

UNIVERSIDAD DEL VALLE DE GUATEMALA

Colegio Universitario



## Laboratorio 11

Edwin de León

Catedrático: Francisco Anzueto

Plataformas móviles

Guatemala, 2023

## Introducción

En el proceso de preparación para el laboratorio, se llevaron a cabo una serie de pasos cruciales para configurar la base de datos que será utilizada en nuestra aplicación de Android, estos pasos incluyeron la generación de un archivo JSON que posteriormente se integraría en nuestro proyecto de Android, y la creación de una base de datos en Cloud Firestore de Firebase.

La base de datos resultante, a la que hemos llamado "Lab11EJGDLG", representa un componente esencial de nuestro proyecto, esta base de datos se concibe como un repositorio centralizado y estructurado destinado a albergar grandes volúmenes de datos, permitiendo un almacenamiento organizado que facilita la recuperación y el análisis de información de manera eficiente.

Uno de los aspectos más destacados de esta base de datos es la capacidad de implementar rigurosos mecanismos de seguridad, esto incluye la autenticación de usuarios y la autorización para determinar quién tiene acceso y quién tiene permisos para modificar los datos, la seguridad de la información confidencial es primordial, y Firebase proporciona las herramientas necesarias para garantizarla.

En este informe, exploraremos con detalle los procesos que se llevaron a cabo para configurar esta base de datos, así como sus ventajas y aplicaciones en el contexto del laboratorio. Además, destacaremos cómo la base de datos "Lab11EJGDLG" desempeñará un papel fundamental en el éxito de nuestra aplicación.

# índice

## Contenido

Introducción .....	2
índice .....	3
Descripción General de la Base de Datos.....	4
Estructura de la Base de Datos.....	4
Campos y Tipos de Datos .....	6
Reglas de Seguridad .....	<b>¡Error! Marcador no definido.</b>
Implementación y Acceso .....	7
Referencias .....	8

## Descripción General de la Base de Datos

Al generar la base de datos que se va a utilizar, para el laboratorio, se realizaron una serie de pasos antes para que la base de datos de Firebase se pudiera comunicar en la app de Android, generando un archivo json, que vamos a guardarlo en nuestro proyecto de Android, después se genera una base de datos en cloud firestore.

La base de datos se llamará Lab11EJGDLG, donde vamos a guardar información de categorías de platillos y los platillos de cada categoría, esta base de datos nos proporciona un lugar centralizado y estructurado para almacenar grandes volúmenes de datos de manera organizada, lo que facilita la recuperación y el análisis de la información.

Además de que se pueden implementar mecanismos de seguridad para proteger la información confidencial, esto incluye autenticación de usuarios y autorización para controlar quién puede acceder y modificar los datos.

## Estructura de la Base de Datos

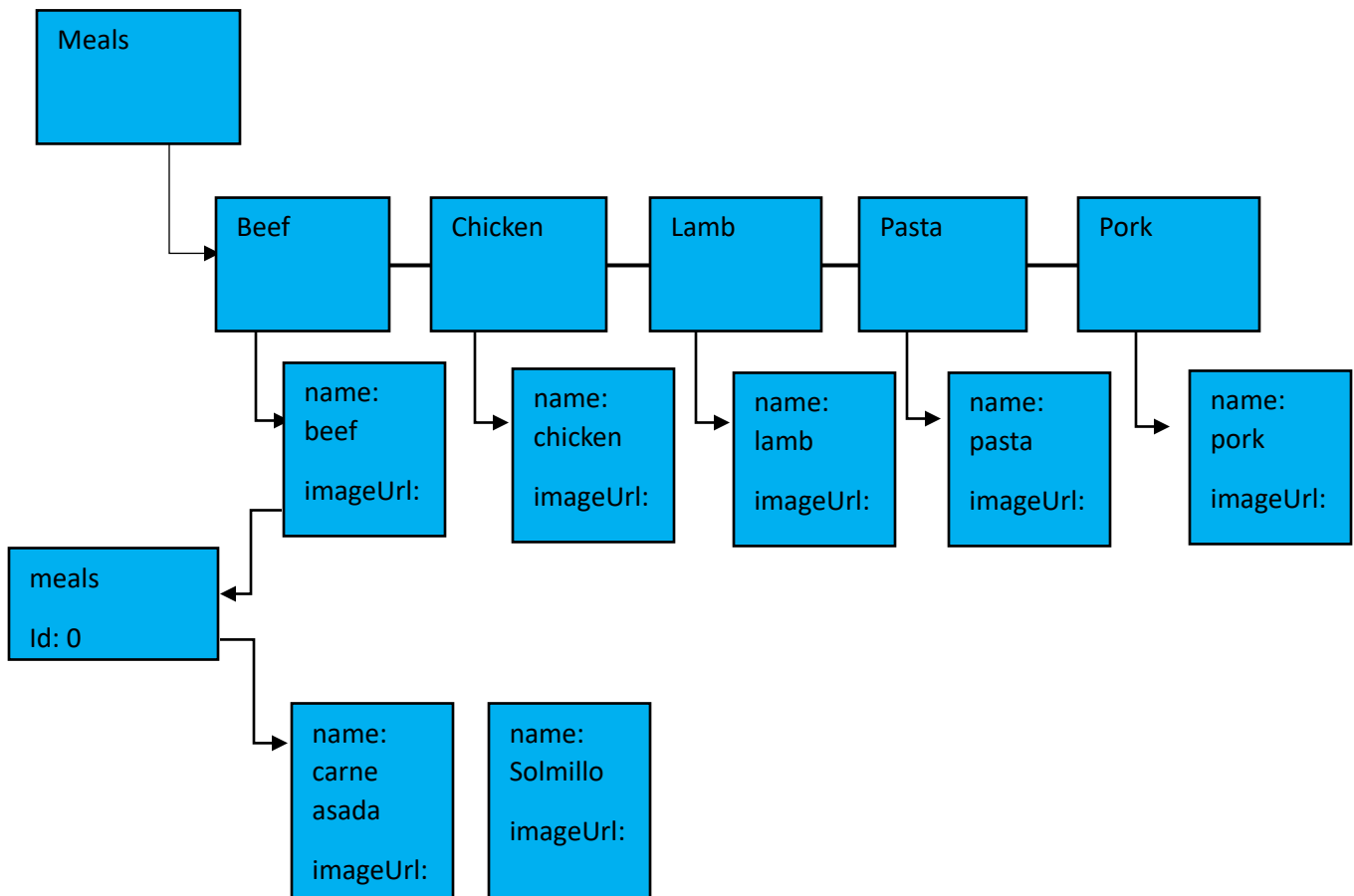
- Presentación de las colecciones principales: nuestra colección principal se va a llamar Meals, que es donde vamos a tener todos nuestros documentos.
- Las otras colecciones se van a llamar meals# (se cambia el numeral por un numero) que va a contener los platos de las cada una de las categorías.
- Documentos dentro de cada colección y sus campos relevantes:

Documentos	campos
Beef	name: beef imageUrl: imagen deseada
Chicken	name: chicken imageUrl: imagen deseada
Lamb	name: lamb imageUrl: imagen deseada
Pasta	name: pasta imageUrl: imagen deseada
Pork	name: pork imageUrl: imagen deseada
Seafood	name: seafood imageUrl: imagen deseada
Side	name: side imageUrl: imagen deseada
Vegan	name: vegan

	imageUrl: imagen deseada
--	--------------------------

Documentos	campos
Carne asada solomillo	name: carne asada name solomillo
Pollo asado pechuga	name: pollo asado name: pechuga
Lam asado Oveja al caldo	name: lam asado name: oveja al caldo
Pasta con asado Pasta alfredo	name: pasta con asado name: pasta Alfredo
Puerco asado Pata de puerco	name:puerco asado name: pata asada
Camarones a la parrilla Camarones salteados	name: camarones a la parrilla name:camarones salteado
Side asado Papas cosidas	name: side asado name: papa cosida
Vegan a la parrilla Verduras salteadas	name: vegan a la parrilla name: Verduras salteadas

- Diagrama para representar la estructura de las colecciones y documentos necesarios.



## Campos y Tipos de Datos

Los campos que va a tener cada una de las categorías van a ser dos, los cuales van a ser name que va a ser de tipo String que va a contener el nombre del document y imageUrl que va a ser de tipo string que va a ser la dirección de la imagen que sea a obtener de internet, tanto para la de Meals que es la de las categorías, y las de los meals que son las de los platillos.

campos	Tipo
name: beef imageUrl: imagen deseada	String
name: chicken imageUrl: imagen deseada	String
name: lamb imageUrl: imagen deseada	String
name: pasta imageUrl: imagen deseada	String
name: pork imageUrl: imagen deseada	String
name: seafood imageUrl: imagen deseada	String
name: side imageUrl: imagen deseada	String
name: vegan imageUrl: imagen deseada	String

Documentos	campos
Carne asada solomillo	String
Pollo asado pechuga	String
Lam asado Oveja al caldo	String
Pasta con asado Pasta alfredo	String
Puerco asado Pata de puerco	String
Camarones a la parrilla Camarones salteadas	String

Side asado Papas cosidas	String
Vegan a la parrilla Verduras salteadas	String

## Implementación y Acceso

Estas clases de datos se utilizan para representar la estructura de respuesta de una API que proporciona información sobre comidas o platillos, al utilizar la anotación “SerializedName”, puedes garantizar que los datos JSON se mapeen correctamente a los campos de las clases de datos durante la interpretación, lo que facilita la manipulación de los datos en tu aplicación.

```
package com.zezzi.eventzezziapp.data.networking.response

import com.google.gson.annotations.SerializedName

data class MealsResponse(val meals: List<Meal>)

data class Meal(
    @SerializedName("idMeal") val id: String,
    @SerializedName("strMeal") val name: String,
    @SerializedName("strMealThumb") val imageUrl: String,
```

Figura No. 01

Esta interfaz define cómo interactuar con una API de comidas mediante solicitudes GET utilizando la biblioteca Retrofit en Kotlin, los métodos definidos en la interfaz representan las operaciones de la API que pueden recuperar información sobre categorías de comidas y platillos.

```

import com.zezzi.eventzezziapp.data.networking.response.MealsCategoriesResponse
import com.zezzi.eventzezziapp.data.networking.response.MealsResponse
import retrofit2.http.GET

interface MealsApi {
    @GET("categories.php")
    suspend fun getCategories(): MealsCategoriesResponse

    @GET("filter.php")
    suspend fun getMealsByCategory(): MealsResponse
}

```

Figura No. 02

## Referencias

*Diccionario informático*. (2023). Lawebdelprogramador.com.

<https://www.lawebdelprogramador.com/diccionario/>

Android API reference. (2023). Android Developers.

<https://developer.android.com/reference>

uses-sdk (2023). Android Developers.

<https://developer.android.com/guide/topics/manifest/uses-sdk-element?hl=es-419>

Rzy. (2022). Como conectar Firebase con Android Studio agregando las dependencias manualmente en Español [YouTube Video]. In YouTube.

<https://www.youtube.com/watch?v=NA70hqggx-k>