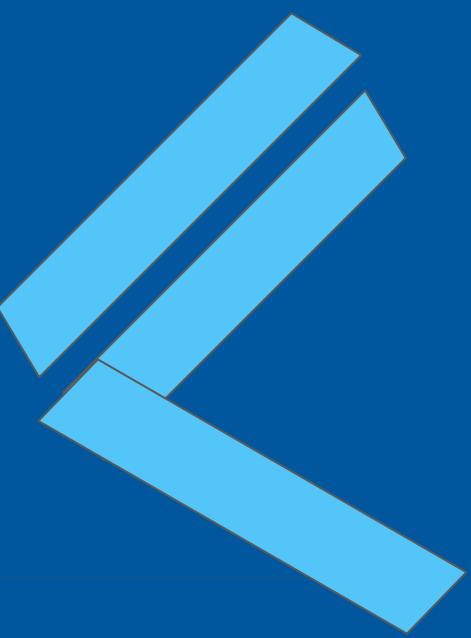


Desenvolvimento
Mobile 1
Aula 09

Prof. Me Daniel Vieira

SENAI

Agenda

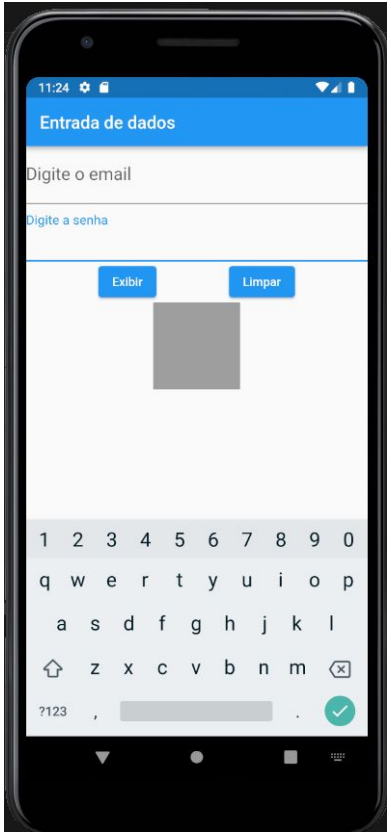
1 - TextField Widget
2 - Criando Aplicativo

Flutter Widgets

```
/*
onChanged:

Callback acionado toda vez que o texto dentro do campo muda. O valor inserido pelo usuário é recebido como parâmetro (text) e, nesse caso, é exibido no console com o comando print.
decoration:

Personaliza a aparência do TextField. Aqui está sendo usado o widget InputDecoration com:
labelText: Exibe um rótulo acima do campo de texto quando ele está em foco ou preenchido.
border: Define um contorno ao redor do campo de texto usando OutlineInputBorder.
*/

TextField(
  onChanged: (text) {
    print('Texto inserido: $text');
  },
  decoration: InputDecoration(
    labelText: 'Digite algo',
    border: OutlineInputBorder(),
  ),
)
```

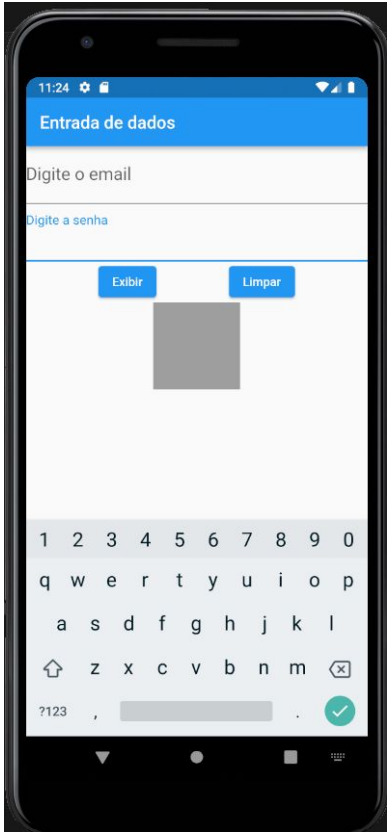
Flutter Widgets

```
1. import 'package:flutter/material.dart';
 2. /* Função main():
 3. A função main() chama o método runApp() para inicializar o aplicativo. O widget principal (Telaprincipal) é
passado como parâmetro para o runApp().
 4. */
 5. void main()
 6.  {
 7.  runApp(Telaprincipal());
 8. }
 9. /*
10. Classe Telaprincipal:
11. Esta classe é o ponto de entrada do aplicativo e é um StatelessWidget, o que significa que ela não possui
estado. Dentro do método build(), o widget MaterialApp é retornado, com a tela principal sendo o Campotexto.
12. */
13. class Telaprincipal extends StatelessWidget {
14.  const Telaprincipal({super.key});
15.
16.  @override
17. Widget build(BuildContext context) {
18. return MaterialApp(
19. home: Campotexto() ,
20. );
21. }
22. }
23.
```



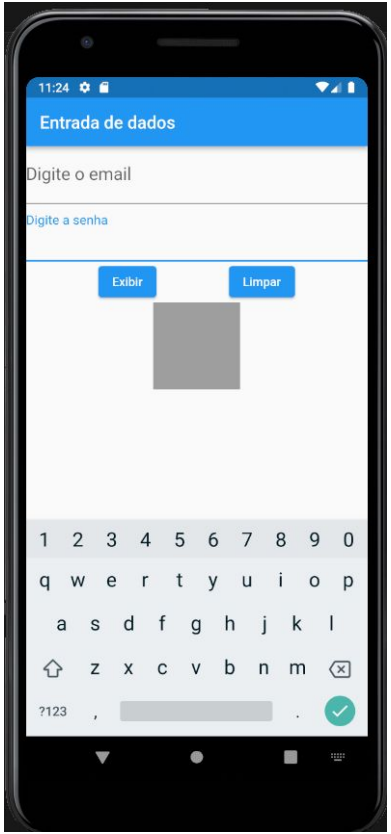
Flutter Widgets

```
24. class Campotexto extends StatefulWidget {
25. Campotexto({super.key});
26. @override
27.  State<Campotexto> createState() => CampotextoState();
28. }
29.
30. class CampotextoState extends State<Campotexto> {
31. // variavel do tipo TextEditing controler para armazenar o que é digitado no campo texto
32. TextEditingController ctexto = TextEditingController();
33. @override
34. Widget build(BuildContext context) {
35. return Scaffold(
36.       appBar: AppBar(
37.        title: Text("Aplicativo Caixa de texto"),
38.       ),
39.       body: Column(
40.        children: [
41.         // Campo texto
42. /*
43. Classe Campotexto:
44. Esta classe é um StatefulWidget, pois ela possui um estado mutável (no caso, o valor do texto inserido no
campo de texto). A criação do estado é feita pela classe CampotextoState.
45. Dentro da classe CampotextoState, temos o controlador de texto (TextEditingController), que é utilizado
para manipular o conteúdo do campo de texto.
46. */
47. /* Uso do TextField:
48. O TextField é utilizado para capturar o texto digitado pelo usuário. No seu código, temos dois campos de
texto:
49. O primeiro campo é utilizado para capturar o nome do usuário e é controlado pela variável ctexto do tipo
TextEditingController.
```



Flutter Widgets

```
50. O segundo campo é um campo para capturar um número (porém, não há controle para ele no momento).
51. Botão "Verificar":
52. O botão chama a função print(ctexto.text), que exibe o texto digitado no primeiro campo no terminal.
53. */
54.         TextField(
55.         keyboardType: TextInputType.name,
56.         decoration: InputDecoration(labelText: "Digite seu nome"),
57.         // On changed exibe no terminal tudo que for digitado
58.        /* onChanged: (String texto){
59.          print(texto);
60.         },*/
61.         /*onSubmitted: (String texto){
62.           print(texto);
63.         },
64.         */
65.         controller: ctexto,
66.         ),
67.         ElevatedButton(onPressed: (){
68.            print(ctexto.text);
69.         }, child: Text("Verificar")),
70.         TextField(
71.         keyboardType: TextInputType.url,
72.         decoration: InputDecoration(labelText: "Digite um numero"),
73.         ),
74.        ],
75.       ),
76.     );
77.   }
78. }
79.
```



Flutter Widgets

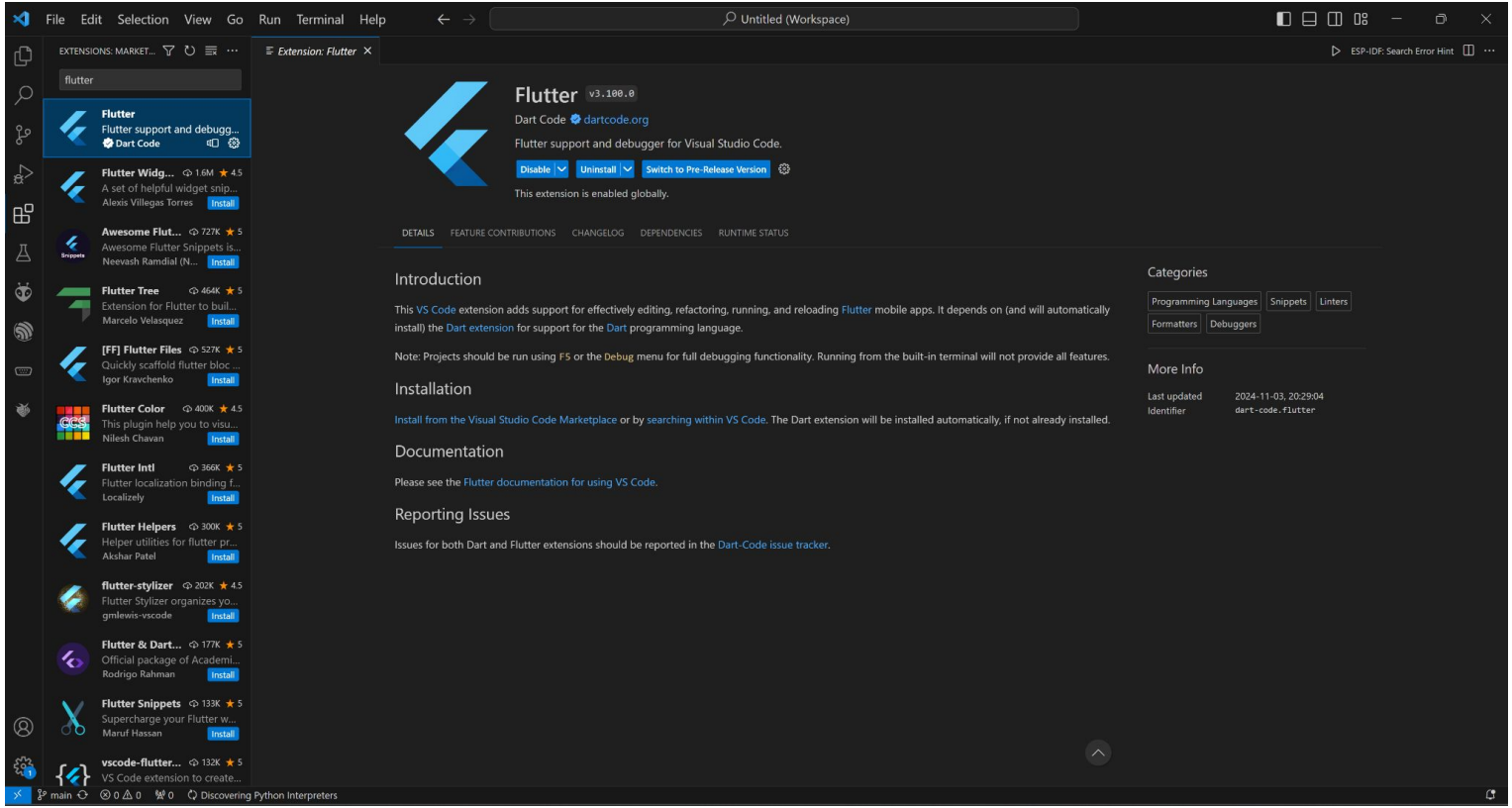
Criando projeto no VSCODE

Para criar um projeto de aplicativo mobile utilizando o framework Flutter com a IDE VSCode é necessário seguir os passos listados a seguir:

1º Abrir o VSCode

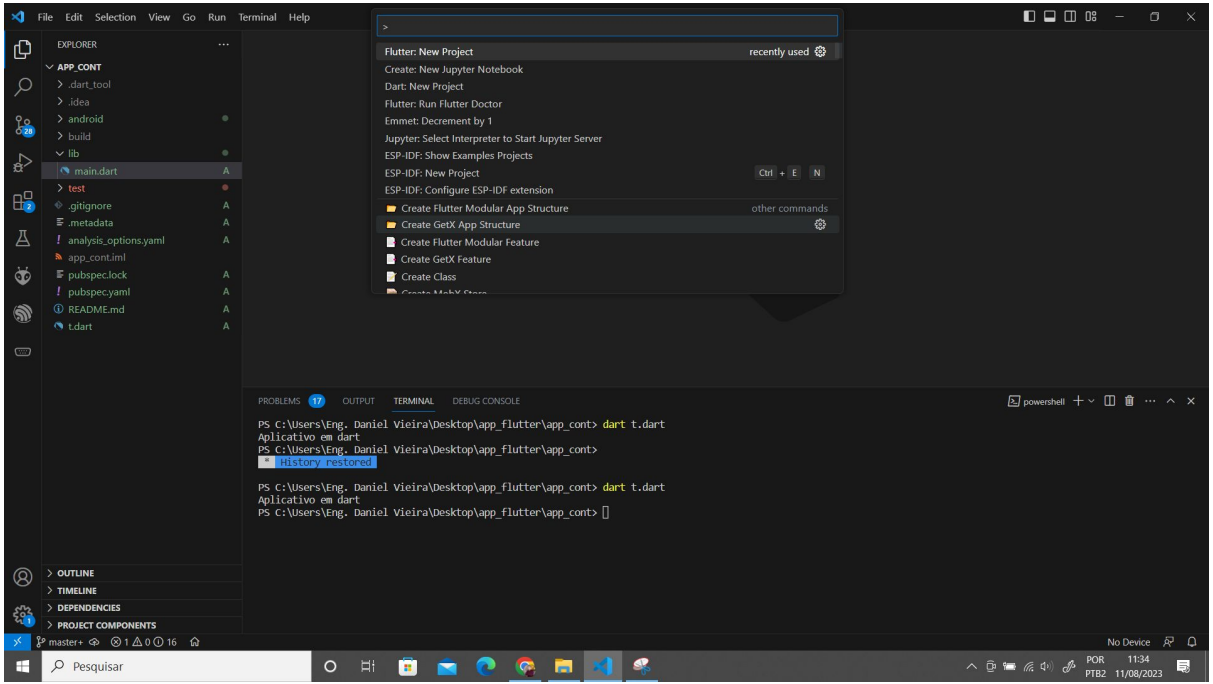
2º Instalar os plugins.

Criando projeto no VSCODE



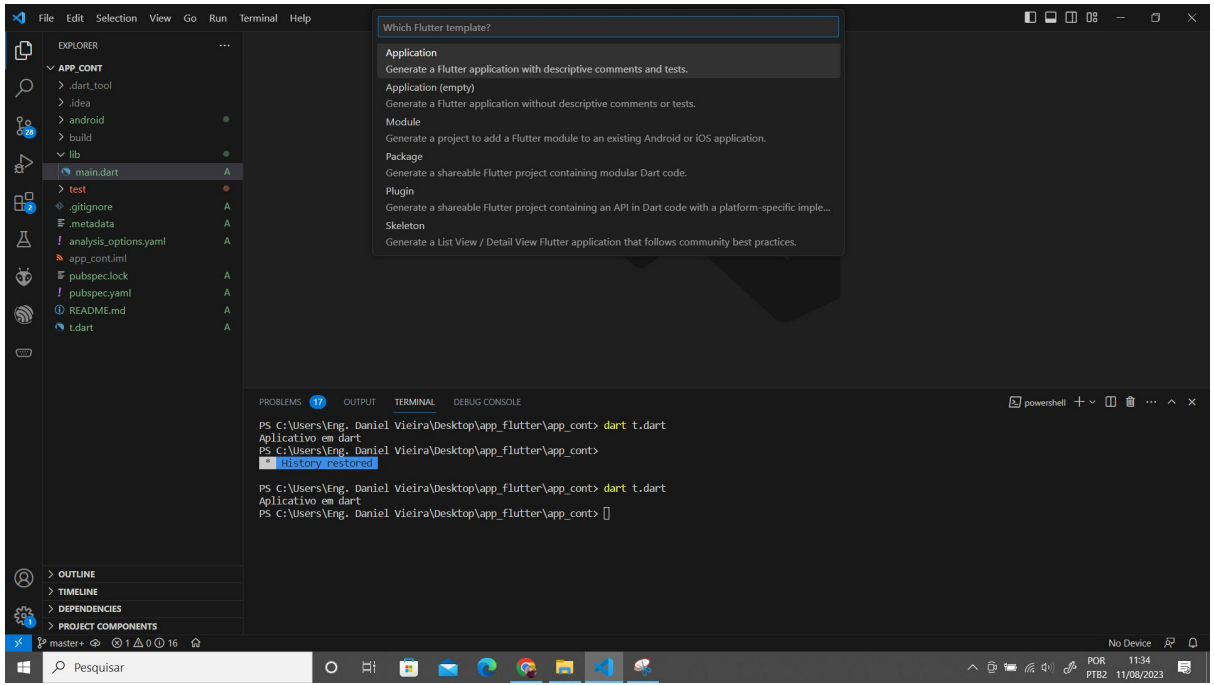
Criando projeto no VSCODE

3º Apertar a tecla F1, após apertar em F1 clicar em Flutter New Project conforme



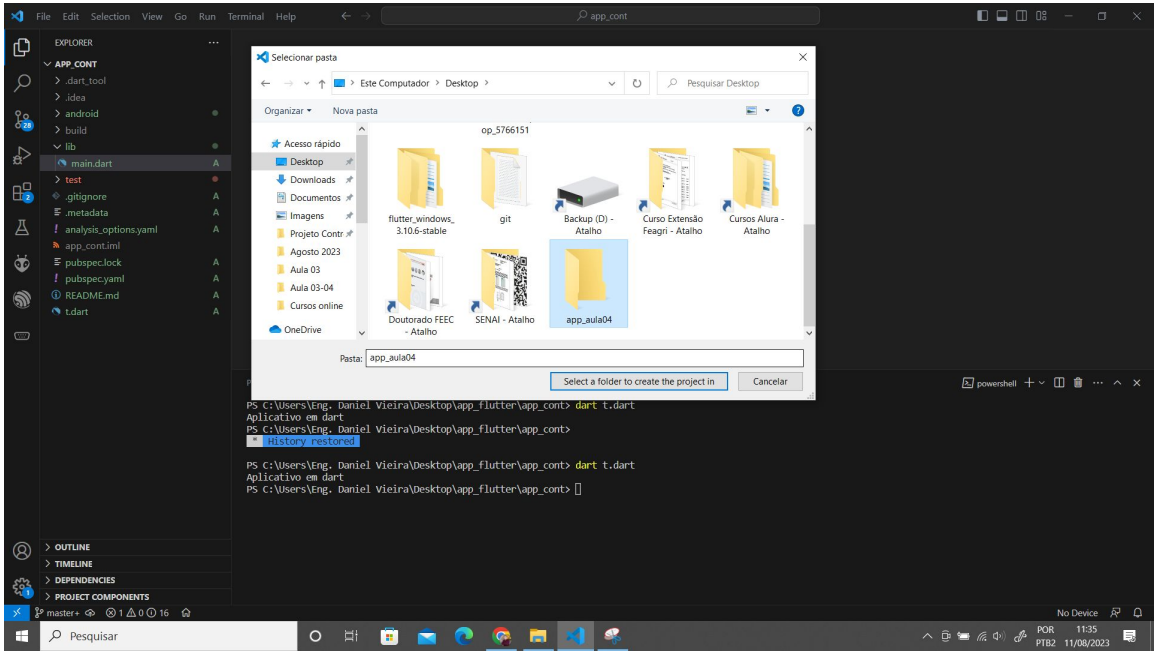
Criando projeto no VSCODE

4º Selecionar a opção Application conforme a figura abaixo



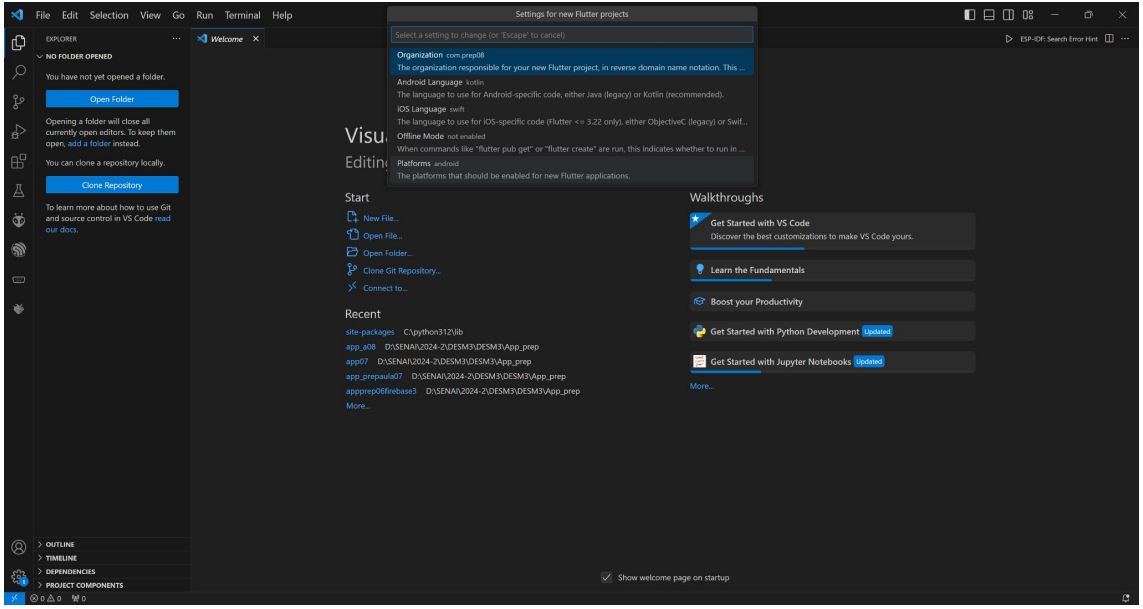
Criando projeto no VSCODE

5º Selecionar uma pasta para salvar o projeto conforme a Figura. A pasta não deve começar com letras maiúsculas, números, não pode ter espaço no nome da pasta



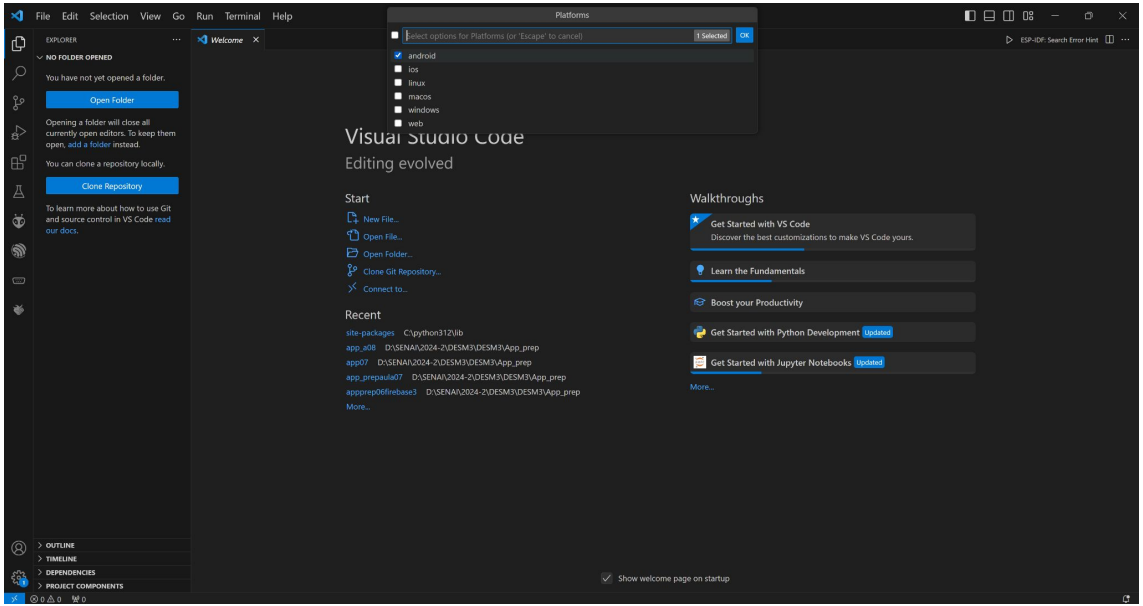
Criando projeto no VSCODE

6º Clicar na engrenagem para configurar as plataformas mobile que o projeto será executado conforme a Figura



Criando projeto no VSCODE

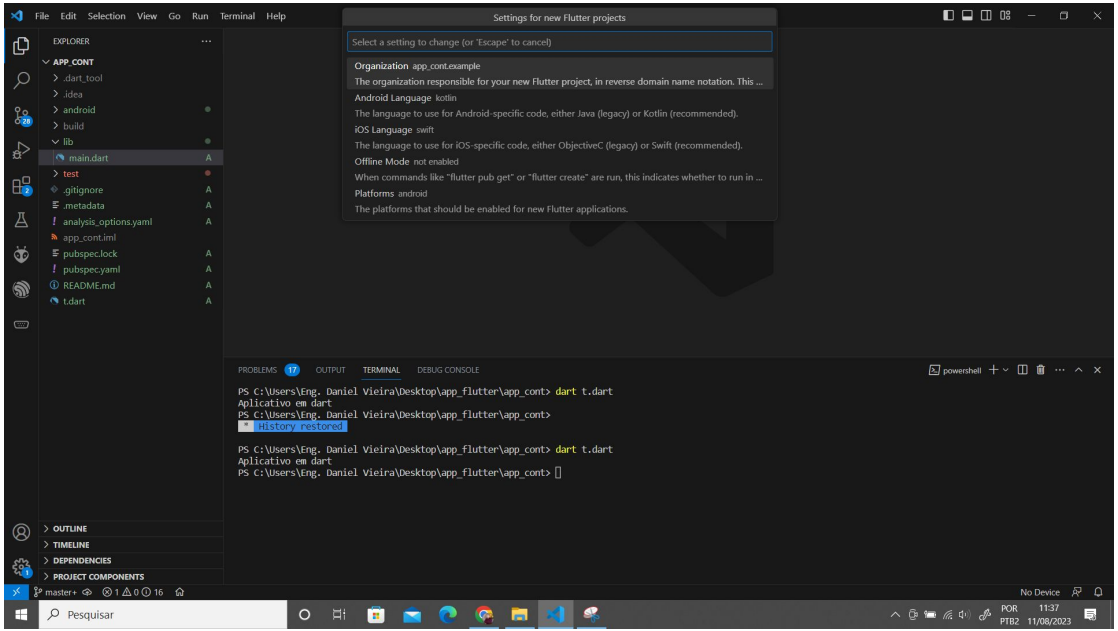
7º Selecionar a opção Android, realizada essa opção apertar enter conforme a Figura



Criando projeto no VSCODE

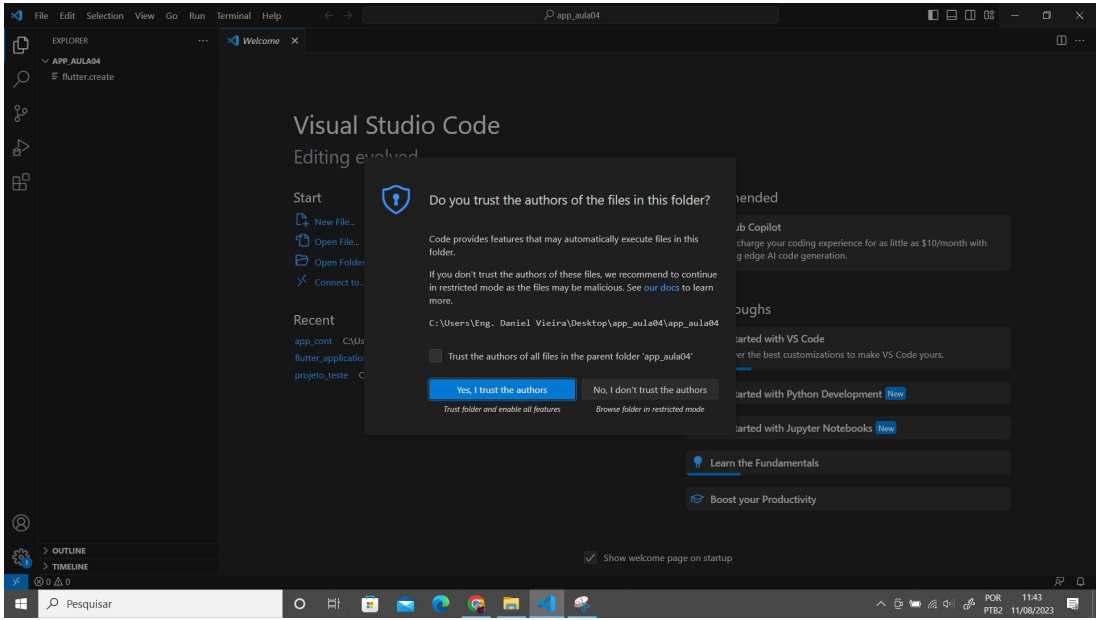
8º Passo Clicar em Organization conforme a Figura 48

Organization é o parâmetro que indica o domínio da empresa que desenvolveu o app. É um parâmetro importante para publicação do APP em lojas virtuais.



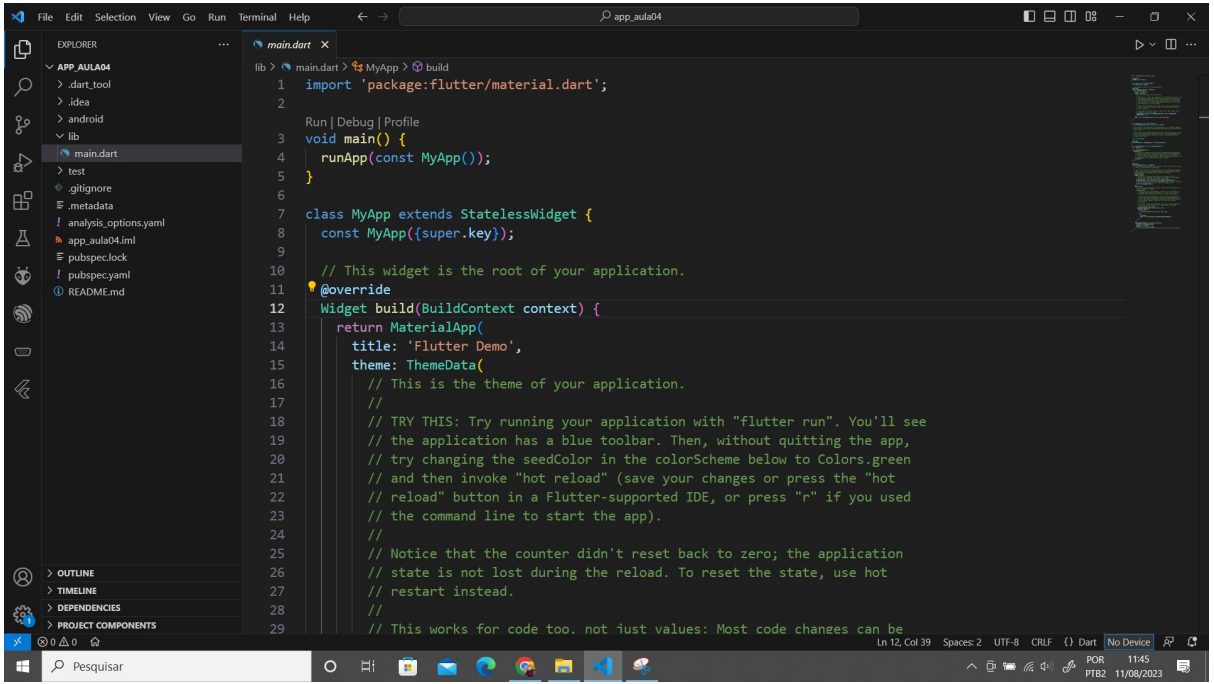
Criando projeto no VSCODE

Após realizar as configurações, apertar Esc e enter
O projeto Flutter será criado Marcar a opção Yes, I Trust the authors conforme a Figura.



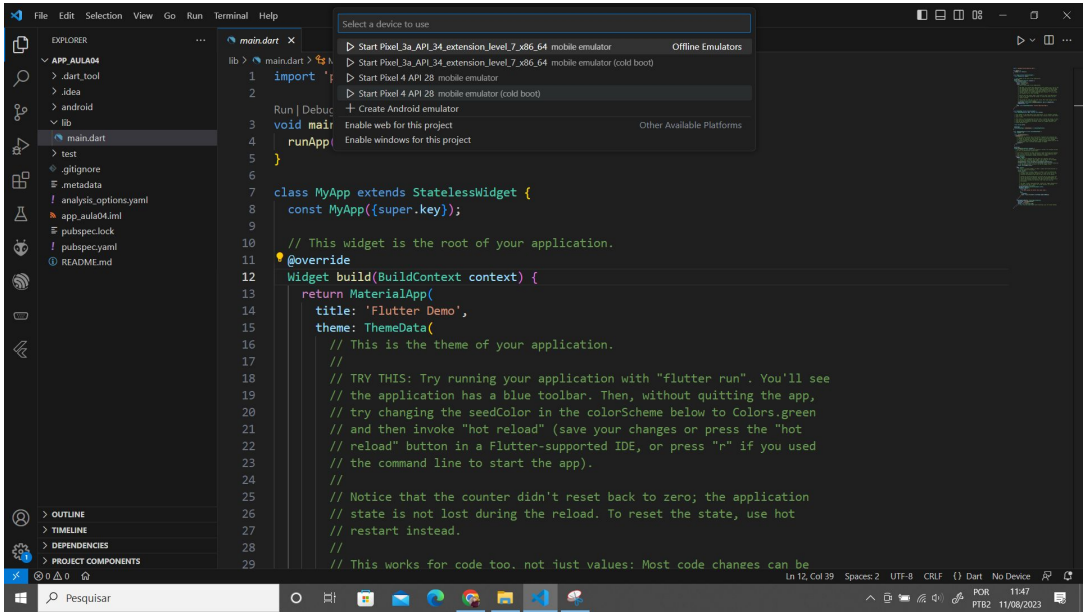
Criando projeto no VSCODE

Para executar o aplicativo no emulador o primeiro passo é clicar em iremos escolher o device para emular nosso telefone, clicar em No Device e selecionar o emulador desejado conforme a Figura.



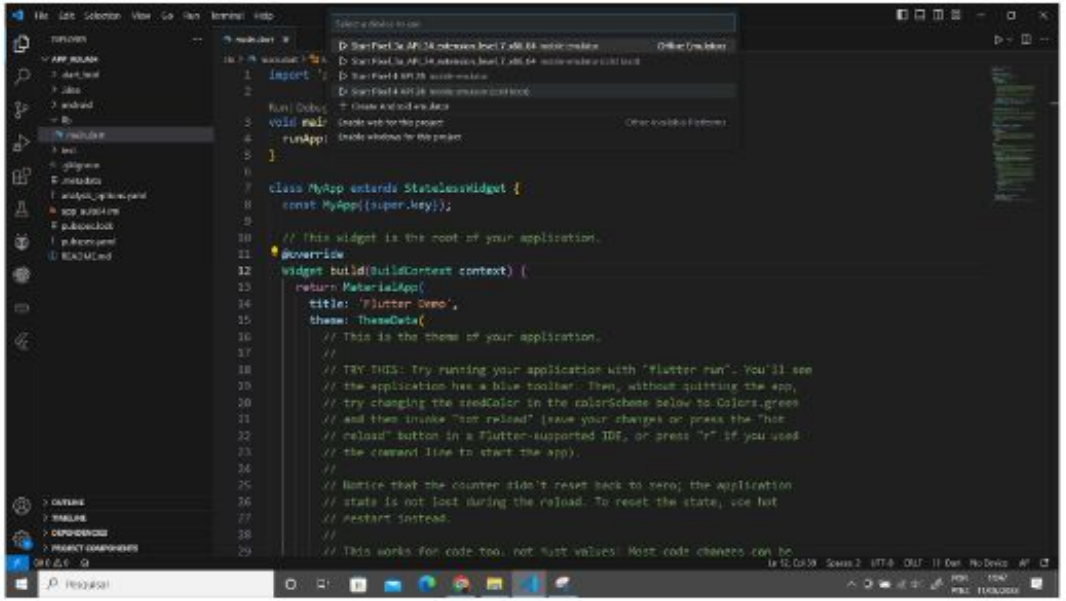
Criando projeto no VSCODE

Após clicar em No Device no editor VSCode irá aparecer para escolher o emulador para executar o aplicativo conforme a Figura, caso não apareça nenhum emulador, então é necessário criá-lo.



Criando projeto no VSCODE

Na Figura, após selecionar o device ele irá aparecer na tela

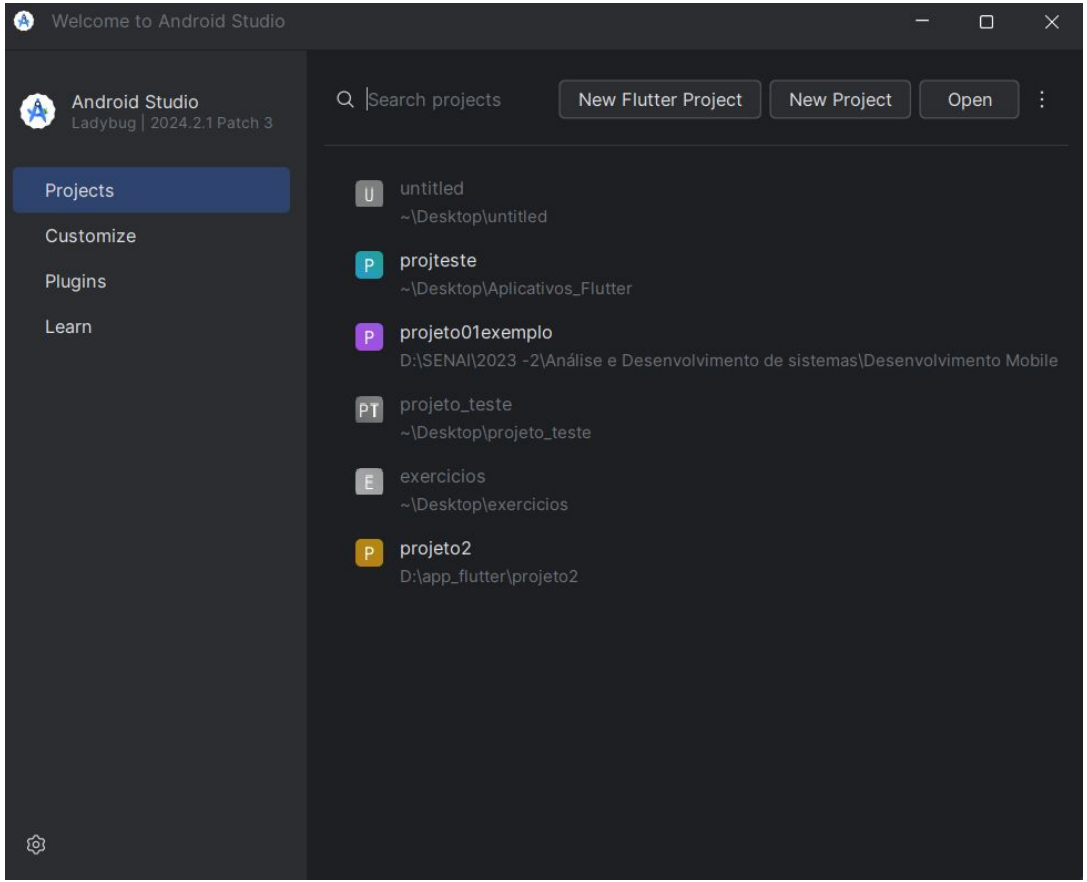


Criando emulador Android

Uma forma muito eficiente e rápida de testar os aplicativos desenvolvidos é utilizando o emulador do Android Studio, para isso é necessário criá-lo no Android Studio conforme os passos a seguir :

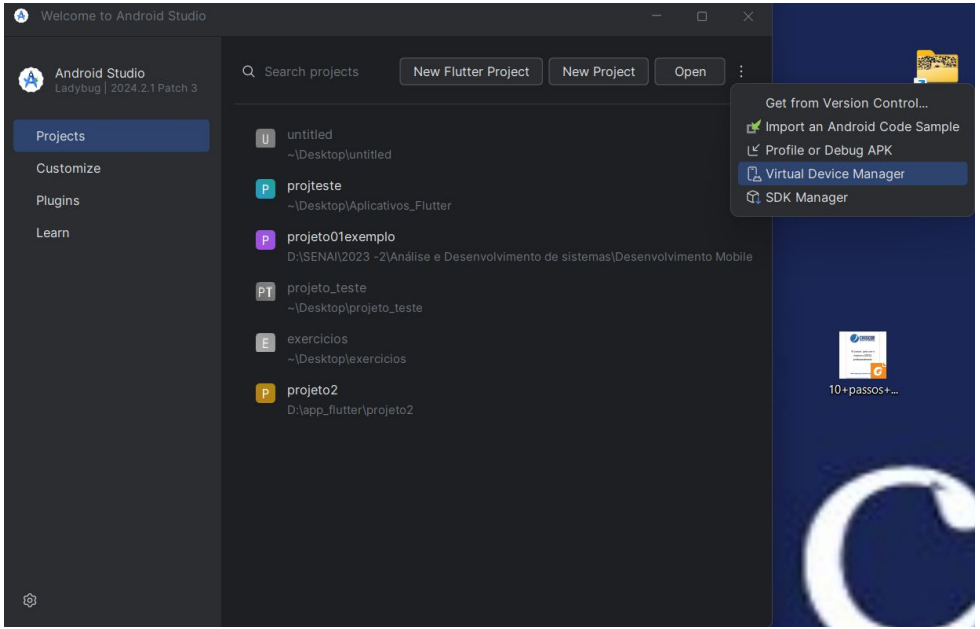
1º Abrir o Android Studio conforme a Figura 54 e clicar nos três pontos ao lado do botão Open

Criando emulador Android



Criando emulador Android

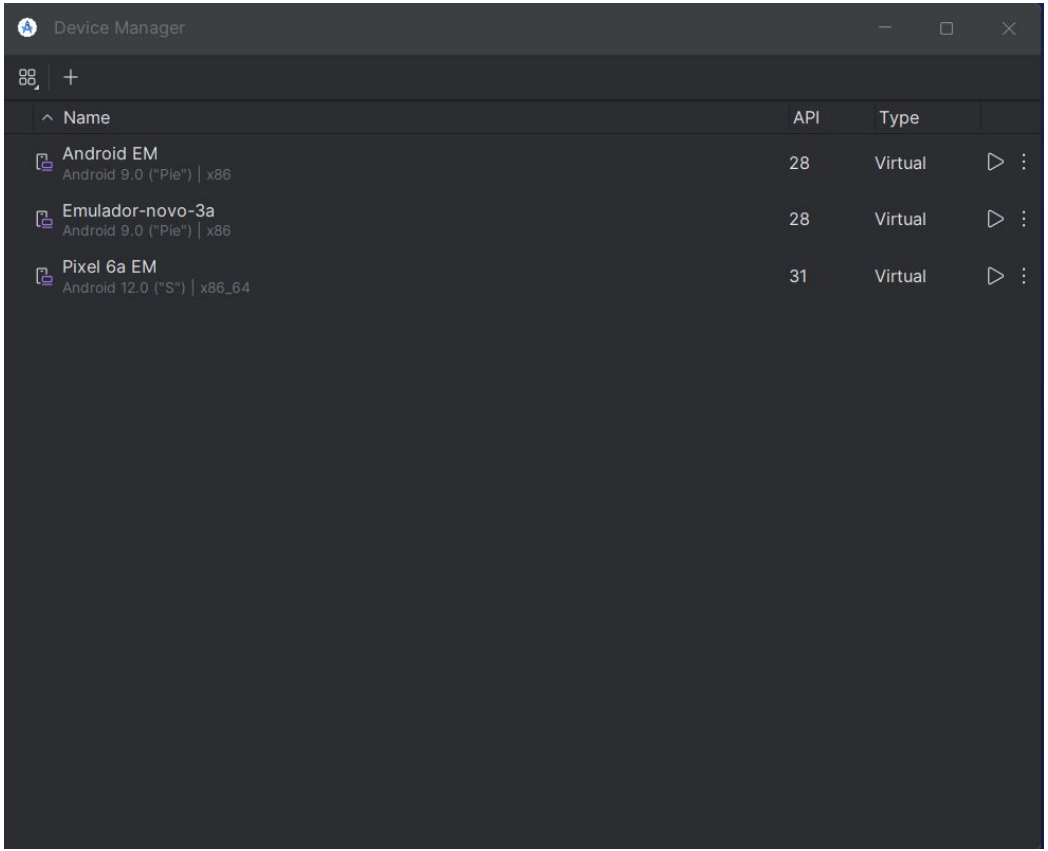
Na Figura após clicar nos três pontos ao lado do botão open, clicamos em Virtual Device Manager



Criando emulador Android

Após clicar em Virtual Device Manager, a tela com os emuladores criados irá aparecer conforme a Figura.

Criando emulador Android

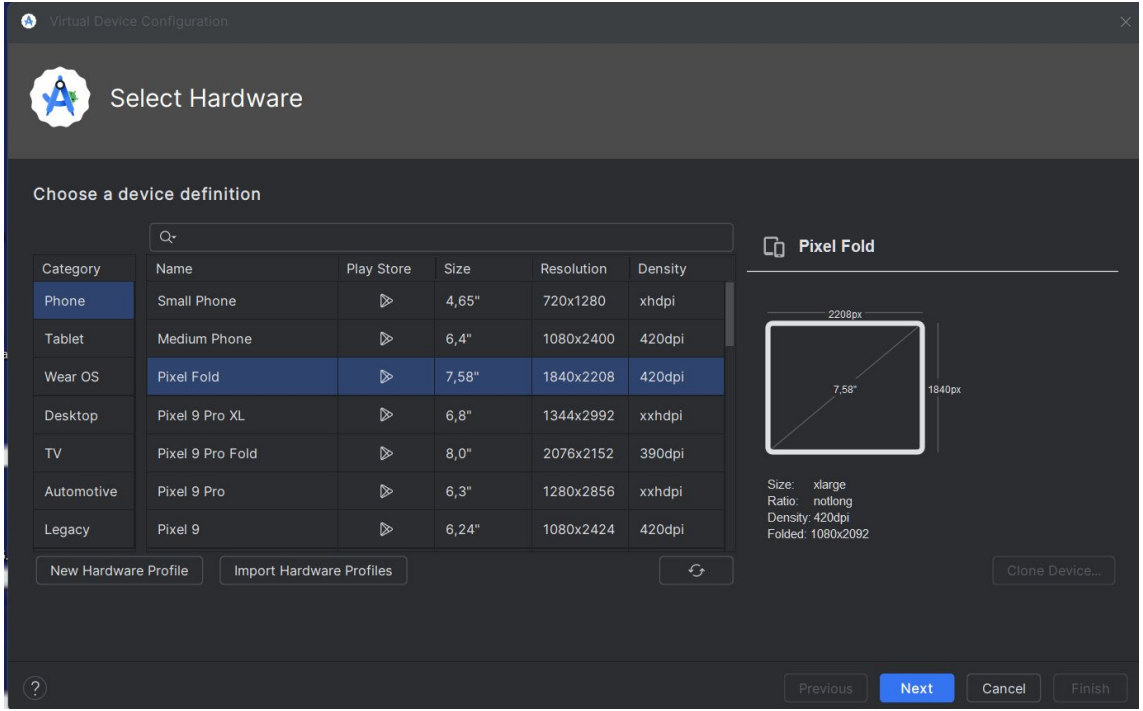


Criando emulador Android

Para criar um novo emulador é necessário clicar no ícone +

Após clicar no ícone + para criar um novo emulador irá aparecer a lista dos dispositivos emuladores disponíveis para serem escolhidos conforme a Figura

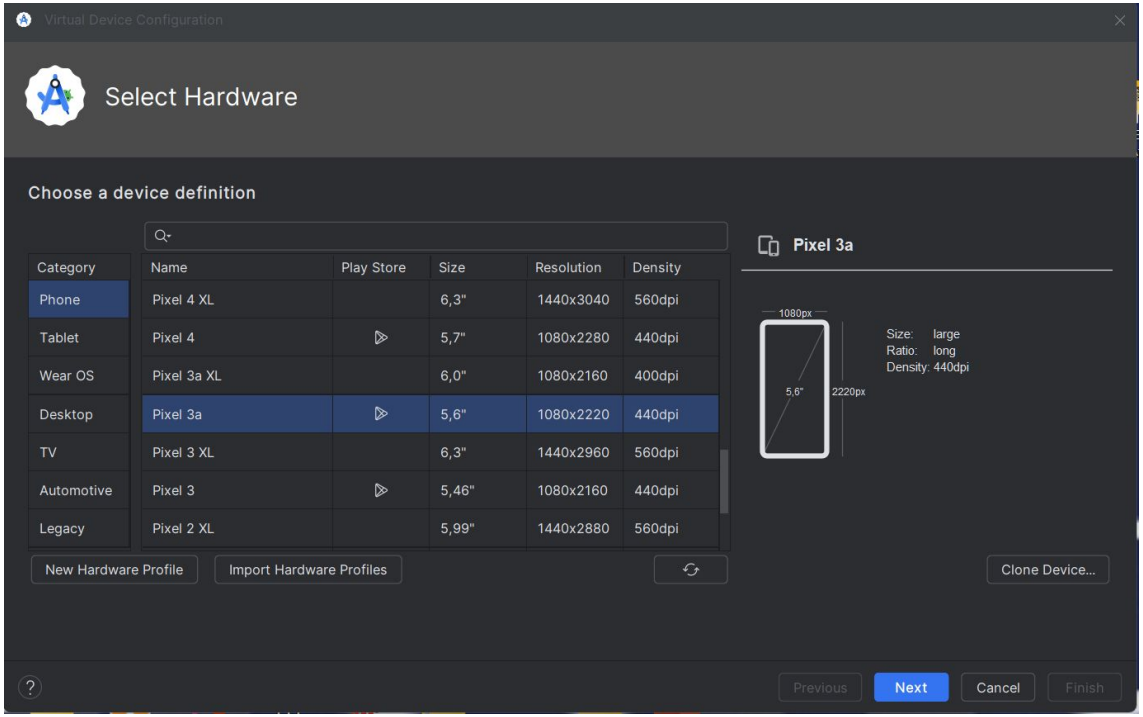
Criando emulador Android



Criando emulador Android

Iremos escolher o emulador Pixel 3a conforme a Figura, ou qualquer outro emulador que esteja com o ícone da Play Store ativado, para caso nós desejemos hospedar o aplicativo na Play Store.

Criando emulador Android

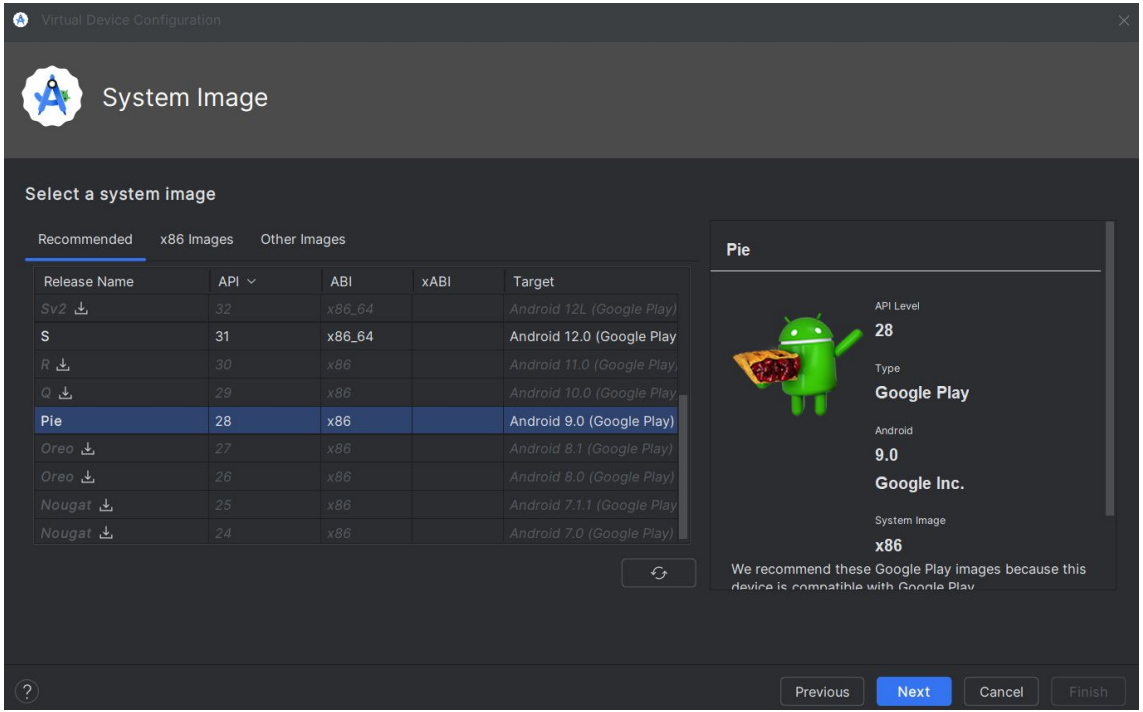


Criando emulador Android

Com o emulador Pixel 3a escolhido, o próximo passo será a escolha da versão do sistema operacional conforme a Figura.

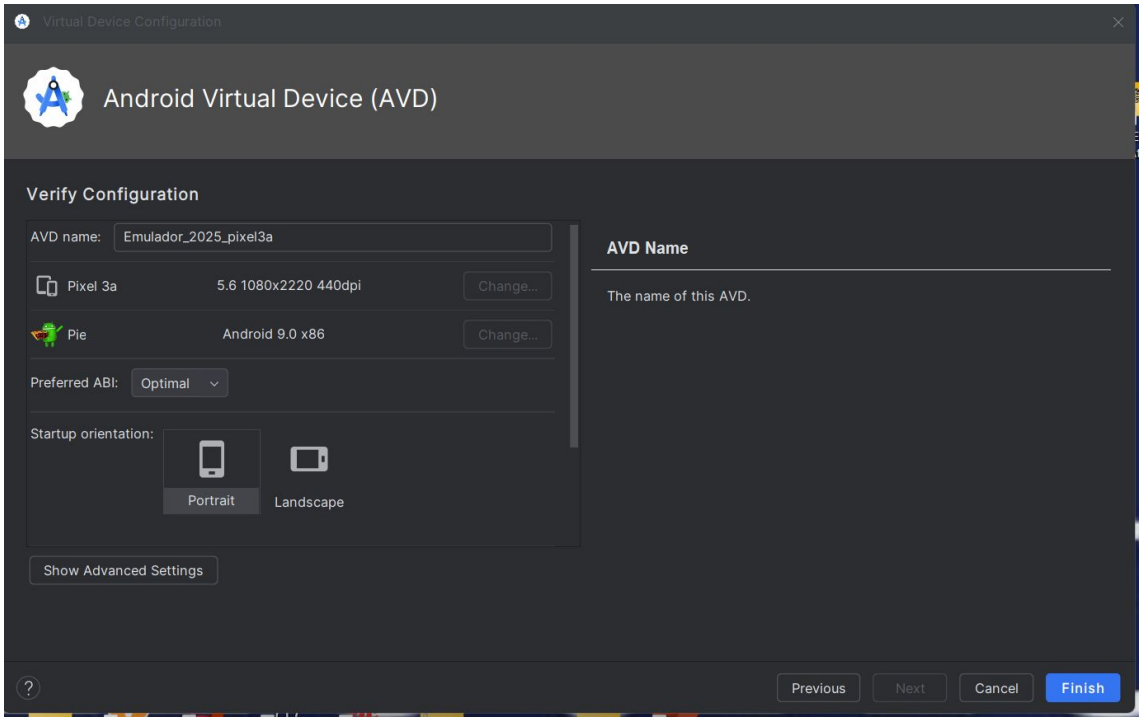
A versão escolhida será a versão Pie por ser mais leve para ser executada em nosso dispositivo.

Criando emulador Android



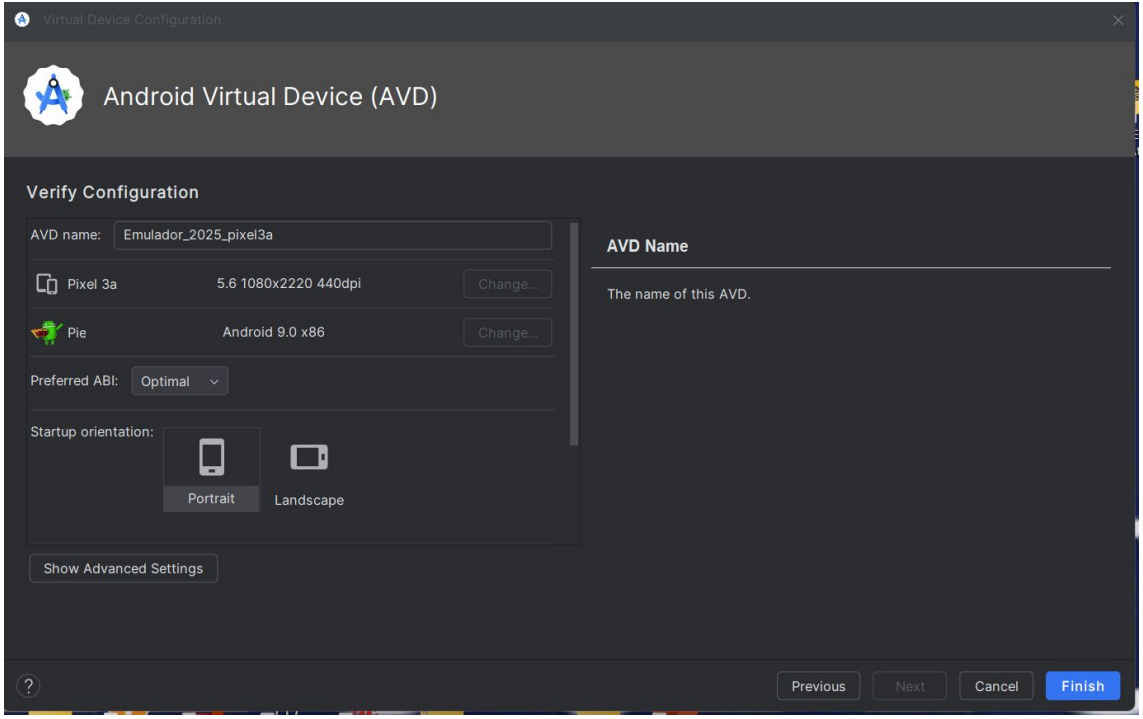
Criando emulador Android

Clicar em Next e nomear o emulador conforme a Figura



Criando emulador Android

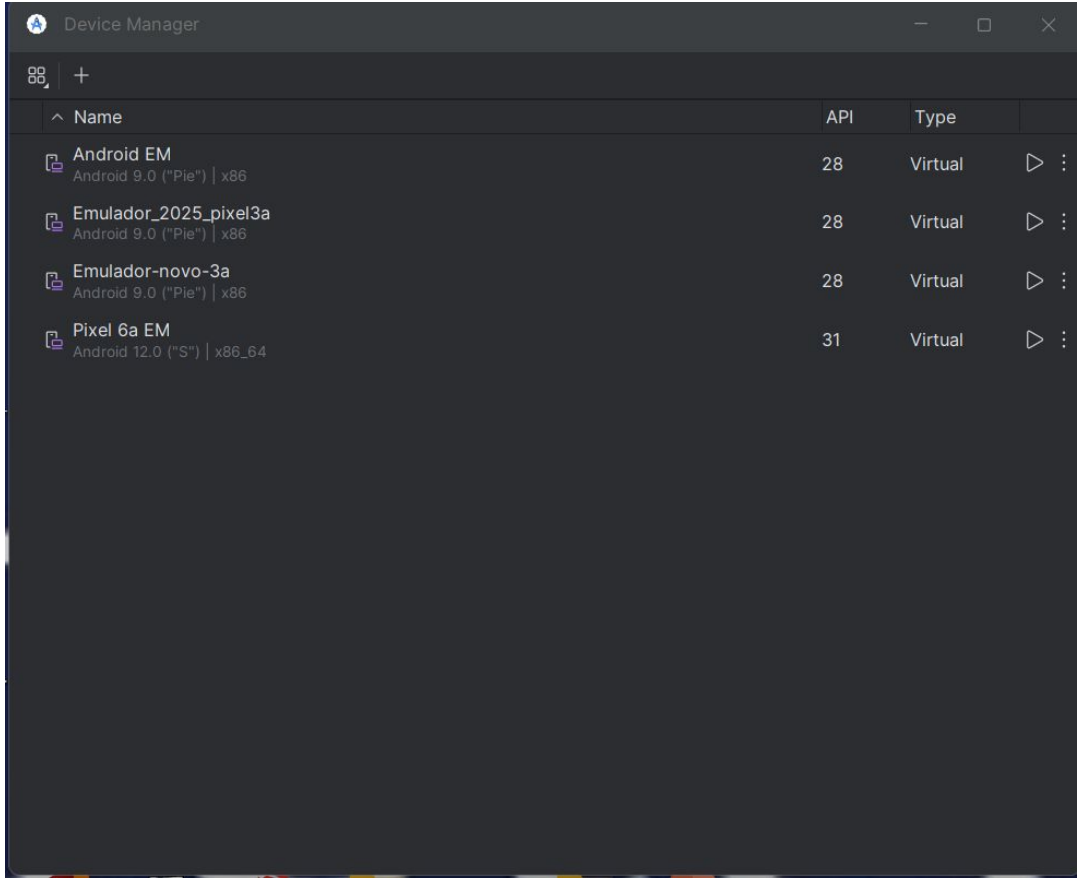
Após atribuir o nome Emulador_2025_pixel3a ao emulador clicar em Finish e o emulador será criado.



Criando emulador Android

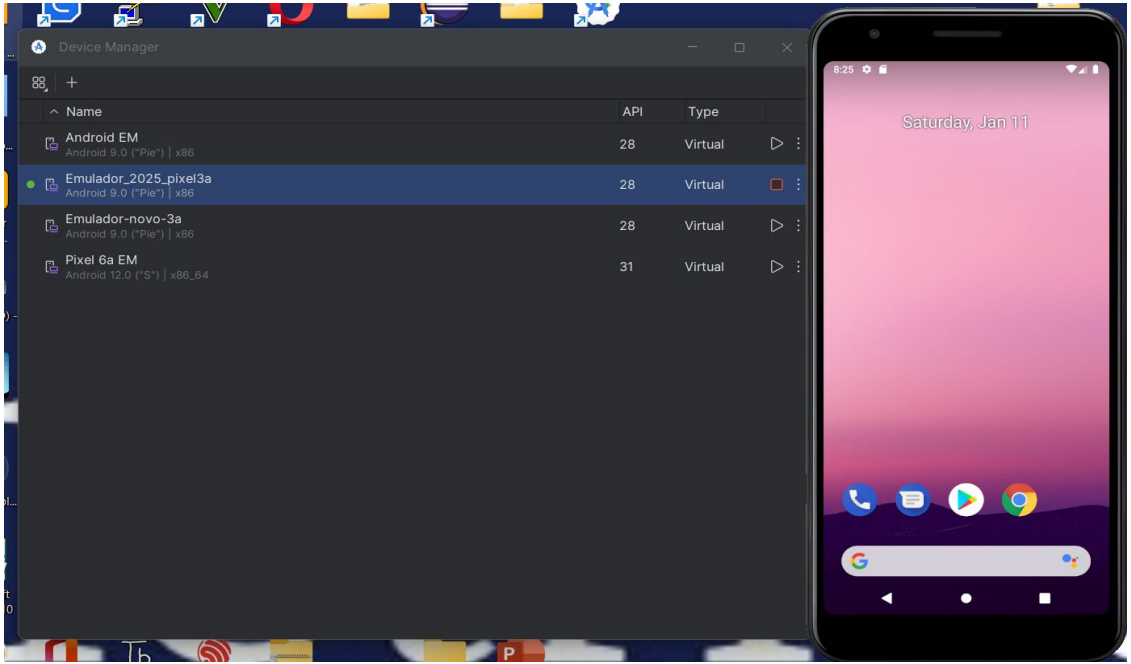
Na Figura é possível visualizar o emulador criado e é só clicar no play para o emulador ser inicializado abrindo o celular na Tela conforme a Figura

Criando emulador Android



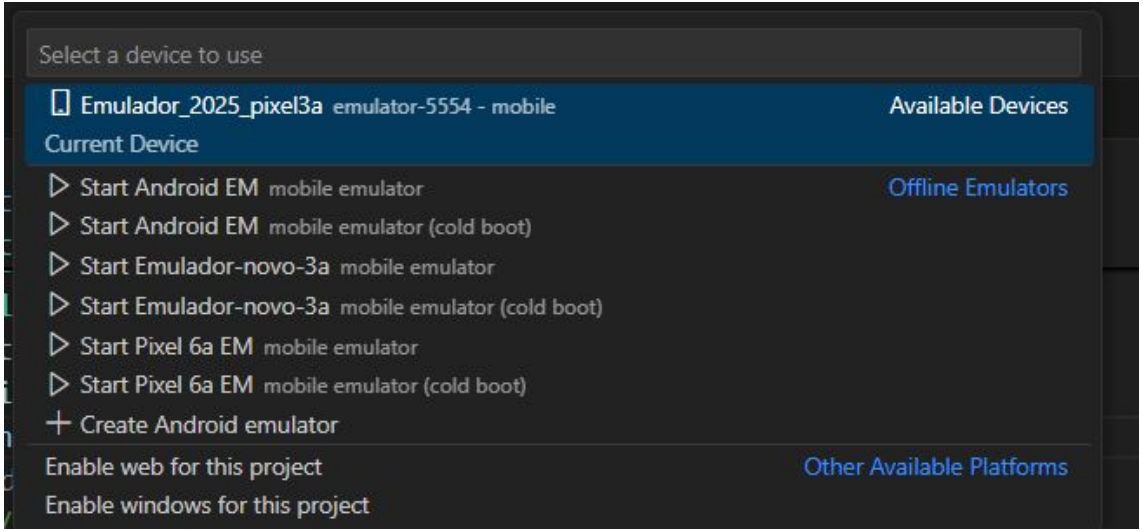
Criando emulador Android

Na Figura é possível ver o emulador sendo executado e agora é possível utilizá-lo para executar os aplicativos.



Criando emulador Android

Pronto, agora temos o emulador criado para executar os nossos aplicativos.



Código APP com texto e Listview

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

void main() {
  runApp(const MyApp());
}

class Jogador {
  final String nome;
  final int idade;
  final int pontuacao;

  Jogador({required this.nome, required this.idade, required this.pontuacao});
}
```

Código APP com texto e Listview

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Jogadores',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const JogadoresPage(),
    );
  }
}
```

Código APP com texto e Listview

```
class JogadoresPage extends StatefulWidget {
  const JogadoresPage({super.key});

  @override
  State<JogadoresPage> createState() => _JogadoresPageState();
}
```

Código APP com texto e Listview

```
class _JogadoresPageState extends State<JogadoresPage> {
  final List<Jogador> jogadores = [];

  final TextEditingController nomeController = TextEditingController();
  final TextEditingController idadeController = TextEditingController();
  final TextEditingController pontuacaoController = TextEditingController();

  void adicionarJogador() {
    final nome = nomeController.text;
    final idade = int.tryParse(idadeController.text);
    final pontuacao = int.tryParse(pontuacaoController.text);

    if (nome.isNotEmpty && idade != null && pontuacao != null) {
      setState(() {
        jogadores.add(Jogador(nome: nome, idade: idade, pontuacao: pontuacao));
      });

      // Limpa os campos
      nomeController.clear();
      idadeController.clear();
      pontuacaoController.clear();
    }
  }
```

Código APP com texto e Listview

```
 @override
void dispose() {
  nomeController.dispose();
  idadeController.dispose();
  pontuacaoController.dispose();
  super.dispose();
}
```

Código APP com texto e Listview

```
@override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(
       title: const Text('Jogadores do Meu Esporte'),
     ),
     body: Padding(
       padding: const EdgeInsets.all(16.0),
       child: Column(
         children: [
           TextField(
             controller: nomeController,
             decoration: const InputDecoration(labelText: 'Nome do jogador'),
           ),
```

Código APP com texto e Listview

```
TextField(
            controller: idadeController,
            decoration: const InputDecoration(labelText: 'Idade'),
            keyboardType: TextInputType.number,
          ),
          TextField(
            controller: pontuacaoController,
            decoration: const InputDecoration(labelText: 'Gols/Pontos'),
            keyboardType: TextInputType.number,
          ),
          const SizedBox(height: 16),
          ElevatedButton(
            onPressed: adicionarJogador,
            child: const Text('Adicionar Jogador'),
          ),
```

Código APP com texto e Listview

```
const SizedBox(height: 20),
        Expanded(
          child: ListView.builder(
            itemCount: jogadores.length,
            itemBuilder: (context, index) {
              final jogador = jogadores[index];
              return ListTile(
                leading: const Icon(Icons.person),
                title: Text(jogador.nome),
                subtitle: Text(
                    'Idade: ${jogador.idade} | Gols/Pontos: ${jogador.pontuacao}'),
              );
            },
          ),
        ),
      ],
    ),
  ),
);
}}
```

Exercícios

1) Criar um aplicativo para receber o nome do aluno, as notas de 4 disciplinas para calcular a média da nota de 3 avaliações por aluno e exibir se o aluno foi aprovado ou não na disciplina e exibir sua média

2) Você foi contratado pela empresa SM Mobile para desenvolver um aplicativo que solicite informações pessoais aos usuários em campos de texto.
Nome, idade, endereço, email, telefone para exibir as informações na tela

3) Criar um aplicativo para o usuário inserir nome de jogadores do seu esporte favorito pode ser futebol, basquete e exibir nome, idade, quantidade de gols ou pontos marcados e exibir em listview .

Obrigado!

Prof. Me Daniel Vieira

Email: danielvieira2006@gmail.com

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95