

Módulo 4: Elementos Visuales y Flujo de Interacción entre Pantallas

Lección: Selección de Valores con Rango en Flutter: Uso del Slider

Objetivos de la Lección

- Comprender el propósito del widget `Slider` para seleccionar valores dentro de un rango definido.
- Aprender a personalizar propiedades de rango, divisiones y pasos en el widget.
- Conocer las funcionalidades y propiedades principales del `Slider`.
- Aplicar un ejemplo práctico de uso del widget en Flutter.

Introducción de la Lección

El widget `Slider` en Flutter permite a los usuarios seleccionar un valor de un rango definido, como volumen, brillo o nivel de configuración. Es muy útil para aplicaciones que requieren ajustes controlados por el usuario en tiempo real. Al mover el control deslizante, el usuario puede seleccionar con precisión un valor dentro del rango establecido. En esta lección, explicaremos las propiedades y funcionalidades principales del `Slider`, y aprenderemos a configurar divisiones, rango y pasos.

Propiedades Principales del `Slider`

- **value**: Representa el valor actual del `Slider`.
- **min** y **max**: Definen el rango de valores del `Slider`.

- **onChanged:** Callback que se ejecuta cuando el usuario mueve el Slider, actualizando el valor en tiempo real.
- **divisions:** Divide el Slider en segmentos discretos, permitiendo al usuario moverse en intervalos específicos.
- **label:** Muestra el valor actual en un cuadro flotante cuando se mueve el Slider.

Ejemplo Simple de Slider:

En este ejemplo, configuramos un Slider que permite al usuario seleccionar un nivel de brillo en una escala de 0 a 100. Al mover el control deslizante, el valor del brillo se muestra en pantalla.

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: SliderExample(),
    );
  }
}

class SliderExample extends StatefulWidget {
  @override
  _SliderExampleState createState() => _SliderExampleState();
}
```

```

class _SliderExampleState extends State<SliderExample> {
    double _brightnessLevel = 50;

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: Text('Ejemplo de Slider'),
            ),
            body: Padding(
                padding: EdgeInsets.all(16.0),
                child: Column(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: <Widget>[
                        Text(
                            'Nivel de Brillo: ${_brightnessLevel.toInt()}',
                            style: TextStyle(fontSize: 20),
                        ),
                        Slider(
                            value: _brightnessLevel,
                            min: 0,
                            max: 100,
                            divisions: 10,
                            label: _brightnessLevel.toInt().toString(),
                            onChanged: (double value) {
                                setState(() {
                                    _brightnessLevel = value;
                                });
                            },
                        ),
                    ],
                ),
            ),
        );
    }
}

```

```
        ),  
    ],  
    ),  
    ),  
    );  
}  
}
```

En este ejemplo:

- **value**: Representa el nivel de brillo actual, almacenado en `_brightnessLevel`.
- **min** y **max**: Definen el rango del `Slider` de 0 a 100.
- **divisions**: Divide el `Slider` en 10 segmentos, permitiendo ajustes en pasos de 10 unidades.
- **label**: Muestra el valor actual del `Slider` como un cuadro flotante.

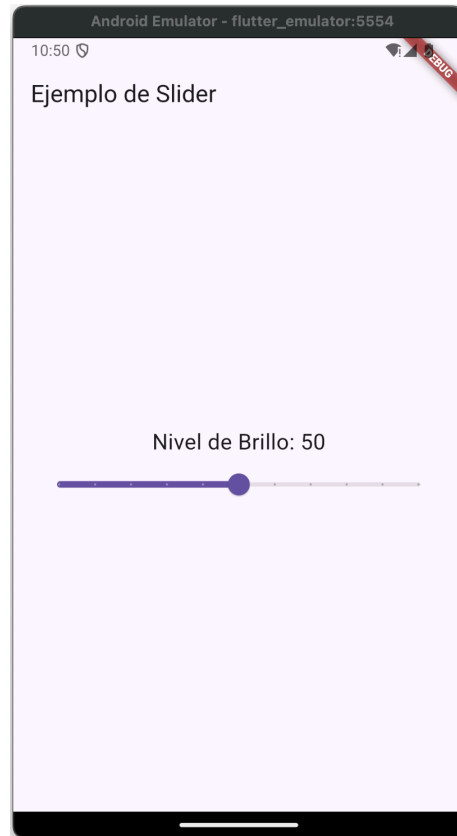


Imagen de la pantalla que incluye el Widget Slider.

Creado por Javier A. Dastas (2024)

Relación con Otros Conceptos

El widget `Slider` es una herramienta útil para la personalización de configuraciones dentro de una aplicación. Al integrar `Slider` en el diseño UX/UI, los desarrolladores pueden ofrecer una experiencia de usuario amigable y sencilla para realizar ajustes en tiempo real, como volumen, brillo o niveles de configuración. Este widget es esencial en aplicaciones que buscan interacción y control intuitivo para ajustes específicos.

Resumen de la Lección

En esta lección, exploramos el widget `Slider` en Flutter, incluyendo sus propiedades y personalización. Aprendimos cómo configurar el rango de valores, divisiones y el valor

actual. Este widget es ideal para interfaces que requieren ajustes de valor en tiempo real, proporcionando un control deslizante fácil de usar.

Actividad de la Lección

En esta actividad, aplicaremos lo aprendido sobre el widget `Slider` para crear una aplicación que permita al usuario ajustar configuraciones como volumen o brillo. Esto ayudará a comprender cómo utilizar `Slider` para controlar valores en tiempo real dentro de un rango específico.

La actividad tiene dos partes:

1. Control de Volumen con `Slider`:

- Implementa un `Slider` que permita al usuario ajustar el volumen de 0 a 100.
- Muestra el nivel de volumen seleccionado en pantalla y actualízalo en tiempo real.
- Esta parte ayuda a reforzar el uso del `Slider` en la configuración de valores continuos dentro de un rango.

2. Ajuste de Brillo con Divisiones:

- Configura un `Slider` para ajustar el nivel de brillo de 0 a 100 con divisiones de 20 en 20.
- Al mover el `Slider`, el nivel de brillo debe actualizarse en pantalla, mostrando los pasos discretos definidos.
- Esta parte permite trabajar con divisiones y comprender cómo ajustar la precisión del `Slider`.

Entrega de la Actividad:

- Desarrolla un documento en formato PDF que incluya el código desarrollado para esta actividad, capturas de pantalla de la aplicación en ejecución, y una breve descripción de cómo el `Slider` mejora la experiencia del usuario al permitir la selección de valores de manera intuitiva.

Con esta actividad, se busca consolidar el conocimiento en la creación de interfaces interactivas que ofrezcan control en tiempo real de valores ajustables, mejorando la experiencia de usuario en aplicaciones Flutter.