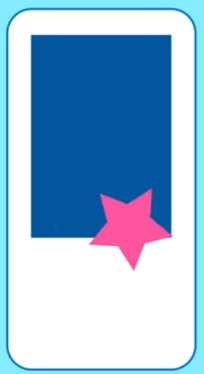
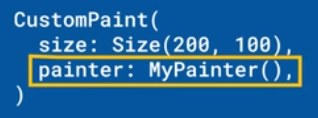
**CustomPainter**

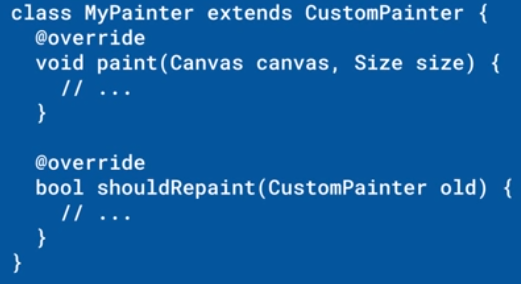
Sirve para hacer interfaces de usuario muy personalizadas. Permite crear gráficos sofisticados:



Creamos el Widget CustomPaint(). Especificamos size y painter:



Luego implementamos la clase MyPainter que extiende de CustomPainter:

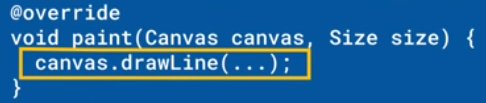


Tiene dos métodos: Paint y shouldRepaint

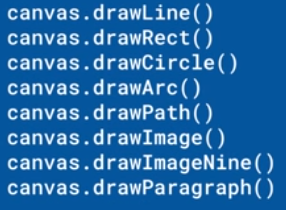
shouldRepaint se invoca cuando se reconstruye el CustomPainter:



El método paint() es donde se pone el lienzo y se dibuja



Se puede dibujar líneas, rectángulos, círculos, arcos, curvasmapas de bits, nine-patchs, y párrafos de texto



Se puede controlar color, filtros, mezclas y más.

|  |  |
| --- | --- |
| **Código** | **Comentarios** |
| import 'package:flutter/material.dart';  class CustomPainterScreen extends StatefulWidget {    const CustomPainterScreen({Key? key}) : super(key: key);    @override    State<CustomPainterScreen> createState() => \_CustomPainterScreenState();  }  class \_CustomPainterScreenState extends State<CustomPainterScreen> {    @override    Widget build(BuildContext context) {      return Scaffold(        appBar: AppBar(          title: const Text('CustomPainter'),          centerTitle: true,        ),        body: Center(          child: CustomPaint(            size: const Size(300, 300),            painter: MyPainter(),          ),        ),      );    }  }  //--------------- MyPainter -------------------  class MyPainter extends CustomPainter {    @override    void paint(Canvas canvas, Size size) {      final lapiz = Paint();      //Propiedades del lápiz      lapiz.color = Colors.red;      lapiz.style = PaintingStyle.fill;      lapiz.strokeWidth = 5.0; //ancho del lápiz      final path = Path();      //Dibujar con el path y el lápiz      path.moveTo(0, 0);      path.lineTo(size.width, size.height);      path.quadraticBezierTo(          size.width \* 0.5, size.height \* 0.45, size.width, size.height \* 0.25);      path.lineTo(size.width, 0);      path.lineTo(0, 0);      canvas.drawPath(path, lapiz);    }    @override    bool shouldRepaint(covariant CustomPainter oldDelegate) {      return true;    }  } |  |

Así se ve:

Logotipo, nombre de la empresa

Descripción generada automáticamente

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -