

```
{"nomeProjeto":"Projeto Seven","funcionarios":[{"nome":"Robinho","dependentes":[{"nome":"Gabriel"}, {"nome":"Lay"}]}, {"nome":"Kleber","dependentes":[{"nome":"Cristiano"}, {"nome":"Geogina"}, {"nome":"Zeli"}]}, {"nome":"Kauan","dependentes":[{"nome":"Carol"}]}]}
```

Search

Files

main.dart

Tools

Recent

All

- AI
- Authentication
- Chat
- Code Search CtrlShiftF
- Console
- Database
- Debugger
- Docs
- Extensions
- Git
- Packages
- PostgreSQL
- Secrets
- Shell CtrlShiftS
- SSH
- System Dependencies
- Threads
- User Settings
- Webview

main.dart

main.dart

Format

```
1 import 'dart:convert';
2
3 class Dependente {
4   late String _nome;
5
6   Dependente(String nome) {
7     this._nome = nome;
8   }
9
10  Map<String, dynamic> toJson() {
11    return {
12      'nome': _nome,
13    };
14  }
15 }
16
17 class Funcionario {
18   late String _nome;
19   late List<Dependente> _dependentes;
20
21   Funcionario(String nome,
22     List<Dependente> dependentes) {
23     this._nome = nome;
24     this._dependentes = dependentes;
25   }
26
27   Map<String, dynamic> toJson() {
28     return {
29       'nome': _nome,
30       'dependentes':
31         _dependentes.map((dependente) =>
32           dependente.toJson()).toList(),
33     };
34   }
35 }
36
37 class EquipeProjeto {
38   late String _nomeProjeto;
39   late List<Funcionario> _funcionarios;
```

Atividade-02 PDM2

main.dart

main.dart

Format

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

```
_nomeProjeto = nomeprojeto;
_funcionarios = funcionarios;
}

Map<String, dynamic> toJson() {
  return {
    'nomeProjeto': _nomeProjeto,
    'funcionarios':
      _funcionarios.map((funcionario) =>
        funcionario.toJson()).toList(),
  };
}

void main() {
  // 1. Criar varios objetos
  Dependente
  Dependente dep1 = new
  Dependente("Gabriel");
  Dependente dep2 = new
  Dependente("Lay");
  Dependente dep3 = new
  Dependente("Cristiano");
  Dependente dep4 = new
  Dependente("Geogina");
  Dependente dep5 = new
  Dependente("Zeli");
  Dependente dep6 = new
  Dependente("Carol");

  // 2. Criar varios objetos
  Funcionario
  // 3. Associar os Dependentes
  criados aos respectivos
  // funcionarios
  Funcionario func1 = new
  Funcionario("Robinho", [dep1, dep2]);
  Funcionario func2 = new
  Funcionario("Kleber", [dep3, dep4,
  dep5]);
  Funcionario func3 = new
```

Tools

Recent

All

AI

Authentication

Chat

Code Search

Console

Database

Debugger

Docs

Extensions

Git

Packages

PostgreSQL

Secrets

Shell

SSH

System Dependencies

Threads

User Settings

Webview


```
66     Funcionario func3 = new
Funcionario("Kauan", [dep6]);
67
68     // 4. Criar uma lista de Funcionarios
69     var myFuncionarios = [func1, func2,
func3];
70
71     // 5. criar um objeto Equipe Projeto
chamando o metodo
72     //     contrutor que da nome ao
projeto e insere uma
73     //     coleção de funcionario
74     var equipeProjeto =
EquipeProjeto("Projeto Seven",
myFuncionarios);
75
76     // 6. Printar no formato JSON o
objeto Equipe Projeto.
77     var printEquipe =
jsonEncode(equipeProjeto.toJson());
78     print(printEquipe);
79 }
80 // FEITO POR LAYANA E GABRIEL
```

