

Desenvolvimento
Mobile
Aula 14

Prof. Me Daniel Vieira



Agenda

- 1 - Login com api
- 2 - Super init state
- 3 - Show dialog
- 4 - Alert Dialog

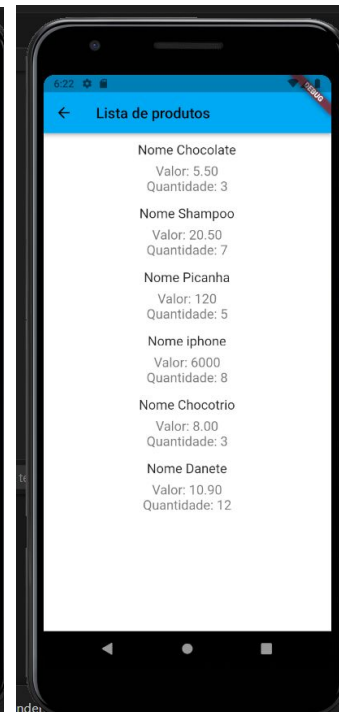
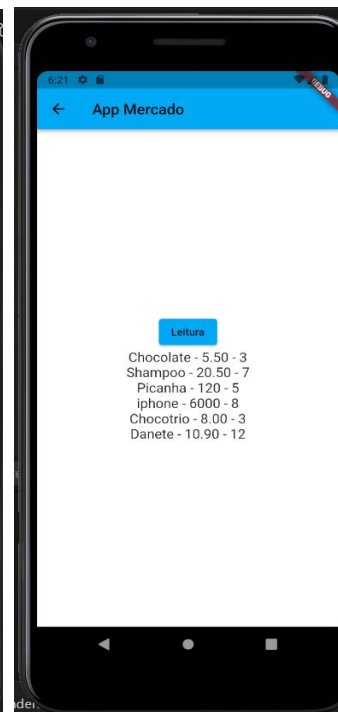
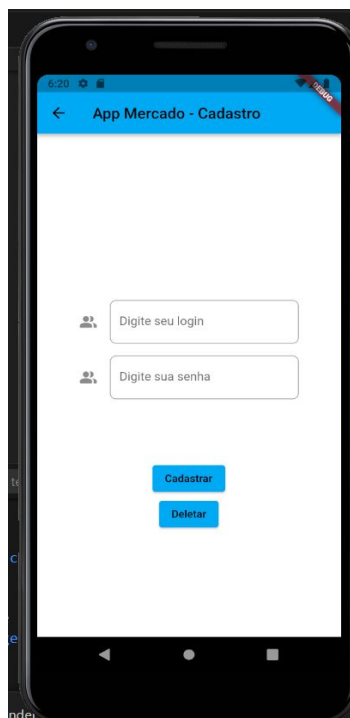
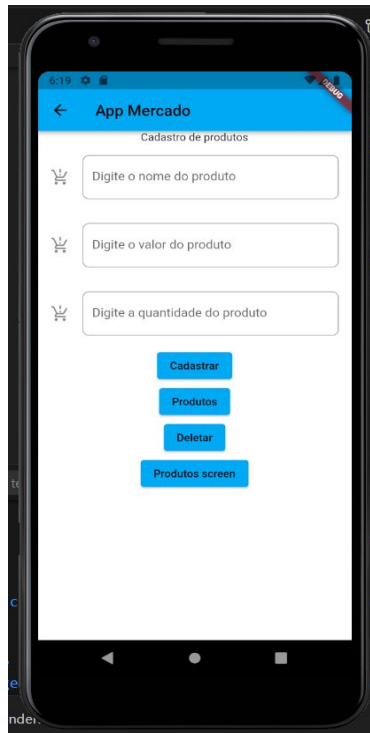
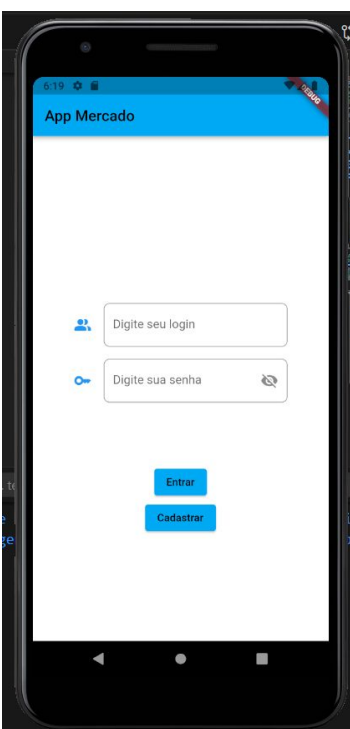
Exercício

1 - Criar um aplicativo para postar dados na api ao serem inseridos pelo TextField, os dados devem ser digitados pelo usuário, os dados devem ser exibidos em uma segunda tela, o aplicativo deve ter três telas, uma de login, outra para permitir que o usuário digite as informações e a terceira tela vai exibir as informações que o usuário digitou após apertar o botão

Criar a api local com o Nodejs

The image displays five sequential screenshots of a mobile application named 'App Mercado'.

- Screenshot 1 (Login):** The screen is titled 'App Mercado'. It features two input fields: 'Digite seu login' and 'Digite sua senha' with a toggle for visibility. Below these are two buttons: 'Entrar' and 'Cadastrar'.
- Screenshot 2 (Product Registration):** The screen is titled 'App Mercado' with a subtitle 'Cadastro de produtos'. It has three input fields, each preceded by a shopping cart icon: 'Digite o nome do produto', 'Digite o valor do produto', and 'Digite a quantidade do produto'. Below these are four buttons: 'Cadastrar', 'Produtos', 'Deletar', and 'Produtos screen'.
- Screenshot 3 (Registration Confirmation):** The screen is titled 'App Mercado - Cadastro'. It contains two input fields: 'Digite seu login' and 'Digite sua senha', each with a user icon. Below them are two buttons: 'Cadastrar' and 'Deletar'.
- Screenshot 4 (Product List):** The screen is titled 'App Mercado'. It displays a list of products under a 'Leitura' header:
 - Chocolate - 5.50 - 3
 - Shampoo - 20.50 - 7
 - Picanha - 120 - 5
 - iphone - 6000 - 8
 - Chocotrio - 8.00 - 3
 - Danete - 10.90 - 12
- Screenshot 5 (Detailed Product List):** The screen is titled 'App Mercado' with a subtitle 'Lista de produtos'. It lists the same products as Screenshot 4, but with more details:
 - Nome Chocolate: Valor: 5.50, Quantidade: 3
 - Nome Shampoo: Valor: 20.50, Quantidade: 7
 - Nome Picanha: Valor: 120, Quantidade: 5
 - Nome iphone: Valor: 6000, Quantidade: 8
 - Nome Chocotrio: Valor: 8.00, Quantidade: 3
 - Nome Danete: Valor: 10.90, Quantidade: 12



Arquivo db.json

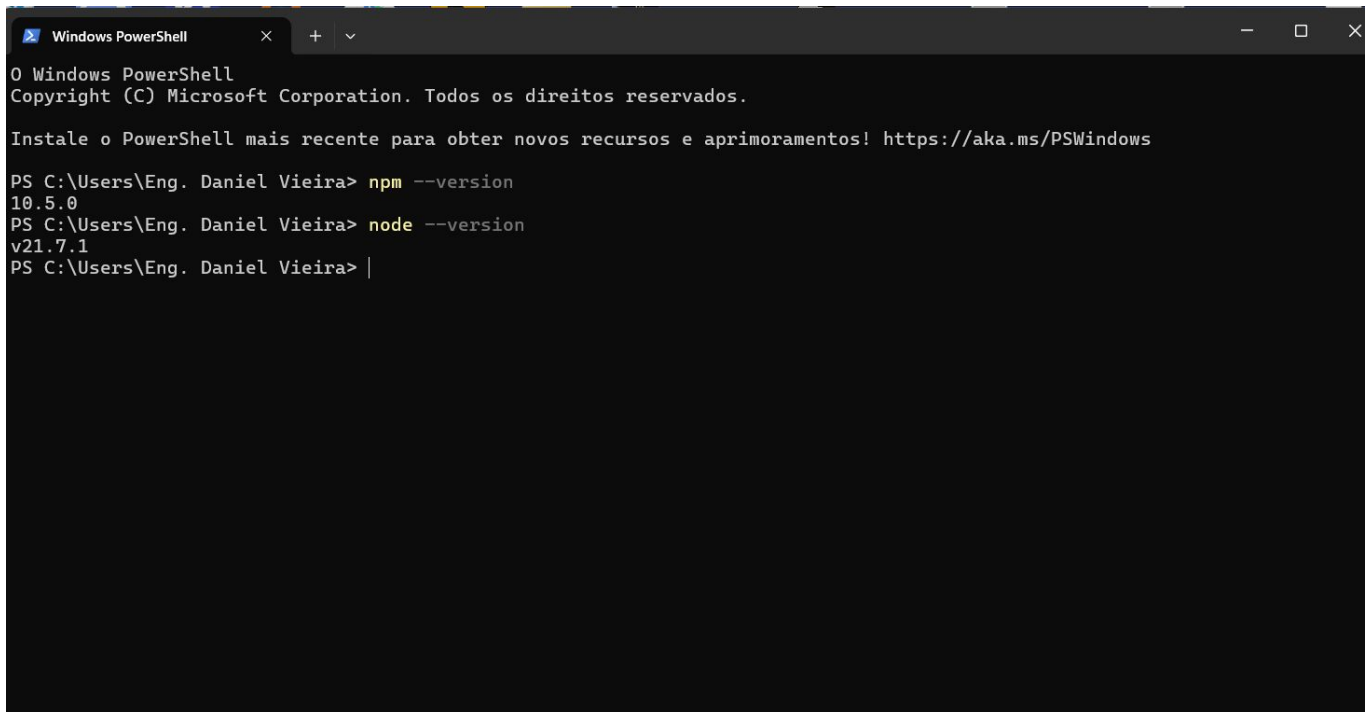
```
1  {
2    "produtos": [
3      {
4        "id": "0",
5        "nome": "Chocolate",
6        "valor": "5.50",
7        "qtde": "3"
8      },
9      {
10       "id": "1",
11       "nome": "Shampoo",
12       "valor": "20.50",
13       "qtde": "7"
14     },
15     {
16       "id": "2",
17       "nome": "Picanha",
18       "valor": "120",
19       "qtde": "5"
20     },
21     {
22       "id": "b352",
23       "nome": "maça",
24       "valor": "2.50",
25       "qtde": "5"
26     },
27   ]
28 }
```

```
27  {
28    "id": "af44",
29    "nome": "iphone",
30    "valor": "6000",
31    "qtde": "8"
32  },
33  {
34    "id": "0e5e",
35    "nome": "Chocotrio",
36    "valor": "8.00",
37    "qtde": "3"
38  },
39  {
40    "id": "04a0",
41    "nome": "Danete",
42    "valor": "10.90",
43    "qtde": "12"
44  }
45 }
```

```
46  "usuarios": [
47    {
48      "login": "admin",
49      "senha": "1234",
50      "id": "fb45"
51    },
52    {
53      "login": "filipe",
54      "senha": "1004",
55      "id": "4e0b"
56    },
57    {
58      "id": "4a9a",
59      "login": "daniel",
60      "senha": "100455"
61    },
62  ]
63 }
```

Instalação Node JS - Json server

O Node JS pode ser feito download através do site: <https://nodejs.org/en/download/current>

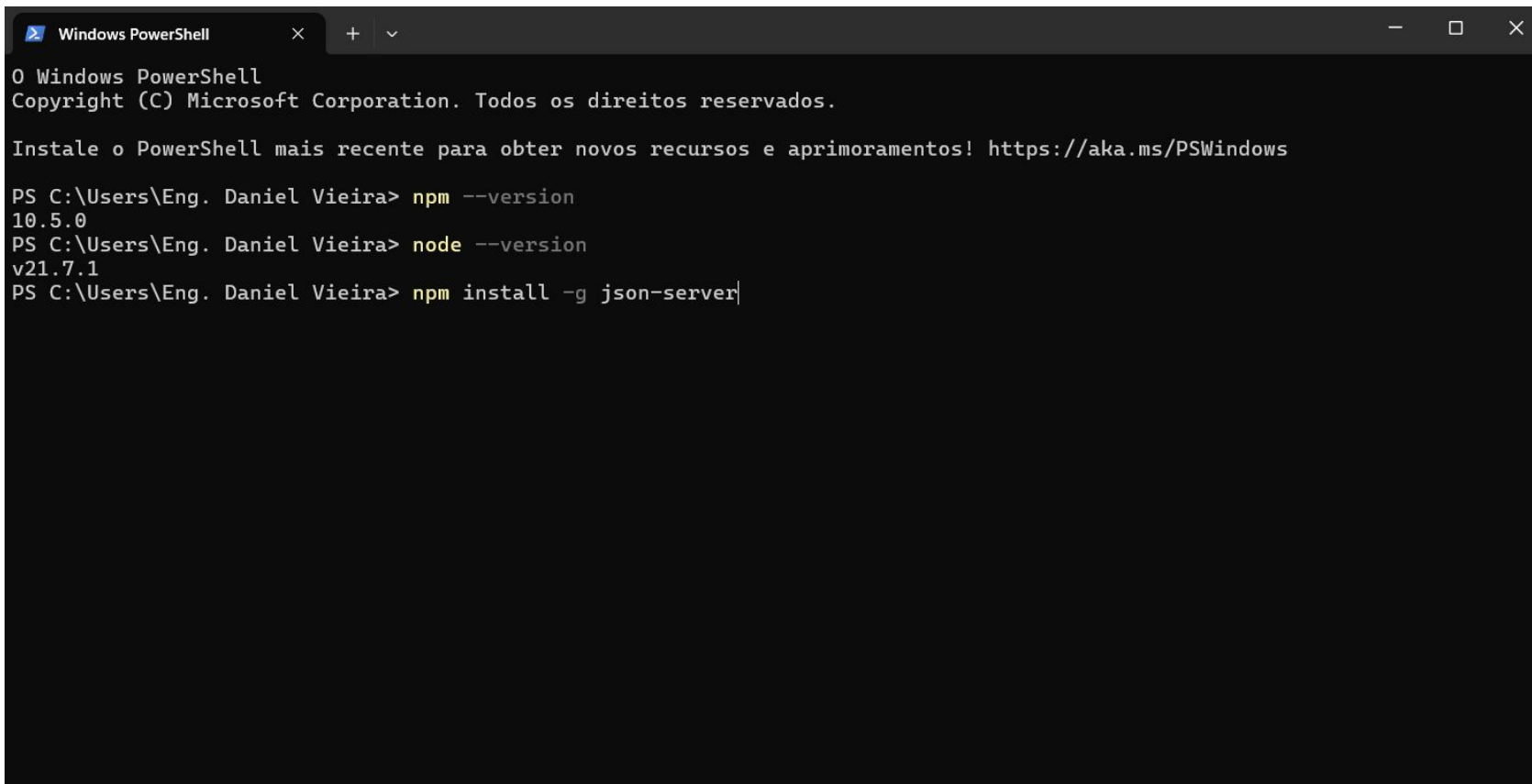


```
Windows PowerShell
O Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows

PS C:\Users\Eng. Daniel Vieira> npm --version
10.5.0
PS C:\Users\Eng. Daniel Vieira> node --version
v21.7.1
PS C:\Users\Eng. Daniel Vieira> |
```

Instalando o Json server



```
Windows PowerShell
O Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows

PS C:\Users\Eng. Daniel Vieira> npm --version
10.5.0
PS C:\Users\Eng. Daniel Vieira> node --version
v21.7.1
PS C:\Users\Eng. Daniel Vieira> npm install -g json-server
```

Iniciar o Json server

Para iniciar o servidor json será necessário descobrir o ip local da sua máquina
`json-server --watch -h numerodoseuip db.json`

```
Prompt de Comando
Estado da mídia. . . . . : mídia desconectada
Sufixo DNS específico de conexão. . . . . :

Adaptador de Rede sem Fio Conexão Local* 2:

Estado da mídia. . . . . : mídia desconectada
Sufixo DNS específico de conexão. . . . . :

Adaptador de Rede sem Fio Conexão Local* 3:

Estado da mídia. . . . . : mídia desconectada
Sufixo DNS específico de conexão. . . . . :

Adaptador de Rede sem Fio Wi-Fi:

Sufixo DNS específico de conexão. . . . . :
Endereço IPv6 . . . . . : 2804:431:cfcc:e09:8568:3017:6009:8bc3
Endereço IPv6 Temporário. . . . . : 2804:431:cfcc:e09:419a:93ff:3862:ca87
Endereço IPv6 de link local . . . . . : fe80::c67a:994:4def:7ff0%22
Endereço IPv4. . . . . : 192.168.15.10
Máscara de Sub-rede . . . . . : 255.255.255.0
Gateway Padrão. . . . . : fe80::9a7e:caff:fe50%22
                        192.168.15.1

Adaptador Ethernet Conexão de Rede Bluetooth:

Estado da mídia. . . . . : mídia desconectada
Sufixo DNS específico de conexão. . . . . :

C:\Users\Eng. Daniel Vieira>
```



Iniciar o Json server

Após descobrir seu ip é necessário entrar na pasta do arquivo json para rodar o servidor

json server

feito isso executar o comando dentro da pasta json com seu respectivo ip

json-server --watch --numerodoseuip db.json



The screenshot shows a Windows PowerShell terminal window with the following content:

```
PROBLEMS 4 OUTPUT TERMINAL PORTS SERIAL MONITOR DEBUG CONSOLE
```

Prefix	Date	Time	File/Path
d----	02/04/2024	15:56	lib
d----	02/04/2024	16:01	server
d----	02/04/2024	15:56	test
-a----	02/04/2024	15:56	745 .gitignore
-a----	02/04/2024	15:56	930 .metadata
-a----	02/04/2024	15:56	1482 analysis_options.yaml
-a----	02/04/2024	15:56	859 app_aula09_post_prep.iml
-a----	02/04/2024	16:06	5951 pubspec.lock
-a----	02/04/2024	16:05	3990 pubspec.yaml
-a----	02/04/2024	15:56	579 README.md

```
PS D:\SENAI\2024-1\Desenvolvimento Mobile 2\DESM-2\App prep\app_aula09_post_prep> cd server
PS D:\SENAI\2024-1\Desenvolvimento Mobile 2\DESM-2\App prep\app_aula09_post_prep\server>
```

The terminal window has a taskbar at the bottom with two instances of 'powershell' open.

Iniciar o Json server

Após descobrir seu ip é necessário entrar na pasta do arquivo json para rodar o servidor

json server

feito isso executar o comando dentro da pasta json com seu respectivo ip

json-server --watch --numerodoseuip db.json

```
d----- 02/04/2024 16:01 server
d----- 02/04/2024 15:56 test
-a---- 02/04/2024 15:56 745 .gitignore
-a---- 02/04/2024 15:56 930 .metadata
-a---- 02/04/2024 15:56 1482 analysis_options.yaml
-a---- 02/04/2024 15:56 859 app_aula09_post_prep.iml
-a---- 02/04/2024 16:06 5951 pubspec.lock
-a---- 02/04/2024 16:05 3990 pubspec.yaml
-a---- 02/04/2024 15:56 579 README.md
```

```
PS D:\SENAI\2024-1\Desenvolvimento Mobile 2\DESM-2\App prep\app_aula09_post_prep> cd server
```

```
PS D:\SENAI\2024-1\Desenvolvimento Mobile 2\DESM-2\App prep\app_aula09_post_prep\server> jsonserver --watch --192.168.15.10 db.json
```

jsonserver : O termo 'jsonserver' não é reconhecido como nome de cmdlet, função, arquivo de script ou programa operável. Verifique a grafia do nome ou, se um caminho tiver sido incluído, veja se o caminho está correto e tente novamente.

No linha:1 caractere:1

+ jsonserver --watch --192.168.15.10 db.json

+ ~~~~~

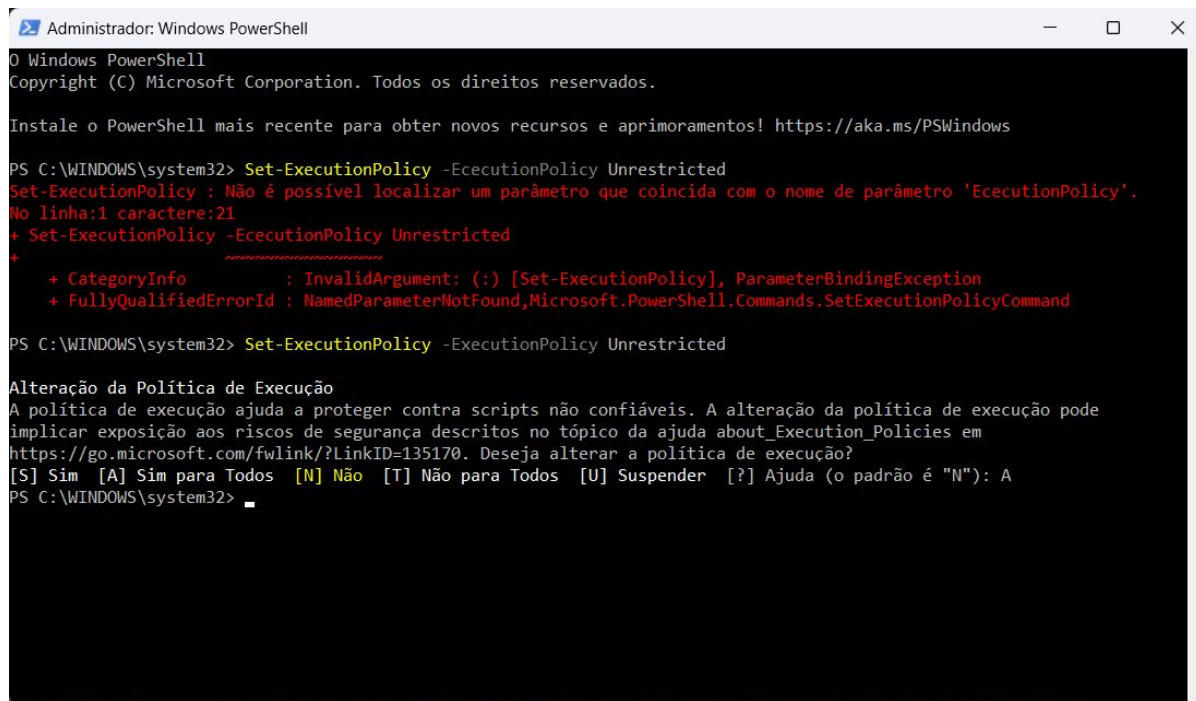
+ CategoryInfo : ObjectNotFound: (jsonserver:String) [], CommandNotFoundException

+ FullyQualifiedErrorId : CommandNotFoundException

```
PS D:\SENAI\2024-1\Desenvolvimento Mobile 2\DESM-2\App prep\app_aula09_post_prep\server> json-server --watch --host 192.168.15.10 db.json
```

Iniciar o Json server

Após esse erro é necessário retirar as políticas de segurança do Windows com o comando `Set-ExecutionPolicy -ExecutionPolicy Unrestricted`



```
Administrador: Windows PowerShell
O Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Instale o PowerShell mais recente para obter novos recursos e aprimoramentos! https://aka.ms/PSWindows

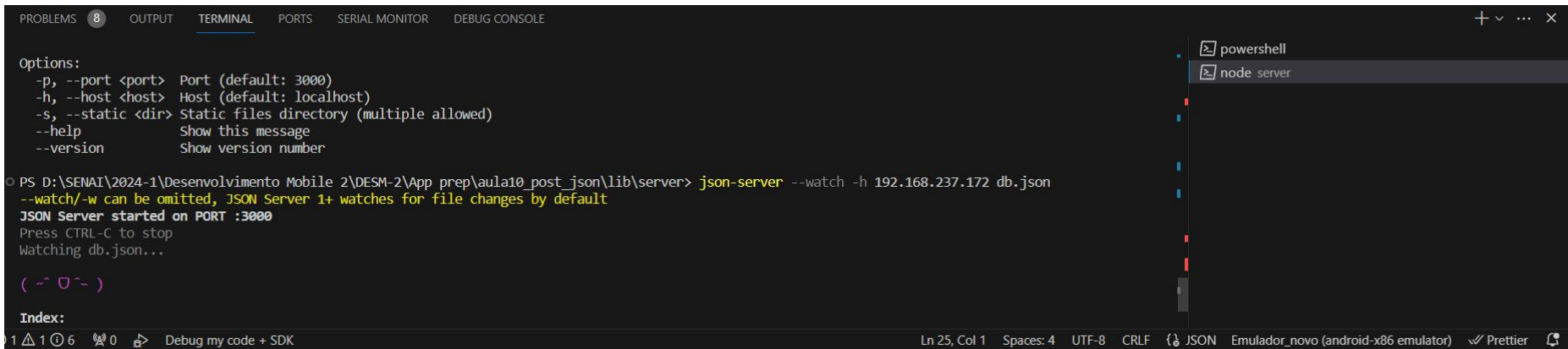
PS C:\WINDOWS\system32> Set-ExecutionPolicy -EecutionPolicy Unrestricted
Set-ExecutionPolicy : Não é possível localizar um parâmetro que coincida com o nome de parâmetro 'EecutionPolicy'.
No linha:1 caractere:21
+ Set-ExecutionPolicy -EecutionPolicy Unrestricted
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (:) [Set-ExecutionPolicy], ParameterBindingException
+ FullyQualifiedErrorId : NamedParameterNotFound,Microsoft.PowerShell.Commands.SetExecutionPolicyCommand

PS C:\WINDOWS\system32> Set-ExecutionPolicy -ExecutionPolicy Unrestricted

Alteração da Política de Execução
A política de execução ajuda a proteger contra scripts não confiáveis. A alteração da política de execução pode
implicar exposição aos riscos de segurança descritos no tópico da ajuda about_Execution_Policies em
https://go.microsoft.com/fwlink/?LinkID=135170. Deseja alterar a política de execução?
[S] Sim [A] Sim para Todos [N] Não [T] Não para Todos [U] Suspender [?] Ajuda (o padrão é "N"): A
PS C:\WINDOWS\system32>
```

Iniciar o Json server

É hora de iniciar o servidor Json



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal displays the command to start the json-server and its output. The command is: `json-server --watch -h 192.168.237.172 db.json`. The output indicates that the server has started on port 3000 and is watching for file changes. The status bar at the bottom shows the file path, encoding, and other settings.

```
PROBLEMS 8 OUTPUT TERMINAL PORTS SERIAL MONITOR DEBUG CONSOLE

Options:
  -p, --port <port>  Port (default: 3000)
  -h, --host <host>  Host (default: localhost)
  -s, --static <dir> Static files directory (multiple allowed)
  --help              Show this message
  --version           Show version number

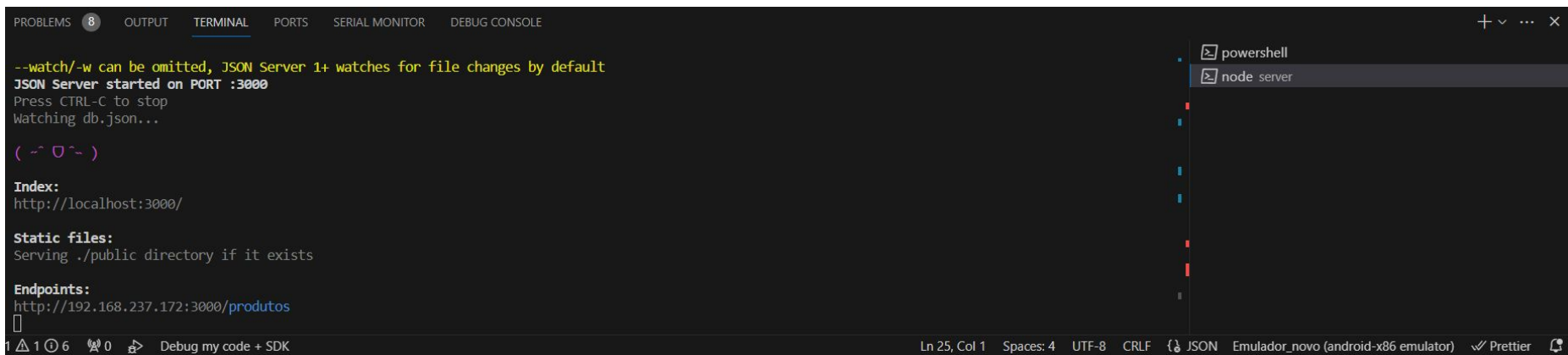
PS D:\SENAT\2024-1\Desenvolvimento Mobile 2\DESM-2\App prep\aula10_post_json\lib\server> json-server --watch -h 192.168.237.172 db.json
--watch/-w can be omitted, JSON Server 1+ watches for file changes by default
JSON Server started on PORT :3000
Press CTRL-C to stop
Watching db.json...

( ^ _ ^ )

Index:
1 1 1 6 0 0 Debug my code + SDK Ln 25, Col 1 Spaces: 4 UTF-8 CRLF { JSON Emulador_novo (android-x86 emulator) Prettier
```

Iniciar o Json server

Servidor iniciado



The screenshot shows the Visual Studio Code interface. The top bar includes tabs for PROBLEMS (8), OUTPUT, TERMINAL, PORTS, SERIAL MONITOR, and DEBUG CONSOLE. The TERMINAL tab is active, displaying the following output:

```
--watch/-w can be omitted, JSON Server 1+ watches for file changes by default
JSON Server started on PORT :3000
Press CTRL-C to stop
Watching db.json...

( ~ 0 ^ )

Index:
http://localhost:3000/

Static files:
Serving ./public directory if it exists

Endpoints:
http://192.168.237.172:3000/produtos
[]
```

On the right side, the Explorer view shows a file tree with two files: powershell and node server. The node server file is selected. The bottom status bar indicates the current position is Ln 25, Col 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and the file is JSON. The active window is Emulador_novo (android-x86 emulator), and Prettier is the active formatter.

Consumindo um api no Flutter

1º Necessário é no arquivo pubspec.yaml e adicionar a dependência no projeto

```
# The following adds the Cupertino Icons font to your application.  
# Use with the CupertinoIcons class for iOS style icons.  
cupertino_icons: ^1.0.2  
http: ^0.13.3
```

2º Importar o método http no arquivo necessário

Código Login

Importando os pacotes para consumo de api e importação das telas

```
1  import 'dart:convert';  
2  
3  import 'package:app_formativa/cadastrocliente.dart';  
4  import 'package:app_formativa/cadastroprodutos.dart';  
5  import 'package:flutter/material.dart';  
6  import 'package:http/http.dart' as http;
```

Código Login

Cria classe Login, textediting controller e url com endereço ip

```
7  class Login extends StatefulWidget {  
8      @override  
9      State<Login> createState() => _LoginState();  
10 }  
11  
12 class _LoginState extends State<Login> {  
13     TextEditingController user = TextEditingController();  
14  
15     TextEditingController senha = TextEditingController();  
16     bool exibir = false;  
17     //String url = "http://10.109.83.10:3000/usuarios";  
18     //String url = "http://100.112.118.243:3000/usuarios";  
19     String url = "http://192.168.15.12:3000/usuarios";  
20 }
```


Lógica para login com api

```
21  _login()async{
22      http.Response response = await http.get(Uri.parse(url));
23      List clientes =<Usuarios>[];
24      bool encuser= false;
25      clientes = json.decode(response.body);
26      for(int i=0; i<clientes.length;i++){
27          print(clientes[0]["login"]);
28      }
29
30      if(user.text==clientes[i]["login"] && senha.text==clientes[i]["senha"]){
31
32          encuser= true;
33
34          break;
35      }
36  }
37
```

Lógica para login com api

Se o usuário for encontrado entra na segunda tela para cadastro dos produtos, se não exibe a mensagem

```
39  if(encuser ==true){
40    Navigator.push(context, MaterialPageRoute(builder: (context)=>Cadastroprodutos()));
41    encuser = false;
42    user.text = "";
43    senha.text = "";
44  }
45
46  else{
47
48    ScaffoldMessenger.of(context).showSnackBar(
49      SnackBar(content: Text("Usuário não cadastrado"),duration: Duration(seconds: 2),),);
50  }
```

Show Dialog


```
54  showDialog(  
55      context: context,  
56      builder: (BuildContext) {  
57          return AlertDialog(  
58              content: Text('Usuário inválido'),  
59              actions: [  
60                  TextButton(  
61                      onPressed: () {  
62                          Navigator.of(context).pop();  
63                      },  
64                      child: Text('Fechar')) // TextButton  
65              ],  
66          ); // AlertDialog  
67      });  
68  }  
69  
70  }  
71
```

Context faz parte da build do código, actions é a função que irá sair do show dialog

Construindo a tela de login

```
77  @override
78  Widget build(BuildContext context) {
79      return Scaffold(
80          appBar: AppBar(
81              title: Text("App Mercado"),
82          ),
83          // AppBar
84          body: Center(
85              child: Column(
86                  mainAxisAlignment: MainAxisAlignment.center,
87                  children: [
88                      SizedBox(
89                          width: 300,
90                          height: 300,
91                          child: Column(
92                              mainAxisAlignment: MainAxisAlignment.center,
93                              children: [
94
```

Construindo a tela de login

```
95     Padding(  
96       padding: const EdgeInsets.all(8.0),  
97       child: TextFormField(  
98         keyboardType: TextInputType.name,  
99         decoration: InputDecoration(border: OutlineInputBorder(  
100           borderRadius: BorderRadius.circular(8),  
101         )), // OutlineInputBorder  
102         icon: Icon(Icons.people_alt_rounded), iconColor:  Colors.blue,  
103         hintText: "Digite seu login",  
104  
105         ), // InputDecoration  
106  
107         controller: user,  
108       ), // TextFormField  
109     ), // Padding
```

Construindo a tela de login

```
110      Padding(  
111        padding: const EdgeInsets.all(8.0),  
112        child: TextFormField(  
113          keyboardType: TextInputType.name,  
114          decoration: InputDecoration(border: OutlineInputBorder(  
115            borderRadius: BorderRadius.circular(8),  
116          )), // OutlineInputBorder  
117          icon: Icon(Icons.key), iconColor: Colors.blue,  
118  
119          hintText: "Digite sua senha",  
120          suffixIcon: IconButton(  
121            icon: Icon(exibir  
122              ? Icons.visibility  
123              : Icons.visibility_off), // Icon  
124            onPressed: () {  
125              setState(() {  
126                exibir = !exibir;  
127              });  
128            },  
129          ),  
130        ) // IconButton
```


Construindo a tela de login

```
132         ), // InputDecoration
133         obscureText: exibir,
134         obscuringCharacter: "*",
135
136         controller: senha,
137     ), // TextFormField
138   ), // Padding
139 ],
140 ), // Column
141 ), // SizedBox
142 ElevatedButton(onPressed: _login, child: Text("Entrar")),
143 ElevatedButton(onPressed: (){
144   Navigator.push(context, MaterialPageRoute(builder: (context)=>CadastroCliente()));
145 }, child: Text("Cadastrar")), // ElevatedButton
146 ],
147 ), // Column
148 ), // Center
149 ); // Scaffold
150
151 }
152
153 }
```

Construindo a tela de login

Factory é um método que permite converter os dados para json

```
156  class Usuarios{
157      String id;
158      String login;
159      String senha;
160      Usuarios(this.id,this.login,this.senha);
161  factory Usuarios.fromJson(Map<String,dynamic> json){
162      |   return Usuarios (json["id"],json["login"],json["senha"]);
163      |
164      |
165      |
166  }
```


Tela de cadastro usuário

```
1  import 'dart:convert';
2  ⚡
3  import 'package:flutter/material.dart';
4  import 'package:http/http.dart' as http;
5  class CadastroCliente extends StatefulWidget {
6      @override
7      State<CadastroCliente> createState() => _CadastroClienteState();
8  }
9
10 class _CadastroClienteState extends State<CadastroCliente> {
11     TextEditingController user_n = TextEditingController();
12
13     TextEditingController senha_n = TextEditingController();
14 }
```

Tela de cadastro usuário

```
15  _cadastrarcliente()async{
16      //String url = "http://10.109.83.10:3000/usuarios";
17      String url = "http://100.112.118.243:3000/usuarios";
18      http.Response response = await http.get(Uri.parse(url));
19      print(user_n.text);
20      print(senha_n.text);
21  Map<String,dynamic> usuarios_n={
22      "login": user_n.text.toString(),
23      "senha": senha_n.text.toString()
24  };
25  http.post(Uri.parse(url),
26  headers: <String,String>{
27      'Content-type': 'application/json; charset=UTF-8',
28  },
29  body: jsonEncode(usuarios_n)
30  );
31  print("Usuario cadastrado");
32  }
```

Função para realizar o post de um novo usuário

Tela de cadastro usuário

```
33  _deletar(){  
34      http.delete(Uri.parse("http://10.109.83.10:3000/usuarios/c381"));  
35      print("Usuario deletado");  
36  }
```

Função para deletar um usuário através do id

Tela de cadastro usuário

```
37 @override
38 Widget build(BuildContext context) {
39   return Scaffold(
40     appBar: AppBar(
41       title: Text("App Mercado - Cadastro"),
42     ), // AppBar
43     body: Center(
44       child: Column(
45         mainAxisAlignment: MainAxisAlignment.center,
46         children: [
47           SizedBox(
48             width: 300,
49             height: 300,
50             child: Column(
51               mainAxisAlignment: MainAxisAlignment.center,
52               children: [
53
54                 Padding(
55                   padding: const EdgeInsets.all(8.0),
56                   child: TextFormField(
57                     keyboardType: TextInputType.name,
58                     decoration: InputDecoration(border: OutlineInputBorder(
59                       borderRadius: BorderRadius.circular(8),
60                     ), // OutlineInputBorder
```

Tela de cadastro usuário

```
61      icon: Icon(Icons.people_alt_rounded),
62      hintText: "Digite seu login",
63
64      ), // InputDecoration
65
66      controller: user_n,
67    ), // TextFormField
68  ), // Padding
69  Padding(
70    padding: const EdgeInsets.all(8.0),
71    child: TextFormField(
72      keyboardType: TextInputType.name,
73      decoration: InputDecoration(border: OutlineInputBorder(
74        |   borderRadius: BorderRadius.circular(8),
75      ), // OutlineInputBorder
76      icon: Icon(Icons.people_alt_rounded),
77      hintText: "Digite sua senha",
78
```

Tela de cadastro usuário

```
79         ), // InputDecoration
80         obscureText: true,
81         obscuringCharacter: "*",
82
83         controller: senha_n,
84     ), // TextFormField
85   ), // Padding
86 ],
87 ), // Column
88 ), // SizedBox
89 ElevatedButton(onPressed: _cadastrarciente, child: Text("Cadastrar")),
90 ElevatedButton(onPressed: _deletar, child: Text("Deletar")),
91 ],
92
```

Tela de cadastro usuário

```
91         ],  
92     ), // Column  
93 ), // Center  
94 ); // Scaffold  
95 }  
96 }  
97 }
```


Tela de cadastro de produto

```
1 import 'package:app_formativa/telaprod.dart';
2 import 'package:app_formativa/telaprodutos.dart';
3 import 'package:flutter/material.dart';
4 import 'package:http/http.dart' as http;
5 import 'dart:convert';
6 class Cadastroprodutos extends StatelessWidget {
7   TextEditingController nomeprod = TextEditingController();
8   TextEditingController valorprod = TextEditingController();
9   TextEditingController qtdeprod = TextEditingController();
10   //String url = "http://10.109.83.10:3000/produtos";
11   String url = "http://192.168.15.12:3000/produtos";
12
13   _cadastroprod() async{
14     http.Response response = await http.get(Uri.parse(url));
15     print(nomeprod.text);
16     print(valorprod.text);
17     print(qtdeprod.text);
18     Map<String,dynamic> produtos={
19       "nome": nomeprod.text.toString(),
20       "valor": valorprod.text.toString(),
21       "qtde": qtdeprod.text.toString()
22     };
23     http.post(Uri.parse(url),
24       headers: <String,String>{
25       'Content-type': 'application/json; charset=UTF-8',
26     },
```


Tela de cadastro de produto

```
27     body: jsonEncode(produtos)
28   );
29   nomeprod.text = "";
30   valorprod.text = "";
31   qtdeprod.text = "";
32 }
33
34 @override
35 Widget build(BuildContext context) {
36   return Scaffold(
37     appBar: AppBar(
38       title: Text("App Mercado"),
39     ), // AppBar
40     body: Column(
41       mainAxisAlignment: MainAxisAlignment.start,
42       children: [
43         Text("Cadastro de produtos"),
44         Padding(
```

Tela de cadastro de produto

```
45 padding: const EdgeInsets.all(16.0),
46 child: TextFormField(
47     keyboardType: TextInputType.name,
48     decoration: InputDecoration(border: OutlineInputBorder(
49         borderRadius: BorderRadius.circular(8),
50     ), // OutlineInputBorder
51     icon: Icon(Icons.production_quantity_limits),
52     hintText: "Digite o nome do produto",
53
54     ), // InputDecoration
55
56     controller: nomeprod,
57 ), // TextFormField
58 ), // Padding
59 Padding(
60     padding: const EdgeInsets.all(16.0),
61     child: TextFormField(
62         keyboardType: TextInputType.name,
63         decoration: InputDecoration(border: OutlineInputBorder(
64             borderRadius: BorderRadius.circular(8),
65         ), // OutlineInputBorder
```

Tela de cadastro de produto

```
66         icon: Icon(Icons.production_quantity_limits),
67         hintText: "Digite o valor do produto",
68
69         ), // InputDecoration
70
71         controller: valorprod,
72     ), // TextFormField
73 ), // Padding
74 Padding(
75     padding: const EdgeInsets.all(16.0),
76     child: TextFormField(
77         keyboardType: TextInputType.name,
78         decoration: InputDecoration(border: OutlineInputBorder(
79             |   borderRadius: BorderRadius.circular(8),
80         ), // OutlineInputBorder
81         icon: Icon(Icons.production_quantity_limits),
82         hintText: "Digite a quantidade do produto",
83
84         ), // InputDecoration
85
86         controller: qtdeprod,
```

Tela de cadastro de produto

```
87         ), // TextFormField
88     ), // Padding
89     ElevatedButton(onPressed: _cadastroprod, child: Text("Cadastrar")),
90     ElevatedButton(onPressed: (){
91         Navigator.push(context, MaterialPageRoute(builder: (context)=>Telaprodutos()));
92     },
93     child: Text("Produtos")), // ElevatedButton
94     ElevatedButton(
95         onPressed: () {
96             http.delete(
97                 Uri.parse('http://192.168.15.12:3000/produtos/b352'));
98         },
99         child: Text("Deletar")), // ElevatedButton
100     ElevatedButton(
101         onPressed: () {
102             Navigator.push(context,
103                 MaterialPageRoute(builder: (context) => Prod_screen()));
104         },
105         child: Text('Produtos screen')) // ElevatedButton
106     ],
107 ), // Column
108 ); // Scaffold
109 }
110 }
```

Tela para exibir os produtos clicando no botão

```
1  import 'package:flutter/material.dart';
2  import 'package:http/http.dart' as http;
3  import 'dart:convert';
4  class Telaprodutos extends StatefulWidget {
5    @override
6    State<Telaprodutos> createState() => _TelaprodutosState();
7  }
8
9  class _TelaprodutosState extends State<Telaprodutos> {
10
11
12    var prod = <Produtos_item>[];
13
14    // String url = "http://10.109.83.10:3000/produtos";
15    String url = "http://192.168.15.12:3000/produtos";
16
17    _leituradados() async{
18      http.Response response = await http.get(Uri.parse(url));
19
20
21      var dado = jsonDecode(response.body) as List;
22      //prod = dado;
23      // print(prod.length);
24      //print(dado);
25      setState(() {
26        prod = dado.map((json) => Produtos_item.fromJson(json)).toList();
```

Tela para exibir os produtos clicando no botão

```
28
29     print(dado);
30
31 }
32
33 @override
34 Widget build(BuildContext context) {
35     return Scaffold(
36         appBar: AppBar(
37             title: Text("App Mercado"),
38         ), // AppBar
39         body: Center(
40             child: Column(
41                 mainAxisAlignment: MainAxisAlignment.center,
42                 children: [
43                     ElevatedButton(onPressed: _leituradados, child: Text("Leitura")),
44                     Column(
45                         children: prod.map((produto) => Text("${produto.nome} - ${produto.valor} - ${produto.qtde}"),
46                         style: TextStyle(fontSize: 18)) .toList(), // Text
47                     ) // Column
48                 ],
49             ), // Column
```

Tela para exibir os produtos clicando no botão

```
49         ), // Column
50     ), // Center
51
52
53     ); // Scaffold
54
55
56 }
57 }
```


Tela para exibir os produtos clicando no botão

```
59  class Produtos_item{  
60      String id;  
61      String nome;  
62      String valor;  
63      String qtde;  
64      Produtos_item(this.id, this.nome, this.valor, this.qtde);  
65  factory Produtos_item.fromJson(Map<String,dynamic>json){  
66      return Produtos_item(json['id'], json['nome'], json['valor'], json['qtde']);  
67  }  
68  
69  
70 }
```


Tela para exibir os produtos clicando no botão

```
88  class Produto_lista {  
89      List prod = [];  
90      Produto_lista(this.prod);  
91  }  
92
```

Tela para exibir os produtos de forma automática

```
1 import 'package:flutter/material.dart';
2 import 'package:http/http.dart' as http;
3 import 'dart:convert';
4
5 class Prod_screen extends StatefulWidget {
6   const Prod_screen({super.key});
7
8   @override
9   State<Prod_screen> createState() => _Prod_screenState();
10 }
11
12 class _Prod_screenState extends State<Prod_screen> {
13   void initState() {
14     super.initState();
15     leituradados();
16   }
17
18   var prod = <Produtos_item>[];
19   List dado = [];
20   // String url = "http://10.109.83.10:3000/produtos";
21   String url = "http://192.168.15.12:3000/produtos";
22 }
```

Tela para exibir os produtos de forma automática

```
23 Future<void> leituraDados() async {
24     http.Response response = await http.get(Uri.parse(url));
25     if (response.statusCode == 200) {
26         setState(() {
27             dado = jsonDecode(response.body) as List<dynamic>;
28         });
29
30         //prod = dado;
31         // print(prod.length);
32         //print(dado);
33
34         //return prod = dado.map((json) => Produtos_item.fromJson(json)).toList();
35     } else {
36         print(response.statusCode);
37         throw Exception('Falha ao consumir api');
38     }
39 }
40
```

Tela para exibir os produtos de forma automática

```
41 @override
42 Widget build(BuildContext context) {
43   return Scaffold(
44     appBar: AppBar(
45       title: Text('Lista de produtos'),
46     ), // AppBar
47     body: Center(
48       child: ListView.builder(
49         itemCount: dado.length,
50         itemBuilder: (context, index) {
51           final item = dado[index];
52           return ListTile(
53             title: Padding(
54               padding: const EdgeInsets.all(8.0),
55               child: Text(
56                 'Nome ${item['nome']}',
57                 style: TextStyle(fontSize: 18),
58                 textAlign: TextAlign.center,
59               ), // Text
60             ), // Padding
61             subtitle: Column(
62               crossAxisAlignment: CrossAxisAlignment.center,
63               children: [
64                 Text(
65                   'Valor: ${item['valor']}',
```

Tela para exibir os produtos de forma automática

```
63  children: [  
64  Text(  
65    'Valor: ${item['valor']}',  
66    style: TextStyle(fontSize: 18),  
67  ), // Text  
68  Text('Quantidade: ${item['qtde']}',  
69    style: TextStyle(fontSize: 18)), // Text  
70  ],  
71  ), // Column  
72  ); // ListTile  
73  }), // ListView.builder  
74  )); // Center // Scaffold  
75  }  
76  }  
77
```

Tela para exibir os produtos de forma automática

```
78 class Produtos_item {  
79     String id;  
80     String nome;  
81     String valor;  
82     String qtde;  
83     Produtos_item(this.id, this.nome, this.valor, this.qtde);  
84     factory Produtos_item.fromJson(Map<String, dynamic> json) {  
85         return Produtos_item(json['id'], json['nome'], json['valor'], json['qtde']);  
86     }  
87 }  
88  
89 class Produto_lista {  
90     List prod = [];  
91     Produto_lista(this.prod);  
92 }  
93
```

Obrigado!

Prof. Me Daniel Vieira

Email: danielvieira2006@gmail.com

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

