



**Universidad Tecnológica
del Norte de Guanajuato**
Organismo Público Descentralizado del Gobierno del Estado
"Educación y progreso para la vida"



Nombre: Cristian Israel Buclon Pedroza.

Num.Control: 1220100646

Grupo: GDS0351

Materia: Aplicaciones Web para I4.0.

Profesor: José De Jesús Eduardo

Barrientos Avalos

Actividad: Ejercicio 3. TextField

1. Propiedad para eliminar el banner DEBUG de las aplicaciones

```
Widget build(BuildContext context) {  
  return MaterialApp(  
    debugShowCheckedModeBanner: false,  
    title: 'Implementación de TextField', //Título de la app  
    home: Scaffold(  
      appBar: AppBar(  
        title: Text('Manejo de TextField'),  
        backgroundColor: Colors.redAccent, //Se especifica un color de fondo para la barra de la aplicación  
      ),  
      body: Center(  
        child: MyTextField(), //Objeto de la clase MyTextField que es una StatefulWidget  
      ),  
    ),  
  );  
}
```

Manejo de TextField

Ingresa el texto aquí por favor

Es una propiedad sencilla de implantar para eliminar el banner, ya que esto hace que cada una de nuestras aplicaciones se pueda ver de manera mas profesional, sin que ese banner se muestre.

2. ¿Cómo se define el color de fondo de la AppBar?

El principal método que implementa el color de fondo de la aplicación es la llamada `backgroundColor`, la cual principal mente establece la estructura previamente establecida para ser adaptada en la app

```
5 class MyApp extends StatelessWidget {  
6   @override  
7   Widget build(BuildContext context) {  
8     return MaterialApp(  
9       title: 'Implementación de TextField', //Título de la app (No visible en pantalla).  
10      home: Scaffold(  
11        appBar: AppBar(  
12          title: Text('Manejo de TextField'),  
13          backgroundColor: Colors.redAccent, //Se especifica un color de fondo para la barra de la aplicación  
14        ), // AppBar  
15        body: Center(  
16          child: MyTextField(), //Objeto de la clase MyTextField que es una StatefulWidget  
17        ), // Center  
18      ), // Scaffold  
19    ); // MaterialApp  
20  }  
21 }
```

3. ¿Para qué es el widget Scaffold? Sirve como contenedor de otros elementos que hace mas dinámica la aplicación ya sea un campo de texto, barras y algunos otros mas.

```

43 }
44 @override
45 Widget build(BuildContext context) {
46   //Se implementa el contenido principal de la pantalla
47   return Container(//contenedor
48     padding: const EdgeInsets.all(10.0),//Margen entre todos los widget
49     child: new Center(//se especifica que este centrado
50       child: new Column(//en forma de columnas
51         crossAxisAlignment: CrossAxisAlignment.stretch,
52         children: <Widget>[]//Conjunto de Widget
53           new TextField(//widget en donde el usuario puede ingresar datos
54             decoration: new InputDecoration(hintText: 'Ingresa el texto aquí porfavor'),//placeholder
55             onSubmitted: (String value){
56               onSubmitted(value);//Se pretende que tome el valor introducido en la caja de texto
57             },
58             controller: controlador, //Controlador asignado al TextField para su manipulación
59           ), // TextField
60           new Text(
61             inputValue,
62             style: new TextStyle(fontSize: 24.0,color: Colors.blueAccent)
63           ) // Text
64         //Se asigna el valor de la variable a un Text para mostrar
65         //Modificar la apariencia del texto resultante
66         //color azul y otro tamaño de texto

```

4. ¿Qué indica la propiedad child? Define los objetos que contiene el widget tal como lo es la forma de una columna.

```

Widget build(BuildContext context) {
  //Se implementa el contenido principal de la pantalla
  return Container(//contenedor
    padding: const EdgeInsets.all(10.0),//Margen entre todos los widget
    child: new Center(//se especifica que este centrado
      child: new Column(//en forma de columnas

```

5. Explica con tus palabras la siguiente línea de código:
 Especifica algunos de los elementos de un nuevo estado a partir de la clase MyTextfield para poder controlar los elementos del createState.
 State<MyTextfield> createState() => _MyTextfieldState();

6. ¿Para qué sirve un controlador en un TextField? Sirve mas que nada para poder controlar un campo específico con una información. ¿A qué clase pertenece?
 Pertenece a la clase en la que indica el text editing para poder cambiar el contenido que pueda contener el campo de texto.

```

class _MyTextFieldState extends State<MyTextField> {
  //Clase en la que se define el estado de la aplicación
  String inputValue=''; //Variable cadena para el contenido que el usuario ingresa
  //Controlador para poder manipular el contenido del TextField
  final TextEditingController controlador= new TextEditingController(); //Objeto controlador
  //Implementación de método onSubmitted, para cambiar el contenido del TextField
  void onSubmitted(String value){
    setState(() {
      print(value); //value se obtiene de la caja de texto o TextField
      inputValue=inputValue+'\n'+value;
      controlador.text=''; //Borrado de datos
    });
  }
}

```

7. Elimina setState en el método onSubmitted

¿Qué diferencia hay al ejecutar la App? La app no muestra los valores introducidos en el campo establecido por el método onSubmitted

```

@override
State<MyTextField> createState() {
  //Clase en la que se define el estado de la aplicación
  String inputValue=''; //Variable cadena para el contenido que el usuario ingresa
  //Controlador para poder manipular el contenido del TextField
  final TextEditingController controlador= new TextEditingController(); //Objeto controlador
  //Implementación de método onSubmitted, para cambiar el contenido del TextField
  void onSubmitted(String value){
    (() {
      print(value); //value se obtiene de la caja de texto o TextField
      inputValue=inputValue+'\n'+value;
      controlador.text=''; //Borrado de datos
    });
  }
}

```

Manejo de TextField

8. ¿Qué es un onSubmitted en un TextField? Tiene el principal objetivo de ir introduciendo los datos cada vez que el usuario desee cuando este presione enter, una vez introducido cualquier tipo de texto en el campo.

9. ¿Qué es un onChange en un TextField? Mas que nada tiene la función de poder cambiar el valor de un elemento o a la hora de introducir un nuevo valor para que después en compañía del Submite es que el nuevo valor es introducido

10. Explica qué contiene en la aplicación, el Widget Column: Contiene las propiedades principales del TextField para poder introducir y subir los datos.

```

46 widget build(BuildContext context) {
47   //Se implementa el contenido principal de la pantalla
48   return Container(//contenedor
49     padding: const EdgeInsets.all(10.0),//Margen entre todos los widget
50     child: new Center(//se especifica que este centrado
51       child: new Column(//en forma de columnas
52         crossAxisAlignment: CrossAxisAlignment.stretch,
53         children: <Widget>[//Conjunto de Widget
54           new TextField(//widget en donde el usuario puede ingresar datos
55             decoration: new InputDecoration(hintText: 'Ingresa el texto aquí porfavor'),//placeholder
56             onSubmitted: (String value){
57               onSubmitted(value);//Se pretende que tome el valor introducido en la caja de texto
58             },
59             controller: controlador,//Controlador asignado al TextField para su manipulación
60           ), // TextField
61           new Text(
62             inputValue,
63             style: new TextStyle(fontSize: 24.0,color: Colors.blueAccent)
64           ) // Text
65           //Se asigna el valor de la variable a un Text para mostrar
66           //Modificar la apariencia del texto resultante
67           //color azul y otro tamaño de texto

```