

# Top level collection con Firebase

El modelo de bases de datos no relacionadas es un poco más cercano al mundo real en su comportamiento.

- Las top level collections son las colecciones que se tienen de inmediato o entrada en el proyecto.
- Firebase es un servicio que tiene múltiples opciones y está pensado principalmente para aplicaciones móviles y web.

## Creando y borrando documentos en Firestore

# Colecciones vs Subcolecciones

La particularidad de las top level collections es que existen en el primer nivel de manera intrínseca. Las subcolecciones ya no vivirán al inicio de la base de datos.

Si tienes una entidad separada que vas a referenciar desde muchos lugares es recomendado usar un top level collection. Por el otro lado si se necesita hacer algo intrínseco al documento es aconsejable usar subcolecciones.

## Recomiendo PlatziBlog

**Nota:** Las bases de datos no redimensionables, no están optimizadas para hacer "querés", sino, mantener el estado de la aplicación por ende si se desean luego hacer reportes en la aplicación de diferentes datos, no es recomendable usar BD NoSQL.

# Construyendo Platziblog en Firestore

Debido a que este es un curso de base de datos no muestran como funciona junto con el backend, al menos no a este punto. En la vida real no hay nadie llenando las bases de datos manualmente, este tipo de base de datos manejan esquemas en inglés **schema** es algo así como una guía para que al momento de guardar la colección sepa como hacerlo adjunto imagen de un schema en MongoDB para un blog sencillo.

- tiene un título
- la etiqueta
- el body

al final retorna un JSON.

```
const blogSchema = new Schema([
  {
    title: {
      type: String,
      required: true,
    },
    tag: {
      type: String,
      required: true,
    },
    body: {
      type: String,
      required: true,
    },
  },
  { timestamps: true }
]);
```

# Proyecto Final: Transformando tu proyecto en una db no relacional

Dentro de las bases de datos relacionales tenemos diferentes niveles de datos. En primer lugar tenemos las Bases de Datos o Esquemas como repositorios donde vivirán los datos que nos interesa guardar. Dentro del esquema existen las Tablas que provienen del concepto de entidades; y a su vez dentro de las tablas tenemos las tuplas o renglones.

Cuando trabajamos con bases de datos basadas en documentos como Firestore, aún existe la figura de la base de datos, sin embargo cambiaremos las tablas en favor de las colecciones y las tuplas en lugar de los documentos.

Recuerda:

Tabla -> Colección

Tupla -> Documento

Dentro de las Colecciones existen 2 grandes tipos. Las Top level collection o colecciones de nivel superior y las subcollections o subcolecciones. Estas últimas viven únicamente dentro de un documento padre.

¿Cómo saber cuál escoger?

Para determinar si tu colección debe ser top level o subcolección no hay una regla escrita en piedra y más bien tiene que ver con el caso de uso en particular y con la experiencia que hayas ganado como desarrollador.

Lo cierto es que no hay una sola forma de estructurar nuestra DB basada en documentos, y por tanto no existe una respuesta correcta, sin embargo a continuación te ofrezco un par de reglas guía que puedes utilizar para transformar tu proyecto que ya trabajaste en bases de datos relacionales en un proyecto no relacional.

**Regla 1. Piensa en la vista de tu aplicación**

La primera pista que te puedo dar es que pienses en un inicio en la manera en que los datos serán extraídos. En el caso de una aplicación, la mejor forma de pensarlo es en términos de las vistas que vas a mostrar a un momento determinado en la aplicación.

Es decir, al armar la estructura en la base de datos que sea un espejo o que al menos contenga todos los datos necesarios para llenar las necesidades que tiene nuestra parte visual en la aplicación.

En el caso de Platziblog por ejemplo si tienes una vista de un blog post individual, generalmente conviene mostrar además de los datos inherentes al post como el contenido, datos adicionales como las etiquetas que tiene o por ejemplo el autor (o autores si es colaborativo), en este caso tal vez convenga guardar estas dos “entidades” (autores y etiquetas) como subcolecciones de cada documento blog post.

**Regla 2. La colección tiene vida propia**

Esta regla se refiere a que la excepción a la regla 1 es cuando tenemos un caso en que la “entidad” que tiene necesidad de vivir y modificarse constantemente de manera independiente a las otras colecciones. Por ejemplo en Platziblog podemos en el ejemplo anterior hacer una excepción a autores porque nos conviene tenerlas como top level collection en el sentido que se añadan, borren, cambien o listen los usuarios sin depender del blog post.

Experimenta aplicando estas dos reglas a un proyecto que ya conozcas en una base de datos relacional y trata de convertirla en un proyecto de Firestore y comentanos los retos a los que te enfrentaste.