



Institución: Universidad de Colima

Escuela: Facultad de Ingeniería Mecánica y Eléctrica

Carrera: Ingeniería en Computación Inteligente

Materia: Cómputo en la nube

Profesor: Carrillo Zepeda Oswaldo

Alumno: Herrera Escareño Kevin Alejandro

Grado y grupo: 6°B

Nombre de la actividad: Cloud Firestore

Fecha: 28 de Febrero de 2022

Lugar: Colima, Col.



Cloud Firestore

Cloud Firestore es una base de datos flexible y escalable para el desarrollo en servidores, dispositivos móviles y la Web desde Firebase y Google Cloud. Firestore mantiene los datos sincronizados entre aplicaciones cliente a través de objetos de escucha en tiempo real y ofrece soporte sin conexión para dispositivos móviles y la Web, por lo es posible compilar apps con capacidad de respuesta que funcionan sin importar la latencia de la red ni la conectividad a Internet. Cloud Firestore también ofrece una integración sin interrupciones con otros productos de Firebase y Google Cloud, incluido Cloud Functions.



Figura 1. Logotipo de Cloud Firestore.

¿Cómo funciona?

Cloud Firestore es una base de datos NoSQL alojada en la nube a la que pueden acceder aplicaciones de Apple, Android y la Web directamente desde los SDK nativos. Cloud Firestore también está disponible en los SDK nativos de Node.js, Java, Python, Unity, C++ y Go, además de las API de REST y RPC.

A partir del modelo de datos NoSQL de Cloud Firestore, es posible almacenar datos en documentos, los cuales contienen campos que se asignan a valores. Estos documentos se almacenan en colecciones, las cuales son contenedores para documentos que pueden ser utilizadas para organizar datos y compilar consultas.

Los documentos admiten varios tipos de datos diferentes, desde strings y números simples, hasta objetos anidados complejos. También es posible crear subcolecciones dentro de documentos, y con ello crear estructuras de datos jerárquicas que se ajustan a escala, a medida que la base de datos crece. El modelo de datos de Cloud Firestore admite cualquier estructura de datos que funcione mejor con la aplicación desarrollada.

Las consultas de Cloud Firestore son expresivas, eficientes y flexibles. Crear consultas superficiales permite recuperar datos en el nivel del documento, sin la necesidad de recuperar la colección completa ni las subcolecciones anidadas. Además, es posible mantener actualizados los datos sin la necesidad de recuperar toda la base de datos cada vez que haya una actualización, esto mediante la adición de agentes de escucha en tiempo real.



Funciones clave

Función	Descripción
Flexibilidad	El modelo de datos de Cloud Firestore admite estructuras de datos flexibles y jerárquicas. Además, los documentos pueden contener objetos anidados complejos y subcolecciones.
Consultas expresivas	Es posible utilizar las consultas para recuperar documentos individuales específicos o para recuperar todos los documentos de una colección que coinciden con los parámetros de la consulta. Las consultas pueden incluir varios filtros en cadena y combinar los filtros con criterios de orden.
Actualizaciones en tiempo real	Cloud Firestore utiliza la sincronización de datos para actualizar los datos de cualquier dispositivo conectado. Sin embargo, también está diseñado para ejecutar consultas de recuperación únicas y sencillas de manera eficiente.
Asistencia sin conexión	Cloud Firestore almacena en caché datos que usa la aplicación de forma activa, por lo que la app puede escribir, leer, escuchar y consultar datos, aunque el dispositivo se encuentre sin conexión.
Diseñado para ajustarse a escala	Ofrece replicación automática de datos multirregión, garantías de coherencia sólida, operaciones atómicas por lotes y asistencia real sobre transacciones.

Figura 2. Funciones clave de Cloud Firestore.

Características

Cloud Firestore dispone de las siguientes características:

- Es una base de datos NoSQL, la cual almacena datos como colecciones de documentos.
- Cuenta con SDK en tiempo real centrado en dispositivos móviles.
- Admite el almacenamiento de datos locales para las apps que funcionan sin conexión.
- Recupera, ordena y filtra datos de la base de datos mediante consultas.
- Cuenta con operaciones avanzadas de escritura y transacción.
- Es una solución regional y multirregional con ajuste de escala automático.
- El escalamiento es automático.
- Dispone de reglas sin formato de cascada que combinan autorización y validación.



Modelo de datos de Cloud Firestore

Cloud Firestore es una base de datos NoSQL orientada a los documentos. A diferencia de una base de datos SQL, no hay tablas ni filas; En su lugar, almacenas los datos en documentos, que se organizan en colecciones. Cada documento contiene un conjunto de pares clave-valor.

Todos los documentos deben ser almacenados en colecciones, las cuales pueden contener subcolecciones y objetos anidados. Además, ambos pueden incluir campos primitivos, como strings, o tipos de objetos complejos, como listas. Las colecciones y los documentos se crean de manera implícita en Cloud Firestore; solo es necesario asignar datos a un documento dentro de una colección. Si la colección o el documento no existen, Cloud Firestore los crea.

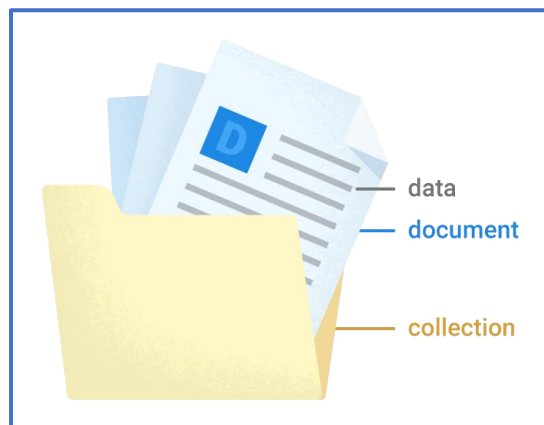


Figura 3. Modelo de datos de Firestore.

Documentos

En Cloud Firestore, la unidad de almacenamiento es el documento. Un documento es un registro que usa pocos recursos y contiene campos con valores asignados. Cada documento se identifica con un nombre.

Un documento que representa a un usuario “alovelace” puede tener el siguiente aspecto:



Figura 4. Ejemplo de un documento.

Los objetos complejos anidados en un documento se llaman mapas. Por ejemplo, es posible estructurar el nombre del usuario del ejemplo anterior con un mapa como este:





Figura 5. Ejemplo de un mapa.

Existen algunas diferencias entre los documentos y los archivos JSON (por ejemplo, los documentos admiten tipos de datos adicionales y su tamaño se limita a 1 MB), pero en general, es posible tratar los documentos como registros JSON livianos.

Colecciones

Los documentos viven en colecciones, que simplemente son contenedores de documentos. Por ejemplo, se podría tener una colección llamada “users” con los distintos usuarios de la app, en la que haya un documento que represente a cada uno:

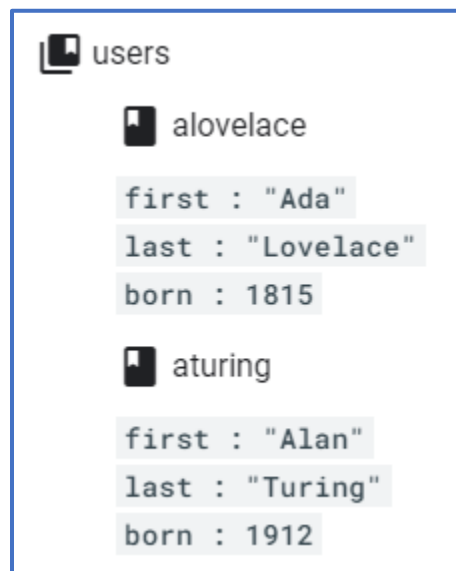


Figura 6. Ejemplo de una colección.

Cloud Firestore no usa esquemas, por lo tanto, se tiene total libertad sobre los campos que se colocan en cada documento y los tipos de datos que se almacenan en esos campos. Los documentos dentro de una misma colección pueden contener campos diferentes o almacenar distintos tipos de datos en esos campos. Sin embargo, se recomienda usar los mismos campos y tipos de datos en varios documentos, de manera que sea posible consultarlos con mayor facilidad.

Una colección contiene solo documentos; no puede contener campos sin procesar con valores de manera directa ni tampoco otras colecciones.



Los nombres de documentos dentro de una colección son únicos. Se pueden proporcionar claves propias, como los ID de usuario, o es posible dejar que Cloud Firestore cree ID aleatorios de forma automática.

No es necesario "crear" ni "borrar" las colecciones; Cuando se crea el primer documento de una colección, esta pasa a existir. Si se borran todos los documentos de una colección, esta deja de existir.

Tipos de datos admitidos

En la siguiente tabla, se enumeran los tipos de datos que admite Cloud Firestore. También se describe el orden de clasificación que se usa para comparar valores del mismo tipo.

Tipo de datos	Orden de clasificación	Notas
Arreglo	Según los valores de los elementos	<p>Un arreglo no puede contener otro valor de arreglo como uno de sus elementos.</p> <p>En los arreglos, los elementos conservan la posición que se les asignó. Para ordenar dos o más arreglos, se debe realizar con base en los valores de sus elementos.</p> <p>Cuando se comparan dos arreglos, se comparan los primeros elementos de cada uno de ellos. Si son iguales, se deben comparar los segundos elementos y así sucesivamente, hasta que se encuentre una diferencia. Si un arreglo no tiene más elementos que comparar, pero es igual al otro arreglo hasta ese punto, debe ordenarse antes del arreglo más largo.</p>
Booleano	False < true	
Bytes	Orden de bytes	Hasta 1,048,487 bytes (de 1 MiB [mebibyte] a 89 bytes). Las consultas solo consideran los primeros 1,500 bytes.
Fecha y hora	Cronológico	Cuando se almacena en Cloud Firestore, la precisión máxima es de microsegundos; cualquier precisión adicional se redondea hacia abajo.
Número de coma flotante	Numérico	Precisión doble de 64 bits, IEEE 754.
Punto geográfico	Por latitud, luego por longitud	No es recomendable usar este tipo de datos debido a las limitaciones de consultas. En general, es mejor almacenar la latitud y longitud como campos numéricos separados.
Número entero	Numérico	64 bits.



Mapa	Por claves y, luego, por valor	<p>Cuando se indexa, es posible hacer consultas por subcampos. Si se excluye este valor de la indexación, también se excluyen todos los subcampos.</p> <p>Los campos de mapas se ordenan por clave y se comparan con pares clave-valor. Primero se comparan las claves y, luego, los valores. Si los primeros pares clave-valor son iguales, se compararán los siguientes, y así sucesivamente. Si dos mapas comienzan con los mismos pares clave-valor, se considerará la longitud del mapa.</p>
Nulo	Ninguno	-
Referencia	Por elementos de ruta de acceso (colección, ID de documento, colección, ID de documento...)	<p>Por ejemplo:</p> <pre>projects/[PROJECT_ID]/databases/[DATABASE_ID]/documents/[DOCUMENT_PATH].</pre>
String de texto	Orden de bytes con codificación UTF-8	Hasta 1,048,487 bytes (de 1 MiB a 89 bytes). Las consultas solo consideran los primeros 1,500 bytes de la representación UTF-8.

Figura 7. Tipos de datos.

Ventajas

Cloud Firestore cuenta con las siguientes ventajas:

- Los datos simples son fáciles de almacenar en documentos, ya que son muy similares a JSON.
- Los datos complejos y jerárquicos son más fáciles de organizar a escala, con subcolecciones dentro de los documentos.
- Necesita menos desnormalización y compactación de datos.
- Soporte sin conexión para clientes de Apple, Android y la Web.
- Es posible encadenar filtros y combinar filtrado con ordenamiento según una propiedad en la misma consulta.
- Las consultas son superficiales: solo muestran documentos de colecciones o grupos de colecciones específicos y no muestran datos de subcolecciones.
- Las consultas se indexan de forma predeterminada: el rendimiento de las consultas es proporcional al tamaño del conjunto de resultados, no del conjunto de datos.
- Aloja los datos en varios centros de datos de distintas regiones, lo que garantiza una escalabilidad global y una confiabilidad sólida.
- Está disponible en configuraciones regionales o multirregionales en todo el mundo.
- Escala de forma completamente automática.
- Las reglas de seguridad de Cloud Firestore protegen las operaciones de lectura y escritura desde los SDK para dispositivos móviles.
- Las reglas pueden restringir consultas.



Desventajas

- Es un servicio de pago.
- La versión gratuita tiene un límite de conexiones simultáneas.
- En la versión gratuita algunas funcionalidades no están disponibles.
- En la versión gratuita se tienen pruebas limitadas en la nube (15 dispositivos por día).

Bibliografía

Google. (2021, 18 noviembre). Cloud Firestore | Firebase Documentation. Firebase. Recuperado 28 de febrero de 2022, de <https://firebase.google.com/docs/firestore?hl=es-419>.

Google. (2021, septiembre 1). Tipos de datos admitidos | Firebase Documentation. Firebase. Recuperado 28 de febrero de 2022, de <https://firebase.google.com/docs/firestore/manage-data/data-types?hl=es-419>.

Google. (2021, noviembre 10). Elige una base de datos: Cloud Firestore o Realtime Database | Firebase Documentation. Firebase. Recuperado 28 de febrero de 2022, de <https://firebase.google.com/docs/firestore/rtdb-vs-firestore?hl=es-419>.

