

# Módulo 4: Interacción de Pantallas y Elementos para Captura de Datos

## Lección 2: Uso de Ventanas de Diálogo y Snackbar en Flutter

### Objetivos de la Lección

- Comprender los diferentes tipos de ventanas de diálogo en Flutter y sus usos.
- Conocer el uso del widget Snackbar en Flutter para mostrar notificaciones breves.
- Identificar los momentos adecuados para utilizar ventanas de diálogo o un snackbar según el contexto de la aplicación.
- Implementar ventanas de diálogo y snackbar utilizando botones para su invocación.

### Introducción de la Lección

En Flutter, los widgets de diálogo y snackbar son herramientas útiles para interactuar con los usuarios, ya sea solicitando confirmaciones, mostrando información o alertas, o notificando sobre acciones breves. Los diálogos son generalmente ventanas emergentes que requieren una interacción directa del usuario, mientras que los Snackbar son notificaciones efímeras que se muestran en la parte inferior de la pantalla.

En esta lección, aprenderás cómo usar ventanas de diálogo y el widget Snackbar, cuándo es apropiado usar cada uno y cómo implementarlos usando solo botones para invocarlos.

## Tipos de Ventanas de Diálogo en Flutter

**AlertDialog:** Es una ventana que generalmente presenta información importante o solicita la confirmación del usuario. Puede tener múltiples botones (e.g., aceptar, cancelar).

```
void _showAlertDialog(BuildContext context) {  
  showDialog(  
    context: context,  
    builder: (BuildContext context) {  
      return AlertDialog(  
        title: Text('Alert Dialog'),  
        content: Text('¿Estás seguro de que quieres  
continuar?'),  
        actions: <Widget>[  
          TextButton(  
            child: Text('Cancelar'),  
            onPressed: () {  
              Navigator.of(context).pop(); // Cierra el diálogo  
            },  
          ),  
          TextButton(  
            child: Text('Aceptar'),  
            onPressed: () {  
              Navigator.of(context).pop();  
              // Acción al aceptar  
            },  
          ),  
        ],  
      );  
    },  
  );  
}
```

```

        );
    },
);
}

```

**SimpleDialog:** Muestra una lista de opciones que los usuarios pueden elegir. Es útil cuando se necesita mostrar múltiples elecciones sin sobrecargar la pantalla principal.

```

void _showSimpleDialog(BuildContext context) {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return SimpleDialog(
        title: Text('Selecciona una opción'),
        children: <Widget>[
          SimpleDialogOption(
            onPressed: () {
              Navigator.pop(context);
            },
            child: Text('Opción 1'),
          ),
          SimpleDialogOption(
            onPressed: () {
              Navigator.pop(context);
            },
            child: Text('Opción 2'),
          ),
        ],
      );
    }
  );
}

```

```

    },
  );
}

```

**Dialog personalizado:** En Flutter puedes crear cualquier tipo de diálogo que se ajuste a tus necesidades personalizando el contenido y los widgets usados en la ventana.

```

void _showCustomDialog(BuildContext context) {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return Dialog(
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(10),
        ),
        child: Padding(
          padding: const EdgeInsets.all(20.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            children: <Widget>[
              Text('Diálogo Personalizado'),
              SizedBox(height: 20),
              TextButton(
                onPressed: () {
                  Navigator.of(context).pop();
                },
                child: Text('Cerrar'),
              ),
            ],
          ),
        ),
      );
    },
  );
}

```

```

        ),
    ),
);
},
);
}

```

## Snackbar en Flutter

Un **Snackbar** es una breve notificación que aparece en la parte inferior de la pantalla y desaparece automáticamente después de unos segundos o al ser descartado por el usuario.

```

void _showSnackBar(BuildContext context) {
    final snackBar = SnackBar(
        content: Text('Este es un SnackBar'),
        duration: Duration(seconds: 3), // Tiempo que el snackbar
estará visible
        action: SnackBarAction(
            label: 'Deshacer',
            onPressed: () {
                // Acción al hacer clic en 'Deshacer'
            },
        ),
    );
};
ScaffoldMessenger.of(context).showSnackBar(snackBar);
}

```

## ¿Cuándo Utilizar Ventanas de Diálogo y SnackBar?

- **AlertDialog:** Utiliza `AlertDialog` cuando necesitas que el usuario tome una decisión crítica o confirme una acción (e.g., eliminar un archivo, cerrar sesión).
- **SimpleDialog:** Es útil para proporcionar una lista de opciones cuando el número de elementos es manejable (e.g., seleccionar el método de pago).
- **Diálogo Personalizado:** Usa un diálogo personalizado cuando necesitas mostrar más información o un contenido específico que no se adapta bien a los diálogos estándar de Flutter.
- **SnackBar:** El `SnackBar` es ideal para notificaciones rápidas que no requieren una interacción directa del usuario (e.g., una notificación de éxito después de enviar un formulario).

## Ejemplos y Explicaciones Detalladas

### Botón para Mostrar un `AlertDialog`:

Este código crea un botón que muestra un `AlertDialog` al ser presionado.

```
ElevatedButton(  
  onPressed: () {  
    _showAlertDialog(context);  
  },  
  child: Text('Mostrar Alert Dialog'),  
)
```

## Botón para Mostrar un SimpleDialog:

Al presionar este botón, se abrirá un SimpleDialog con opciones.

```
ElevatedButton(  
  onPressed: () {  
    _showSimpleDialog(context);  
  },  
  child: Text('Mostrar Simple Dialog'),  
)
```

## Botón para Mostrar un SnackBar:

Este botón mostrará un snackbar cuando se presione.

```
ElevatedButton(  
  onPressed: () {  
    _showSnackBar(context);  
  },  
  child: Text('Mostrar SnackBar'),  
)
```

## Ejemplos y Explicaciones Detalladas

A continuación, se presenta un ejemplo simple donde se implementan cuatro botones en una aplicación Flutter. Cada botón muestra un diálogo diferente cuando es presionado: AlertDialog, SimpleDialog, un diálogo personalizado y un SnackBar.

```

import 'package:flutter/material.dart';

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

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

class ButtonScreen extends StatefulWidget {
  @override
  _ButtonScreenState createState() => _ButtonScreenState();
}

class _ButtonScreenState extends State<ButtonScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Diálogos en Flutter'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,

```



```

children: <Widget>[
  ElevatedButton(
    onPressed: () {
      showDialog(
        context: context,
        builder: (BuildContext context) {
          return AlertDialog(
            title: Text('AlertDialog'),
            content: Text('Este es un AlertDialog.'),
            actions: <Widget>[
              TextButton(
                child: Text('OK'),
                onPressed: () {
                  Navigator.of(context).pop();
                },
              ),
            ],
          );
        },
      );
    },
    child: Text('Mostrar AlertDialog'),
  ),
  SizedBox(height: 10),
  ElevatedButton(
    onPressed: () {
      showDialog(
        context: context,
        builder: (BuildContext context) {
          return SimpleDialog(

```



```

        child: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: <Widget>[
            Text('Diálogo Personalizado'),
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: () {
                Navigator.of(context).pop();
              },
              child: Text('Cerrar'),
            ),
          ],
        ),
      ),
    );
  },
);
},
child: Text('Mostrar Diálogo Personalizado'),
),
SizedBox(height: 10),
ElevatedButton(
  onPressed: () {
    final snackBar = SnackBar(
      content: Text('Este es un SnackBar.'),
    );

    ScaffoldMessenger.of(context).showSnackBar(snackBar);
  },
  child: Text('Mostrar SnackBar'),
),

```

```

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

```

### Explicación:

1. **AlertDialog:** Este tipo de diálogo muestra una alerta con un título, contenido y acciones, como el botón "OK".
2. **SimpleDialog:** Muestra opciones sencillas dentro de un cuadro de diálogo, con dos opciones para elegir.
3. **Diálogo Personalizado:** Es un diálogo que permite añadir cualquier widget, en este caso, un texto y un botón para cerrarlo.
4. **SnackBar:** Muestra un mensaje en la parte inferior de la pantalla que desaparece automáticamente después de unos segundos o puede ser deslizado fuera de la vista por el usuario.

Cada botón llama a una ventana de diálogo diferente cuando es presionado, permitiendo que los estudiantes vean cómo implementar distintos tipos de interacciones visuales en Flutter.

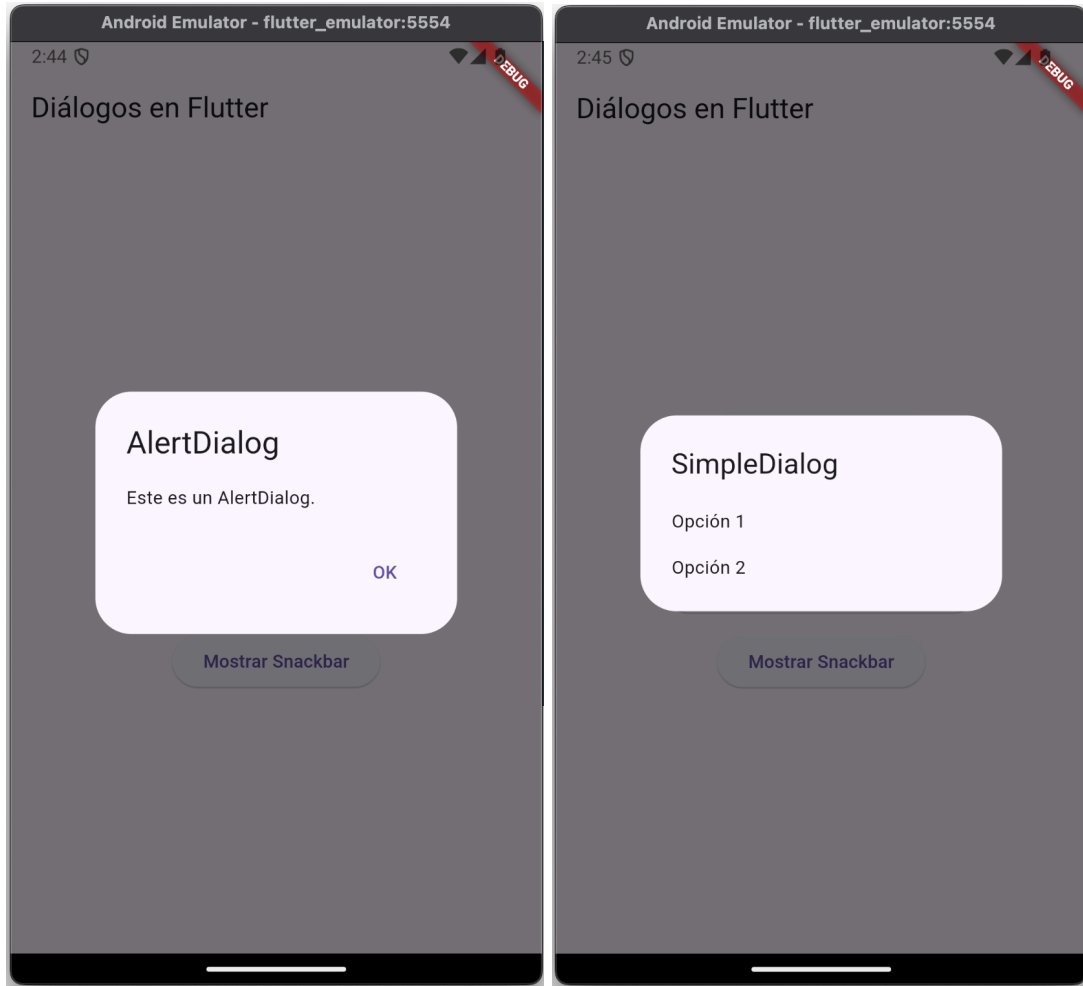


Imagen con dos capturas de pantalla de la aplicación creada con el código del ejemplo anterior.  
Creado por Javier A. Dastas (2024)

## Relación con Otros Conceptos o Lecciones

Este concepto de interacción con el usuario es clave para mejorar la experiencia de usuario (UX). Los diálogos y snackbars son elementos visuales que informan o piden confirmación de manera efectiva, y aprender a utilizarlos correctamente es un aspecto esencial del desarrollo de interfaces en Flutter. Además, la estructura de widgets en Flutter sigue siendo fundamental, por lo que comprender el manejo de estados y el contexto sigue siendo relevante en este tipo de lecciones.

## Resumen de la Lección

En esta lección, hemos explorado los tipos de diálogos en Flutter (`AlertDialog`, `SimpleDialog` y diálogos personalizados), cómo y cuándo utilizarlos, así como el uso del `SnackBar` para notificaciones rápidas. Ahora deberías tener una mejor comprensión de cómo proporcionar interacciones de usuario eficientes en una aplicación Flutter utilizando botones para invocar diálogos y snackbars.

## Actividad de la Lección

Esta actividad te permitirá aplicar los conceptos aprendidos sobre diálogos y snackbars, y demostrar que comprendes cuándo utilizar cada uno de estos elementos en una aplicación Flutter

### Instrucciones:

1. Crea una aplicación Flutter con 4 botones:
  - Un botón que muestre un `AlertDialog` cuando se presione.
    - i. Ajusta las propiedades del `AlertDialog` para presentar información o un mensaje que incluirías en una aplicación real.
  - Un botón que muestre un `SimpleDialog` con al menos dos opciones.
    - i. Ajusta las propiedades del `SimpleDialog` para presentar información o un mensaje que incluirías en una aplicación real.
  - Un botón que muestre un `Dialog` Personalizado .
    - i. Ajusta las propiedades para presentar información o un mensaje que incluirías en una aplicación real.
  - Un botón que muestre un `SnackBar` con un mensaje y una opción de "Deshacer".
    - i. Ajusta las propiedades del `SimpleDialog` para presentar información o un mensaje que incluirías en una aplicación real.
2. Asegúrate de que los diálogos y snackbar se comporten como se explicó, y agrega comentarios en tu código que describan qué hace cada parte.
3. Documenta tu código e imágenes de la pantalla en documento en un formato de PDF. Entrega el PDF en el enlace provisto por el profesor para esta actividad.