

# FIREBASE: DESARROLLO DE UN CRUD PARA UNA BASE DE DATOS DE MÚSICA

Por Mikel Jorge y Eneko Martinez

## Antes de empezar...

Realiza la actividad guiada [UT5\\_A1 Inicio rápido](#) para aprender cómo poner en marcha un proyecto con **Spring Boot y Firebase** y crear una colección. Revisa los apuntes sobre creación de un CRUD y escribe el código de ejemplo para familiarizarte con las clases que usaremos para un endpoint RESTful.

## Creación del proyecto

Sigue los pasos de la actividad guiada [UT5\\_A1 Inicio rápido](#) para generar el proyecto Spring Boot y para crear una base de datos Firestore en Firebase si es necesario.

## ¿Qué queremos que haga nuestro programa?

Nuestro programa se encargará de gestionar, mediante un sistema **RESTful** en **Spring Boot**, una base de datos de música almacenada en **Firestore** de Firebase. Implementará un CRUD completo para **grupos musicales**, así como para sus **discos y canciones**.

## Requisitos del sistema

### 1. Entidad "Grupo"

- **id** (String) → Autogenerado por Firestore
- **nombre** (String) → Nombre del grupo
- **genero** (String) → Género musical (Rock, Pop, Metal, etc.)
- **pais** (String) → País de origen
- **discos** (Lista de Discos)

### 2. Entidad "Disco" (Documentos anidados dentro de cada Grupo)

- **id** (String) → Autogenerado
- **titulo** (String) → Nombre del disco
- **anio** (Integer) → Año de lanzamiento
- **canciones** (Lista de Canciones)

### 3. Entidad "Canción" (Documentos anidados dentro de cada Disco)

- **id** (String) → Autogenerado
- **titulo** (String) → Nombre de la canción
- **duracion** (Double) → Duración en minutos

## Funcionalidades a implementar

### ✦ Operaciones CRUD para la colección "Grupos":

1. **Crear un grupo** → **POST** /api/grupos
2. **Obtener todos los grupos** → **GET** /api/grupos
3. **Obtener un grupo por ID** → **GET** /api/grupos/{id}
4. **Actualizar un grupo** → **PUT** /api/grupos/{id}
5. **Eliminar un grupo** → **DELETE** /api/grupos/{id}

### ✦ Operaciones CRUD para Discos dentro de un Grupo:

6. **Añadir un disco a un grupo**  
→ **POST** /api/grupos/{grupold}/discos
7. **Obtener todos los discos de un grupo**  
→ **GET** /api/grupos/{grupold}/discos
8. **Eliminar un disco de un grupo**  
→ **DELETE** /api/grupos/{grupold}/discos/{discold}

### ✦ Operaciones CRUD para Canciones dentro de un Disco:

9. **Añadir una canción a un disco**  
→ **POST** /api/grupos/{grupold}/discos/{discold}/canciones
10. **Obtener todas las canciones de un disco**  
→ **GET** /api/grupos/{grupold}/discos/{discold}/canciones
11. **Eliminar una canción de un disco**  
→ **DELETE** /api/grupos/{grupold}/discos/{discold}/canciones/{cancionId}

## Requisitos técnicos

- ✓ Usar **Firebase Firestore** como base de datos.
- ✓ Probar los endpoints con **Postman** y validar los resultados.
- ✓ Respuestas con **ResponseEntity** siguiendo buenas prácticas REST.
- ✓ Estructura de datos en **Firestore** correctamente modelada (Grupos → Discos → Canciones).