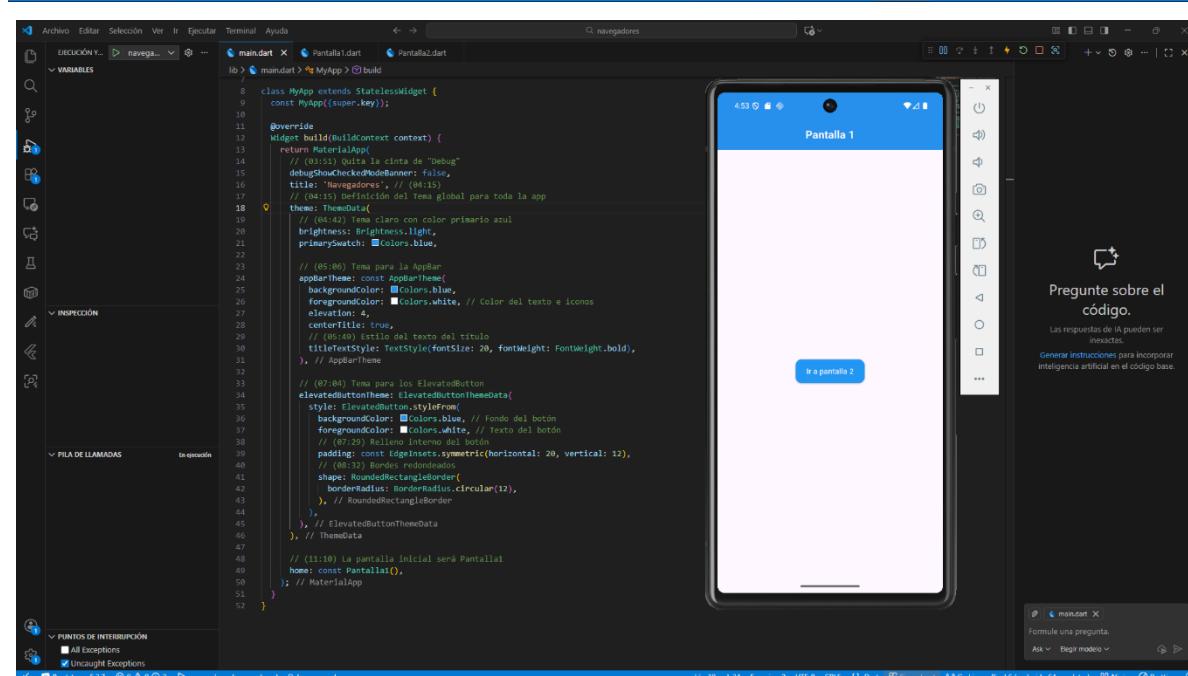
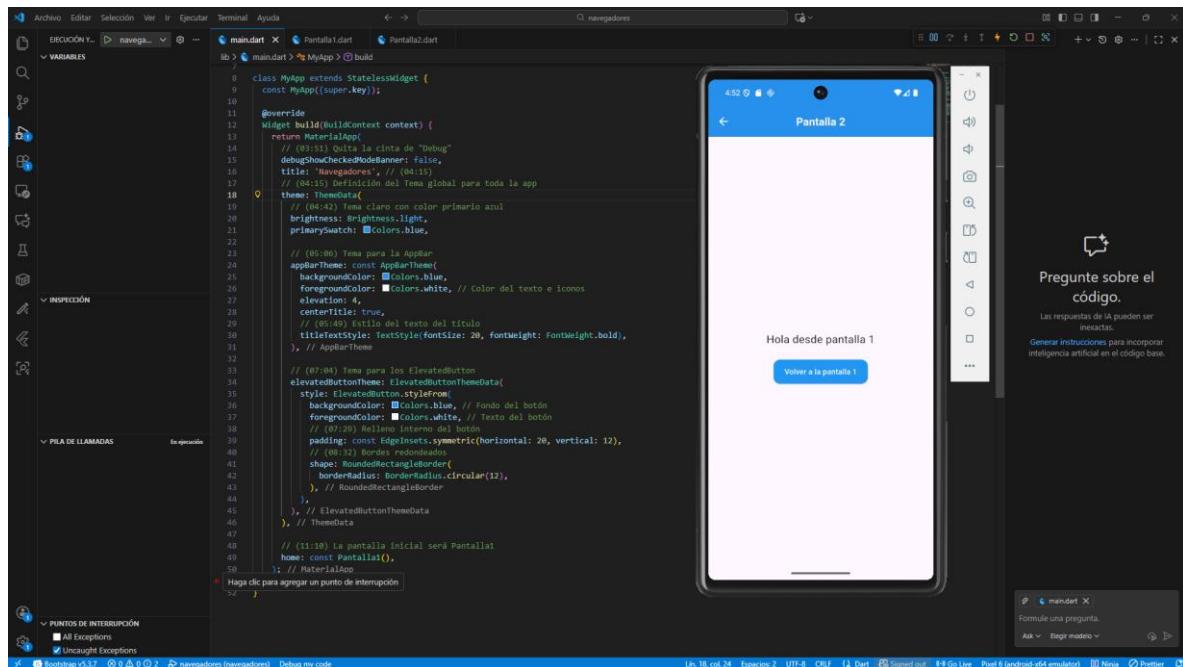


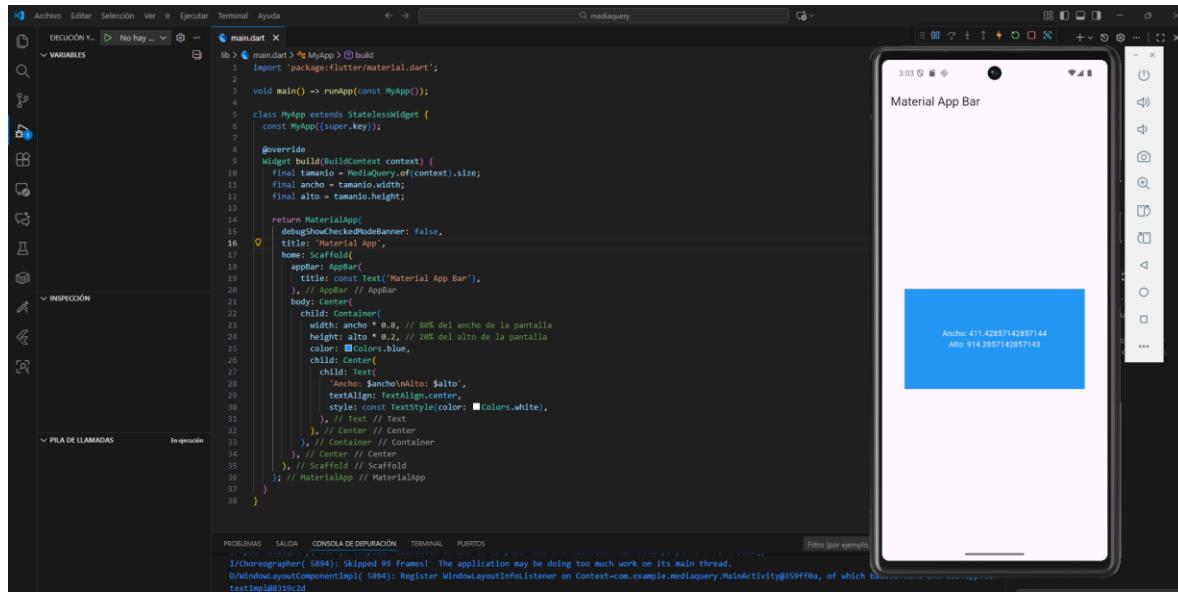
# CURSO DE FLUTTER-DART DESDE CERO | ENVIAR DATOS ENTRE PANTALLAS (video 52)

Se implementó el envío de datos entre pantallas en Flutter, donde la Pantalla 1 envía un mensaje "Hola desde pantalla 1" a la Pantalla 2 mediante el constructor de la pantalla destino durante la navegación. La Pantalla 2 recibe este dato como un parámetro requerido y lo muestra en su interfaz, demostrando el patrón fundamental de comunicación entre pantallas en Flutter a través del paso de parámetros durante la transición entre vistas.



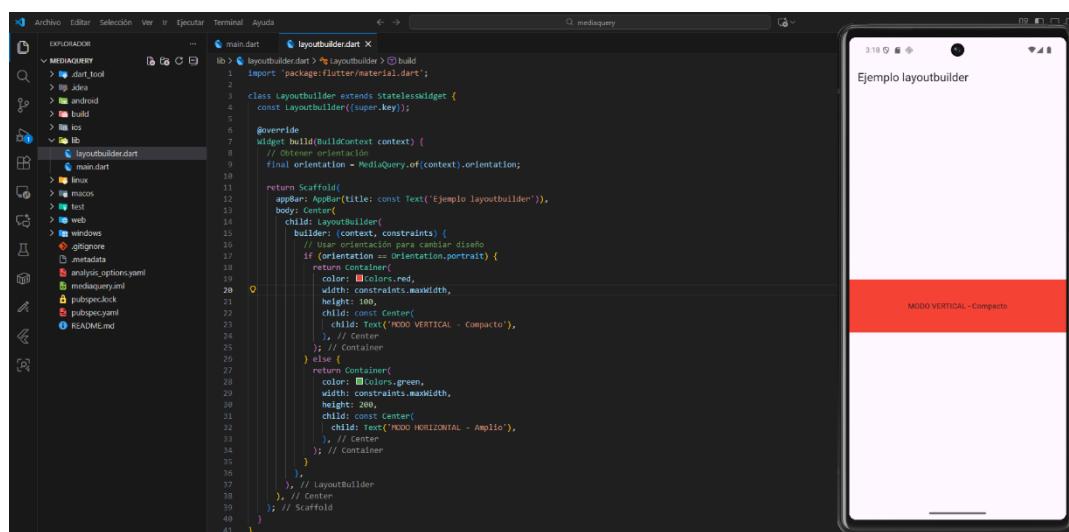
## CURSO DE FLUTTER-DART DESDE CERO | QUERYBUILDER (video 53)

En el código se implementó un diseño responsive utilizando MediaQuery como herramienta principal. Se capturaron las dimensiones totales de la pantalla del dispositivo mediante MediaQuery.of(context).size, obteniendo el ancho y alto reales. Con estos valores se creó un contenedor azul que ocupa el 80% del ancho total y el 20% del alto total de la pantalla.



## CURSO DE FLUTTER-DART DESDE CERO | LAYOUTBUILDER (video 54)

Ahora se implementó layoutbuilder para crear diseños adaptativos que responden al espacio disponible. El LayoutBuilder detecta el ancho máximo del contenedor padre y según si es menor o mayor a 400 píxeles, muestra un diseño compacto (rojo) o amplio (verde), cambiando automáticamente el layout cuando la pantalla se redimensiona.



```

lib > layoutbuilder.dart > LayoutBuilder > build
1 import 'package:flutter/material.dart';
2
3 class LayoutBuilder extends StatelessWidget {
4   const LayoutBuilder({super.key});
5
6   @override
7   Widget build(BuildContext context) {
8     // Obtener orientación
9     final orientation = MediaQuery.of(context).orientation;
10
11    return Scaffold(
12      appBar: AppBar(title: const Text('Ejemplo layoutbuilder')),
13      body: Center(
14        child: LayoutBuilder(
15          builder: (context, constraints) {
16            // Usar orientación para cambiar diseño
17            if (orientation == Orientation.portrait) {
18              return Container(
19                color: Colors.red,
20                width: constraints.maxWidth,
21                height: 100,
22                child: const Center(
23                  child: Text('MODO VERTICAL - Compacto'),
24                ), // Center
25              ); // Container
26            } else {
27              return Container(
28                color: Colors.green,
29                width: constraints.maxWidth,
30                height: 200,
31                child: const Center(
32                  child: Text('MODO HORIZONTAL - Amplio'),
33                ), // Center
34              ); // Container
35            }
36          },
37        ), // LayoutBuilder
38      ), // center
39    ); // Scaffold
40  }
41}

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

I/ImeTracker( 5894): com.example.mediaquery:3c7f38aa: onCancelled at PHASE\_CLIENT\_ALREADY\_HIDDEN  
D/InsetsController( 5894): hide(ime), fromIme=false  
I/ImeTracker( 5894): com.example.mediaquery:fee01c23: onCancelled at PHASE\_CLIENT\_ALREADY\_HIDDEN  
D/InsetsController( 5894): hide(ime), fromIme=false  
I/ImeTracker( 5894): com.example.mediaquery:da9ce0db: onCancelled at PHASE\_CLIENT\_ALREADY\_HIDDEN

Lín. 20, col. 45 Espacios: 2 UTF-8 CRLF { Dart Signed out 8 Go Live Pixel 6 (android-x64 emulator) 0 Ninja ⚡ Prettier

## CURSO DE FLUTTER-DART DESDE CERO | WIDGETS RESPONSIVOS, EXPANDED (video 55)

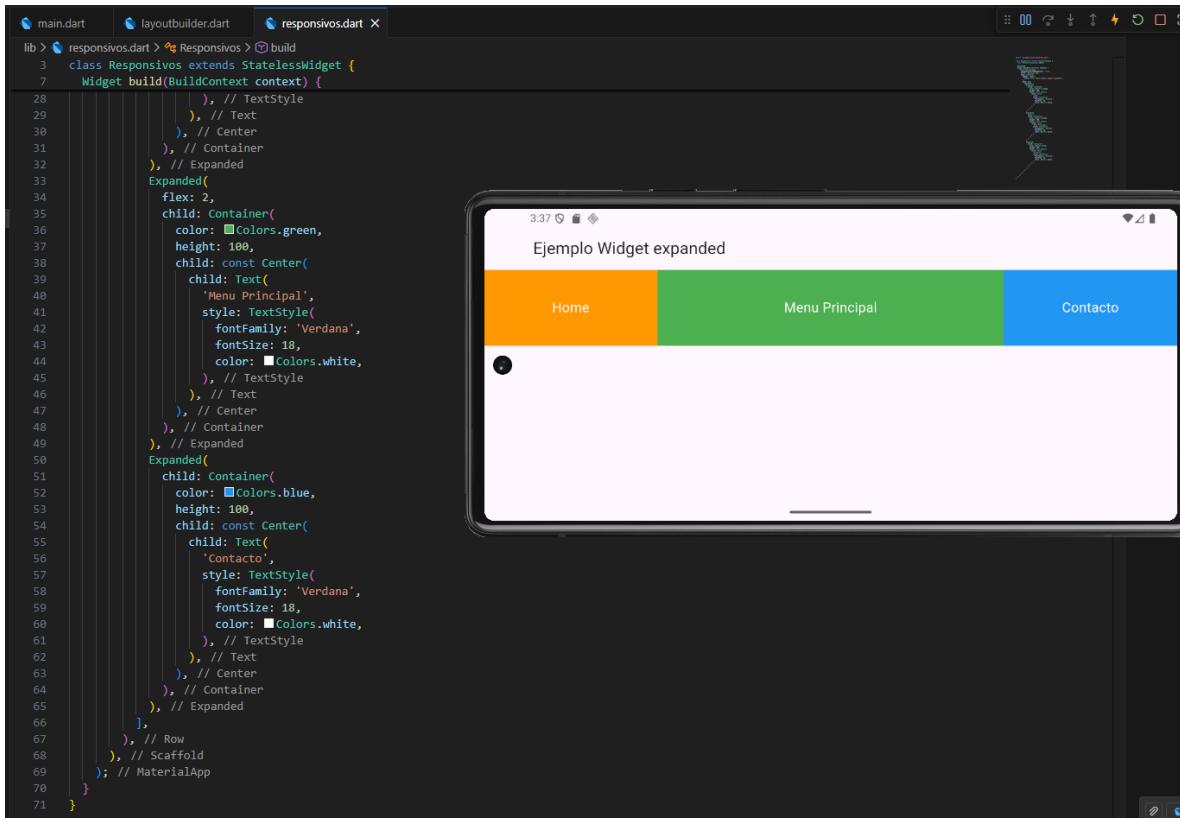
Se implementaron widgets responsivos usando expanded para crear un layout flexible. tres contenedores en una fila se distribuyen el espacio disponible: el primero y tercero ocupan 1 parte cada uno (flex: 1 implícito), mientras el central ocupa 2 partes (flex: 2), haciendo que el "menu principal" sea el doble de ancho que "home" y "contacto", adaptándose automáticamente al ancho de pantalla.

```

EXPLORADOR
> MEDIAQUERY
> dart_tool
> idea
> android
> build
> ios
> lib
  > layoutbuilder.dart
  main.dart
  responsive.dart
  responsive.dart
> linux
> macos
> test
> web
> windows
> gitignore
  metadata
> analysis_options.yaml
  mediquery.yaml
  pubspec.lock
  pubspec.yaml
  README.md

main.dart > responsive.dart > Responsivos > build
1 import 'package:flutter/material.dart';
2
3 class Responsivos extends StatelessWidget {
4   const Responsivos({super.key});
5
6   @override
7   Widget build(BuildContext context) {
8     return MaterialApp(
9       debugShowCheckedModeBanner: false,
10       title: 'Material App',
11       home: Scaffold(
12         appBar: AppBar(
13           title: const Text('Ejemplo Widget expanded'),
14         ), // AppBar
15         body: Row(
16           children: [
17             Expanded(
18               child: Container(
19                 color: Colors.orange,
20                 height: 100,
21                 child: Center(
22                   child: Text(
23                     'Home',
24                     style: TextStyle(
25                       fontFamily: 'Verdana',
26                       fontSize: 18,
27                       color: Colors.white,
28                     ), // TextStyle
29                   ), // Center
30               ), // Expanded
31             ), // Expanded
32             Expanded(
33               flex: 2,
34               child: Container(
35                 color: Colors.green,
36                 height: 100,
37                 child: Center(
38                   child: Text(
39                     'Menu Principal',
40                     style: TextStyle(
41                       fontFamily: 'Verdana',
42                     ), // TextStyle
43                   ), // Center
44               ), // Container
45             ), // Container
46           ], // children
47         ), // Row
48       ), // Scaffold
49     ); // MaterialApp
50   }
51}

```



## CURSO DE FLUTTER-DART DESDE CERO | WIDGETS LISTVIEW (video 56)

Se implementó una lista desplazable usando ListView.builder que muestra un catálogo de frutas. Cada elemento de la lista se construye con ListTile incluyendo un ícono de compras, número de fruta, nombre de la fruta y un ícono de flecha, donde al hacer tap en cualquier fruta se muestra un Snackbar con el nombre de la fruta seleccionada, demostrando la creación de listas dinámicas.

