

FORO 1

BRANDON DENILSON AZAHAR ROJAS



ASIGNATURA
DPS

Indice ;

Tabla de contenido

Introduccion :	3
2. ¿Cuál es la definición de Cloud Firestore?	4
Características clave:	4
Consultas y Flexibilidad de Búsqueda:	5
Las principales diferencias entre las bases de datos SQL y NoSQL:	5
Consistencia y Transacciones:	5
Conclusión:	6
Capturas:	6
.....	7

Introduccion :

Muchas aplicaciones modernas requieren un manejo eficiente de datos en tiempo real, y Firebase de Google ofrece dos soluciones principales para este problema: Cloud Firestore y Database Realtime. Ambos servicios tienen características, ventajas y usos distintos. Este trabajo examinará en detalle Cloud Firestore y Realtime Database, discutiendo sus diferencias fundamentales, comparando bases de datos SQL y NoSQL y decidiendo cuál sería la mejor opción para implementar en una aplicación desarrollada en React Native.

2. ¿Cuál es la definición de Cloud Firestore?

Definición y funcionamiento: Cloud Firestore es una base de datos NoSQL en tiempo real que almacena datos en documentos JSON dentro de colecciones. Permite consultas avanzadas y utiliza una estructura de datos flexible.

Características destacadas:

Flexible estructura de datos

Preguntas Avanzadas

Autoescalabilidad

Sync en tiempo real

Sistema de seguridad integrado

¿Qué es una base de datos en tiempo real?

Definición y aplicación:

Los datos se sincronizan instantáneamente entre los clientes y los servidores a través de una base de datos en tiempo real. sincroniza y almacena los datos en una estructura JSON en tiempo real.

Características clave:

Formato de datos JSON en tiempo real sincronizado

Escalabilidad reducida

Las reglas de seguridad que se pueden personalizar

Las diferencias entre Cloud Firestore y la base de datos en tiempo real:

Estructura de datos y modelado: Cloud Firestore ofrece una estructura de datos más adaptable que admite tipos de datos complejos y objetos anidados. La base de datos realtime utiliza una estructura JSON plana.

Escalabilidad y rendimiento: Cloud Firestore proporciona escalabilidad automática y mejora la gestión de cargas de trabajo cambiantes. La escalabilidad de Realtime Database es limitada, especialmente para aplicaciones que requieren mucha escritura.

Consultas y Flexibilidad de Búsqueda:

Cloud Firestore ofrece consultas de búsqueda avanzadas y flexibilidad. Las capacidades de consulta de la base de datos en tiempo real son más limitadas.

Sincronización en tiempo real y Seguridad: Aunque ambos servicios ofrecen sincronización en tiempo real, Cloud Firestore ofrece una seguridad más avanzada a través de reglas de seguridad más avanzadas.

Las principales diferencias entre las bases de datos SQL y NoSQL:

Modelado de Datos y Flexibilidad:

El modelo de datos de las bases de datos SQL es estructurado y rígido, mientras que las bases de datos NoSQL ofrecen una mayor flexibilidad.

Escalabilidad y Rendimiento: Las bases de datos NoSQL generalmente son más escalables y funcionan mejor en entornos distribuidos que las bases de datos SQL.

Consistencia y Transacciones: Las bases de datos SQL tienen una mayor consistencia y soportan transacciones ACID, mientras que las bases de datos NoSQL pueden tener modelos de consistencia más antiguos y no soportan transacciones ACID.

¿Cuál es el método más efectivo para aplicar en una aplicación hecha con React Native?
Exigencias y casos de uso:

La elección entre Realtime Database y Cloud Firestore dependerá de las necesidades y casos de uso de cada aplicación.

Consideraciones relacionadas con el desarrollo y el mantenimiento: La estructura de

datos flexible de Cloud Firestore y la capacidad de consulta avanzada facilitan la expansión y el mantenimiento.

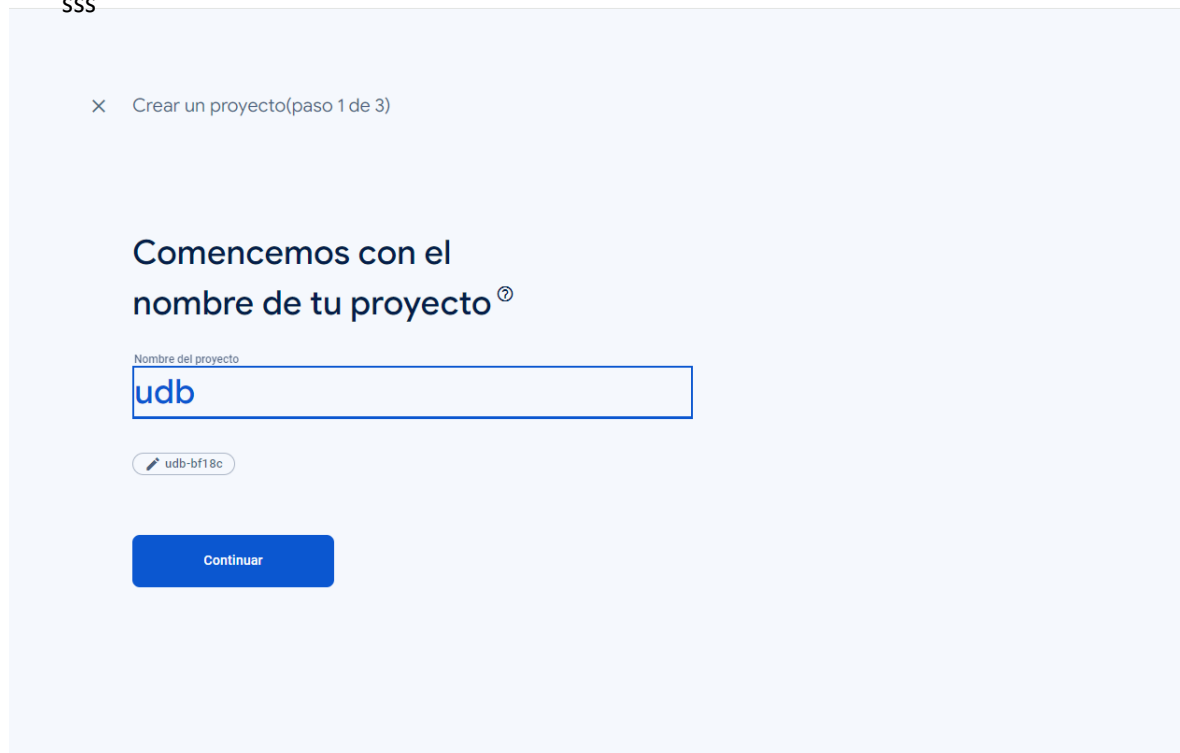
Conclusión:

Dependiendo de la necesidad se puede usar uno u otro lo cierto es que el servicio de Google tanto de firebase realtime database como el fire estore ofrece una salida para los programadores facilitando mucho el Back end

A diferencia de una base de datos relacional como SQL que requiere mas trabajo pero a la vez es mas solida y a parte de soportar mayor volumen de información también se pueden hacer varias cosas como “gatillos” “procedimientos almacenado” “vistas” entre otros

Capturas:

SSS



× Crear un proyecto(paso 1 de 3)


Comencemos con el
nombre de tu proyecto [?]


Nombre del proyecto

udb


udb-bf18c


Continuar


 Firebase


Descripción gene... 


Categorías de producto

Compilación 


Lanzamiento y supervisión 

Analytics 

Participación 






Todos los productos 

Spark
Sin costo \$0 por mes Actualizar

udb 

Plan Spark

Comienza por agregar
Firebase a tu app




Agrega una app para comenzar

Gemini en Firebase


Obtén respuestas a preguntas sobre productos, casos de uso y funciones de Firebase. Genera código para el desarrollo y acorta el proceso de solución de problemas con estadísticas nuevas gracias a una interfaz de chat de lenguaje natural directamente en Firebase console.


Probar sin costo





Almacena y sincroniza datos de app en milisegundos

Crashlytics

 Segmentación de usuarios y orientación a ellos en los productos de Firebase

 Activadores de Cloud Functions basados en eventos


 Informes ilimitados y gratuitos


 **Habilitar Google Analytics para este proyecto**
Recomendado

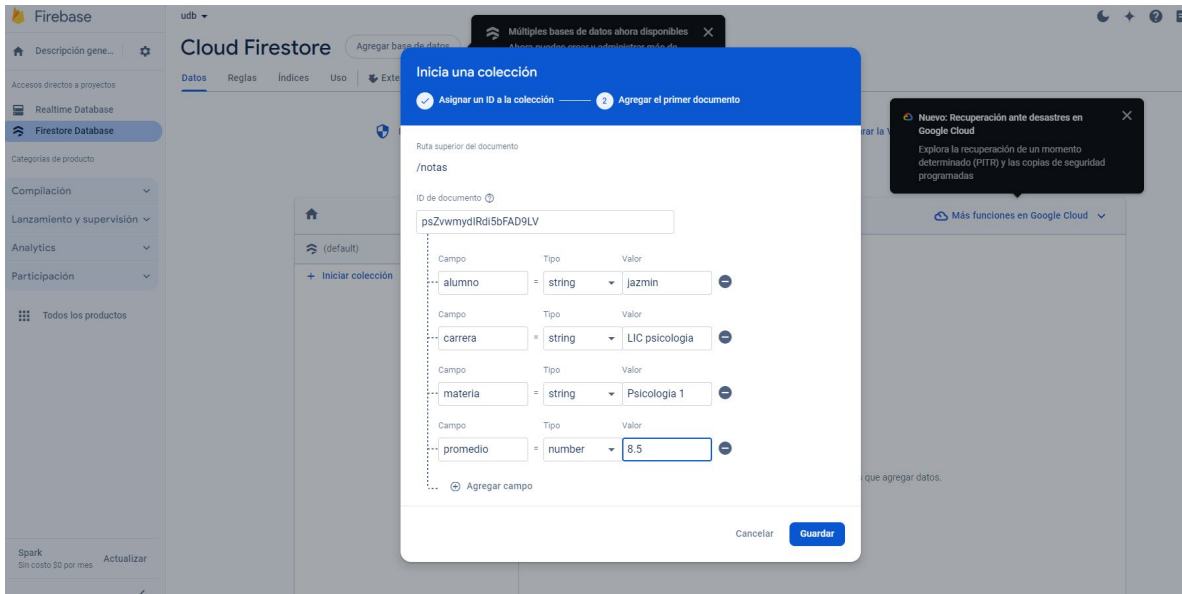
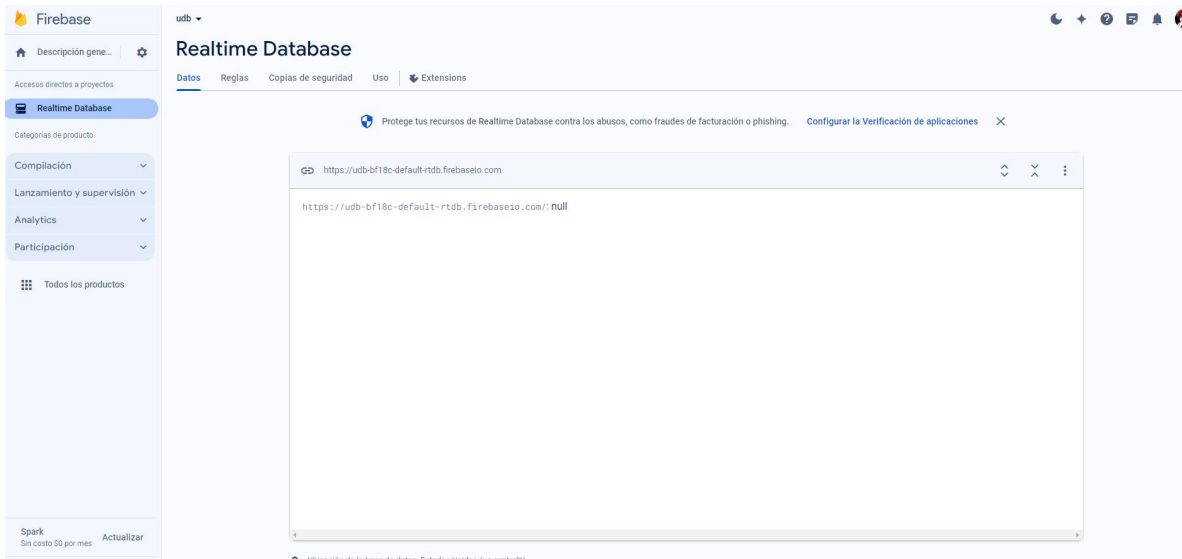
Realtime Database

Almacena y sincroniza datos en tiempo real

Crear una base de datos



 ¿Realtime Database es adecuado para ti? [Comparar bases de datos](#)



🏠

>

notas

>

psZvwmydIRdI5..

Más funciones en Google Cloud

(default)

📁 notas

☰

⋮

psZvwmydIRdISbFAD9LV

⋮

+ Iniciar colección

notas

>

+ Agregar documento

psZvwmydIRdISbFAD9LV

>

+ Iniciar colección

+ Agregar campo

alumno: "Jazmin"

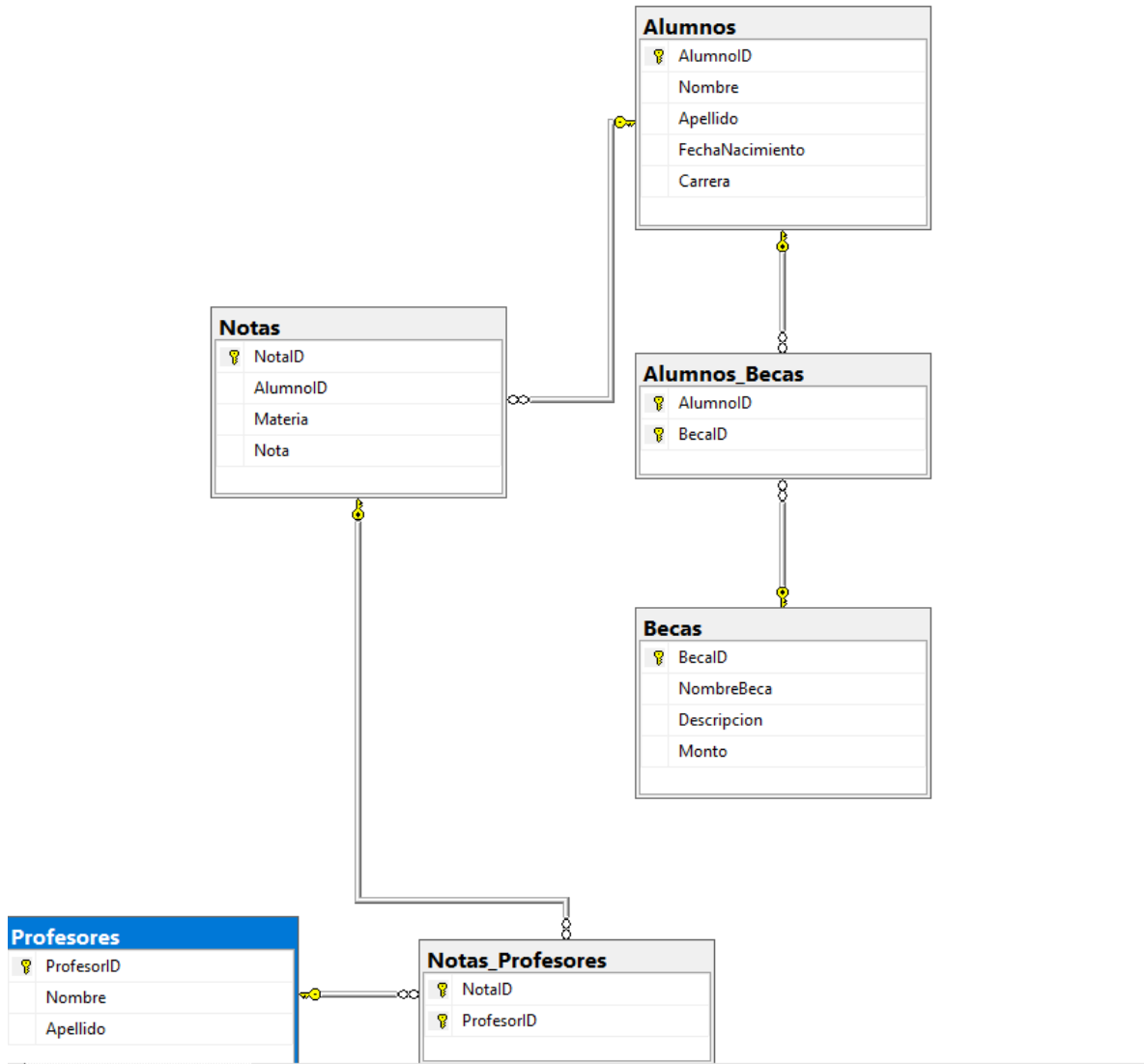
carrera: "LIC psicologia"

materia: "Psicologia 1"

promedio: 8.5

📍 Ubicación de la base de datos: nam5

31/09




```

(3, 'Carlos', 'García'),
(4, 'Ana', 'Martínez'),
(5, 'Pedro', 'Ruiz');

-- Insertar relaciones entre notas y profesores
INSERT INTO Notas_Profesores (NotaID, ProfesorID)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5);

SELECT dbo.Alumnos.Nombre, dbo.Alumnos.Apellido, dbo.Alumnos.FechaNacimiento, dbo.Alumnos.Carrera, dbo.Alumnos_Becas.BecaID, dbo.Becas.NombreBeca, dbo.Becas.Descripcion, dbo.Becas.Monto, dbo.Notas.Materia,
FROM
    dbo.Alumnos INNER JOIN
    dbo.Alumnos_Becas ON dbo.Alumnos.AlumnoID = dbo.Alumnos_Becas.AlumnoID INNER JOIN
    dbo.Becas ON dbo.Alumnos_Becas.BecaID = dbo.Becas.BecaID INNER JOIN
    dbo.Notas ON dbo.Alumnos.AlumnoID = dbo.Notas.AlumnoID INNER JOIN
    dbo.Notas_Profesores ON dbo.Notas.NotaID = dbo.Notas_Profesores.NotaID INNER JOIN
    dbo.Profesores ON dbo.Notas_Profesores.ProfesorID = dbo.Profesores.ProfesorID
  
```

00 %  Results Messages

	Nombre	Apellido	FechaNacimiento	Carera	BecaID	NombreBeca	Descripcion	Monto	Materia	Nota
1	Juan	Pérez	2000-05-15	Ingeniería Informática	1	Beca de Excelencia	Para alumnos con promedio superior a 9.0	500	Matemáticas	9.5
2	Maria	González	1999-09-23	Matemáticas	3	Beca Socioeconómica	Para alumnos en situación económica vulnerable	200	Historia	8
3	Carlos	López	2001-02-10	Administración de Empresas	2	Beca Deportiva	Para alumnos destacados en deportes	300	Programación	9.2
4	Ana	Martínez	2000-11-30	Derecho	4	Beca de Investigación	Para alumnos participantes en proyectos de inves...	400	Derecho Penal	7.8
5	Pedro	Ruiz	1998-07-08	Medicina	5	Beca Cultural	Para alumnos destacados en actividades culturales	250	Anatomía	9.7