# fluter

Al crear un proyecto nuestro código se va a encontrar en la carpeta /lib

Se van a ver un carpeta por cada plataforma

Los test van a estar un carpeta /test

# pubspec.lock

Almacena información sobre las librerias usadas

# pubspec.yaml

tienes varias configuraciones generales del proyecto, como el nombre o la versión de andriod objetivo

# main

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

runApp recibe un Widget como argumento

# Widget

Hay de dos tipos

## Stateless

Es una pieza de lego que se construye muy rápido y no mantiene el estado por si mismo. Flutter sabe cuando se de be de volver a dibujar, es recomendado que en lugar de crear funciones y métodos que retornan Widget, es mejor crear Clases que extiendan de estos StatelesWidgets.

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    // TODO: implement build

    return MaterialApp(

      home: Center(child: Text("hola mundo")),

    );

  }

}

### Constructor con constante

Por alguna razón es buena idea declarar el constructor como a continuación

class MyApp extends StatelessWidget {

  const MyApp({super.key});

  @override

  Widget build(BuildContext context) {

    // TODO: implement build

    return const MaterialApp(

      debugShowCheckedModeBanner: false,

      home: CounterScreem(),

    );

  }

}

## Stateful

Es similar al stateless en cuanto a que es un Widget, pero este permite mantener un estado interno y ciclo de vida como su inicialización y destrucción. Muchos lo tachan de que jamás se deben de usar pero eso no es cierto, los stateful básicamente son el corazón de cualquier animación que suceda.

## Text

Text("hola mundo")

## Center

Center(child: Text("hola mundo"))

## MaterialApp

MaterialApp(

      home: Center(child: Text("hola mundo")),

    );

### Con temas y sin debug

MaterialApp(

      debugShowCheckedModeBanner: false,

      home: CounterScreem(),

      theme: ThemeData(useMaterial3: true, colorSchemeSeed: Colors.red),

    );

## Column

Column(

          mainAxisAlignment: MainAxisAlignment.center,

          children: [Text("a"), Text("b")],

        )

## Icon

Icon(Icons.plus\_one)

## Scaffold

Scaffold(

      appBar: AppBar(

        title: const Text('el titulo'),

      ),

      body: Center(

        child: Column(

          mainAxisAlignment: MainAxisAlignment.center,

          children: [

            Text(

              "a",

              style: TextStyle(fontSize: 10, fontWeight: FontWeight.w100),

            ),

            Text("b")

          ],

        ),

      ),

      floatingActionButton: FloatingActionButton(

        onPressed: () {},

        child: Icon(Icons.plus\_one),

      ),

    );

## AppBar

AppBar(

        title: const Text('el titulo'),

      )

# return const

si en una funcion se pone de esta manera en el return flutter optimiza el código resultante para no esperar un cambio de este código ?

class MyApp extends StatelessWidget {

  const MyApp({super.key});

  @override

  Widget build(BuildContext context) {

    // TODO: implement build

    return const MaterialApp(

      home: Scaffold(

        body: Center(

          child: Text('hola mundo'),

        ),

      ),

    );

  }

}

# TextStyle

Algunos componentes los reciben el parámetro style

Text("el texto",

              style: TextStyle(fontSize: 10, fontWeight: FontWeight.w100),

            ),

# ThemeData

ThemeData(useMaterial3: true, colorSchemeSeed: Colors.red),

# Estado global de la aplicación

Es un estado que todos los widgets sin importar el lugar donde se encuentren dentro del árbol de widtges