la elección entre Cloud Firestore y Realtime Database dependerá de los requisitos específicos de la aplicación y de la familiaridad del equipo con cada base de datos. Puede ser útil realizar una prueba de concepto en ambas bases de datos antes de tomar una decisión final.

Ejercicio práctico – Diseño de Bases de Datos.

1. En esta fase, deben elaborar dos estructuras de bases de datos: una utilizando SQL y otra utilizando NoSQL (Firebase Firestone o Realtime Database). Ambas bases de datos deben permitir almacenar las notas de los alumnos becarios de UDB VIRTUAL.
2. En el caso de la base de datos SQL, se requiere que esté normalizada y que siga las buenas prácticas de diseño de bases de datos relacionales.

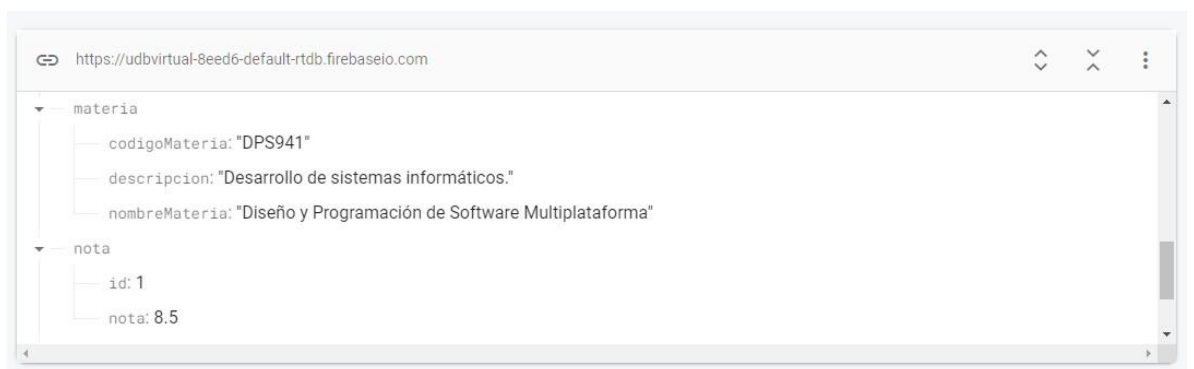
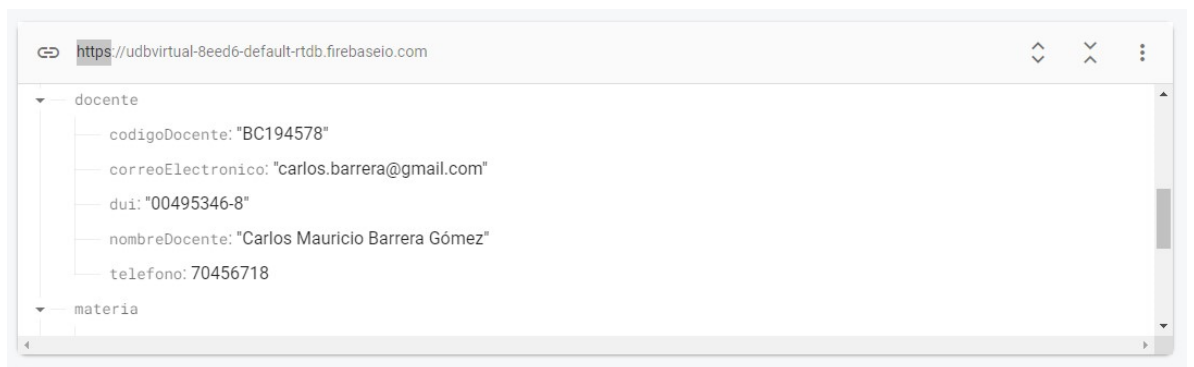
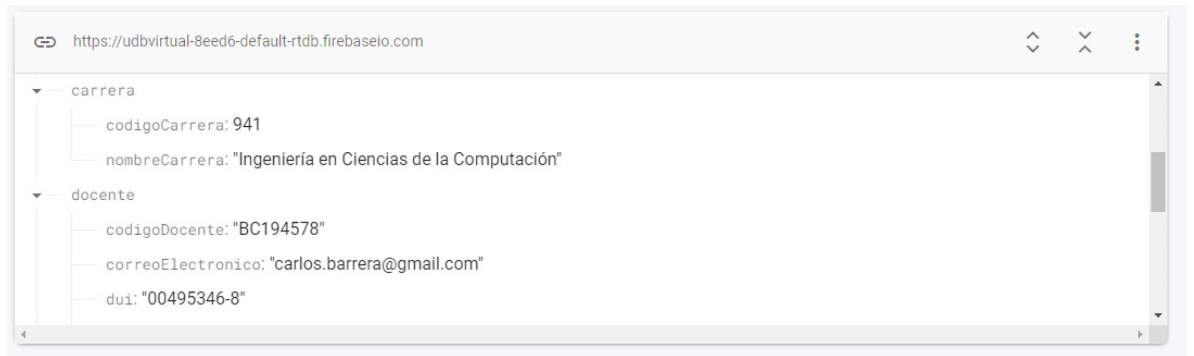
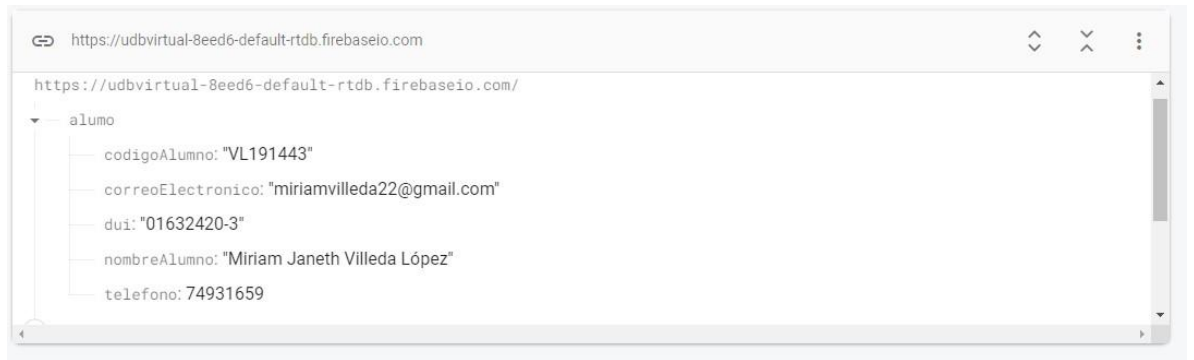
Base de Datos SQL.

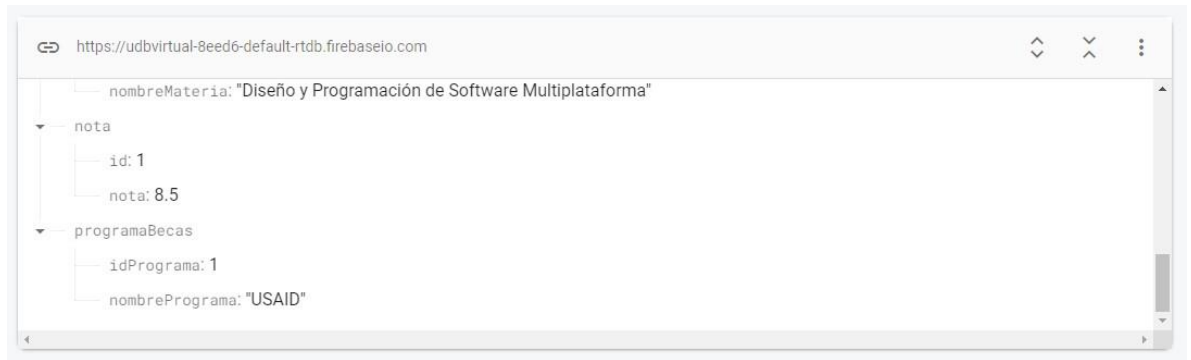


3. Para la base de datos NoSQL, se espera que diseñen una estructura que aproveche las ventajas de Firestone o Realtime Database en Firebase.

Base de Datos NoSQL.







4. Finalmente, se les solicita realizar una comparación de ambas bases de datos y proporcionar conclusiones basadas en su experiencia y en la investigación realizada.

Comparación entre Bases de Datos.

Modelo de Datos.

- **SQL:** Utiliza un modelo de datos tabular con esquema fijo. Los datos se organizan en tablas relacionales con columnas predefinidas.
- **NoSQL:** Puede utilizar varios modelos de datos, como documento, columnas, grafos o clave-valor. Los datos pueden ser semiestructurados o no estructurados

Esquema.

- **SQL:** Requiere un esquema estricto y predeterminado, lo que significa que se debe definir la estructura de la tabla antes de insertar datos.
- **NoSQL:** No requiere un esquema fijo. Se puede agregar campos y estructuras de datos de forma flexible sin necesidad de un esquema rígido.

Rendimiento.

- **SQL:** Buen para consultas complejas y transacciones, pero puede tener un rendimiento inferior en aplicaciones de alta velocidad de lectura/escritura.

- **NoSQL:** Ofrece un rendimiento rápido en aplicaciones con alto volumen de lectura/escritura, como aplicaciones web en tiempo real.

Conclusiones.

- Las bases de datos SQL son altamente eficientes para almacenar grandes cantidades de datos y recuperarlos de manera rápida y precisa. Esto las hace ideales para aplicaciones donde se necesita un acceso rápido y a la información.
- La integridad de los datos es una característica fundamental de las bases de datos SQL. Se pueden establecer restricciones y reglas para garantizar que los datos sean precisos y coherentes, lo que es esencial en entornos donde la precisión de los datos es crítica.
- Las bases de datos SQL ofrecen una variedad de medidas de seguridad, como autenticación, autorización y cifrado, para proteger los datos contra accesos no autorizados. Esto es esencial para la protección de la información confidencial.
- La mayoría de las bases de datos SQL son compatibles con múltiples lenguajes de programación y se integran bien con otras aplicaciones y sistemas, lo que facilita su implementación en una variedad de entornos.
- Las bases de datos NoSQL permiten el almacenamiento de datos de diferentes tipos y formatos, lo que es especialmente útil en aplicaciones con esquemas de datos cambiantes o no estructurados.
- Las bases de datos NoSQL están diseñadas para escalarse horizontalmente, lo que significa que puedan manejar grandes volúmenes de datos y cargas de trabajo distribuidas de manera más eficiente que las bases de datos SQL tradicionales.
- Las bases de datos NoSQL están diseñadas para escalarse horizontalmente, lo que significa que pueden manejar grandes

volúmenes de datos y cargas de trabajo distribuidas de manera mas eficiente que las bases de datos SQL tradicionales.

- En situaciones con altos volúmenes de lectura y escritura, las bases de datos, como documentos, columnas, grafos y clave-valor, lo que permite adaptar la base datos al tipo de datos y consulta específicos.
- Las bases de datos NoSQL ofrecen la flexibilidad de elegir el modelo de datos y el enfoque de escalabilidad que mejor se adapte a las necesidades de una aplicación o sistema.
- Las bases de datos NoSQL pueden ofrecer tiempos de respuesta más bajos en comparación con las bases de datos SQL en determinados casos, lo que es fundamental en aplicaciones en tiempo real.