

Documentação do Projeto Flutter - Demonstração de Layout

Objetivos da Prática

- Instalar e configurar o Flutter SDK e o ambiente de desenvolvimento integrado (IDE) de acordo com as melhores práticas.
- Empregar widgets fundamentais, como `MaterialApp`, `Scaffold`, `AppBar`, `Text` e `RaisedButton`.
- Criar interfaces visuais atraentes e funcionais com Flutter.
- Utilizar o widget `ListView` para exibir e gerenciar listas de dados.
- Implementar funcionalidades personalizadas em widgets.

Estrutura do Projeto

Arquivo `main.dart`

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

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

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    // Função auxiliar para criar colunas de botões
    Column _buildButtonColumn(Color color, IconData icon, String label) {
      return Column(
        mainAxisAlignment: MainAxisAlignment.min,
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Icon(icon, color: color),
          Container(
            margin: const EdgeInsets.only(top: 8),
            child: Text(
              label,
              style: TextStyle(
                fontSize: 12,
                fontWeight: FontWeight.w400,
                color: color,
              ),
            ),
          ),
        ],
      );
    }
  }
}
```

```
    ),  
  ],  
);  
}
```

```
// Cor principal do tema  
Color color = Theme.of(context).primaryColor;
```

```
// Seção de título
```

```
Widget titleSection = Container(  
  padding: const EdgeInsets.all(32),  
  child: Row(  
    children: [  
      Expanded(  
        child: Column(  
          crossAxisAlignment: CrossAxisAlignment.start,  
          children: [  
            Container(  
              padding: const EdgeInsets.only(bottom: 8),  
              child: const Text(  
                'Oeschinen Lake Campground',  
                style: TextStyle(fontWeight: FontWeight.bold),  
              ),  
            ),  
            Text(  
              'Kandersteg, Switzerland',  
              style: TextStyle(color: Colors.grey[500]),  
            ),  
          ],  
        ),  
      ),  
      Icon(  
        Icons.star,  
        color: Colors.red[500],  
      ),  
      const Text('41'),  
    ],  
  ),  
);
```

```
// Seção de botões
```

```
Widget buttonSection = Row(  
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
  children: [  
    _buildButtonColumn(color, Icons.call, 'CALL'),  
    _buildButtonColumn(color, Icons.near_me, 'ROUTE'),  
    _buildButtonColumn(color, Icons.share, 'SHARE'),  
  ],  
);
```

```

);

// Seção de texto
Widget textSection = Container(
  padding: const EdgeInsets.all(32),
  child: const Text(
    'O Lago Oeschinen fica aos pés do Blümlisalp nos Alpes Berneses. Situado a 1.578 metros acima do nível do mar, é um dos lagos alpinos mais amplos. Um passeio de teleférico a partir de Kandersteg, seguido por meia hora de caminhada por pastagens e floresta de pinheiros, leva você ao lago, que aquece até 20 graus Celsius no verão. As atividades desfrutadas aqui incluem remo e andar no tobogã de verão.',
    softWrap: true,
  ),
);

// Estrutura do aplicativo
return MaterialApp(
  title: 'Demonstração de layout Flutter',
  home: Scaffold(
    appBar: AppBar(
      title: const Text('Demonstração de layout Flutter'),
    ),
    body: ListView(
      children: [
        Image.asset(
          'images/lake.jpg',
          width: 600,
          height: 240,
          fit: BoxFit.cover,
        ),
        titleSection,
        buttonSection,
        textSection,
      ],
    ),
  ),
);
}
}

```

Configuração do pubspec.yaml

Para adicionar a imagem ao projeto, edite o arquivo `pubspec.yaml` localizado na raiz do projeto:

```

flutter:
  assets:

```

- images/lake.jpg

Após editar o arquivo, execute o comando abaixo para aplicar as mudanças:

```
flutter pub get
```

Estrutura de Pastas

Certifique-se de que o arquivo da imagem esteja localizado no seguinte diretório:

```
my_layout_demo/  
├── images/  
│   └── lake.jpg  
├── lib/  
│   └── main.dart  
├── pubspec.yaml  
└── ...
```

Funcionalidades e Widgets Utilizados

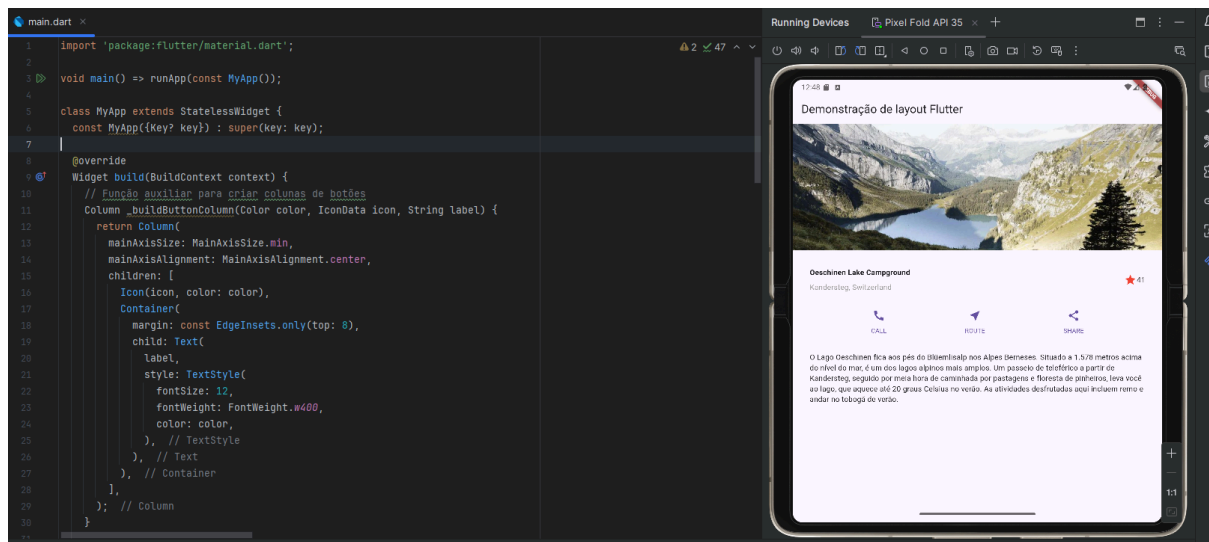
1. **Imagem:**
 - Utiliza `Image.asset` para carregar a imagem e `BoxFit.cover` para ajustar o tamanho.
2. **Seção de Título:**
 - Usa `Row` e `Column` para organização e alinhamento dos textos e ícones.
3. **Seção de Botões:**
 - Implementada com um método auxiliar `_buildButtonColumn`.
4. **Seção de Texto:**
 - Mostra informações com o widget `Text`.
5. **ListView:**
 - Garante a rolagem em dispositivos com telas menores.

Testando o Aplicativo

Para executar o aplicativo, use o comando:

```
flutter run
```

Um dispositivo físico ou emulador deve estar conectado.



Resultados Esperados

- Uma interface responsiva que suporta rolagem em dispositivos menores.
- Seções claramente organizadas: imagem, título, botões e texto.
- Utilização eficiente de widgets fundamentais do Flutter.

Este documento detalha a implementação do projeto de Demonstração de Layout no Flutter, alinhando-se aos objetivos da prática do 2º nível de conhecimento. RPG0024 - Posso criar um App de outra forma.