

The background features a repeating geometric pattern of solid and dashed lines forming a grid of triangles. A large, light blue circle is centered on the page, serving as a backdrop for the title text.

DESENVOLVIMENTO FLUTTER

Prof. Ravel Teixeira de Vasconcelos



Vamos aproveitar a base do app anterior para criar um novo app.

Dessa vez, o novo app, além de gerar e apresentar um número aleatório, deverá informar se o número gerado é par ou ímpar.

```

NumeroAleatorio createState() => NumeroAleatorioState {}

class NumeroAleatorioState extends State<NumeroAleatorio> {
  int numero = 0;

  void gerarNumero() {
    setState(() {
      Random numeroAleatorio = new Random();
      numero = numeroAleatorio.nextInt(1000);
    });
  }

  @override
  Widget build(BuildContext context) {

    return Container(
      child: Column(
        children: [
          Text("$numero", style: TextStyle(fontSize: 28)),
          SizedBox(height: 30),
          RaisedButton(
            onPressed: gerarNumero,
            child: Text("Gerar número")
          )
        ]
      )
    );
  }
}

```

▶ RUN

Gerador aleatório

Primeiro, vamos atualizar a classe de estado.

Gerador de Números

288

O número é par

Gerar número

```
NUMEROALEATORIO JUSTICE createState() => NumeroAleatorioState() {  
}  
  
class NumeroAleatorioState extends State<NumeroAleatorio> {  
  
  int numero = 0;  
  String mensagem = "";  
  
  void gerarNumero() {  
    setState(() {  
      Random numeroAleatorio = new Random();  
      numero = numeroAleatorio.nextInt(1000);  
    });  
  }  
  
  @override  
  Widget build(BuildContext context) {  
  
    return Container(  
      child: Column(  
        children: [  
          Text("$numero", style: TextStyle(fontSize: 28)),  
          SizedBox(height: 30),  
          RaisedButton(  
            onPressed: gerarNumero,  
            child: Text("Gerar número")  
          )  
        ]  
      )  
    );  
  }  
}
```

▶ RUN

Gerador aleatório

Inclua um atributo que representará a mensagem de feedback a ser apresentada ao usuário.

Gerador de Números

288

O número é par

Gerar número

```
NUMEROALEATORIO 103146 createState() -- NUMEROALEATORIO 103146
```

RUN

```
class NumeroAleatorioState extends State<NumeroAleatorio> {  
  
  int numero = 0;  
  String mensagem = "";  
  
  void gerarNumero() {  
    setState(() {  
      Random numeroAleatorio = new Random();  
      numero = numeroAleatorio.nextInt(1000);  
  
      if(numero % 2 == 0) {  
        mensagem = "O número é par";  
      }  
      else {  
        mensagem = "O número é ímpar";  
      }  
    });  
  }  
  
  @override  
  Widget build(BuildContext context) {  
  
    return Container(  
      child: Column(  
        children: [  
          Text("$numero", style: TextStyle(fontSize: 28)),  
          SizedBox(height: 30),  
          RaisedButton(  

```

Gerador aleatório

A partir de uma condição, podemos verificar se o número gerado é par ou ímpar, atualizando o atributo mensagem de acordo com a condição.

numeros

Gerar número

```

if(numero % 2 == 0) {
  mensagem = "0 número é par";
}
else {
  mensagem = "0 número é impar";
}
});
}

```

▶ RUN

Gerador aleatório

```

@override
Widget build(BuildContext context) {

  return Container(
    child: Column(
      children: [
        Text("$numero", style: TextStyle(fontSize: 28)),
        SizedBox(height: 30),
        RaisedButton(
          onPressed: gerarNumero,
          child: Text("Gerar número")
        )
      ]
    )
  );
}

```

Agora, precisamos incluir um texto que permita exibir o conteúdo do atributo mensagem no corpo da página principal.

meros

Gerar número

```
if(numero % 2 == 0) {  
  mensagem = "O número é par";  
}  
else {  
  mensagem = "O número é ímpar";  
}  
});  
}
```

▶ RUN

```
@override  
Widget build(BuildContext context) {
```

```
  return Container(  
    child: Column(  
      children: [  
        Text("$numero", style: TextStyle(fontSize: 28)),  
        SizedBox(height: 30),  
        Text("$mensagem"),  
        RaisedButton(  
          onPressed: gerarNumero,  
          child: Text("Gerar número")  
        )  
      ]  
    )  
  );  
}
```

Gerador aleatório

Antes do botão, incluímos um Widget de texto, inserindo a variável mensagem no texto a ser exibido.

números

Gerar número

```

if(numero % 2 == 0) {
  mensagem = "O número é par";
}
else {
  mensagem = "O número é ímpar";
}
});
}

```

▶ RUN

```

@override
Widget build(BuildContext context) {

return Container(
  child: Column(
    children: [
      Text("$numero", style: TextStyle(fontSize: 28)),
      SizedBox(height: 30),
      Text("$mensagem"),
      SizedBox(height: 30),
      RaisedButton(
        onPressed: gerarNumero,
        child: Text("Gerar número")
      )
    ]
  )
);
}
}

```

Gerador aleatório

E incluímos um novo Widget espaçador só para questões de organização de layout.

meros

Gerar número

```
if(numero % 2 == 0) {  
  mensagem = "O número é par";  
}  
else {  
  mensagem = "O número é impar";  
}  
});  
}
```

▶ RUN

Teste o app!

```
@override  
Widget build(BuildContext  
  
return Container(  
  child: Column(  
    children: [  
      Text("$numero",  
        SizedBox(height  
      Text("$mensagem",  
        SizedBox(height  
      RaisedButton(  
        onPressed: ge  
        child: Text("
```

Gerador aleatório

Gerador de Números

556

O número é par

Gerar número

Trabalhando com formulários no Flutter

Criando um formulário de boas vindas

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Boas vindas'),
        ),
      ),
    );
  }
}
```

▶ RUN

Boas vindas

DEBUG

Vamos primeiramente codificar a base da nossa aplicação.

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Boas vindas'),
        ),
        body: Column(
          children: [
            Text("Digite seu nome:"),
          ]
        ),
      ),
    );
  }
}
```

▶ RUN

Boas vindas

DEBUG

No corpo do Scaffold,
vamos incluir um texto
(Widget Text).

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Boas vindas'),
        ),
        body: Column(
          children: [
            Text("Digite seu nome:"),
          ]
        )
      ),
    );
  }
}
```

```
class BoasVindas extends StatefulWidget {
  BoasVindasState createState() => new BoasVindasState();
}

class BoasVindasState extends State<BoasVindas> {
}
```

▶ RUN

Boas vindas

DEBUG

Digite seu nome:

Vamos criar o Widget que conterà um botão, um evento a ser executado a partir do mesmo, além do texto que irá dar boas vindas baseado no que o usuário digitar. Também precisamos criar uma classe de estado associada, já que iremos alterar estados de Widgets e executar métodos.

```
home: Scaffold(  
  appBar: AppBar(  
    title: Text('Boas vindas'),  
  ),  
  body: Column(  
    children: [  
      Text("Digite seu nome:"),  
    ]  
  )  
);  
}
```

▶ RUN

Boas vindas

Digite seu nome:

```
class BoasVindas extends StatefulWidget {  
  BoasVindasState createState() => new BoasVindasState();  
}  
  
class BoasVindasState extends State<BoasVindas> {  
  
  @override  
  Widget build(BuildContext context) {  
    return Container(  
      child: Column(  
        children: []  
      )  
    );  
  }  
}
```

Na classe de estado, somos obrigado a implementar o método build. Nele, utilizaremos uma coluna dentro de um container para incluir o botão e o rótulo de boas vindas.

```
    appBar: AppBar(  
      title: Text('Boas vindas'),  
    ),  
    body: Column(  
      children: [  
        Text("Digite seu nome:"),  
      ],  
    ),  
  );  
}  
  
class BoasVindas extends StatefulWidget {  
  BoasVindasState createState() => new BoasVindasState();  
}  
  
class BoasVindasState extends State<BoasVindas> {  
  @override  
  Widget build(BuildContext context) {  
    return Container(  
      child: Column(  
        children: [  
          TextField(),  
        ],  
      ),  
    );  
  }  
}
```

▶ RUN

Boas vindas

DEBUG

Digite seu nome:

Incluimos um campo de texto a partir do Widget TextField().

```
    title: Text('Boas vindas'),
  ),
  body: Column(
    children: [
      Text("Digite seu nome:"),
    ]
  )
);
}

class BoasVindas extends StatefulWidget {
  BoasVindasState createState() => new BoasVindasState();
}

class BoasVindasState extends State<BoasVindas> {

  @override
  Widget build(BuildContext context) {
    return Container(
      child: Column (
        children: [
          TextField(),
          SizedBox(height: 30),
        ]
      )
    );
  }
}
```

▶ RUN

Boas vindas

DEBUG

Digite seu nome:

Incluimos um separador.

```
        Text("Digite seu nome:"),
      ],
    ),
  );
}

class BoasVindas extends StatefulWidget {
  BoasVindasState createState() => new BoasVindasState();
}

class BoasVindasState extends State<BoasVindas> {

  @override
  Widget build(BuildContext context) {
    return Container(
      child: Column (
        children: [
          TextField(),
          SizedBox(height: 30),
          RaisedButton(
            child: Text('Enviar'),
            onPressed: () {},
          )
        ],
      ),
    );
  }
}
```

▶ RUN

Boas vindas

DEBUG

Digite seu nome:

E um botão, que por enquanto não faz nada.

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Boas vindas'),
        ),
        body: Column(
          children: [
            Text("Digite seu nome:"),
            BoasVindas()
          ]
        )
      )
    );
  }
}

class BoasVindas extends StatefulWidget {
  BoasVindasState createState() => new BoasVindasState();
}

class BoasVindasState extends State<BoasVindas> {
  @override
```

▶ RUN

Boas vindas

Digite seu nome:

DEBUG

Inclua o Widget recém-criado no corpo do Scaffold.

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Boas vindas'),
        ),
        body: Column(
          children: [
            Text("Digite seu nome:"),
            BoasVindas()
          ]
        )
      ),
    );
  }
}
```

▶ RUN

Boas vindas

DEBUG

Digite seu nome:

Enviar

Teste o app. Caso esteja tudo ok, vamos capturar os textos digitados no TextField para processamento via evento.

```
class BoasVindas extends StatefulWidget {
  BoasVindasState createState() => new BoasVindasState();
}
```

```
class BoasVindasState extends State<BoasVindas> {
```

```
  @override
```



“That’s all Folks!”