

## Guía de Laboratorio 3

### Responsividad en Flutter (Teléfono y Tablet)

#### 1. Objetivo General

Implementar una aplicación en Flutter que adapte su diseño de forma automática al tamaño de pantalla del dispositivo, diferenciando entre **teléfonos y tabletas**, aplicando principios de **diseño responsivo**.

#### 2. Objetivos Específicos

- Comprender el concepto de **diseño responsivo** en el desarrollo móvil.
- Identificar y aplicar **breakpoints** usando MediaQuery.
- Usar **LayoutBuilder** para construir interfaces que se adapten dinámicamente.
- Implementar una **grilla adaptable** que cambie el número de columnas según el tamaño de pantalla.
- Ajustar tipografía, espaciados y distribución de widgets según el dispositivo.

#### 3. Marco Teórico

##### Diseño Responsivo

El **diseño responsivo** consiste en crear interfaces que se adapten a diferentes tamaños de pantalla, manteniendo una buena experiencia de usuario en cualquier dispositivo (teléfono, tablet o escritorio).

En Flutter, esto se logra utilizando herramientas que permiten **detectar el tamaño del dispositivo** y **reorganizar los widgets** de forma automática.

##### Herramientas principales

###### MediaQuery

Permite acceder a las propiedades del dispositivo, como:

- `size.width`: ancho de la pantalla.
- `size.height`: alto de la pantalla.
- `orientation`: orientación (vertical u horizontal).

Ejemplo:

```
final ancho = MediaQuery.of(context).size.width;
```

```
final esTablet = ancho >= 600;
```

**LayoutBuilder**

Permite construir widgets adaptativos dentro de un contenedor según su tamaño disponible.

LayoutBuilder(

```
builder: (context, constraints) {  
  if (constraints.maxWidth < 600) {  
    return VistaMovil();  
  } else {  
    return VistaTablet();  
  }  
},  
);
```

**GridView Adaptable**

Con `SliverGridDelegateWithFixedCrossAxisCount` podemos ajustar el número de columnas (`crossAxisCount`) según el tamaño disponible.

Ejemplo:

GridView.builder(

```
gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(  
  crossAxisCount: esTablet ? 3 : 2,  
),  
);
```

**Principios de diseño aplicados**

Principio	Descripción
<b>Claridad visual</b>	Se muestran los mismos elementos, pero reorganizados.
<b>Consistencia</b>	Colores y tipografía se mantienen entre dispositivos.
<b>Adaptabilidad</b>	La UI reacciona dinámicamente al espacio disponible.

**4. Materiales y requisitos**

- Tener instalado **Flutter** y **Visual Studio Code**.
- Emulador Android o iOS configurado.
- SDK de Flutter actualizado.

**5. Desarrollo del laboratorio**

Copia el siguiente código en `lib/main.dart`

```
import 'package:flutter/material.dart';

void main() => runApp(const ResponsiveDemoApp());

class ResponsiveDemoApp extends StatelessWidget {
  const ResponsiveDemoApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Demo Responsiva (Teléfono/Tablet)',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(useMaterial3: true, colorSchemeSeed: Colors.teal),
      home: const ResponsiveHome(),
    );
  }
}

class ResponsiveHome extends StatelessWidget {
  const ResponsiveHome({super.key});

  @override
  Widget build(BuildContext context) {
    // Detectamos ancho del dispositivo
    final double width = MediaQuery.of(context).size.width;
    final bool isTablet = width >= 600; // breakpoint entre teléfono y tablet

    return Scaffold(
      appBar: AppBar(
        title: Text(isTablet ? 'Vista Tablet' : 'Vista Teléfono'),
        centerTitle: true,
      ),
      body: Padding(
        padding: EdgeInsets.all(isTablet ? 24 : 12),
        child: LayoutBuilder(
          builder: (context, constraints) {
            return isTablet
              ? _buildTabletLayout(context)
              : _buildPhoneLayout(context);
          },
        ),
      ),
    );
  }

  /// Diseño para pantallas de teléfono
  Widget _buildPhoneLayout(BuildContext context) {
```

```
    return Column(
      children: [
        _HeaderSection(isTablet: false),
        const SizedBox(height: 12),
        Expanded(child: _GridSection(isTablet: false)),
      ],
    );
  }

  /// Diseño para pantallas de tablet
  Widget _buildTabletLayout(BuildContext context) {
    return Row(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Expanded(flex: 1, child: _HeaderSection(isTablet: true)),
        const SizedBox(width: 20),
        Expanded(flex: 2, child: _GridSection(isTablet: true)),
      ],
    );
  }
}

/// Encabezado adaptable (texto y tamaño de fuente)
class _HeaderSection extends StatelessWidget {
  final bool isTablet;
  const _HeaderSection({required this.isTablet});

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: EdgeInsets.all(isTablet ? 24 : 12),
      decoration: BoxDecoration(
        color: Colors.teal.shade100,
        borderRadius: BorderRadius.circular(12),
      ),
      child: Column(
        crossAxisAlignment: isTablet
          ? CrossAxisAlignment.start
          : CrossAxisAlignment.center,
        children: [
          Text(
            'Dashboard de Ventas',
            style: TextStyle(
              fontSize: isTablet ? 26 : 20,
              fontWeight: FontWeight.bold,
            ),
          ),
        ],
      ),
    );
  }
}
```

```
        const SizedBox(height: 8),
        Text(
          'Demostración de diseño responsivo con Flutter.',
          textAlign: isTablet ? TextAlign.left : TextAlign.center,
          style: TextStyle(fontSize: isTablet ? 18 : 14),
        ),
      ],
    ),
  );
}
}

/// Sección de tarjetas adaptables con GridView
class _GridSection extends StatelessWidget {
  final bool isTablet;
  const _GridSection({required this.isTablet});

  @override
  Widget build(BuildContext context) {
    final crossCount = isTablet ? 3 : 2;

    return GridView.builder(
      gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: crossCount,
        crossAxisSpacing: 10,
        mainAxisSpacing: 10,
        childAspectRatio: isTablet ? 4 / 3 : 1,
      ),
      itemCount: 6,
      itemBuilder: (context, index) {
        return Card(
          elevation: 2,
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(12),
          ),
          color: Colors.teal.shade50,
          child: Center(
            child: Text(
              'Tarjeta ${index + 1}',
              style: TextStyle(
                fontSize: isTablet ? 18 : 14,
                fontWeight: FontWeight.w600,
                color: Colors.teal.shade900,
              ),
            ),
          ),
        );
      },
    );
  }
};
```

```

    },
  );
}
}

```

## 6. Explicación de los bloques de código

Bloque	Descripción
main()	Punto de entrada de la aplicación.
ResponsiveDemoApp	Define el tema y la estructura base de la app.
MediaQuery.of(context).size.width	Obtiene el ancho de pantalla para decidir si es teléfono o tablet.
LayoutBuilder	Evalúa el espacio disponible y selecciona el layout apropiado.
_buildPhoneLayout()	Diseño vertical con encabezado y grilla de 2 columnas.
_buildTabletLayout()	Diseño horizontal con encabezado lateral y grilla de 3 columnas.
_HeaderSection	Encabezado adaptable (alineación, tamaño de texto y espaciado).
_GridSection	Grilla de tarjetas que cambia el número de columnas según isTablet.

## 7. Resultados esperados

Dispositivo	Distribución esperada
<b>Teléfono</b>	Encabezado centrado + grilla de 2 columnas
<b>Tablet</b>	Encabezado a la izquierda + grilla de 3 columnas

## 8. Trabajo independiente

Desarrolla una aplicación Flutter titulada **“Mi Perfil Responsivo”** que cambie su diseño y distribución de elementos dependiendo del dispositivo en el que se ejecute (teléfono o tablet).

La aplicación debe incluir:

1. Una **sección de encabezado** con foto de perfil (puedes utilizar una fotografía de internet), nombre y profesión.
2. Una **sección de información personal** (edad, correo, ubicación, etc.).
3. Una **sección de habilidades o intereses** representada con tarjetas.

La aplicación debe presentar los elementos de la siguiente manera:

Vista Teléfono

- El diseño debe mostrarse en **columna (vertical)**.
- El encabezado debe estar **centrado**.
- Las tarjetas de habilidades deben mostrarse en **2 columnas**.
- Tipografía y paddings más compactos (fontSize entre 14 y 18).

Vista Tablet

- El diseño debe mostrarse en **fila (horizontal)** con el encabezado a la izquierda.
- Las tarjetas de habilidades deben mostrarse en **3 columnas**.
- Tipografía más grande (fontSize entre 18 y 24).
- Márgenes amplios (EdgeInsets.all(24)).

Toma en cuenta las siguientes restricciones técnicas

- Utiliza **MediaQuery** para obtener el ancho de pantalla.
- Define un **breakpoint en 600 píxeles** para diferenciar entre teléfono y tablet.
- Usa **LayoutBuilder** o condicionales if (isTablet) para cambiar la estructura.
- Aplica **GridView.builder** para la sección de tarjetas.

Envía INBOX mediante WhatsApp dos capturas de pantalla de tu aplicación en el emulador, toma en cuenta su sentido vertical y horizontal, los datos mostrados en el perfil deben ser los tuyos.