

# UNIVERSIDAD DON BOSCO



## FORO I

Docente: Alexander Alberto Sigüenza Campos

Integrantes:

Avalos Avalos, Josué Ezequiel: AA200509

Flores Ramírez, Carlo Heriberto FR230818

Vargas Villatoro, Bryan Eduardo VV110806

Diseño y Programación de Software Multiplataforma DPS941 G01T

San Salvador 28 de abril de 2024

## Contenido

Introducción.....	2
¿Qué es Cloud Firestore? .....	2
¿Qué es Realtime Database? .....	3
¿Cuáles son las diferencias entre Cloud Firestore y Realtime Database? .....	4
¿Cuáles son las diferencias fundamentales entre las bases de datos SQL y NoSQL? .....	5
¿Cuál de estas bases de datos consideran que sería la mejor opción para implementar en una aplicación desarrollada en ReactNative?.....	6
Conclusiones de investigación .....	9
Conclusiones de implementación .....	9

## Introducción

Conoceremos sobre las dos bases de datos NoSQL que nos ofrece FireBase las cuales tenemos Cloud Firestore y Realtime Database, donde ambas nos ofrecen almacenamiento en tiempo y real y soporte sin conexión a internet tanto para dispositivos móviles y la web, daremos a conocer la diferencia que presentan entre sí, así como sus diferencias fundamentales entre las bases de datos SQL y NoSQL, para luego poder poner el conocimiento adquirido en la práctica sin ninguna temor a elegir cuál de las dos serian la mejor opción.

## ¿Qué es Cloud Firestore?

Cloud Firestore es una base de datos de documentos NoSQL caracterizada por su flexibilidad y escalabilidad en la nube, que nos permite almacenar y sincronizar los datos necesarios para el desarrollo del cliente y el servidor tanto en dispositivos móviles como en la web.

A partir del modelo de datos NoSQL de Cloud Firestore, almacena la información en documentos que contienen campos que se asignan a valores. Estos documentos se almacenan en colecciones que son contenedores para que los documentos se pueden usar para organizar los datos y compilar consultas. Los documentos admiten diferentes tipos de datos desde cadenas y números simples, hasta objetos animados complejos.

Esta base de datos tiene la opción de seleccionar un modo nativo o un modo llamado Datastore, este modo que no cambia la ubicación de la base de dato y usa la misma estructura que la modalidad nativa.

Dentro de las características importantes que destaca en su modo nativo podemos encontrar:

- Que esta base de dato gestionada completamente por Google no necesita utilizar de un servidor externo, lo que facilita escalar los recursos para ajustarse a alguna demanda que realice el cliente.
- Cuenta con Asistencia sin conexión esta característica se encarga de guardar los datos que la aplicación utiliza de manera frecuente en el caché, esto permite que se pueda realizar operaciones de escritura incluso de realizar consulta de datos, incluso cuando el dispositivo se encuentre sin conexión.
- Cuenta con Actualización y sincronización en tiempo real esto contribuye en el desarrollo de diferentes tipos de aplicaciones, especialmente la de tipo multiusuario o colaborativa en la web para dispositivos móviles.
- Replicación multirregional esto permite que los datos estén seguros y cuenten con una garantía de disponibilidad de hasta el 99.99%, incluso en situaciones problemáticas o cuando se presenten desastres.

Firestore también cuenta con la capacidad de integrarse con la Firebase Authentication e Identity Platform con el objetivo de habilitar controles de acceso de seguridad que tenga como base la identidad personalizable y la validación de datos a través del uso de un lenguaje de configuración.

## ¿Qué es Realtime Database?

Realtime Database es una base de datos NoSQL alojada en la nube. Sincroniza todos los datos con el usuario en tiempo real y se mantiene disponible aun cuando la aplicación se encuentra sin conexión. Permite al usuario tener acceso desde un dispositivo móvil o web.

Los datos los almacena en formato JSON y al actualizarlos se almacenan en la nube, y La plataforma envía una notificación a todos los dispositivos conectados con la aplicación que usamos en segundos.

Otras de las ventajas con la que cuenta es que, si el usuario pierde la conexión a internet, el SDK de la base de datos hará uso de la cache local en el dispositivo que se esté usando en ese momento, este con el fin de almacenar y procesar la información en tiempo real, así cuando el usuario recupere la conexión la información pueda ser almacenada en la nube.

Realtime Database cuenta con una norma llamada reglas de seguridad de Firebase Realtime Database, con la cual podemos decidir a quien le brindamos el acceso a los datos y la manera en que ingresara a ellos.

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

	Realtime Database	Firestore
Desarrollada por	Firebase	Google
Estructura de datos	Almacena sus datos en formato JSON, pero está organizado a manera de árbol jerárquico, lo que la vuelven un poco compleja conforma van creciendo las anidaciones.	De igual manera, se almacena en formato JSON, muestra una estructura visible mucho más sencilla de escalar, sobre todo en jerarquías grandes.

<b>Soporte Offline</b>	Brinda soporte offline para Apple y Android.	Además de brindar soporte offline para Apple y Android, también se incluye para la web.
<b>Consultas</b>	Sus consultas solo pueden incluir un filtro y estas devuelven todo el subárbol.	Se pueden aplicar múltiples filtros y son indexadas por defecto.
<b>Rendimiento y confiabilidad</b>	Lleva varios años en el mercado y es utilizada por muchas empresas, lo que nos brinda el respaldo de su fiabilidad y baja latencia.	Lleva poco tiempo en el mercado, por lo cual aún puede estar sujeta a errores, sin embargo, sus datos son almacenados en múltiples datacenters y distintas regiones lo que nos da a su vez escalabilidad y confiabilidad.
<b>Precios</b>	Se cobra según el espacio de almacenamiento utilizado y ancho de banda.	Posee dos modelos de cobro, según operaciones realizadas y según espacio de almacenamiento y ancho de banda. Pero es más económico que Realtime Database.

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

SQL	NoSQL
Almacenan los datos de manera estructurada en tablas relacionadas.	Almacena los datos en el formato original.
Su escalabilidad es baja, ya que, al poseer una estructura preestablecida, resulta complicado aplicar cambios radicales.	Alta capacidad de escalabilidad, al estar pensadas para trabajar con grandes volúmenes de datos, pueden estar distribuidas en diferentes equipos, consumiendo pocos recursos.
Su adaptación a los cambios es casi inexistente.	Se adaptan fácilmente al ser totalmente flexibles.
Las bases de datos SQL nos brindan consistencia en los datos almacenados.	No poseen consistencia de datos, ya que su principal objetivo es la rapidez.
En volúmenes bajos de datos, la estructura y organización de las SQL nos brindan rapidez al realizar consultas y búsquedas complejas, sin embargo, con forme esta se	Al funcionar con Big Data no estructurado, es mucho más rápido para realizar las búsquedas.

va llenando de Data, se vuelve más lenta al tener que leer grandes cantidades.	
--	--

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

Para recomendar una base de datos en específico se necesitan aspectos y objetivos específicos que tendrá la app que se desea desarrollar, pero en términos generales y en la mayoría de los escenarios se recomienda el uso de Cloud Firestore, debido a las siguientes características:

- Flexibilidad de documentos: Esto significa que se adapta bien a datos no estructurados y en constante cambio, lo cual es muy común en aplicaciones móviles.
- Sincronización en tiempo real: Todas las actualizaciones o creación de data nueva se verá reflejada en tiempo real en todos los dispositivos que estén conectados, esto es ideal para experiencias colaborativas y claro actualizaciones instantáneas.
- Escalabilidad robusta: Una de las grandes ventajas y por lo cual en los casos generales se opta por esta opción es porque gestiona automáticamente el crecimiento de la base de datos sin necesidad de intervención manual.
- Integración con Firebase: Nos brinda soporte para aprovechar otros servicios de Firebase como autenticación, funciones y almacenamiento.

## Evidencias base de datos NOSQL con Firestore

### Colecciones de la BD NOSQL



Colección de alumnos con sus documentos y respectivos campos, además de una subcolección para las notas de la materia o materias que este cursando.

(default)	Alumnos	AL200300
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
Alumnos >	AL200300 >	Notas
Evaluaciones	FA200301	+ Agregar campo Apellidos: "Turing" Email: "alan@gmail.com" Nombre: "Alan"
Materias	GA200302	
Profesores	KL200304 PA200303	

Subcolección de notas.

AL200300	Notas	RE1
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
Notas >	RE1 >	+ Agregar campo
+ Agregar campo Apellidos: "Turing" Email: "alan@gmail.com" Nombre: "Alan"		MateriaID: /Materias/DWS901
		Nota1: 10 Nota2: 10 Nota3: 10 NotaFinal: 10

Colección de evaluaciones.

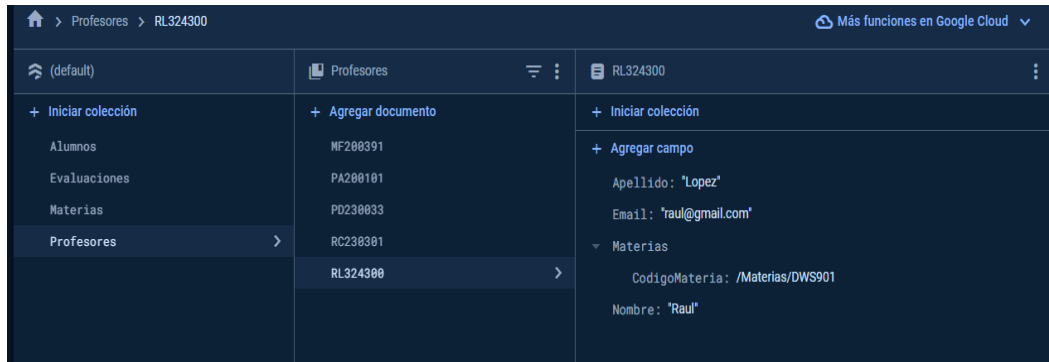
(default)	Evaluaciones	EVAL1
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
Alumnos	EVAL1 >	+ Agregar campo
Evaluaciones >	EVAL2	Fecha: 25 de abril de 2024, 10:30:00 a.m. UTC-6
Materias	EVAL3	Materia: /Materias/DWS901
Profesores	EVAL4	Nombre: "Parcial 1"
	EVAL5	Porcentaje: 0.3

Colección de materias.

(default)	Materias	DWS901
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
Alumnos	ABD950	+ Agregar campo
Evaluaciones	DPS941	▼ Evaluaciones
Materias >	DWS901 >	CodigoEval: /Evaluaciones/EVAL1
Profesores	P00900 SPP930	Nombre: "Desarrollo de Aplicaciones Web con Software Interpretado en el Servidor"
		ProfesorID: /Profesores/RL324300



### Colección de profesores.



## Conclusiones de investigación

Respecto a la investigación Firestore es la opción general para la mayoría de los casos a la hora de elegir una base de datos NOSQL, pero en casos específicos es mejor utilizar Firebase, ejemplo de ello puede ser una aplicación de mensajería.

SQL es una base de datos más rígida con capacidad de brindarnos la posibilidad de consultas muy complejas, son muy fiables respecto a salvaguardar la integridad de los datos por ellos se utilizan en sistemas financieros ya tiene consistencia ACID.

Por otro lado, tenemos NOSQL que nos da flexibilidad ya que almacena datos no estructurados, ideal para grandes volúmenes de datos, fiable a la hora de concurrencia masiva y con sincronización en tiempo real, lo mejor para aplicaciones colaborativas.

## Conclusiones de implementación

Respecto a la implementación, podemos determinar qué, las bases de dato NoSQL son fáciles de implementar, pero debemos saber estructurarlas para optimizar su funcionamiento, ya que el árbol jerárquico debe llevar un orden específico para la indexación de la data.

Por otro lado, debemos conocer el objetivo final de nuestro proyecto, para de esta manera, determinar cuál es la mejor opción, en cuanto al tipo de base de datos (SQL o NoSQL) que debemos implementar. Si se pretende manejar grandes volúmenes de datos (Big Data), lo más conveniente es utilizar NoSQL, por otro lado, si se proyecta que no sea tan grande la cantidad de datos, es más conveniente usar la base de datos SQL, ya que su estructura es más funcional para indexar las consultas anidadas.