



Estácio

RPG 0024 - Vamos Criar um App

- Nome: Aliffer Farias
- Matrícula: 202302439251
- Turma: 2023.

Relatório de Conclusão: Ambiente de Desenvolvimento Flutter

Introdução: O presente relatório descreve as etapas e resultados obtidos na configuração do ambiente de desenvolvimento Flutter, abrangendo desde a instalação das ferramentas até a execução de um projeto básico utilizando widgets fundamentais do framework. O objetivo principal foi garantir a preparação de um ambiente eficiente, seguro e funcional para o desenvolvimento de aplicativos móveis multiplataforma.

1. Preparação do Ambiente

Material Utilizado:

- **Flutter SDK:** Versão estável mais recente foi baixada e instalada conforme o sistema operacional utilizado (Windows, Linux ou macOS).
- **Git:** Utilizado para controle de versionamento e integração contínua.
- **IntelliJ IDEA e Android Studio:** Instalação dos editores de texto e IDE para facilitar o processo de codificação, teste e depuração.
- **Emulador Android:** Configurado para simular dispositivos Android, garantindo que o ambiente estivesse apto a rodar e testar os aplicativos.

Procedimentos Realizados:

- **Instalação do Flutter:** Todas as etapas de download, extração e configuração de variáveis de ambiente foram seguidas conforme cada sistema operacional. O

comando `flutter doctor` foi executado com sucesso, e qualquer dependência pendente foi resolvida.

- **Instalação do IntelliJ IDEA e Android Studio:** Foram instalados e configurados adequadamente em todas as plataformas testadas (Windows, Linux, macOS). O **AVD Manager** foi configurado para simular diferentes dispositivos móveis.

Resultados: O ambiente foi configurado com sucesso, conforme esperado, sem pendências ou erros de instalação. A preparação do sistema permite agora o desenvolvimento ágil e sem interrupções, garantindo flexibilidade e suporte tanto para Android quanto para iOS.

2. Desenvolvimento de Aplicações com Widgets Flutter Básicos

Material Utilizado:

- **Editor de texto ou IDE:** IntelliJ IDEA e VS Code.
- **Flutter SDK:** Framework utilizado para criação dos aplicativos.
- **Simulador Android:** Para verificar o funcionamento do código em ambiente de teste.

Procedimentos Realizados:

- Foi criado um novo projeto Flutter utilizando o comando `flutter create my_app`.
- No arquivo `main.dart`, foram introduzidos os widgets básicos como `MaterialApp`, `Scaffold` e `AppBar`, que compõem a estrutura inicial do aplicativo.
- O código foi testado e rodado com sucesso no emulador, exibindo o texto inserido no widget `Text`, conforme esperado.
- O comando `flutter run` foi executado e confirmou que o aplicativo estava funcionando corretamente.

Código Utilizado:

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

void main() {
  return runApp(
    MaterialApp(
      home: StatelessWidgetExemplo("Olá Flutter - MaterialApp"),
    )
  );
}

class StatelessWidgetExemplo extends StatelessWidget {
  final String _appBarTitle;
  StatelessWidgetExemplo(this._appBarTitle) : super();
```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(_appBarTitle),
    ),
    body: Center(
      child: Text('Macoratti .net'),
    ),
  );
}
}

```

Resultados: O aplicativo foi executado com sucesso no emulador Android, exibindo a estrutura básica com a barra de navegação (AppBar) e o conteúdo centralizado. A estrutura de widgets foi compreendida e implementada conforme esperado.

3. Criação de Layouts com Widgets Básicos

Material Utilizado:

- **Editor de Texto ou IDE:** IntelliJ IDEA.
- **Simulador Android:** Para testar o layout gerado.

Procedimentos Realizados:

- Foi criado um layout básico utilizando a combinação de widgets `Column`, `Row`, `Icon` e `Text`.
- O código foi composto por três colunas organizadas horizontalmente em uma linha (`Row`), cada uma contendo um ícone e um texto.
- A execução foi bem-sucedida no simulador Android, exibindo corretamente os ícones e textos organizados em uma estrutura horizontal.

Código Utilizado:

```

dart
Copiar código
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    Column(
      children: <Widget>[Icon(Icons.call), Text('Call')],
    ),
    Column(
      children: <Widget>[Icon(Icons.directions), Text('Route')],
    ),
    Column(
      children: <Widget>[Icon(Icons.share), Text('Share')],
    ),
  ],
)

```

Resultados: O layout foi exibido corretamente no emulador, com os widgets organizados de maneira consistente, utilizando a estrutura `Row` e `Column`. O resultado final foi satisfatório e alinhado com o que se esperava.

Conclusão: Com a finalização destas microatividades, concluímos com êxito a configuração do ambiente de desenvolvimento Flutter e a implementação de funcionalidades básicas utilizando widgets. A estrutura montada está pronta para evoluir para projetos mais complexos, garantindo que tanto o ambiente quanto os conceitos fundamentais foram devidamente implementados e compreendidos.

Este processo proporcionou uma base sólida para o desenvolvimento de aplicativos multiplataforma, com uma configuração robusta do ambiente e domínio dos widgets básicos do Flutter.

Recomendações Finais:

- Manter o ambiente atualizado, especialmente as versões do Flutter SDK e do Android Studio, para garantir compatibilidade com futuras implementações.
- Continuar praticando a criação de layouts e a utilização de widgets mais avançados para explorar todo o potencial do Flutter.