

NOME: VINICIUS ALIPETI DANTAS
RA: 32410139

CÓDIGO

```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';

// Modelo para a previsão do tempo
class Previsao {
  final String data;
  final double temperatura;
  final double umidade;
  final double luminosidade;
  final double vento;
  final double chuva;
  final String unidade;

  Previsao({
    required this.data,
    required this.temperatura,
    required this.umidade,
    required this.luminosidade,
    required this.vento,
    required this.chuva,
    required this.unidade,
  });

  factory Previsao.fromJson(Map<String, dynamic> json) {
    return Previsao(
      data: json['data'],
      temperatura: json['temperatura'].toDouble(),
      umidade: json['umidade'].toDouble(),
      luminosidade: json['luminosidade'].toDouble(),
      vento: json['vento'].toDouble(),
      chuva: json['chuva'].toDouble(),
      unidade: json['unidade'],
    );
  }
}

void main() {
  runApp(PrevisaoApp());
}

class PrevisaoApp extends StatelessWidget {
```

```

@override
Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Previsão do Tempo',
    theme: ThemeData(
      primarySwatch: Colors.blue,
    ),
    home: PrevisaoPage(),
  );
}

class PrevisaoPage extends StatefulWidget {
  @override
  _PrevisaoPageState createState() => _PrevisaoPageState();
}

class _PrevisaoPageState extends State<PrevisaoPage> {
  late Future<List<Previsao>> previsoes;

  @override
  void initState() {
    super.initState();
    previsoes = fetchPrevisao();
  }

  // Função para buscar as previsões do endpoint
  Future<List<Previsao>> fetchPrevisao() async {
    final response =
      await http.get(Uri.parse('https://demo3520525.mockable.io/previsao'));

    if (response.statusCode == 200) {
      List<dynamic> data = jsonDecode(response.body);
      return data.map((item) => Previsao.fromJson(item)).toList();
    } else {
      throw Exception('Falha ao carregar a previsão do tempo');
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Previsão do Tempo'),
      ),
      body: FutureBuilder<List<Previsao>>(
        future: previsoes,
        builder: (context, snapshot) {

```

```

if (snapshot.connectionState == ConnectionState.waiting) {
  return Center(child: CircularProgressIndicator());
} else if (snapshot.hasError) {
  return Center(child: Text('Erro: ${snapshot.error}'));
} else if (snapshot.hasData) {
  return ListView.builder(
    itemCount: snapshot.data!.length,
    itemBuilder: (context, index) {
      final previsao = snapshot.data![index];
      return Card(
        margin: EdgeInsets.all(10),
        child: ListTile(
          title: Text('Data: ${previsao.data}'),
          subtitle: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                'Temperatura: ${previsao.temperatura}°${previsao.unidade}'),
              Text('Umidade: ${previsao.umidade}%'),
              Text('Luminosidade: ${previsao.luminosidade} lux'),
              Text('Vento: ${previsao.vento} m/s'),
              Text('Chuva: ${previsao.chuva} mm'),
            ],
          ),
        ),
      );
    },
  );
} else {
  return Center(child: Text('Nenhuma previsão disponível.));
}
},
);
};
}
}

```

PRINT CELULAR

