

# Base de datos en entornos móviles

---

## **Integrantes:**

- María Dolores Martínez León ML202278
- Omar Alexander Rivas Serrano RS060867
- Diego Alberto Vásquez Arias VA172033
- José Luis Bardales Paniagua BP171968

## **Asignatura:**

Diseño y Programación de Software Multiplataforma

## **Docente:**

Ing. Alexander Sigüenza

## **Fecha de entrega:**

30 de abril de 2023

## Firestore



Firebase es una plataforma digital que esta diseñada para optimizar el desarrollo de las aplicaciones móviles y web de alta calidad, de una manera eficiente y rápida, esto con el objetivo de escalar el rendimiento de dichas aplicaciones para la implementación de los módulos que la conformen, y así la aplicación sea mucho mas segura, administrable y fácil de usar para los usuarios.

Esta plataforma esta alojada en la nube y es desarrollada actualmente por Google, disponible para Android, iOS y sitio web. Cuenta con funcionalidades que se adaptan a las necesidades y en su mayoría son gratuitas.

Una vez teniendo la información básica sobre Firebase y lo que esta ofrece, la investigación se enfocara en el uso de bases de datos en entornos móviles NoSQL las cuales serán: **Cloud Firestore** y **Realtime Database**.

## Cloud Firestore

En una aplicación puede ser necesario que se requiera almacenar información que se gestiona dentro de ella en la nube.



Pero uno de los inconvenientes puede ser todo el proceso de instalación y mantenimiento de servidores, también gestionar los problemas en la red y que la infraestructura sea escalable según la cantidad de usuarios que utilicen la aplicación.

Por lo que una solución para ese problema puede ser Cloud Firestore, que es una base de datos NoSQL de Firebase, que ofrece flexibilidad, escalabilidad y esta alojada en la nube, con el objetivo que se puedan almacenar y sincronizar información en tiempo real para el desarrollo en los servidores y dispositivos que se puedan conectar hacia ella. Los datos se estructuran como documentos que están organizados en colecciones.

Cloud Firestore está disponible para los SDK nativos de Android, también disponible para iOS, Web, Node.js, Java y también en las API de Rest y RPC.

### Ventajas

1. Asistencia sin conexión, lo que permite a la aplicación que los datos que utiliza puedan leer, escribir y realizar consultas de información sin necesidad que el dispositivo se encuentre conectado a internet, y cuando se recupera la conexión se sincronizan los datos locales a la nube.
2. Amplias normas de seguridad.
3. Maneja cargas de trabajo que pueden ser exigentes, según la cantidad de usuarios, pero que se escala automáticamente para soportar la aplicación.

## Realtime Database

Es la base de datos original de Firebase. Siendo una solución bastante eficiente y de muy baja latencia.



Realtime Database es una base de datos NoSQL que de igual manera esta alojada en la nube. Los datos que se almacenan están en formato JSON los cuales se sincronizan en tiempo real con cada usuario que se encuentra conectado y también se mantienen disponibles en el momento que la aplicación no tiene conexión a internet.

Este modulo de Firebase esta disponible para iOS, Android, Web, Flutter y Unity.

### Ventajas

1. La sincronización de los datos JSON es en tiempo real, lo que permite almacenar y sincronizar información entre los usuarios inmediatamente.
2. Con la sincronización en tiempo real, permite a los usuarios acceder a los datos desde cualquier dispositivo con facilidad y trabajar simultáneamente.
3. Permite crear aplicaciones sin la necesidad de utilizar servidores, de igual manera se puede ejecutar código de backend que responda a los eventos que activa la base de datos.
4. Seguridad basada en usuarios, en el cual se pueden permitir el acceso según la identidad de los usuarios.

## **Diferencias de base de datos SQL y NoSQL**

### **Definición:**

#### **SQL**

La base de datos SQL funciona con el **lenguaje estructurado** de consulta que permite tener un gran alcance sobre la información almacenada, siendo un sistema de gestión de bases de datos que es relacional además de multihilo y multiusuario.

Los datos que se almacenan son en tablas relacionadas, y cada una de ellas se componen por filas que contienen campos de información en las que se insertan datos, por lo que es necesario que se defina el esquema de estas tablas con un análisis previamente hecho para su correcto funcionamiento.

#### **NoSQL**

En este tipo de base de datos, se plantean modelos de datos que son específicos y flexibles que sean adaptables a los requisitos de las nuevas aplicaciones. Esto permite que tengan alto rendimiento y baja exigencia, porque se pueden ejecutar en equipos con recursos mínimos.

Permite almacenar distintos tipos de información no estructurada, como documentos (PDF, Word), emails, audios y también en distintos formatos, como JSON, por ejemplo.

### **Principales diferencias:**

- En las bases de datos SQL los datos se almacenan de manera estructurada en tablas, y las NoSQL en su formato original.
- SQL proporcionan una capacidad para escalar baja, caso contrario las NoSQL, ya que están pensadas para grandes volúmenes de información que puede considerarse Big Data.
- Las SQL son muy poco adaptable a los cambios una vez realizado el análisis y que este en funcionamiento, en cambio las NoSQL son muy flexibles.

- La rapidez dependerá en la forma de su uso, en datos estructurados las SQL serán mucho más rápidas, pero en grandes cantidades de información donde no es estructurado, las NoSQL tienen más rapidez.

## Diferencias entre Cloud Firestore y Realtime Database

- a) En cuanto a la función de la base de datos, con Cloud Firestore se pueden realizar búsquedas, operaciones de insertar, actualizar y eliminar datos al mismo tiempo y ordenamientos de los datos avanzados, mientras que en Realtime Database se sincronizan datos con consultas principalmente básicas. Esto debido que soporta escritura de manera individual.
- b) En el uso de la base de datos, Cloud Firestore se maneja gran cantidad de datos (desde cientos de GB hasta TB) que se leen con mayor frecuencia de cambios realizados, y en Realtime Database la cantidad de datos son menor, pero cambian con mayor frecuencia.
- c) Realtime Database permite el almacenamiento offline para Android y iOS, mientras que Cloud Firestore da soporte offline para móviles y web.
- d) En la estructura de datos almacenados, Realtime Database son en formato JSON organizados en una jerarquía de árbol, siendo una forma compleja cuando crecen las uniones y la información, en Cloud Firestore se tiene una estructura mas visible en la consola siendo sencilla y fácil de escalar cuando se maneja con colecciones.
- e) Realtime Database esta limitado a una sola región que ofrece gran fiabilidad y baja latencia, pero Cloud Firestore almacena los datos en muchos datacenters y en varias regiones lo cual brinda mayor confiabilidad.
- f) En cuestión de precios, con Realtime Database el cobro se realiza por tamaño de almacenamiento y ancho de banda siendo mas costoso que Cloud Firestore, ya que el modelo principal de cobro este ultimo es por operaciones que se realizan en la base de datos y el secundario es por tamaño de almacenamiento y ancho de banda.

- g) Para la seguridad en Cloud Firestore, realizar las reglas es más fácil y robusto, y en Realtime Database las reglas son en cascada además que la validación de datos hay que definirlos mientras en que Cloud Firestore es automático.



## **Mejor opción para implementar React Native**

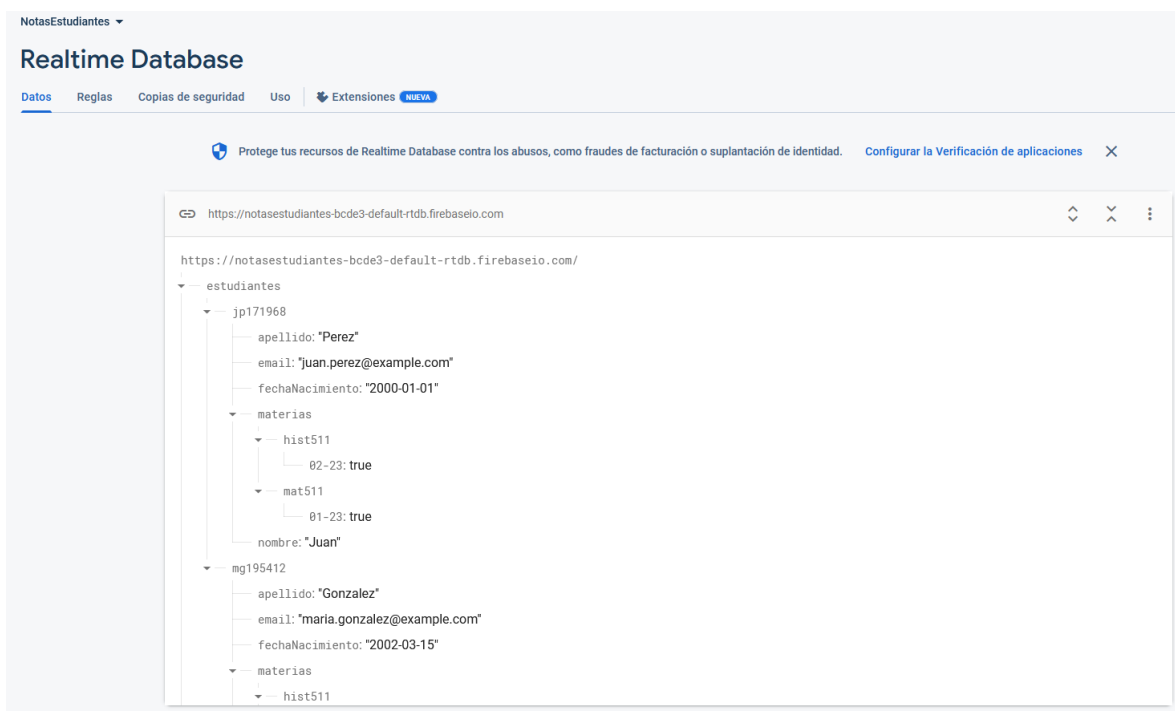
La mejor opción para implementar con React Native dependerá mucho mas de las necesidades que se requiera según la aplicación a desarrollar, ya que, aunque ambas bases de datos ofrecen funciones en común. Los puntos para considerar son justamente las diferencias listadas en el desarrollo de la investigación que va desde el uso de la base de datos, las operaciones que se vayan a realizar y con que frecuencia se realizaran, la estructura de los datos que se van a almacenar según la complejidad de la aplicación, la disponibilidad de la información que es muy importante si la información es crítica, que los dispositivos tengan siempre conexión o sea limitada y la cantidad de bases de datos que se vayan a crear.

Realmente ambas opciones son muy importantes a considerar por lo que, en el análisis de toda esta información, no existe una mejor opción para implementar, sino mas bien implementar la base de datos que mejor se adapte a las necesidades de la aplicación.

## Base de datos para almacenar las notas de los alumnos becarios de UDB Virtual

Al realizar un análisis entre cada una de las ventajas y diferencias entre las 2 bases de datos estudiadas, se elegirá Realtime Database, ya que siguiendo la necesidad se requiere únicamente almacenar las notas de los alumnos becarios, lo que significa que no se requiere realizar consultas ni transacciones de manera avanzada como ordenamiento o filtros, los datos serán simples porque no serán documentos ni datos complejo, la disponibilidad no es crítica, de igual manera la conexión de los usuarios no será constantemente en línea y la cantidad de datos no será tan grande y no cambiarán con frecuencia como si se tratara de los alumnos que no son becarios que representan mayor porcentaje para la universidad.

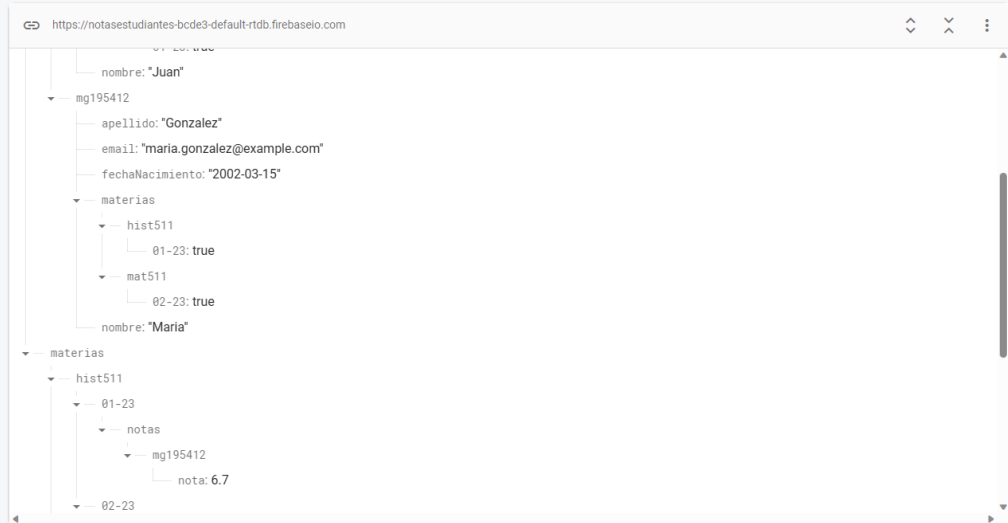
### Base de datos NoSQL en Realtime Database:



## Realtime Database

[Datos](#) [Reglas](#) [Copias de seguridad](#) [Uso](#) | [Extensiones](#) **NUEVA**

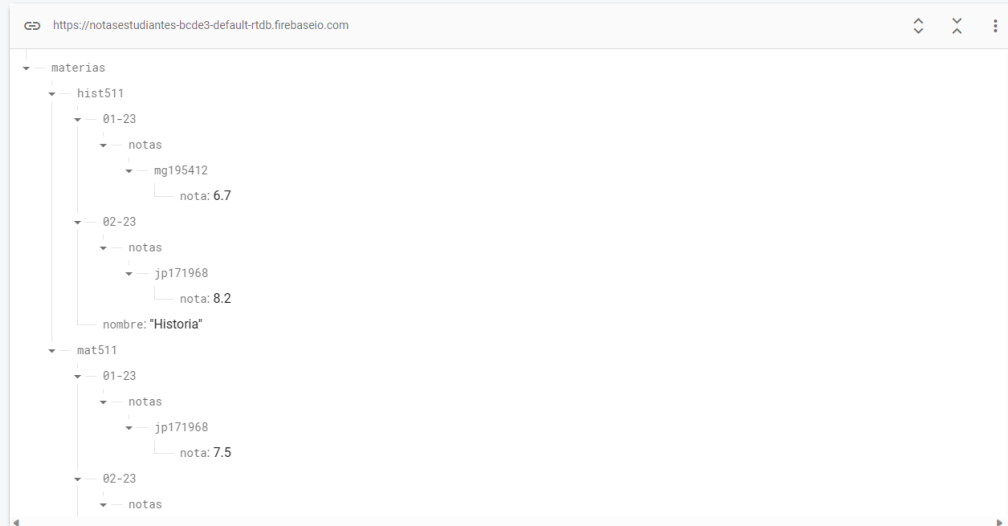
Protege tus recursos de Realtime Database contra los abusos, como fraudes de facturación o suplantación de identidad.

[Configurar la Verificación de aplicaciones](#)

## Realtime Database

[Datos](#) [Reglas](#) [Copias de seguridad](#) [Uso](#) | [Extensiones](#) **NUEVA**

Protege tus recursos de Realtime Database contra los abusos, como fraudes de facturación o suplantación de identidad.

[Configurar la Verificación de aplicaciones](#)

## Realtime Database

[Datos](#)[Reglas](#)[Copias de seguridad](#)[Uso](#)[Extensiones](#)[NUEVA](#)

Protege tus recursos de Realtime Database contra los abusos, como fraudes de facturación o suplantación de identidad.

[Configurar la Verificación de aplicaciones](#)<https://notasestudiantes-bcd3-default-rtdb.firebaseio.com>

```
└─ notas
  └─ mg195412
    └─ nota: 6.7
└─ 02-23
  └─ notas
    └─ jp171968
      └─ nota: 8.2
  └─ nombre: "Historia"
└─ mat511
  └─ 01-23
    └─ notas
      └─ jp171968
        └─ nota: 7.5
  └─ 02-23
    └─ notas
      └─ mg195412
        └─ nota: 9
    └─ nombre: "Matematicas"
```

## Conclusiones

- En las bases de datos SQL se evita la duplicidad de registros, así como la repetición de las columnas dentro de una misma tabla, caso contrario en la NoSQL en las que incluso se duplican datos para realizar las relaciones bidireccionales, pero a su vez esto permite obtener la información más rápido, ya que se puede consultar las materias cursadas y las notas de un estudiante eficazmente, aunque existan muchos registros en la base de datos.
- Se puede invertir la enumeración de los ID comúnmente conocidas, a ser las claves y que su valor de asignación sea true, lo que permite una búsqueda de clave más sencillo.
- En el ejercicio realizado las materias están indexados con los estudiantes según hayan sido inscritos en el ciclo correspondiente, por lo que, si se requiere borrar a un estudiante del registro de una materia, es necesario actualizar tanto en el estudiante como en el nodo notas de cada materia el cual estuvo asociado. Es una de las desventajas de las relaciones bidireccionales a pesar de que brinden mayor rapidez en la búsqueda de información.
- Tomando en cuenta el punto anterior, las SQL son mas fiables ya que si existe un error en un proceso, la operación no se realiza, en cambio en NoSQL la operación se puede realizar sin el resultado correcto esperado, lo que generaría información no confiable.
- Mientras menos anidada este la información en la base de datos NoSQL se puede obtener de manera separada los nodos por lo que tener los datos desnormalizados para no descargar todo el árbol y lo que contenga.
- En cuanto a la relación de escalabilidad y costos, SQL es escalable pero su costo económico es mas elevado, en cambio NoSQL es escalable y se puede aumentar su capacidad fácilmente además que ofrece un mejor rendimiento porque necesita menos recursos de hardware.