

Universidad Nacional Autónoma de Honduras



Asignación:

Documentación sobre app de un Recetario

Asignatura:

Base de Datos II 1000 || Lenguajes de Programación 1100

Catedrático:

Elmer Danuari Padilla

Alumnos:

20201001693

Lissandro Adolfo Cantarero

20211920084

Ángel Alexander Portillo

20221900031

Carlos Eduardo Castro

## **Introducción**

A continuación, le daremos a conocer los avances, problemas y demás sucesos ocurridos durante nuestro desarrollo de nuestra app de un recetario en flutter.

Tomando en cuenta que este lenguaje no lo hemos explotado al máximo y no hemos conocido de sus capacidades y posibles contras tuvimos que buscar varias fuentes y videos para poder empezar, sin más comenzamos.

El primer inconveniente era qué elegir, al final del consenso y más la ayuda del ingeniero y los compañeros de aula se terminó eligiendo un recetario, pero ¿Con qué lo haríamos? Y más importante ¿Cómo lo haríamos?

A primer vistazo elegimos un video para guiarnos,

<https://www.youtube.com/watch?v=VTR5HpRfS0A>



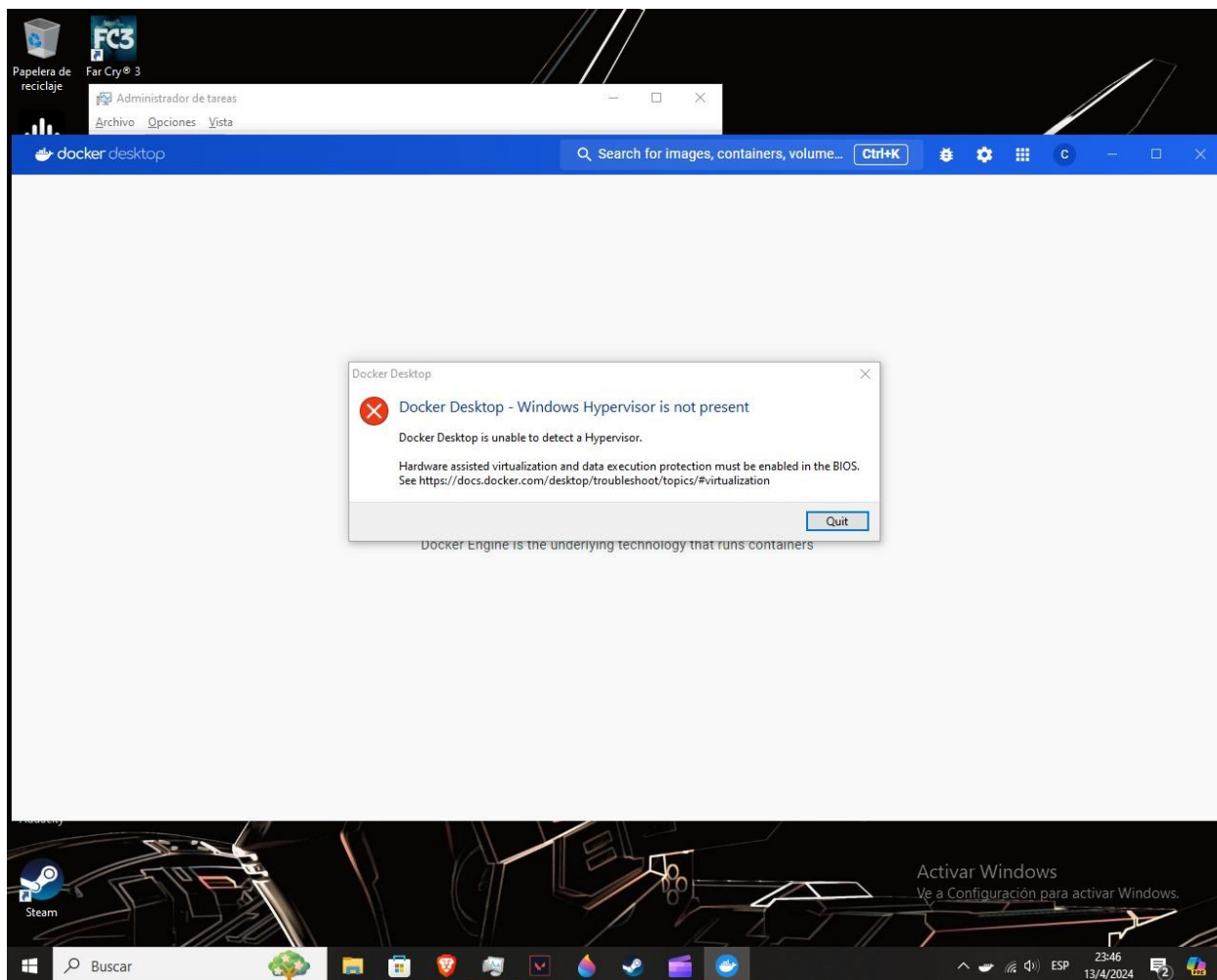
Sin embargo, el problema es que estaba construido con api y nosotros buscábamos integrar “Bases de Datos” así que después de una búsqueda más encontramos otro vídeo que nos pareció interesante y aplicaba nuevas tecnologías que era el principal objetivo de la actividad,

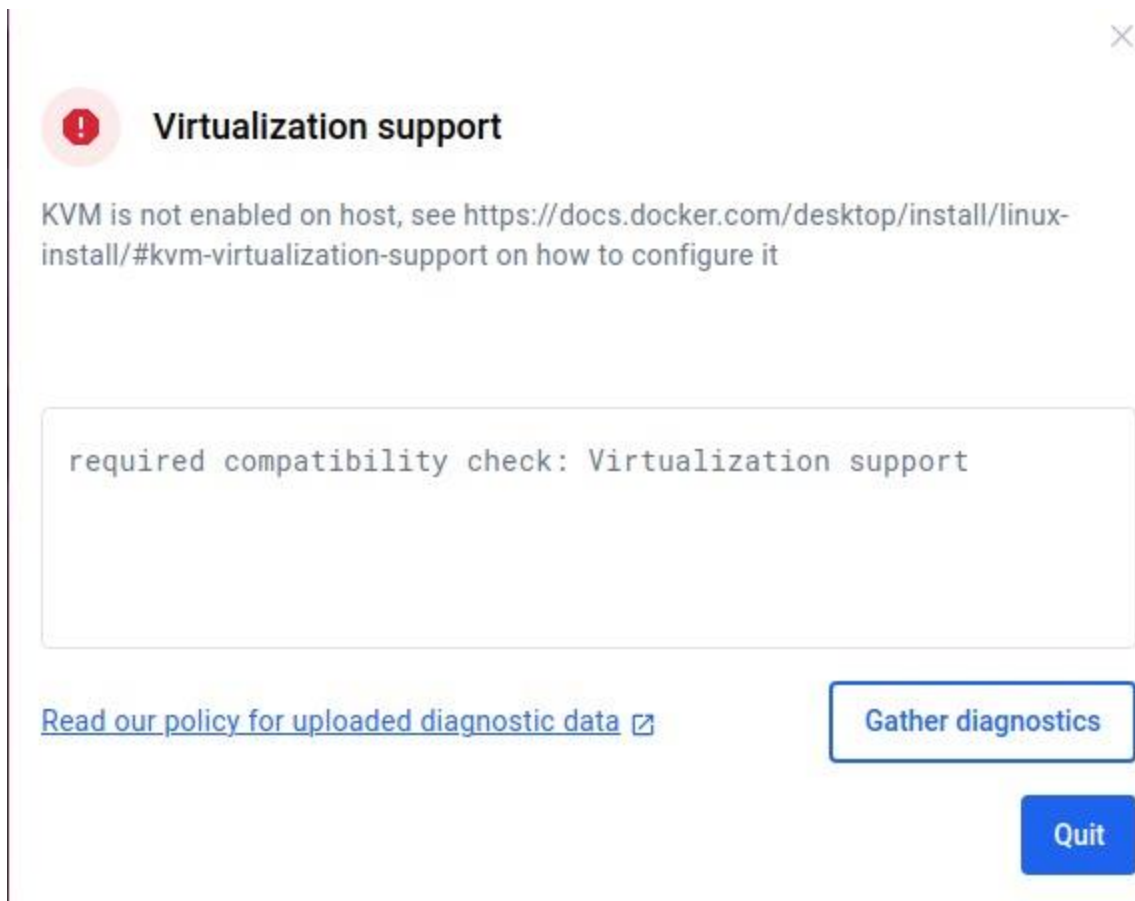
[https://www.youtube.com/watch?v=TBshP1eDVr0&list=PL\\_FZWtS2C0pEXHVucM6VEOc6VDDFx8na4&index=8](https://www.youtube.com/watch?v=TBshP1eDVr0&list=PL_FZWtS2C0pEXHVucM6VEOc6VDDFx8na4&index=8)

# Recipe Management Fullstack App



Pero el siguiente inconveniente que nos encontramos es que este necesitaba de “Docker” y solo 1 de los 3 integrantes lo pudo instalar correctamente a continuación se puede observar los errores.





A pesar de esto se intentó dar apoyo al compañero y finalizar el vídeo y ejecutarlo solamente en su computadora, pero acá surge otro problema.

Surgía este error:

```
PS C:\Users\lisan\Downloads\WORDPRESS\recipe_management_app-main\rm_graphql_client> flutter pub get
Resolving dependencies... (1.1s)
  gql_code_builder 0.10.0 (0.11.0 available)
  pedantic 1.11.1 (discontinued replaced by lints)
Got dependencies!
1 package is discontinued.
1 package has newer versions incompatible with dependency constraints.
Try `flutter pub outdated` for more information.
PS C:\Users\lisan\Downloads\WORDPRESS\recipe_management_app-main\rm_graphql_client>
```

```
name: rm_graphql_client
description: A graphql client for connecting to Recipes database.
environment:
  sdk: '>=3.3.3 <4.0.0'
dependencies:
  ferry: '^0.16.0+1'
  gql_http_link: '^1.0.1+1'
dev_dependencies:
  pedantic: '^1.11.1' # Consider replacing with `lints`
  ferry_generator: '^0.10.0'
  build_runner: '^2.4.9'
  gql_code_builder: '^0.11.0'
```

El cual ni con ayuda de diversas herramientas se pudo resolver, así que con todo en contra y a contra reloj empezamos de 0 con otra serie de videos

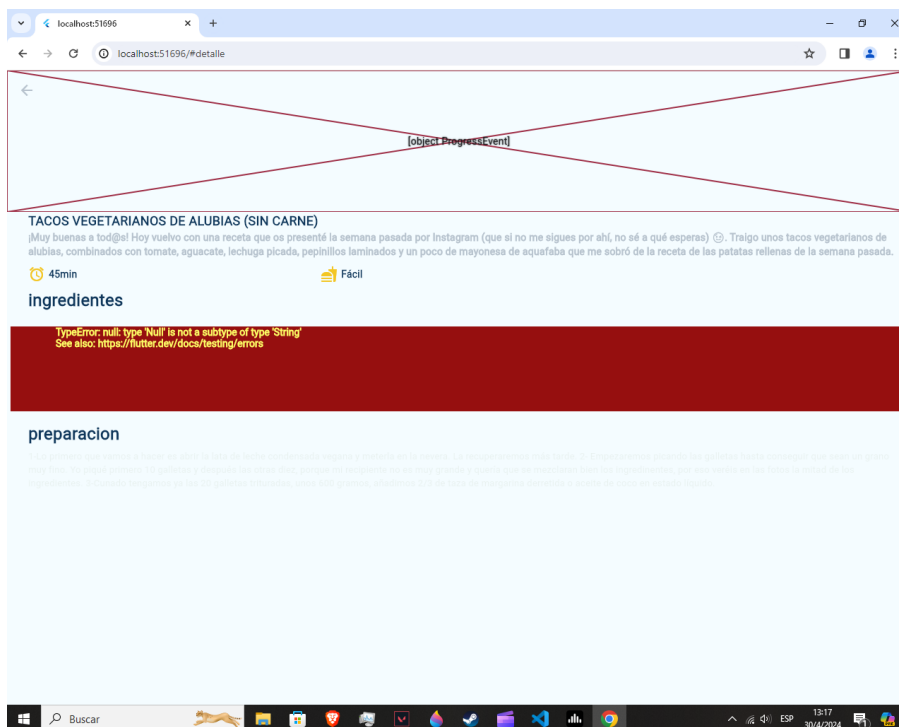
<https://youtube.com/playlist?list=PLQt9RdTywVtnLtsYrb-IA-Z2wHH1ArYkK&si=5u6rk-S2RL6YBEWH>



El único inconveniente con esta es que era de paga ya que para acceder a los videos teníamos que ser miembros del canal sin embargo logramos cubrir el pago y seguir adelante, todo iba bien para los 3 miembros cada uno avanzaba una parte del

código en su computadora lo enviaba a un drive y anunciaba hasta donde había avanzado, luego otro avisaba que parte iba a trabajar y luego subirla al drive y así hasta que a 1 de los 3 por importar lo que otro hizo le aviso el “VS code” que tenía que actualizar sus paquetes y eso ocasionó que el “Flutter” dejara de funcionar en su computadora, así que aun faltando la mitad del proyecto los 2 compañeros avanzaban.

Otro inconveniente es que muchas de las imágenes que el sujeto tenía estaban en un repositorio privado que sí, era de paga también.



Así que hasta ese momento todo salía como en la imagen, sin embargo, el proyecto se ejecutaba “correctamente” y podíamos ver el avance, otro problema es que debíamos tener un archivo .json el cual localizar en un canal de “Discord” de él fue algo tardado no estaba completo a como él le aparecía en el video así que tuvimos que editarlo a modo que saliera algo similar al video.

Uno de los inconvenientes finales es que el vídeo es de hace 3 años lo cual puede parecer poco pero tecnológicamente hablando es mucho desfase y desactualización así que para conectar flutter con “Firebase” tuvimos que buscar otro video que estuviera más actualizado para poder guiarnos correctamente.

[https://youtu.be/ZgiW\\_IUeFII?si=Wlz2HmxI60Gn4-Dm](https://youtu.be/ZgiW_IUeFII?si=Wlz2HmxI60Gn4-Dm)



# CRUD



Flutter



Firebase

Parte #1 - Conexión con **FlutterFire**

Sin embargo, en ningún video se mostraba como instalar flutterfire en el proyecto así que nos daba 1 problema, no mostraba nada y en consola se mostraba el error: "Error: Assertion failed:

```
file:///C:/Users/carlo/AppData/Local/Pub/Cache/hosted/pub.dev/firebase_core_web-2.16.0/lib/src/firebase_core_web.dart:273:11
```

```
options != null
```

```
"FirebaseOptions cannot be null when creating the default app."
```

```
dart-sdk/lib/internal/js_dev_runtime/private/ddc_runtime/errors.dart 297:3 throw
```

```
dart-sdk/lib/_internal/js_dev_runtime/private/ddc_runtime/errors.dart 38:3
```

```
assertFailed
```

```
packages/firebase_core_web/src/firebase_core_web.dart 273:18
```

```
initializeApp
```

```
dart-sdk/lib/_internal/js_dev_runtime/patch/async_patch.dart 45:50
```

Que, en resumen, nos dice que hay un error de inserción y nosotros nos preguntábamos que habíamos hecho mal si seguíamos todo lo del video, pues que no instalamos flutterfire para eso primero tiene que instalar node js

<https://nodejs.org/en>

Luego descargar el CLI de flutter fire

<https://firebase.google.com/docs/cli?hl=es#install-cli-windows>

Y luego ejecutar el siguiente comando en consola normal

*“npm install -g firebase-tools”*

Lo siguiente a realizar es en el proyecto ejecutamos por consola “flutter pub add firebase\_core”

Luego # Install the CLI if not already done so

dart pub global activate flutterfire\_cli

Lo siguiente es cualquiera de los siguientes comandos:

*“Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass”*

*“Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned”*

Luego ejecutamos “flutterfire login” accedemos a nuestra cuenta de google y una vez hecho eso ejecutamos

Y por último “flutterfire configure” que nos permitirá acceder a la base de datos de nuestro proyecto.

Y por último en el main.dart tenemos que importar la librería

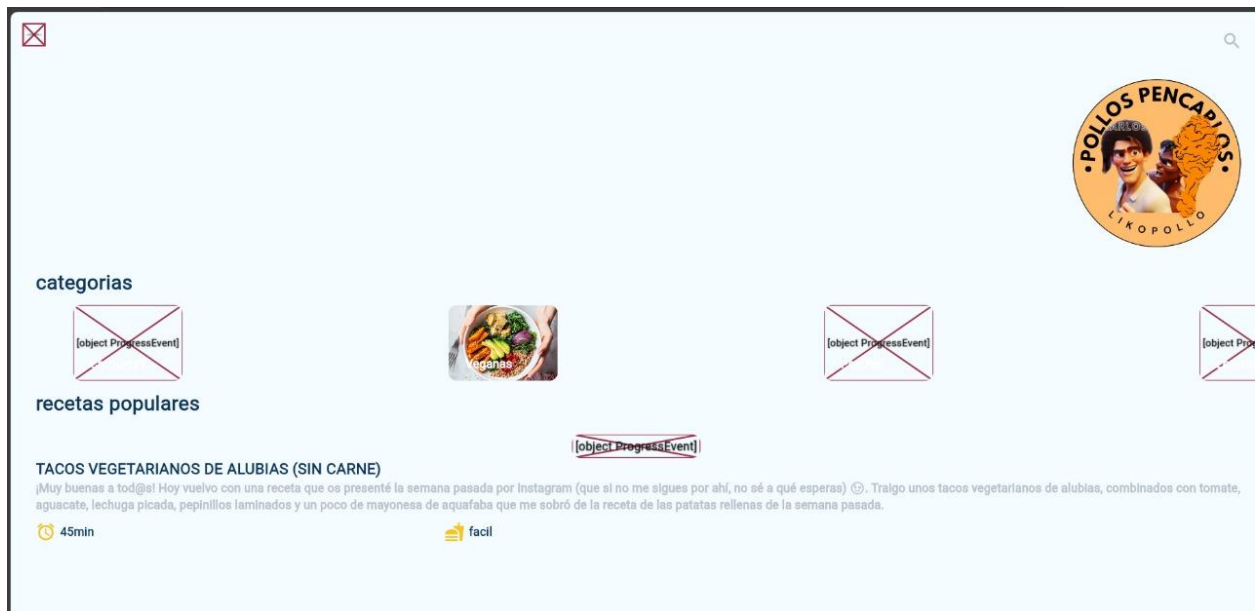
```
import 'package:firebase_core/firebase_core.dart';
```

```
import 'firebase_options.dart';
```

Y cambiar la manera de inicializar por:

```
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firebase.initializeApp(  
    options: DefaultFirebaseOptions.currentPlatform,  
  );  
  runApp(MyApp());  
}
```

Ya luego de eso si se mostraba todo con normalidad, o bueno en su mayoría ya que algunos assets no cargaban.



Todo parece indicar que el problema es de compatibilidad ya que las imágenes si están en la Base de Datos.

WhatsApp

recetas-app-flutter - Descripción

Recetario\_Documentación.docx

console.firebase.google.com/.../7b4

VPN

Firebase

Tus proyectos de Firebase

+

Agregar proyecto

recetas-app-flutter

recetas-app-flutter 67cb4

ios </>

Explorar un proyecto de demostración

Los proyectos de Firebase son contenedores para tus apps

Idioma

español (Latinoamérica)

Asistencia

Condiciones

Política de Privacidad

recetas-app-flutter

Plan Spark

5 apps

Recetario

recetario (android)

recetario (ios)

+

Gemini en Firebase

Obtén respuestas a preguntas sobre productos, casos de uso y funciones de Firebase. Genera código para el desarrollo y acorta el proceso de solución de problemas con estadísticas nuevas gracias a una interfaz de chat de lenguaje natural directamente en Firebase console.

Probar sin costo

Compilación

Firestore

Operaciones de lectura (actual)

339

Operaciones de escritura (actual)

18

Actualizar

