

# DGT2812 Desenvolvimento de Aplicativos Móveis com Flutter



**Jacson Souza de Andrade - Mat. 2023 1204 4479**

Campus: Polo Piabetá - Magé - RJ

Turma: 2025.4

Semestre Letivo: 4º

## Desenvolvimento de Aplicativos Móveis Com Flutter

Início meu projeto importando a biblioteca principal flutter, e a dart:async, que uso para o timer do slideshow

```
1 import 'dart:async';
2 import 'package:flutter/material.dart';
```

O trecho a seguir contempla Home Page da aplicação, que precisa ser StatefulWidget, porque o slideshow e a avaliação mudam.

late PageController \_pageController controla o pageview(slideshow)

final ScrollController \_destinosScrollController = ScrollController(); Controla o scroll horizontal dos cards

int \_currentPage = 0; Guarda qual slide está ativo

Em final List<Map<String, String>> destinos tenho os dados

Conforme o código da imagem abaixo, o slide show executa uma vez, quando a tela nasce, e depois a cada x segundos, com troca suave de slides.

termino adicionando o trecho para liberar memória | Evitar vazamento (memory leak).

```
//(SLIDESHOW AUTOMÁTICO)
@Override
void initState() {
  super.initState();
  _pageController = PageController();

  Timer.periodic(const Duration(seconds: 3), (timer) { //Executa a cada 3 segundos
    if (_pageController.hasClients) {
      _currentPage = (_currentPage + 1) % destinos.length; //Avança slide e % garante looping infinito

      //Anima a troca de slide suavemente
      _pageController.animateToPage(
        _currentPage,
        duration: const Duration(milliseconds: 500),
        curve: Curves.easeInOut,
      );
    }
  }); // Timer.periodic
}
```

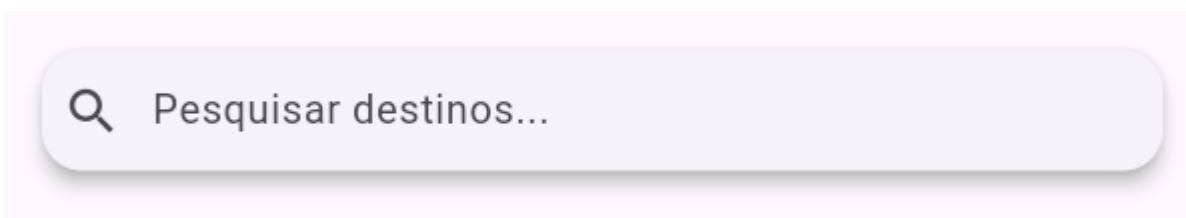
Efeito visual, mostrando a rolagem automática para troca de slides:



Abaixo do slideshow adicionei uma barra de pesquisa. Com um Widget isolado. Campo pronto para busca. UX moderna

```
199 <Widget searchBarSection() {
200   return Padding(
201     padding: const EdgeInsets.symmetric(horizontal: 24),
202     child: Card(
203       elevation: 6,
204       shape: RoundedRectangleBorder(
205         borderRadius: BorderRadius.circular(16),
206       ), // RoundedRectangleBorder
207       child: TextField(
208         decoration: InputDecoration(
209           hintText: 'Pesquisar destinos...',
210           prefixIcon: const Icon(Icons.search),
211           border: InputBorder.none,
212           contentPadding: const EdgeInsets.symmetric(
213             horizontal: 16,
214             vertical: 14,
215           ), // EdgeInsets.symmetric
216         ), // InputDecoration
217       ), // TextField
218     ), // Card
219   ); // Padding
220 }
221 }
```

Abaixo mostro o resultado visual



O próximo passo foi adicionar minha Build principal.

Esse método `build` é responsável por **construir a interface principal da HomePage**.

Ele descreve **como a tela deve ser renderizada com base no estado atual do widget**, especialmente:

- `_currentPage` (slide ativo)
- `destinos` (fonte de dados)

- navegação entre telas

O Flutter **reexecuta esse método automaticamente** sempre que `setState` é chamado.

usei `final color = Theme.of(context).primaryColor` para obter a cor principal do tema ativo.

appBar: AppBar(

title: const Text('Explore Mundo 🌎'),

), é uma Barra superior fixa.

body: ListView(... Permite **scroll vertical automático**, ideal para telas longas, como é o caso da aplicação

```

225    @override
226    Widget build(BuildContext context) {
227      final color = Theme.of(context).primaryColor;
228
229      return Scaffold(
230
231        appBar: AppBar(
232          title: const Text('Explore Mundo 🌎'),
233        ), // AppBar
234        body: ListView(
235          children: [
236            bannerSection(context),
237            const SizedBox(height: 16),
238            searchBarSection(),
239

```

A seguir, o código do título e avaliação dinâmica

Padding(

padding: const EdgeInsets.all(32),

child: Row(. `Row` : organiza conteúdo horizontal. `Expanded` : texto ocupa o máximo possível. Ícone + avaliação alinhados à direita

A avaliação é dinâmica: `destinos[_currentPage]['rating']!`

```

241 |   |   | Padding()
242 |   |   |   padding: const EdgeInsets.all(32),
243 |   |   |   child: Row(
244 |   |   |     children: [
245 |   |   |       Expanded(
246 |   |   |         child: Column(
247 |   |   |           crossAxisAlignment: CrossAxisAlignment.start,
248 |   |   |           children: [
249 |   |   |             const Text(
250 |   |   |               'Explore Mundo',
251 |   |   |               style: TextStyle(
252 |   |   |                 fontWeight: FontWeight.bold,
253 |   |   |                 fontSize: 16,
254 |   |   |               ), // TextStyle
255 |   |   |             ), // Text
256 |   |   |             const SizedBox(height: 8),
257 |   |   |             Text(
258 |   |   |               'Os melhores destinos para você',
259 |   |   |               style: TextStyle(color: Colors.grey[500]),
260 |   |   |             ), // Text
261 |   |   |           ],
262 |   |   |         ), // Column
263 |   |   |       ), // Expanded
264 |   |   |       Icon(Icons.star, color: Colors.orange[500]),
265 |   |   |       const SizedBox(width: 4),
266 |   |   |       Text(
267 |   |   |         | destinos[_currentPage]['rating']!, // Validação muda conforme slide, tudo sincronizado via _currentPage
268 |   |   |         | style: const TextStyle(fontWeight: FontWeight.bold),
269 |   |   |       ), // Text
270 |   |   |     ],
271 |   |   |   ), // Row

```

Logo depois temos os Botões de ação (menu rápido)

## Row

**mainAxisAlignment: MainAxisAlignment.spaceEvenly**, Menu de navegação rápida

**Usa \_buildButtonColumn** para: reduzir duplicação, padronizar UI e facilitar manutenção.

## Navigator.push

context,

## MaterialPageRoute

builder: (\_) => const SobreNosPage(),

),

); Empilha uma nova rota, Mantém histórico de navegação e usa MaterialPageRoute (transição nativa)

```

274 // BOTÕES
275 Row(
276   mainAxisAlignment: MainAxisAlignment.spaceEvenly,
277   children: [
278     _buildButtonColumn(color, Icons.search, 'DESTINOS'),
279     _buildButtonColumn(color, Icons.calendar_month, 'PACOTES'),
280     _buildButtonColumn(color, Icons.phone, 'CONTATO'),
281     _buildButtonColumn(
282       color,
283       Icons.info_outline,
284       'SOBRE NÓS',
285       onTap: () {
286         Navigator.push(
287           context,
288           MaterialPageRoute(
289             builder: (_) => const SobreNosPage(),
290           ), // MaterialPageRoute
291         );
292       },
293     ),
294   ],
295 ), // Row
296

```

E para finalizar, o Texto institucional, onde é apresentada a proposta da empresa

const Padding(  
padding: EdgeInsets.all(32),  
child: Text(...)  
), Conteúdo estático

```

297 // TEXTO
298 const Padding(
299   padding: EdgeInsets.all(32),
300   child: Text(
301     'A Explore Mundo é uma agência de viagens especializada em experiências
302     inesquecíveis. Explore destinos incríveis, consulte pacotes exclusivos
303     e realize sua reserva com praticidade e segurança.',
304   ), // Text
305   ), // Padding
306   destinosPopularesSection(),
307 ],
308 ), // ListView
309 ); // Scaffold
310 }

```

A seguir, o método `destinosPopularesSection()` constrói uma **seção reutilizável** da tela que: Exibe um **título**, mostra uma **lista horizontal de cards**, permite **rolagem horizontal com barra visível**.

Cada card: Exibe imagem, título, local e avaliação, é **clicável**, redireciona para uma página de detalhes (`DestinoPage`) e trabalha com **dados dinâmicos** (`destinos`)

**return Padding(**

**padding: const EdgeInsets.symmetric(vertical: 24),** cria espaçamento acima e abaixo da seção e evita que o conteúdo fique colado em outros widgets. O uso de const melhora performance

**child: Column(**

**crossAxisAlignment: CrossAxisAlignment.start,** organiza os elementos verticalmente. `crossAxisAlignment.start`: Alinha tudo à esquerda.

**SizedBox(**

**height: 220,** obrigatório para listas horizontais

**Scrollbar(**

**controller: \_destinosScrollController,** exibe feedback visual de rolagem

**thumbVisibility: true,**

**trackVisibility: true,** barra sempre visível. Trilho visível. Usa o mesmo `ScrollController` da lista

```

316   Widget destinosPopularesSection() {
317     return Padding(
318       padding: const EdgeInsets.symmetric(vertical: 24),
319       child: Column([
320         // TÍTULO
321         const Padding(
322           padding: EdgeInsets.symmetric(horizontal: 24),
323           child: Text(
324             'Destinos populares', // título da seção
325             style: TextStyle(
326               fontSize: 18,
327               fontWeight: FontWeight.bold,
328             ), // TextStyle
329           ), // Text
330         ), // Padding
331         const SizedBox(height: 16),
332         // LISTA HORIZONTAL
333         SizedBox(
334           height: 220,
335           //Exibe barra de rolagem horizontal
336           child: Scrollbar(
337             controller: _destinosScrollController,
338             thumbVisibility: true, // ↗ barra sempre visível
339             trackVisibility: true, // ↗ trilho (opcional)
340           ),
341         ),
342       ],
343     );
344   }

```

## ListView.builder(

scrollDirection: Axis.horizontal,

itemCount: destinos.length, Usando o builder para criar os cards sob demanda. É mais performático para listas dinâmicas

### **Scroll horizontal**

Axis.horizontal Cada item ocupa uma largura fixa.

itemBuilder: (context, index) {

final destino = destinos[index]; Para cada índice: Obtém o mapa do destino. Evita repetição de código. Facilita leitura

return Container(

width: 160, Função para definir largura fixa dos cards. Assim garanto um layout consistente

left: index == 0 ? 24 : 12, Primeiro item → margem maior à esquerda

right: index == destinos.length - 1 ? 24 : 24, Último item → margem maior à direita

Resultado: lista não cola nas bordas da tela

InkWell(

borderRadius: BorderRadius.circular(16),

onTap: () { Função que detecta toque, adiciona efeito visual (ripple) e torna o card interativo

Navigator.push(

context,

MaterialPageRoute(

builder: (\_) => DestinoPage(destino: destino),

},

); Envia o mapa completo. Assim, a página de destino fica: genérica, reutilizável, desacoplada da Home

```

316   Widget destinosPopularesSection() {
343     trackVisibility: true, // 📺 trilho (opcional)
344     //Lista horizontal
345     child: ListView.builder(
346       controller: _destinosScrollController,
347       scrollDirection: Axis.horizontal,
348       itemCount: destinos.length,
349       itemBuilder: (context, index) {
350         final destino = destinos[index];
351
352         return Container(
353           width: 160,
354           margin: EdgeInsets.only(
355             left: index == 0 ? 24 : 12,
356             right: index == destinos.length - 1 ? 24 : 24,
357           ), // EdgeInsets.only
358           child: Inkwell(
359             borderRadius: BorderRadius.circular(16),
360             onTap: () {
361               //Envia o mapa completo .Página de destino fica genérica e reutilizável
362               Navigator.push(
363                 context,
364                 MaterialPageRoute(
365                   builder: (_) => DestinoPage(
366                     destino: destino,
367                   ), // DestinoPage
368                 ), // MaterialPageRoute
369               );
370             },

```

## Column(

**crossAxisAlignment: CrossAxisAlignment.start**, Conteúdo empilhado verticalmente e alinhado à esquerda.

## ClipRRect(

**borderRadius: BorderRadius.vertical(**

**top: Radius.circular(16),**

**),** ClipRRect para recorte da imagem, e manter bordas arredondadas apenas no topo

## Image.asset(

**destino['image']!,**

**fit: BoxFit.cover,**

**BoxFit.cover:** Preenche o espaço. Mantém proporção.

```

370 },
371 child: Card(
372   shape: RoundedRectangleBorder(
373     borderRadius: BorderRadius.circular(16),
374   ), // RoundedRectangleBorder
375   elevation: 4,
376   child: Column(
377     mainAxisAlignment: MainAxisAlignment.start,
378     children: [
379       // IMAGEM
380       ClipRRect(
381         borderRadius: const BorderRadius.vertical(
382           top: Radius.circular(16),
383         ), // BorderRadius.vertical
384         child: Image.asset(
385           destino['image']!,
386           height: 110,
387           width: double.infinity,
388           fit: BoxFit.cover,
389         ), // Image.asset
390       ), // ClipRRect
391

```

## Text

destino['title']!, Nome do destino. Destaque visual

## Row

children: [

Icon(Icons.star),

Text(destino['rating']!)

],

Ícone + nota. Reforça credibilidade. Informação rápida e escaneável

```
407     Text(                                destino['local']!,  
408         style: TextStyle(  
409             fontSize: 12,  
410             color: Colors.grey[600],  
411         )), // TextStyle  
412     ), // Text  
413  
414     const SizedBox(height: 8),  
415  
416     Row(                                children: [  
417         const Icon(  
418             Icons.star,  
419             color: Colors.orange,  
420             size: 16,  
421         )), // Icon  
422         const SizedBox(width: 4),  
423         Text(                                destino['rating']!,  
424             style: const TextStyle(fontSize: 12),  
425         )), // Text  
426     ],  
427     ), // Row  
428     ],  
429     ), // Column  
430     ), // Padding  
431     ],  
432     ), // Column  
433     ), // Padding
```

abaixo, segue o resultado visual:

## Explore Mundo

★ 4.9

Os melhores destinos para você



DESTINOS



PACOTES



CONTATO



SOBRE NÓS

A Explore Mundo é uma agência de viagens especializada em experiências inesquecíveis. Explore destinos incríveis, consulte pacotes exclusivos e realize sua reserva com praticidade e segurança.

### Destinos populares



Paris

França

★ 4.9



Roma

Itália

★ 4.8



Nova York

EUA

★ 4.7

Quando o usuário clica em alguma imagem do slideshow ou do card, é redirecionado para página de destino, que recebe todos os dados. Não depende de estado global

```
457 class DestinoPage extends StatelessWidget {  
458   final Map<String, String> destino;  
459  
460   const DestinoPage({super.key, required this.destino});  
461  
462   @override  
463   Widget build(BuildContext context) {  
464     return Scaffold(  
465       appBar: AppBar(  
466         title: Text(destino['title']!), //Título dinâmico  
467       ), // AppBar  
468  
469       body: ListView(  
470         children: [  
471           // IMAGEM DO DESTINO  
472           Image.asset(  
473             destino['image']!, //Imagem dinâmica  
474             height: 250,  
475             width: double.infinity,  
476             fit: BoxFit.cover,  
477           ), // Image.asset  
478  
479           // CONTEÚDO  
480           Padding(  
481             padding: const EdgeInsets.all(24),  
482             child: Column(  
483               crossAxisAlignment: CrossAxisAlignment.start,  
484               children: [  
485                 // TÍTULO  
486                 Text(  
487                   destino['title']!,  
488                   style: const TextStyle(  
489             ),  
490           ),  
491         ),  
492       ),  
493     );  
494   }  
495 }  
496  
497 // ignore_for_file: prefer_const_constructors
```

```
488 |         style: const TextStyle(  
489 |             fontSize: 26,  
490 |             fontWeight: FontWeight.bold,  
491 |                 ), // TextStyle  
492 |             ), // Text  
493 |  
494 |             const SizedBox(height: 8),  
495 |  
496 |             // AVALIAÇÃO  
497 |             Row(  
498 |                 children: const [  
★ 499 |                     Icon(Icons.star, color: Colors.orange),  
500 |                     SizedBox(width: 4),  
501 |                     Text(  
502 |                         '4.8 (1.254 avaliações)',  
503 |                         style: TextStyle(fontSize: 14),  
504 |                     ), // Text  
505 |                 ],  
506 |             ), // Row  
507 |  
508 |             const SizedBox(height: 24),  
509 |  
510 |             // DESCRIÇÃO  
511 |             const Text(  
512 |                 'Descrição',  
513 |                 style: TextStyle(  
514 |                     fontSize: 18,  
515 |                     fontWeight: FontWeight.bold,  
516 |                 ), // TextStyle  
517 |             ), // Text
```

```
517 ), // Text
518
519     const SizedBox(height: 8),
520
521     const Text(
522         'Este destino oferece experiências inesquecíveis, '
523         'paisagens deslumbrantes, cultura rica e opções incríveis '
524         'de lazer, gastronomia e hospedagem. Ideal para quem busca '
525         'momentos únicos e memoráveis.',
526         style: TextStyle(
527             fontSize: 16,
528             height: 1.5,
529         ), // TextStyle
530     ), // Text
531
532     const SizedBox(height: 32),
533
534     // BOTÃO RESERVAR
535     SizedBox(
536         width: double.infinity,
537         height: 50,
538         child: ElevatedButton(
539             //Simula reserva, Feedback imediato ao usuário
540             onPressed: () {
541                 ScaffoldMessenger.of(context).showSnackBar(
542                     const SnackBar(
543                         content: Text('Reserva iniciada com sucesso!'),
544                     ), // SnackBar
545                 );
546             },
547         ),
548     ),
549 
```

Resultado visual:

[!\[\]\(dacbe48c94a2ea5ede00a2df31864c5c\_img.jpg\) Paris](#)



## Paris

 4.8 (1.254 avaliações)

### Descrição

Este destino oferece experiências inesquecíveis, paisagens deslumbrantes, cultura rica e opções incríveis de lazer, gastronomia e hospedagem. Ideal para quem busca momentos únicos e memoráveis.

[Reservar agora](#)

Quando o usuário clica no botão Sobre nós, é redirecionado para página "Sobre a Explore o mundo"

```
class SobreNosPage extends StatelessWidget {
  const SobreNosPage({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Sobre a Explore Mundo'),
      ), // AppBar
      body: Padding(
        padding: const EdgeInsets.all(24),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: const [
            Text(
              'Quem somos',
              style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold),
            ), // Text
            SizedBox(height: 16),
            Text(
              'A Explore Mundo é uma agência de viagens especializada em criar experiências únicas e inesquecíveis, oferecendo pacotes personalizados com segurança e praticidade.',
              style: TextStyle(fontSize: 16),
            ), // Text
            SizedBox(height: 24),
            Text(
              'Nossa missão',
              style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
            ), // Text
            SizedBox(height: 8),
            Text(

```

## Quem somos

A Explore Mundo é uma agência de viagens especializada em criar experiências únicas e inesquecíveis, oferecendo pacotes personalizados com segurança e praticidade.

## Nossa missão

Conectar pessoas a experiências incríveis ao redor do mundo.