

Firestore (III)

Inserciones

En los ejemplos que hemos visto, la búsqueda de **Documentos** dentro de una colección consiste básicamente en identificar el nodo, añadirle una o más condiciones, y luego obtener los documentos de la query.

El proceso de inserción de un nuevo documento es similar... aunque antes tienes que recordar que:

- Firebase NO controla las ID ni las autoincrementa.
- Firebase NO controla si ese Documento debería estar en esa colección.
- Firebase NO controla el formato de los Documentos.
- Firebase NO controla si has puesto campos de más o de menos.

Por lo tanto, este tipo de cosas las tienes que controlar tú a mano.

Un ejemplo de una inserción simple de un Documento.

```
public void addEmpresa(Empresa empresa) throws FireBaseException {
    try {
        Firestore dataBase = FirestoreClient.getFirestore();






        // Query... donde vamos a colocar el documento de empresa
        CollectionReference query = dataBase.collection(COLLECTION_EMPRESA);

        // El Map que representa el documento con sus campos...
        Map<String, Object> empresaMap = new HashMap<>();
        empresaMap.put("nombre", empresa.getNombre());
        empresaMap.put("localizacion", empresa.getLocalizacion());

        // Le cargamos la ID (de Firebase) de forma manual desde Java
        DocumentReference documentReference = query.document(empresa.getId());
        documentReference.set(empresaMap);

    } catch (Exception e) {
        throw new FireBaseException("Error - " + e.getLocalizedMessage());
    }
}
```

Las **ID a las que se refiere** en este caso NO son el campo ID del Documento, sino la ID interna que genera el propio Firebase. ¿Recuerdas que, al crear un documento nuevo, Firebase te obligaba a meter una ID como mínimo? Pues es esa. A las E1 y E3 que ves en la imagen.

 (default)	 EmpleadoBD  	 E1
+ Iniciar colección	+ Agregar documento	+ Iniciar colección
EmpleadoBD >	E1 >	+ Agregar campo
EmpresaBD	E3	Apellido: "Perez" Comision: "No"

También puedes optar por dejar a Firebase que se encargue, obviamente:

```
public void addEmpresaAutogen(Empresa empresa) throws FireBaseException {
    try {
        Firestore dataBase = FirestoreClient.getFirestore();

        // Query... donde vamos a colocar el documento de empresa
        CollectionReference query = dataBase.collection(COLLECTION_EMPRESA);

        // El Map que representa el documento con sus campos...
        Map<String, Object> empresaMap = new HashMap<>();
        empresaMap.put("nombre", empresa.getNombre());
        empresaMap.put("localizacion", empresa.getLocalizacion());

        // Le cargamos la ID autogenerada por firebase
        DocumentReference documentReference = query.document();
        documentReference.set(empresaMap);
    } catch (Exception e) {
        throw new FireBaseException("Error - " + e.getLocalizedMessage());
    }
}
```

Finalmente, tenemos el problema de las **referencias**. No deberían insertarse a mano, sino obteniendo primero la referencia al objeto que queremos referenciar y después asignándosela al campo correspondiente del nuevo Documento. Es decir... como si tuviésemos que asignar las claves primarias y ajenas de una relación 1:N **manualmente** en SQL.

```
Firestore dataBase = FirestoreClient.getFirestore();

// La referencia a la empresa...
ApiFuture<QuerySnapshot> future = dataBase.collection(COLLECTION_EMPRESA)
    .whereEqualTo("nombre", nameEmpresa).get();

// Procesamos la query...
QuerySnapshot querySnapshot = future.get();
List<QueryDocumentSnapshot> empresas = querySnapshot.getDocuments();
if (!empresas.isEmpty()) {
    // La empresa existe
    ret = true;

    // Referencia a esa empresa...
    DocumentSnapshot documentSnapshot = empresas.get(0);
    DocumentReference empresaReference = documentSnapshot.getReference();

    // -- Preparamos el Documento del Empleado

    // Query... donde vamos a colocar el documento de empresa
    CollectionReference query = dataBase.collection(COLLECTION_EMPLEADO);

    // El Map que representa el documento con sus campos...
    Map<String, Object> empresaMap = new HashMap<>();
    empresaMap.put("Nombre", empleado.getNombre());
    empresaMap.put("Apellido", empleado.getApellido());
    empresaMap.put("Direccion", empleado.getDireccion());
    empresaMap.put("Fecha", empleado.getFecha());
    empresaMap.put("Salario", empleado.getSalario());
    empresaMap.put("Comision", empleado.getComision());
    empresaMap.put("puesto", empresaReference);

    // Le cargamos la ID (de Firebase) de forma manual desde Java
    DocumentReference documentReference = query.document(empleado.getId());
    documentReference.set(empresaMap);
} else {
    // La empresa no existe
    ret = false;
}
```

Otras inserciones

Otras operaciones de inserción que puedes realizar:

- **document()**: Crea un documento con Id autogenerado.
- **put("nombreCampo","valorCampo")**: Modifica el valor del campo
- **set(mapa)**: Modifica el documento con los valores de los campos establecidos
- **colection("nombreCol")**: Crea una colección en un determinado documento
- **getDouble("nombreCampo")**: Obtiene un campo de tipo Double.
- **add(mapa)**: Añade un documento a una colección.
- **close()**: Cierra la base de datos y actualiza las modificaciones