



UNIVERSIDAD DON BOSCO

**DISEÑO Y PROGRAMACION DE SOFTWARE
MULTIPLATAFORMA
DPS941 G01T**

ING. ALEXANDER SIGÜENZA

FORO 1

No	INTEGRANTES	CARNÉ
1	Caleb Verenice López Gutiérrez	LG211551
2	René Saúl Jovel Calderón	JC211517
3	Bryan Antonio Mena Cortez	MC211787

29 De octubre Del 2023

PARTE I

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

Las bases de datos SQL tienen un volumen de datos limitado, por lo que son utilizadas por organizaciones, empresas y también instituciones con una cantidad de datos determinada, como por ejemplo el registro de una escuela de 1000 alumnos, en cuanto a las NoSQL también conocidas como "Not Only SQL", se almacenan grandes cantidades de datos como la Big Data, por lo que también permite crear aplicaciones o modelos de datos mucho más interactivos en la nube, por ejemplo, juegos en línea y aplicaciones web, además estas están orientadas a documentos y no a tablas, los formatos de estos son JSON o BSON, un ejemplo es MongoDB y Couchbase. En el caso de las bases de datos NoSQL, se debe tener en cuenta que pueden ser de columnas, en las que se almacenan en lugar de filas, siendo esto lo que las hace eficientes para consultas analíticas y agregaciones, como Apache, Cassandra y Hbase, dentro de las mismas bases de datos NoSQL, tenemos las de clave-valor, las cuales almacenan datos en pares clave-valor, son altamente eficiente para operaciones de lectura y escritura, como por ejemplo Redis y Riak, y tenemos las bases de datos en grafos, las cuales son para datos relacionales en forma de nodos y relaciones, como Neo4j y amazon Neptune.

En cuanto a otros tipos de bases de datos NoSQL, tenemos:

- Bases de datos en memoria.
- Bases de datos de tiempo real.
- Bases de datos multimodelo.

Conociendo el concepto de ambas y sus implementaciones, nos damos cuenta de las diferencias claras entre las mismas.

2. ¿Cuáles son las diferencias específicas entre Cloud Firestore y Real time Database?

Cuadro comparativo de las diferencias específicas entre Cloud Firestore y Real time Database	
Cloud Firestore	Real time
Almacena datos como colecciones de documentos.	Almacena datos como un gran árbol JSON.
Soporte sin conexión para clientes de Apple, Android y la Web.	Se admite la presencia.
Consultas indexadas con ordenamiento y filtrado compuestos.	Consultas directas con funciones de ordenamiento y filtrado limitadas.

Operaciones avanzadas de escritura y transacción.	Operaciones básicas de escritura y transacción. Permite escribir datos mediante operaciones de configuración y actualización. Las transacciones son atómicas en un subárbol de datos específico.
Cloud Firestore es una solución regional y multirregional con ajuste de escala automático. Aloja los datos en varios centros de datos de distintas regiones, lo que garantiza una escalabilidad global y una confiabilidad sólida.	Realtime Database es una solución regional
Las tasas de escritura en índices o documentos individuales tienen límites.	Las tasas de escritura en piezas individuales de datos no tienen límites locales.
Reglas sin formato de cascada que combinan autorización y validación: Las reglas de seguridad de Cloud Firestore protegen las operaciones de lectura y escritura desde los SDK para dispositivos móviles. Identity and Access Management (IAM) protege las operaciones de lectura y escritura desde los SDK para servidores.	Latencia extremadamente baja, es la opción ideal para sincronizar estados con frecuencia. El escalamiento necesita fragmentación. Lenguaje de reglas en cascada que separa la autorización de la validación. Las reglas de seguridad de Realtime Database protegen las operaciones de lectura y escritura desde los SDK para dispositivos móviles. Transmisión en cascada de reglas de lectura y escritura
Se cobra principalmente por operaciones ejecutadas en la base de datos (lecturas, escrituras y eliminaciones) y, con una tarifa menor, por ancho de banda y almacenamiento. Cloud Firestore admite límites de gasto diarios para proyectos de App Engine a fin de garantizar que no excedas los costos esperados.	Se cobra solo por ancho de banda y almacenamiento, pero con una tarifa mayor.

Cuadro comparativo de las diferencias específicas entre Cloud Firestore y Real time Database basado en la fuente de consulta

Fuente de consulta: <https://firebase.google.com/docs/firestore/rtdb-vs-firestore?hl=es-419>

3. Basándose en su investigación, ¿cuál de estas bases de datos consideran que sería la mejor opción para implementar en una aplicación desarrollada en ReactNative?

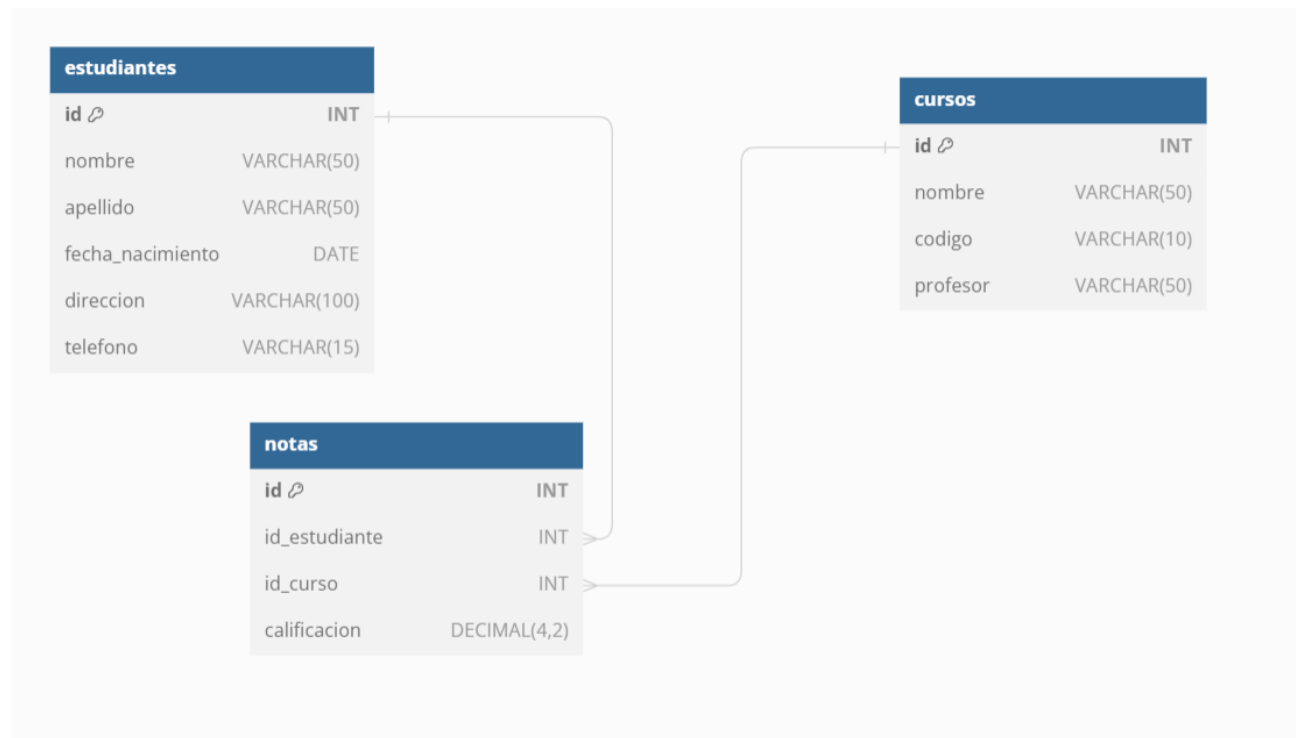
Sabiendo que esta tecnología es para aplicaciones nativas, y también se pueden crear aplicaciones cuyo enfoque es web, podríamos hacer uso de una base de datos NoSQL, teniendo en cuenta que, estas deben ser evaluadas según el objeto de creación, un ejemplo es firebase, cuyo almacenamiento es en la nube.

PARTE II

En esta fase, deben elaborar dos estructuras de base de datos: una utilizando SQL y otra utilizando NoSQL (Firebase Firestore o Realtime Database). Ambas bases de datos deben permitir almacenar las notas de los alumnos becarios de UDB VIRTUAL.

Diagrama de estructura de Base de Datos SQL

La base de datos diseñada para almacenar notas de estudiantes de la UDB consta de tres tablas: "estudiantes" que contiene información personal de los estudiantes, "cursos" que registra los detalles de los cursos y "notas" que relaciona calificaciones con estudiantes y cursos. Estas tablas están interconectadas mediante claves foráneas, lo que permite un seguimiento detallado del desempeño académico de los estudiantes en cursos específicos sin repetir información.

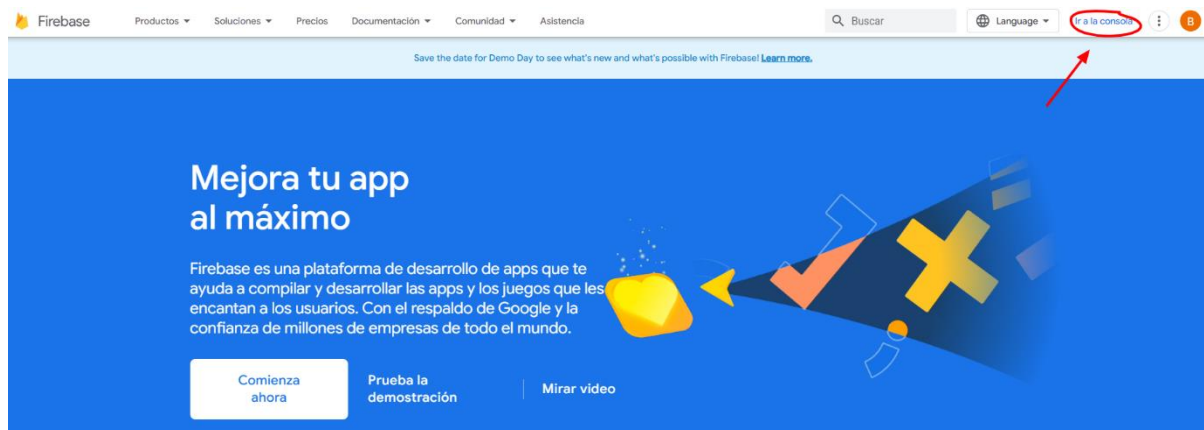


Estructura base de datos NoSQL con Cloud Firestore

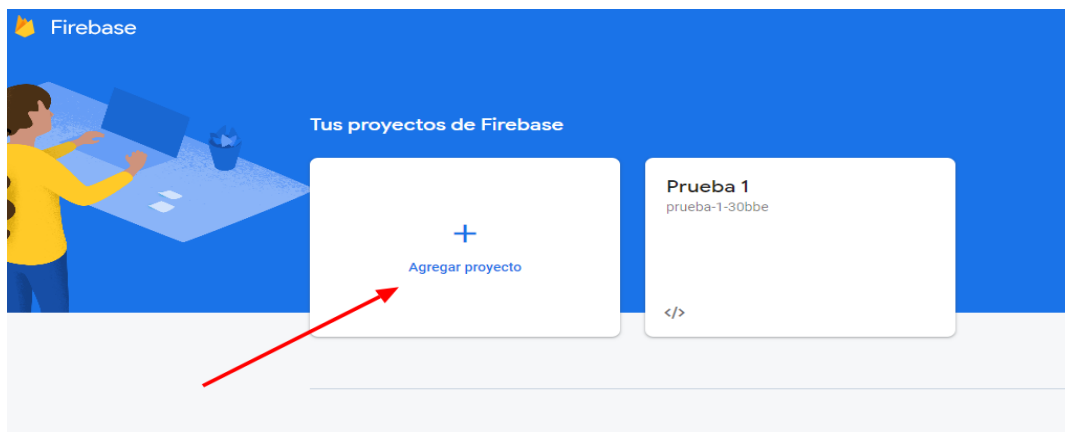
Aunque existen varios tipos de bases de datos NoSQL como, por ejemplo: clave-valor, grafos, tiempo real, entre otros. Para este caso utilizaremos un tipo de bases de datos NoSQL orientado a documentos, ya que esta estructura es flexible y permite organizar y consultar los datos de manera eficiente. En este tipo de estructura los datos están organizados en colecciones y cada colección contiene los diferentes campos necesarios para almacenar dichos datos.

Para poder crear una base de datos NoSQL seguiremos los siguientes pasos:

- 1- Con nuestra cuenta Google vamos a firebase.google.com y ahí crearemos una cuenta o ingresaremos con una ya existente y nos dirigimos al botón “ir a la consola”.



- 2- Una vez dentro de la consola seleccionaremos “agregar proyecto”.



- 3- Pondremos el nombre de nuestro proyecto, en este caso se llamará “DPS base de datos NoSQL” y damos clic en continuar.

× Crear un proyecto(paso 1 de 2)

Comencemos con el nombre de tu proyecto [?]

Nombre del proyecto

DPS base de datos NoSQL

✎ dps-base-de-datos-nosql

Continuar

- 4- Firebase nos preguntará si queremos usar métricas de análisis para nuestra base de datos, en este caso pondremos que no y luego damos clic en “crear proyecto”

× Crear un proyecto(paso 2 de 2)

Google Analytics para tu proyecto de Firebase

Google Analytics es una solución de analítica ilimitada y gratuita que permite usar la segmentación, los informes y otras funciones en Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing y Cloud Functions.

Google Analytics habilita las siguientes funciones:

× Pruebas A/B [?]

× Usuarios que no experimentan fallos [?]

× Segmentación de usuarios y orientación a ellos en los productos de Firebase [?]

× Activadores de Cloud Functions basados en eventos [?]

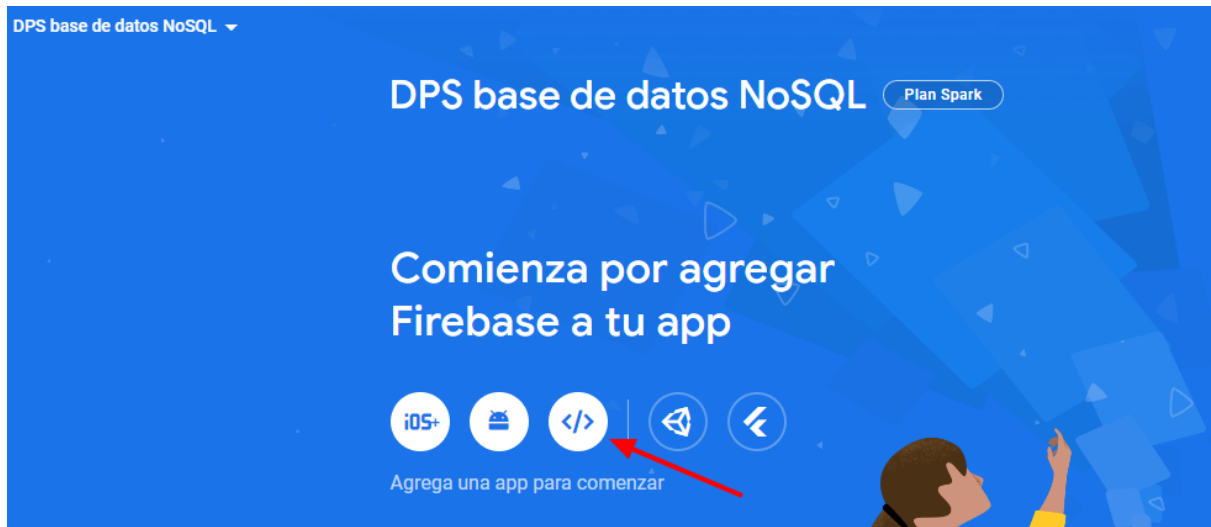
× Informes ilimitados y gratuitos [?]

☒ Habilitar Google Analytics para este proyecto
Recomendado

Anterior

Crear proyecto

- 5- Una vez creado el proyecto podemos seleccionar si queremos que sea par IOS, Android o Web, en este caso escogeremos Web para el ejemplo.



- 6- Le pondremos un nombre a nuestra web app.

A screenshot of the 'Agregar Firebase a tu aplicación web' (Add Firebase to your web application) form. The title is at the top with a close button (X). Below the title, there is a section labeled '1 Registrar app'. Under this section, there is a label 'Sobrenombre de la app' with a help icon (?). Below the label is a text input field containing 'Notas Becarios UDB Virtual'. Further down, there is a checkbox that is currently unchecked, followed by the text 'Además, configura **Firestore** para esta app.' and a link 'Más información'. Below this, there is a paragraph: 'Hosting también se puede configurar más adelante. Puedes comenzar cuando quieras sin costo alguno.' At the bottom of the form is a blue button labeled 'Registrar app'.

- 7- Al darle clic al botón “Registrar App” tendremos nuestro SDK para poner en el código de nuestra app.

2 Agrega el SDK de Firebase

☒ Usar npm ☐ Usar una etiqueta <script>

Si ya usas [npm](#) y un agrupador de módulos como [Webpack](#) o [Rollup](#), puedes ejecutar el siguiente comando para instalar la versión más reciente del SDK ([más información](#)):

```
$ npm install firebase
```



Luego, inicializa Firebase y comienza a usar los SDK de los productos que quieres utilizar.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyAfRS659v82vpe8ppxGwVUCUavH6D-ZtFk",
  authDomain: "dps-base-de-datos-nosql.firebaseio.com",
  projectId: "dps-base-de-datos-nosql",
  storageBucket: "dps-base-de-datos-nosql.appspot.com",
  messagingSenderId: "1070087178559",
  appId: "1:1070087178559:web:728718dc8a6aa9304beb8a"
};

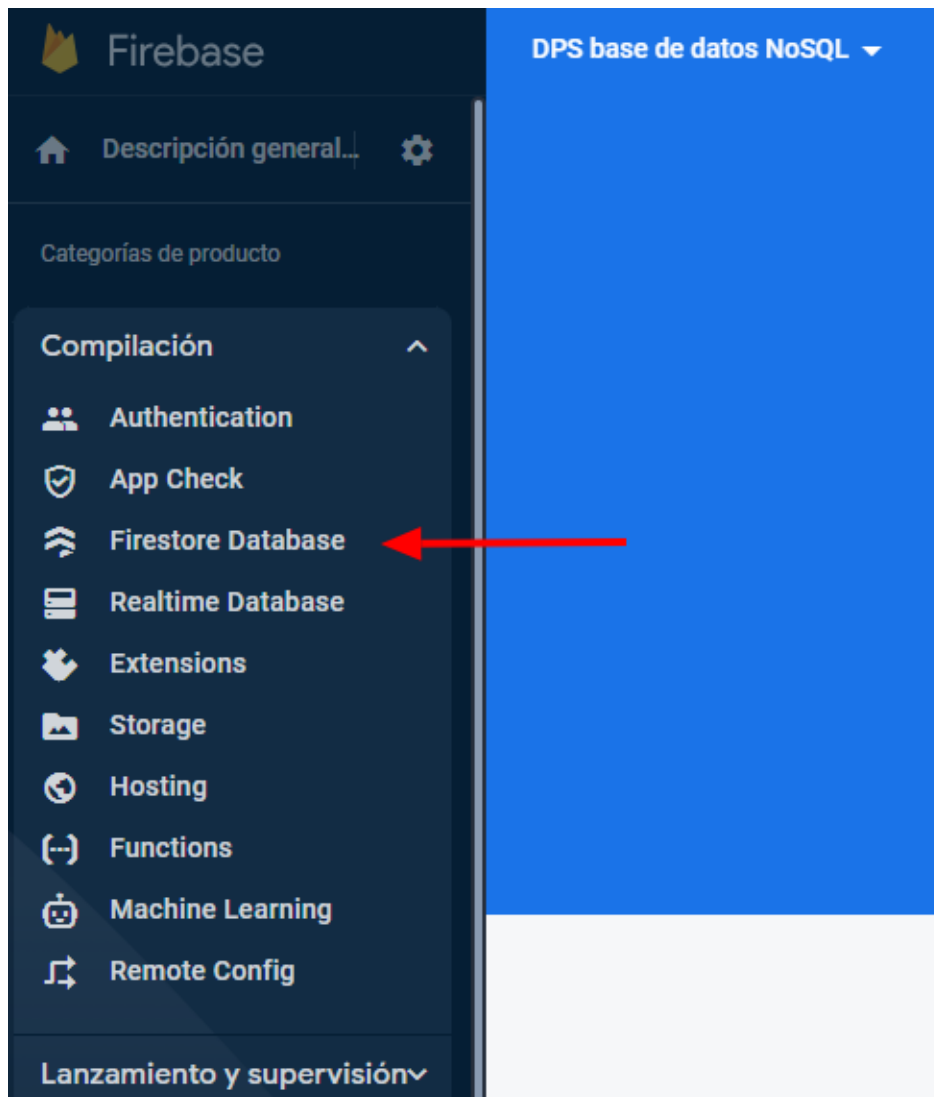
// Initialize Firebase
const app = initializeApp(firebaseConfig);
```



Nota: Esta opción utiliza el [SDK de JavaScript modular](#), que proporciona un tamaño reducido del SDK.

Obtén más información sobre Firebase para la Web: [primeros pasos](#), [referencia de la API del SDK web](#) y [muestras](#)

- 8- Al irnos a la consola ya podremos crear nuestra base de datos, para este caso escogeremos realizarla con Firestore.



Cloud Firestore

Actualizaciones en tiempo real, consultas potentes y ajuste de escala automático

Crear base de datos

9- Escogemos si usaremos el modo producción o el modo prueba.

Crear base de datos

1 Configura la ubicación de Cloud Firestore

2 Reglas de seguridad de Cloud Firestore

Después de definir la estructura de tus datos, debes crear reglas para protegerlos.
[Más información](#)

☐ Iniciar en **modo de producción**

Tus datos son privados de forma predeterminada. El acceso de lectura/escritura de los clientes solo se otorgará como se indica en tus reglas de seguridad.

☒ Comenzar en **modo de prueba**

Para permitir una configuración rápida, los datos se abren de forma predeterminada. Sin embargo, debes actualizar las reglas de seguridad en un plazo de 30 días a fin de habilitar el acceso de lectura/escritura a largo plazo para los clientes.

```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2023, 11, 24);
    }
  }
}
```

! Las reglas de seguridad predeterminadas del modo de prueba permiten que cualquier usuario con acceso a tu referencia de base de datos pueda ver, editar y borrar todos los datos durante los siguientes 30 días.

Si habilitas Cloud Firestore, no podrás usar Cloud Datastore en este proyecto

Cancelar **Habilitar**

10- Para la estructura de nuestra base de datos en la cual usaremos colecciones y documentos, lo organizaremos así:

Colección “alumnos”:

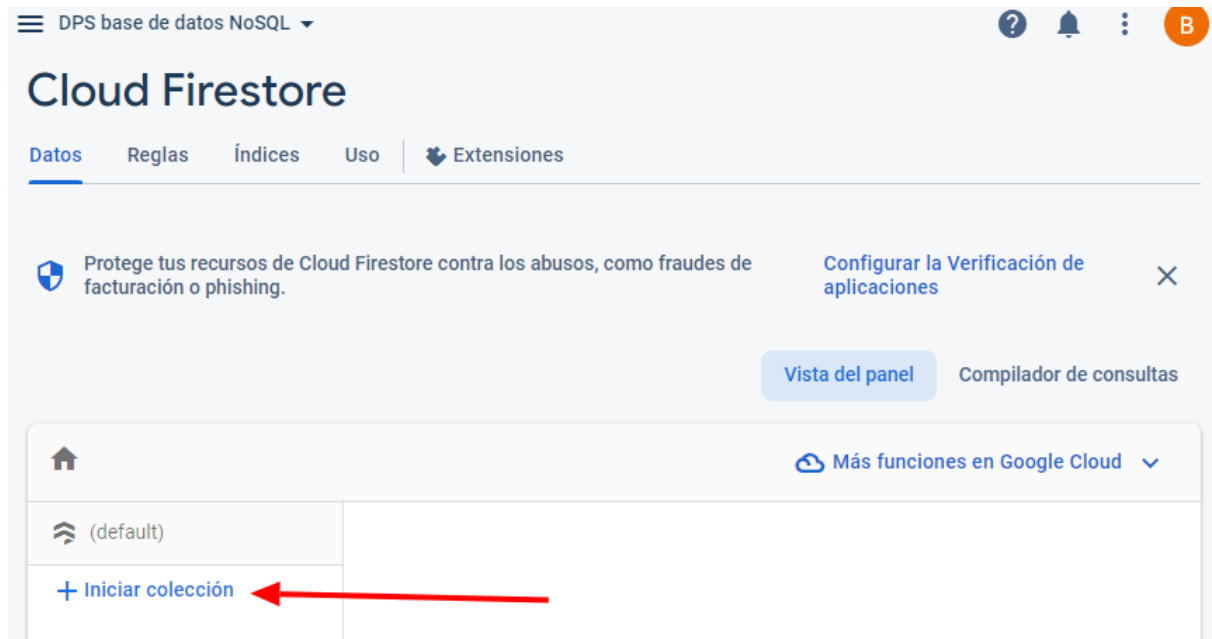
Cada documento de esta colección representaría a un alumno y tendría los campos: “id_alumno”, “nombres”, “apellidos”, “carnet”, “estado_beca”.

Colección “notas”:

En esta colección cada documento representa las notas obtenidas por el alumno, los campos serán los siguientes: “id_alumno”, “materia”, “examen_1”, “examen_2”, “examen_3”, “promedio”.

El campo “id_alumno” nos servirá para relacionar las notas con los alumnos.

- 11- En Firebase usando Firestore crearemos dicha estructura comenzando por las colecciones antes mencionadas.



Inicia una colección

1

Asignar un ID a la colección

2

Agregar el primer documento

Cancelar

Siguiente

12- Podemos crear un documento de prueba para dicha colección, creando los campos y agregándoles datos desde Firebase.

Inicia una colección

✓ Asignar un ID a la colección

2 Agregar el primer documento

Ruta superior del documento

/alumnos

ID de documento ?

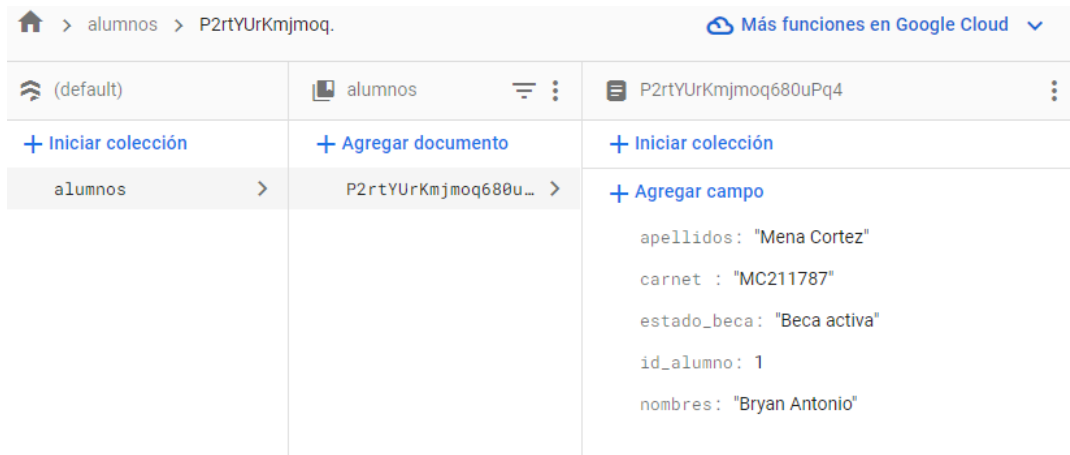
P2rtYUrKmjmoq680uPq4

Campo	Tipo	Valor	
id_alumno	number	1	−
nombres	string	Bryan Antonio	−
apellidos	string	Mena Cortez	−
carnet	string	MC211787	−
estado_beca	string	Beca activa	−
+ Agregar campo			

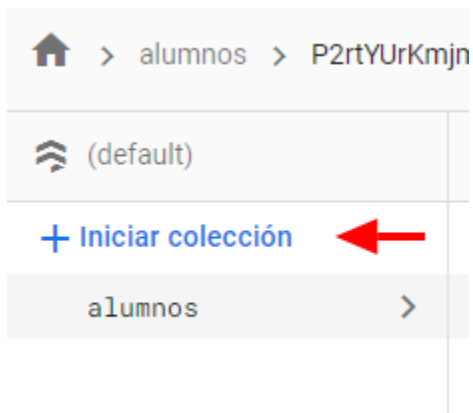
Cancelar

Guardar

Aquí vemos los datos ya almacenados



13- Para crear nuestra segunda colección llamada “notas, damos clic aquí:



Y colocamos el nombre de la colección:

Inicia una colección

1

Asignar un ID a la colección

2

Agregar el primer documento

Ruta superior

/

ID de la colección ⓘ

notas

Cancelar

Siguiente

14- De igual manera ingresaremos datos de prueba con los campos de dicha colección:

Inicia una colección

✓ Asignar un ID a la colección

2 Agregar el primer documento

Ruta superior del documento ⓘ

/notas

ID de documento ⓘ

NhUEokj6NwiN08kb2hxc

Campo	Tipo	Valor
id_alumno	= number	1
materia	= string	Calculo
examen_1	= number	10
examen_2	= number	10
examen_3	= number	10
promedio	= number	10

+ Agregar campo

Obtendremos el siguiente resultado, donde podemos ver que están las colecciones “alumnos” y “notas” y cada documento son los datos ingresados. Y nuestra base NoSQL está lista.

🏠 > notas > NhUEokj6NwiN...

Más funciones en Google Cloud

(default)

notas

NhUEokj6NwiN08kb2hxc

+ Iniciar colección

alumnos

notas >

+ Agregar documento

NhUEokj6NwiN08kb2... >

+ Iniciar colección

+ Agregar campo

examen_1: 10

examen_2: 10

examen_3: 10

id_alumno: 1

materia: "Calculo"

promedio: 10

Conclusiones en base a comparación de las 2 bases de datos.

En la practica realizada podemos observar que, en ambos casos, tanto en la base de datos SQL, como en la NoSQL, logramos cumplir con el fin, el cual es, poder llevar un registro de los datos que necesitamos guardar, por lo que la elección del tipo de bases de datos que usaremos ya depende de factores como las herramientas de software que usará el proyecto, la visión a futuro de la aplicación que llevará la base de datos e incluso el presupuesto con el que cuenta nuestro proyecto.

Las bases de datos en la nube como Firebase nos facilitan a desplegar nuestra aplicación de manera rápida en cuanto al almacenamiento de la información, si al pasar del tiempo nuestra aplicación crece en usuarios e información, podemos optar por migrar nuestra aplicación a utilizar un servidor donde procese la información y guarde en bases de datos NoSQL o SQL, donde tenemos mayor flexibilidad en escalar recursos como memoria o CPU, particionar las bases de datos, y reducir costos operativos.