

# Lección: Cambiar el Color de Fondo de una Pantalla en Flutter

## Objetivos de la Lección

- Comprender cómo cambiar el color de fondo de una pantalla en Flutter utilizando el widget **Scaffold**.
- Aprender a utilizar colores predeterminados y personalizados en Flutter.
- Implementar ejemplos prácticos para establecer colores de fondo en una aplicación sencilla.

## Introducción

En Flutter, el widget **Scaffold** proporciona una propiedad llamada `backgroundColor` que permite cambiar el color de fondo de la pantalla. Esto es útil para personalizar la apariencia de la aplicación y adaptarla a los requisitos de diseño. Flutter ofrece una amplia gama de colores predeterminados en la clase `Colors`, así como la opción de definir colores personalizados.

## Cambiar el Color de Fondo Usando Colores Predeterminados

La clase `Colors` de Flutter contiene una serie de colores predefinidos que se pueden usar para cambiar rápidamente el fondo de la pantalla. Estos colores incluyen tonos básicos como `Colors.blue`, `Colors.red`, `Colors.green`, entre otros.

### Ejemplo Básico: Cambiar el Fondo a Azul

En este ejemplo, cambiaremos el color de fondo de la pantalla a azul utilizando la propiedad `backgroundColor` en el widget **Scaffold**.

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

```

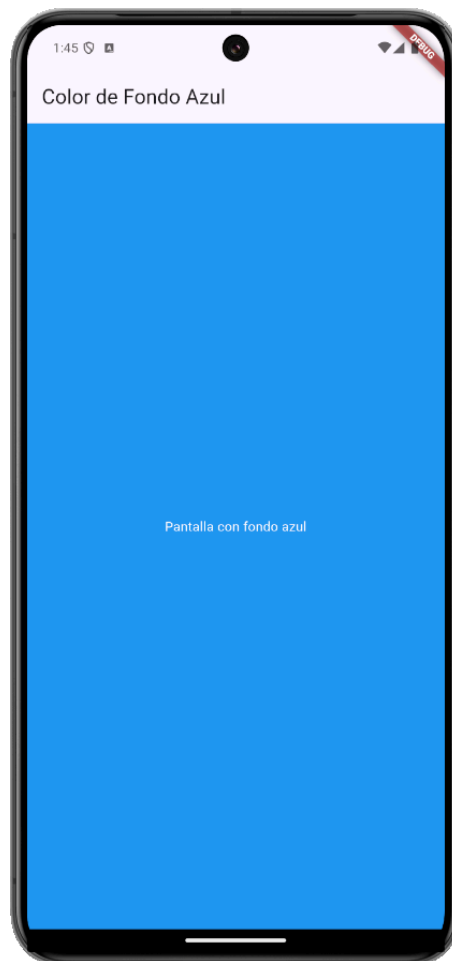
void main() {
    runApp(MyApp());
}

class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            home: Scaffold(
                backgroundColor: Colors.blue, // Cambia el color de
fondo a azul
                appBar: AppBar(
                    title: Text('Color de Fondo Azul'),
                ),
                body: Center(
                    child: Text(
                        'Pantalla con fondo azul',
                        style: TextStyle(color: Colors.white), // Cambia el
color del texto a blanco
                    ),
                ),
            ),
        );
    }
}

```

## Explicación del Código:

1. **backgroundColor: Colors.blue:** Cambia el color de fondo de la pantalla a azul.
2. **TextStyle(color: Colors.white):** Cambia el color del texto a blanco para que contraste con el fondo azul.



Captura de la pantalla del ejemplo completo presentando todas las opciones para cambio de color de una página o pantalla en Flutter.

Creado por Javier A. Dastas (2024)

## Cambiar el Color de Fondo Usando Tonos y Opacidad

Flutter permite personalizar los tonos de los colores usando valores de `.shade`, como `Colors.blue.shade200`, `Colors.blue.shade900`, etc. También se puede agregar opacidad usando el método `withOpacity`.

## Ejemplo: Cambiar el Fondo a un Tono Claro de Verde

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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.green.shade200, // Fondo en un
        // tono claro de verde
        appBar: AppBar(
          title: Text('Color de Fondo Verde Claro'),
        ),
        body: Center(
          child: Text(
            'Pantalla con fondo verde claro',
            style: TextStyle(color: Colors.black),
          ),
        ),
      ),
    );
  }
}
```

## Explicación del Código:

- **Colors.green.shade200**: Usa un tono claro de verde para el fondo de la pantalla, lo que permite una mayor variedad de colores personalizados.
- **TextStyle(color: Colors.black)**: Cambia el color del texto a negro para crear un buen contraste con el fondo claro.



Captura de la pantalla del ejemplo completo presentando todas las opciones para cambio de color de una página o pantalla en Flutter.

Creado por Javier A. Dastas (2024)

## Personalizar el Color de Fondo Usando Valores ARGB

Además de los colores predefinidos, Flutter permite definir colores personalizados usando el método `Color.fromARGB` o `Color.fromRGBO`, especificando los valores

de transparencia, rojo, verde y azul. Esto permite una personalización completa de colores, incluyendo niveles de transparencia.

### Ejemplo: Usar un Color Personalizado con Transparencia

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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Color.fromARGB(255, 100, 150, 200), //
Color personalizado
        appBar: AppBar(
          title: Text('Color de Fondo Personalizado'),
        ),
        body: Center(
          child: Text(
            'Pantalla con fondo personalizado',
            style: TextStyle(color: Colors.white),
          ),
        ),
      ),
    );
  }
};
```

```
}  
}
```

### Explicación del Código:

- **Color.fromARGB(255, 100, 150, 200)**: Define un color personalizado con valores específicos de rojo, verde y azul (RGB), y un valor de opacidad de 255 (sin transparencia).
- **TextStyle(color: Colors.white)**: El color del texto se ajusta a blanco para mejorar la visibilidad sobre el fondo personalizado.

### Usar Opacidad con withOpacity

El método `withOpacity` permite aplicar un nivel de transparencia a un color ya definido en Flutter. El valor de opacidad va de `0.0` (totalmente transparente) a `1.0` (opaco).

### Ejemplo: Fondo Azul con Transparencia

```
import 'package:flutter/material.dart';  
  
void main() {  
  runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: Scaffold(  

```

```

        backgroundColor: Colors.blue.withOpacity(0.5), // Fondo
azul con opacidad
        appBar: AppBar(
            title: Text('Color de Fondo con Opacidad'),
        ),
        body: Center(
            child: Text(
                'Pantalla con fondo semitransparente',
                style: TextStyle(color: Colors.black),
            ),
        ),
    ),
);
}
}

```

Explicación del Código:

- **Colors.blue.withOpacity(0.5):** Aplica una transparencia del 50% al color azul, haciendo que el fondo sea semitransparente.
- **TextStyle(color: Colors.black):** Cambia el color del texto a negro para que se vea claramente sobre el fondo semitransparente.

## Ejemplo Completo: Pantalla con Fondo, Tono y Opacidad

En este ejemplo, combinamos varios conceptos para mostrar cómo personalizar completamente el color de fondo de la pantalla, utilizando un tono personalizado, opacidad y contraste de texto.

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



```

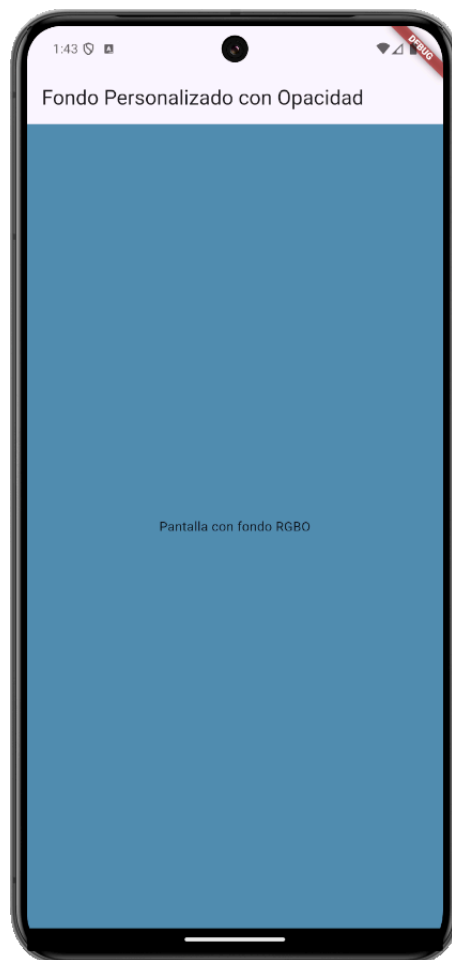
void main() {
    runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Color.fromRGBO(120, 200, 255, 0.7), //
Fondo personalizado con opacidad
        appBar: AppBar(
          title: Text('Fondo Personalizado con Opacidad'),
        ),
        body: Center(
          child: Text(
            'Pantalla con fondo RGB0',
            style: TextStyle(color: Colors.black87), // Texto en
un tono oscuro de negro
          ),
        ),
      ),
    );
  }
}

```

## Explicación del Código:

- **Color.fromRGBO(120, 200, 255, 0.7):** Define un color personalizado utilizando RGB y un valor de opacidad de 0.7, lo que proporciona un fondo de color azul claro semitransparente.
- **TextStyle(color: Colors.black87):** Ajusta el color del texto a un negro suave para contrastar con el fondo semitransparente.



Captura de la pantalla del ejemplo completo presentando todas las opciones para cambio de color de una página o pantalla en Flutter.

Creado por Javier A. Dastas (2024)

## Resumen de la Lección

En esta lección, hemos aprendido a cambiar el color de fondo de una pantalla en Flutter utilizando la propiedad `backgroundColor` del widget **Scaffold**. Hemos explorado varias maneras de establecer el color de fondo:

- Usando colores predefinidos de la clase `Colors`.
- Personalizando el color utilizando tonos y opacidad (`shade` y `withOpacity`).
- Definiendo colores personalizados con `Color.fromARGB` y `Color.fromRGB0`.

Con estos métodos, podemos adaptar el fondo de la pantalla para ajustarlo a los requisitos estéticos y de diseño de la aplicación.

## Actividad de la Lección

Estas actividades te permitirán aplicar los conceptos aprendidos sobre el cambio de color de fondo en Flutter y experimentar con diferentes combinaciones de tonos y opacidad.

### Instrucciones:

1. Crea una aplicación en Flutter que muestre un fondo de pantalla en un tono personalizado de color (usando `Color.fromARGB`), y ajusta el color de texto para que contraste con el fondo.
2. Implementa un color de fondo usando `Colors.red.withOpacity(0.3)`, y ajusta el color del texto para que sea visible y estéticamente agradable sobre el fondo semitransparente.
3. Entrega un documento en formato PDF con copia de tu código y copia de imágenes o capturas de pantalla demostrando que tu código funciona.