

En esta sección se verá cómo instalar las herramientas necesarias para el correcto funcionamiento y desarrollo de las aplicaciones a realizar.

- **Instalación de Visual Studio Code:** editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS.
- **Instalación de Flutter en Visual Studio Code:** Flutter es un SDK de código fuente abierto de desarrollo de aplicaciones móviles creado por Google
- **Creación de dispositivos virtuales:** dispositivos virtuales que permiten emular el comportamiento de un dispositivo físico. Utilizado para poder ver y testear la funcionalidad de la aplicación.
- **Dart upgrade:** actualización del sdk de Dart.
- **Dart upgrade comando:** actualización del sdk de Dart mediante un comando.

En VSCode presionamos Ctrl + Shift +P

Seleccionamos Flutter: New Application Project, seleccionamos la ubicación del proyecto, le asignamos un nombre y esperamos a que se cree el proyecto.

También se puede con el comando flutter create nombre_app

El archivo principal que genera el proyecto es el “main.dart”, este archivo lo borramos y creamos uno desde cero con el emmet “mateapp”.

Con el comando flutter doctor identificamos todo lo que contiene flutter incluido si poseemos emuladores

Sobre el código main.dart hacemos clic y presionamos F5, esto nos muestra los emuladores o maneras de visualizar el app en construcción

También podemos ejecutar el proyecto con el comando “flutter run” y luego seleccionar una opción

Con el emmet st creamos estructuras de nuestros widgets

<https://esflutter.dev/docs/development/tools/vs-code>

Ctrl + . para crear métodos como el override

Comentarios ctrl+k, ctrl+c

//Comentario de una línea

Alt+shift+a

/*Comentario multilínea*/

Material app (material design)

bottomNavigationBar

es la barra de navegación típica de las apps

Drawer es como un menú lateral o endDrawer

Layout es el contenedor de widgets

Dev tolos es una herramienta para comprender la lógica de dart y su diseño

Columna y fila son widgets altamente utilizados

Text este widget tiene sus propiedades como style: donde definimos como deseamos que se visualice

```
//2.Texto enriquecido. Permite mostrar cadenas de texto con diferentes estilos.
RichText(
  text: TextSpan(
    text: 'texto enriquecido ',
    style: TextStyle(fontSize: 16, fontStyle: FontStyle.normal, color: Colors.black),
    children: <TextSpan>
    [
      TextSpan(text: 'bold ', style: TextStyle(fontWeight: FontWeight.bold, color: Colors.blue)),
      TextSpan(text: 'texto regular'),
    ] // <TextSpan>[]
  ), // TextSpan
) // RichText
```

```
//2.Texto enriquecido. Permite mostrar cadenas de texto con diferentes estilos.
RichText(
  text: TextSpan(
    text: 'texto enriquecido ',
    style: DefaultTextStyle.of(context).style,
    children: <TextSpan>
    [
      TextSpan(text: 'bold ', style: TextStyle(fontWeight: FontWeight.bold)),
      TextSpan(text: 'texto regular'),
    ] // <TextSpan>[]
  ), // TextSpan
) // RichText
```

Container se utiliza para añadir una capa al fondo del widget

```

class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int counter = 0;

  void incrementCounter() {
    setState(() {});
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(),
      body: Container(
        color: Colors.yellow,
        margin: EdgeInsets.only(top: 24, left: 16),

        padding: EdgeInsets.all(16),
        child: Icon(
          Icons.accessible_forward,
          color: Colors.white,
        ), // Icon
      )); // Container // Scaffold
  }
}

```

Compilación

Generando un archivo tipo APK

Abrimos una nueva terminal ubicándonos siempre en la ruta del proyecto.

Escribimos el comando flutter build apk --release

Esperamos el proceso y al final copiamos la ruta que nos genera el compilador precedido por la palabra Build , esa ruta la pegamos después de la ruta principal del proyecto y allí obtenemos el archivo apk que necesitamos.

Pasar la apk a formato AAB para playstore

Firmar y subir la app

Comando flutter build appbundle