



Asignatura: Diseño y Programación de Software Multiplataforma.

Docente: Alexander Alberto Siguenza Campos.

.

Integrantes

- Jorge Miguel Alberto Cruz - AC221717.
- Diego Emerson Varela Linares - VL181980.
- Daniel Antonio Marroquin Granados MG161914
- Daniel Adonay Molina Jovel MJ150737

Actividad: Foro 1.

Fecha de entrega: 29/10/2023

Índice

Bases de Datos en Firebase	2
Conclusiones	6
Bibliografía	7
Firestore Realtime database. (s. f.).	
Firestore. https://firebase.google.com/docs/database?hl=es-419	7

Bases de Datos en Firebase

Cloud Firestore

Introducción a Cloud Firestore

Cloud Firestore, una base de datos flexible y escalable de Firebase y Google Cloud, es una solución poderosa para el desarrollo en servidores, dispositivos móviles y la Web. Similar a Firebase Realtime Database, facilita la sincronización de datos entre aplicaciones cliente a través de objetos de escucha en tiempo real. Además, ofrece soporte sin conexión para dispositivos móviles y la Web, lo que te permite construir aplicaciones responsivas que funcionan incluso en condiciones de red intermitente.

Además, Cloud Firestore proporciona una integración sin interrupciones con otros productos de Firebase y Google Cloud, incluyendo Cloud Functions.

Funciones clave de Cloud Firestore:

1. Flexibilidad: El modelo de datos de Cloud Firestore permite estructuras de datos flexibles y jerárquicas, almacenando datos en documentos organizados en colecciones. Estos documentos pueden contener objetos anidados complejos y subcolecciones.

2. Consultas expresivas: Cloud Firestore facilita el uso de consultas para recuperar documentos específicos o conjuntos de documentos de una colección en función de ciertos parámetros de consulta. Las consultas pueden incluir múltiples filtros encadenados y se pueden combinar con criterios de orden. Además, las consultas se indexan de forma predeterminada, lo que garantiza un rendimiento eficiente.

3. Actualizaciones en tiempo real: Al igual que Realtime Database, Cloud Firestore actualiza los datos de dispositivos conectados de manera eficiente, además de permitir consultas de recuperación sencillas y únicas.

4. Asistencia sin conexión: Cloud Firestore almacena en caché datos activamente utilizados por la aplicación, lo que permite operaciones de escritura, lectura, escucha y consulta incluso cuando el dispositivo está sin conexión. Una vez que el dispositivo se conecta nuevamente, Cloud Firestore sincroniza todos los cambios locales de vuelta a la base de datos en la nube.

5. Escalabilidad diseñada: Con una base en la robusta infraestructura de Google Cloud, Cloud Firestore garantiza replicación automática de datos multirregión, consistencia sólida, operaciones atómicas por lotes y un soporte sólido para transacciones.

¿Cómo funciona Cloud Firestore?

Cloud Firestore es una base de datos NoSQL alojada en la nube que tus aplicaciones pueden acceder directamente desde los SDK nativos, así como a través de SDK para Node.js, Java, Python, Unity, C++, y Go, además de las APIs de REST y RPC. El modelo de datos NoSQL de Cloud Firestore almacena los datos en documentos que contienen campos con valores asignados y se organizan en colecciones. Además, permite la creación de subcolecciones y estructuras de datos jerárquicas que escalan eficientemente a medida que la base de datos crece.

Con consultas expresivas, eficientes y flexibles, puedes recuperar datos específicos a nivel de documento, agregar criterios de orden, filtros y límites, y usar agentes de escucha en tiempo real para mantener actualizados los datos de tus aplicaciones.

Para proteger el acceso a tus datos, Cloud Firestore se integra perfectamente con Firebase Authentication y las reglas de seguridad de Cloud Firestore para Android, iOS, y JavaScript, así como con la administración de identidades y accesos (IAM) para entornos de servidor.

Ruta de implementación de Cloud Firestore:

Integración de SDK: Añade clientes rápidamente mediante Gradle, CocoaPods o un script.

Protección de datos: Utiliza las reglas de seguridad de Cloud Firestore o Identity and Access Management (IAM) para proteger tus datos en el desarrollo para dispositivos móviles, la Web y servidores.

Gestión de datos: Crea documentos y colecciones en tu base de datos.

Acceso a datos: Crea consultas o utiliza agentes de escucha en tiempo real para recuperar datos de la base de datos.

Ejercicio práctico

Realtime Database

¿Qué es?

Firebase 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 apps 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. (*Firebase Realtime database*, s. f.)

Funciones clave

- Tiempo real
- Sin conexión
- Acceso de dispositivos cliente
- Escalamiento de varias bases de datos.

¿Cómo funciona?

Firebase Realtime Database ofrece una manera potente de desarrollar aplicaciones colaborativas y adaptables. Permite el acceso seguro a la base de datos directamente desde el código del cliente, con persistencia local de datos. Incluso sin conexión, los eventos en tiempo real continúan activándose, proporcionando una experiencia continua para el usuario. Cuando se restablece la conexión, Realtime Database sincroniza automáticamente los cambios locales con las actualizaciones remotas, gestionando conflictos de forma automática.

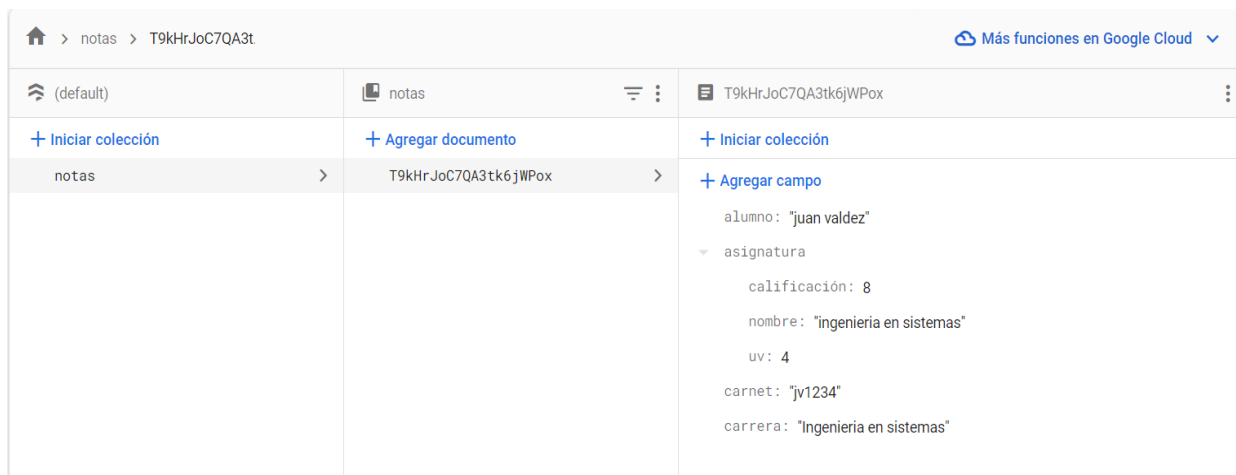
Las reglas de seguridad de Firebase Realtime Database, un lenguaje flexible basado en expresiones, permiten definir la estructura de datos y cuándo se pueden leer o escribir. Al integrar Firebase Authentication, los desarrolladores pueden controlar el acceso a los datos.

Como base de datos NoSQL, Realtime Database presenta optimizaciones y funcionalidades específicas. Su API está diseñada para ejecutar operaciones rápidas, proporcionando una experiencia en tiempo real para millones de usuarios sin sacrificar la capacidad de respuesta. La clave es pensar en cómo acceden los usuarios a los datos y estructurarlos de manera adecuada.

Ruta de implementación

1. Integrar los SDK de Firebase Realtime Database.
2. Crear referencias de Realtime Database.
3. Configurar datos y detectar cambios.
4. Habilitar la persistencia sin conexión.
5. Proteger los datos.

Ejercicio práctico



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

Aspecto	Bases de Datos SQL	Bases de Datos NoSQL
Estructura	Tablas con filas y columnas.	Puede ser basada en documentos, columnares, en grafos o de tipo clave-valor. Flexibilidad en la estructura de datos.
Escalabilidad	Escalabilidad vertical común.	Escalabilidad horizontal más sencilla y eficiente.
Transacciones	Transacciones ACID para integridad.	Algunas pueden sacrificar ACID por rendimiento. Pueden ofrecer consistencia eventual.

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

Modelo de datos y consultas:

Cloud Firestore: Utiliza un modelo de datos más estructurado y jerárquico que permite almacenar datos en documentos organizados en colecciones. Puede manejar consultas más complejas, incluyendo consultas compuestas, ordenación y filtrado de campos.

Realtime Database: Almacena datos en formato JSON en una estructura de árbol simple. Las consultas están limitadas en comparación con Cloud Firestore y solo admiten operaciones de consulta simples como ordenación y filtrado.

Sincronización y actualizaciones en tiempo real:

Cloud Firestore: Proporciona una sincronización eficiente de datos en tiempo real para dispositivos conectados y consultas de recuperación. Es particularmente útil para aplicaciones que requieren una gran cantidad de datos con una estructura compleja.

Realtime Database: Ofrece sincronización en tiempo real para datos en formato JSON con una estructura de árbol simple. Es más adecuado para aplicaciones que necesitan actualizaciones en tiempo real de datos simples y no requieren operaciones de consulta complejas.

Escalabilidad y rendimiento:

Cloud Firestore: Diseñado para escalar de manera más eficiente en aplicaciones que requieren estructuras de datos jerárquicas y consultas complejas. Ofrece una mejor escalabilidad y rendimiento para aplicaciones con grandes volúmenes de datos y consultas complejas.

Realtime Database: Aunque puede manejar una carga considerable, puede tener limitaciones en cuanto a escalabilidad y rendimiento en comparación con Cloud Firestore para aplicaciones con requisitos de consulta más complejos y volúmenes de datos más grandes.

¿Cuál es la mejor opción para implementar en una aplicación desarrollada en React Native?

La elección entre Cloud Firestore y Realtime Database depende de la naturaleza y los requisitos específicos de tu aplicación. Sin embargo, teniendo en cuenta las capacidades y las necesidades típicas de una aplicación desarrollada en React Native, Cloud Firestore suele ser la opción más recomendada. Esto se debe a su capacidad para manejar una estructura de datos más compleja, consultas flexibles y su mejor escalabilidad en comparación con Realtime Database.

Además, considerando que las aplicaciones modernas tienden a tener requisitos más complejos de datos y consultas, Cloud Firestore proporciona un enfoque más estructurado y escalable que se alinea mejor con las demandas de aplicaciones más sofisticadas y de mayor escala.

Conclusiones

Firebase Realtime Database se centra en proporcionar actualizaciones en tiempo real, lo que es ideal para aplicaciones colaborativas y experiencias en tiempo real. En contraste, las bases de datos SQL siguen un enfoque más relacional y estructurado, donde las consultas explícitas son necesarias para obtener los datos más recientes .

Bibliografía

Firebase Realtime database. (s. f.).

Firebase. <https://firebase.google.com/docs/database?hl=es-419>

Cloud Firestore : <https://firebase.google.com/docs/firestore?hl=es-419>