

DESARROLLO DE APP MOVILES I
GONZALEZ DIAZ ANSELMO ALEXIS
Aarón Hernández García

Textfields (Investigación)

Introducción

En Flutter, los `TextField` son los widgets utilizados para la entrada de texto por parte del usuario. Para manejar y obtener el texto ingresado de manera eficiente, se utilizan **controladores de texto** (`TextEditingController`).

Uso de `TextEditingController`

- Un `TextEditingController` permite:
 - Obtener el valor del `TextField`.
 - Modificar el texto programáticamente.
 - Escuchar cambios en el campo de texto.

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

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

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

class TextFieldExample extends StatefulWidget {
  @override
  _TextFieldExampleState createState() => _TextFieldExampleState();
}

class _TextFieldExampleState extends State<TextFieldExample> {
  final TextEditingController _controller = TextEditingController();

  @override
  void dispose() {
    _controller.dispose(); // Liberar el controlador al destruir el widget
    super.dispose();
  }
}
```

```

void _mostrarTexto() {
  showDialog(
    context: context,
    builder: (context) {
      return AlertDialog(
        title: Text("Texto ingresado"),
        content: Text(_controller.text),
        actions: [
          TextButton(
            onPressed: () => Navigator.pop(context),|
            child: Text("Cerrar"),
          ),
        ],
      );
    },
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text("TextField con Controller")),
    body: Padding(
      padding: EdgeInsets.all(16.0),
      child: Column(
        children: [
          TextField(
            controller: _controller,
            decoration: InputDecoration(
              labelText: "Escribe algo",
              border: OutlineInputBorder(),
            ),
          ),
        ],
      ),
    ),
  );
}

```

Explicación del Código

1. Se crea un TextEditingController llamado _controller.
2. Se asigna al TextField para controlar el texto ingresado.
3. Al presionar el botón, se muestra un diálogo con el texto ingresado.
4. Se usa dispose() para liberar el controlador cuando el widget deja de existir.

Escuchar Cambios en el TextField

Para reaccionar cuando el usuario escribe, puedes agregar un listener al controlador:

```
_controller.addListener(() {  
  print("Texto actual (Aaron): ${_controller.text}");  
});  
|
```

Modificar el Texto del TextField Programáticamente

Puedes modificar el contenido del TextField usando:

```
_controller.text = "Nuevo texto (Apps Moviles)";  
|
```

Uso de TextField en Formularios (Form y TextFormField)

Si trabajas con formularios, usa TextFormField junto con un GlobalKey<FormState> para validaciones más avanzadas.

```
final _formKey = GlobalKey<FormState>();  
  
TextFormField(  
  controller: _controller,  
  validator: (value) {  
    if (value == null || value.isEmpty) {  
      return "Este campo es obligatorio (Aaron)";  
    }  
    return null;  
  },  
);  
|
```

Propiedades de Textfield

1. Control del Texto

- **controller** → `TextEditingController`: Maneja el texto dentro del campo.
- **initialValue** (solo en `TextFormField`) → Texto inicial en el campo.
- **onChanged** → Función que se ejecuta cuando el usuario escribe.
- **onSubmitted** → Se ejecuta cuando el usuario presiona "Enter" o "Done".
- **onEditingComplete** → Se dispara cuando el usuario termina de editar.
- **inputFormatters** → Lista de reglas para modificar la entrada del usuario

2. Diseño y Estilos

- **decoration** → Personaliza el estilo del TextField con `InputDecoration`

```
TextField(  
  decoration: InputDecoration(  
    labelText: "Apps",  
    hintText: "Ingresa tu App favorita",  
    border: OutlineInputBorder(),  
  ),  
)
```

-
- **style** → Modifica el estilo del texto.

```
style: TextStyle(fontSize: 18, color: Colors.blue),
```

- **textAlign** → Alineación del texto dentro del TextField.

```
textAlign: TextAlign.center,
```

- **textDirection** → Dirección del texto (LTR o RTL).
- **maxLines** → Número máximo de líneas permitidas (por defecto es 1).
- **minLines** → Número mínimo de líneas visibles.
- **maxLength** → Límite de caracteres.

3. Comportamiento y Accesibilidad

- **enabled** → Habilita o deshabilita el TextField.

```
enabled: false, // Deshabilitado
```

- **readOnly** → Hace que el campo solo sea de lectura.

```
readOnly: true,
```

- **keyboardType** → Tipo de teclado a mostrar.

```
keyboardType: TextInputType.emailAddress,
```

- **autofocus** → Si es true, el TextField obtiene foco automáticamente.
- **obscureText** → Oculta el texto (para contraseñas).

```
obscureText: true,
```

- **autocorrect** → Habilita/deshabilita la autocorrección.

4. Personalización Avanzada

- **cursorColor** → Cambia el color del cursor.
- **cursorWidth** → Modifica el ancho del cursor.
- **cursorRadius** → Define el radio de la esquina del cursor.
- **showCursor** → Muestra u oculta el cursor.
- **expands** → Hace que el TextField ocupe todo el espacio disponible.

```
TextField(  
  controller: TextEditingController(),  
  decoration: InputDecoration(  
    labelText: "gmail",  
    hintText: "ksamsarojam333@gmail.com",  
    prefixIcon: Icon(Icons.email),  
    border: OutlineInputBorder(),  
  ),  
  keyboardType: TextInputType.emailAddress,  
  maxLength: 50,  
  obscureText: false,  
  textAlign: TextAlign.left,  
  style: TextStyle(fontSize: 16, color: Colors.black),  
)  
|
```

Conclusión

El uso de TextEditingController en TextField es fundamental para gestionar la entrada de texto en Flutter, permitiendo manipular, validar y escuchar cambios en tiempo real. ¡Con esto puedes construir formularios y entradas de usuario más avanzadas!

Referencias

Documentación oficial de Flutter - TextField

<https://api.flutter.dev/flutter/material/TextField-class.html>

Documentación oficial de Flutter - TextEditingController

<https://api.flutter.dev/flutter/widgets/TextEditingController-class.html>

Flutter Dev - Manejo de Formularios

<https://docs.flutter.dev/cookbook/forms/validation>