Desenvolver Interface Gráfica Para Dispositivos Móveis

Aula 18

Professor: Henrique Augusto Maltauro

Fecomércio PR
Sesc Senac IFPD
Senac

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder**

O OutlinedBorder é uma classe abstrata do Flutter que implementa um componente com um formato definido por uma borda.

Ele não é classificado como um componente, ou seja, ele sempre vai precisar estar relacionado a algum outro componente.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder**

Existem oito classes concretas que implementam o OutlinedBorder:

➔ **BeveledRectangleBorder**

➔ **CircleBorder**

➔ **ContinuousRectangleBorder**

➔ **LinearBorder**

Flutter/Dart

- ShapeBorder ➤ OutlinedBorder

➔ MaterialStateOutlinedBorder

➔ RoundedRectangleBorder

➔ StadiumBorder

➔ StarBorder

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder**

É com o OutlinedBorder o strokeAlign do BorderSide vai funcionar corretamente.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ BeveledRectangleBorder**

O BeveledRectangleBorder é uma classe do Flutter que implementa um borda em um formato retangular, com cantos retos e chanfrados.

Ele não é classificado como um componente, ou seja, ele sempre vai precisar estar relacionado a algum outro componente.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ BeveledRectangleBorder**

As principais propriedades do BeveledRectangleBorder são:

➔ **side**

- Que recebe um BorderSide para definir a coloração e o tamanho das bordas.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ BeveledRectangleBorder**

➔ **borderRadius**

- Que recebe um BorderRadiusGeometry para definir os raios de cada canto.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ BeveledRectangleBorder**

```
BeveledRectangleBorder(
    side: BorderSide,
    borderRadius: BorderRadiusGeometry
)  // BeveledRectangleBorder
```

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ CircleBorder**

O CircleBorder é uma classe do Flutter que implementa uma borda em um formato circular.

Ele não é classificado como um componente, ou seja, ele sempre vai precisar estar relacionado a algum outro componente.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ CircleBorder**

As principais propriedades do CircleBorder são:

➔ **side**

- Que recebe um BorderSide para definir a coloração e o tamanho das bordas.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ CircleBorder**

➔ **eccentricity**

- Que recebe um double, que deve ter o valor entre 0.0 e 1.0, para definir como a borda do círculo irá ser formada para caber dentro de um retângulo.
- Quando o valor for 0.0, será formado um círculo perfeito, quando o valor 1.0 será formado um círculo oval.

Flutter/Dart

- ShapeBorder ➤ OutlinedBorder ➤ CircleBorder

```
CircleBorder(
    side: BorderSide,
    eccentricity: double
)  // CircleBorder
```

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ ContinuousRectangleBorder**

O ContinuousRectangleBorder é uma classe do Flutter que implementa um borda em um formato retangular, com suaves e contínuas conexões entre os lados e os cantos arredondados.

Ele não é classificado como um componente, ou seja, ele sempre vai precisar estar relacionado a algum outro componente.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ ContinuousRectangleBorder**

As principais propriedades do ContinuousRectangleBorder são:

➔ **side**

- Que recebe um BorderSide para definir a coloração e o tamanho das bordas.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ ContinuousRectangleBorder**

➔ **borderRadius**

- Que recebe um BorderRadiusGeometry para definir os raios de cada canto.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ ContinuousRectangleBorder**

```
ContinuousRectangleBorder(
    side: BorderSide,
    borderRadius: BorderRadiusGeometry
)  // ContinuousRectangleBorder
```

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

O LinearBorder é uma classe do Flutter que implementa uma borda com um formato retangular, similar ao BoxBorder, que permite definir separadamente os detalhes de cada lado da borda.

Porém em termos praticos essa borda é renderizada como uma coisa só.

Ele não é classificado como um componente, ou seja, ele sempre vai precisar estar relacionado a algum outro componente.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

As principais propriedades do LinearBorder são:

➔ **side**

- Que recebe um BorderSide para definir a coloração e o tamanho das bordas.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

➔ **start**

- Que recebe um LinearBorderEdge? para definir a borda da esquerda.

➔ **end**

- Que recebe um LinearBorderEdge? para definir a borda da direita.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

➔ **top**

- Que recebe um LinearBorderEdge? para definir a borda de cima.

➔ **bottom**

- Que recebe um LinearBorderEdge? para definir a borda de baixo.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

```
LinearBorder(
    side: BorderSide,
    start: LinearBorderEdge?,
    end: LinearBorderEdge?,
    top: LinearBorderEdge?,
    bottom: LinearBorderEdge?,
)  // LinearBorder
```

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

O LinearBorder também possui outros quatro construtores mais especificos:

➔ **bottom**

- Que implementa somente a borda de baixo.

➔ **end**

- Que implementa somente a borda da direita.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

➔ **start**

- Que implementa somente a borda da esquerda.

➔ **top**

- Que implementa somente a borda de cima.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

Todos esses outros construtores recebem:

➔ **side**

- Que recebe um BorderSide para definir a coloração e o tamanho das bordas.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

➔ **alignment**

- Que recebe um double para definir o alinhamento da borda.

➔ **size**

- Que recebe um double para definir o tamanho da borda.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

```
LinearBorder.bottom(
    side: BorderSide,
    alignment: double,
    size: double
)  // LinearBorder.bottom
```

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

```
LinearBorder.top(
    side: BorderSide,
    alignment: double,
    size: double
)  // LinearBorder.top
```

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

```
LinearBorder.start(
    side: BorderSide,
    alignment: double,
    size: double
)  // LinearBorder.start
```

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ LinearBorder**

```
LinearBorder.end(
    side: BorderSide,
    alignment: double,
    size: double
)  // LinearBorder.end
```

Flutter/Dart

- **LinearBorderEdge**

O LinearBorderEdge é uma classe do Flutter que implementa o tamanho e o alinhamento de uma borda linear.

Ele não é classificado como um componente, ou seja, ele sempre vai precisar estar relacionado a algum outro componente.

Flutter/Dart

- **LinearBorderEdge**

As principais propriedades do LinearBorderEdge são:

➔ **size**

- Que recebe um double para definir o tamanho da borda.

➔ **alignment**

- Que recebe um double para definir o alinhamento da borda.

Flutter/Dart

- **LinearBorderEdge**

```
LinearBorderEdge(
    alignment: double,
    size: double
)  // LinearBorderEdge
```

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ MaterialStateOutlinedBorder**

O MaterialStateOutlinedBorder é uma classe do Flutter que implementa uma borda cujo formato depende de um conjunto MaterialStates, permitindo que o componente seja interativo.

Ele não é classificado como um componente, ou seja, ele sempre vai precisar estar relacionado a algum outro componente.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ MaterialStateOutlinedBorder**

Ele é um pouco mais complexo de ser trabalhado, então eu vou deixar mais pra frente a formulação de como se trabalhar com ele.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ RoundedRectangleBorder**

O RoundedRectangleBorder é uma classe do Flutter que implementa uma borda com formato retangular e cantos arredondados.

Ele não é classificado como um componente, ou seja, ele sempre vai precisar estar relacionado a algum outro componente.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ RoundedRectangleBorder**

As principais propriedades do RoundedRectangleBorder são:

➔ **side**

- Que recebe um BorderSide para definir a coloração e o tamanho das bordas.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ RoundedRectangleBorder**

➔ **borderRadius**

- Que recebe um BorderRadiusGeometry para definir o arredondamento de cada canto.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ RoundedRectangleBorder**

```
RoundedRectangleBorder(
    side: BorderSide,
    borderRadius: BorderRadiusGeometry
)   // RoundedRectangleBorder
```

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StadiumBorder**

O StadiumBorder é uma classe do Flutter que implementa uma borda com formato de estádio.

Ou seja, ele possui semi-circulos nos lados verticais.

Ele não é classificado como um componente, ou seja, ele sempre vai precisar estar relacionado a algum outro componente.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StadiumBorder**

As principais propriedades do StadiumBorder são:

➔ **side**

- Que recebe um BorderSide para definir a coloração e o tamanho das bordas.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StadiumBorder**

```
StadiumBorder(
    side: BorderSide
)  // StadiumBorder
```

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StarBorder**

O StarBorder é uma classe do Flutter que implementa uma borda com formato de estrela ou de polígono.

Ele não é classificado como um componente, ou seja, ele sempre vai precisar estar relacionado a algum outro componente.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StarBorder**

As principais propriedades do StarBorder são:

➔ **side**

- Que recebe um BorderSide para definir a coloração e o tamanho das bordas.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StarBorder**

➔ **point**

- Que recebe um double para definir a quantidade de pontas que a estrela possui.

➔ **innerRadiusRatio**

- Que recebe um double para definir a proporção entre o raio externo e o raio interno da estrela.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StarBorder**

```
StarBorder(
    side: BorderSide,
    point: double,
    innerRadiusRatio: double
)  // StarBorder
```

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StarBorder**

➔ **pointRounding**

- Que recebe um double para definir a quantidade de arredondamento das pontas da estrela.

➔ **valleyRounding**

- Que recebe um double para definir a quantidade de arredondamento dos cantos internos da estrela.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StarBorder**

➔ **rotation**

- Que recebe um double para definir a rotação em graus no sentido horário.

➔ **squash**

- Que recebe um double para definir a proporção da estrela em relação ao componente em anexo.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StarBorder**

```
StarBorder(
    pointRounding: double,
    valleyRounding: double,
    rotation: double,
    squash: double
)  // StarBorder
```

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StarBorder**

Para definirmos um polígno, deve-se fazer uso do construtor polygon.

Quando utilizado esse construtor, os parâmetros recebidos são:

➔ **side**

- Que recebe um BorderSide para definir a coloração e o tamanho das bordas.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StarBorder**

➔ **sides**

- Que recebe um double para definir a quantidade de lados que o polígno possui.

➔ **pointRounding**

- Que recebe um double para definir a quantidade de arredondamento dos cantos do polígno

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StarBorder**

➔ **rotation**

- Que recebe um double para definir a rotação em graus no sentido horário.

➔ **squash**

- Que recebe um double para definir a proporção do polígno em relação ao componente em anexo.

Flutter/Dart

- **ShapeBorder ➤ OutlinedBorder ➤ StarBorder**

```
StarBorder.polygon(
    side: BorderSide,
    pointRounding: double,
    rotation: double,
    squash: double
)  // StarBorder.polygon
```

Exercício

Exercício

1. Construir uma tela de login
    a. Usar ElevatedButton
    b. Usar InputBorder nos campos de texto
    c. Usar StarBorder para definir um logo central

github.com/hmaltaurodev/slides