

# **Projeto: Sistema de Análise Meteorológica Inteligente**

## **Objetivo do Projeto**

Criar um sistema em Java que analisa dados meteorológicos de múltiplas cidades, identifica padrões, gera alertas e calcula estatísticas complexas.

## **Descrição do Exercício**

Os alunos devem desenvolver um programa que processa dados de temperatura e umidade de 5 cidades diferentes. O sistema deve ser capaz de:

### **Parte 1: Arquitetura Complexa**

Criar uma classe `AnaliseMeteorologica` com:

1. Método `main` que coordena todo o fluxo do programa
2. Pelo menos 8 métodos com responsabilidades específicas e bem definidas
3. Organização modular em seções claras (inicialização, processamento, saída)

### **Parte 2: Estruturas de Dados Complexas**

Utilizar arrays bidimensionais para armazenar dados:

1. Matriz 5x2 para temperaturas (máxima e mínima diária de cada cidade)
2. Matriz 5x3 para umidades (manhã, tarde, noite de cada cidade)
3. Cálculos complexos envolvendo médias ponderadas, variações e desvios

### **Parte 3: Lógica Condicional Avançada**

Implementar sistemas de decisão complexos:

1. Classificações múltiplas baseadas em combinações de condições
2. Hierarquia de alertas com prioridades (usando `if-else if-else` encadeados)
3. Métodos booleanos complexos com múltiplas condições lógicas (`&&`, `||`, `!`)
4. Validação de dados com tratamento de valores inválidos e limites

## **Parte 4: Formatação de Saída Profissional**

Gerar relatórios formatados como:

1. Tabelas alinhadas usando `printf` com width specifiers
2. Relatórios comparativos entre cidades
3. Gráficos ASCII simples para visualização de dados
4. Códigos de cores (usando sequências de escape ANSI) - opcional

## **Especificações Técnicas Detalhadas**

Dados Iniciais (Hard-coded):

Arquivo: *dados-iniciais.txt*

### **Métodos Obrigatórios:**

1. `calcularMediaPonderadaTemperatura(double max, double min)`

- Retorna média ponderada (70% máxima, 30% mínima)
- Deve validar se temperaturas estão em intervalo razoável (-50°C a 60°C)

2. `classificarClima(double tempMedia, int umidadeMedia)`

- Retorna String com classificação baseada em múltiplas condições:

"MUITO QUENTE E ÚMIDO" se temp > 30 e umidade > 75

"CONFORTAVEL" se temp entre 20-25 e umidade 50-70

"FRIO E SECO" se temp < 15 e umidade < 50

3. `identificarCidadeComMaiorAmplitudeTermica()`

- Percorre o array de temperaturas
- Calcula amplitude (máxima - mínima) para cada cidade
- Retorna índice da cidade com maior amplitude

4. `calcularIndiceCalor(double temp, int umidade)`

- Implementa fórmula simplificada:

```
'indiceCalor = temp + 0.5 * (umidade/100) * (temp - 20)'
```

- Retorna valor arredondado para 1 decimal

5. `gerarAlertas(int cidadeIndex)`

- Analisa múltiplas condições:

- \* Alerta VERMELHO: temp > 35 OU umidade > 90
  - \* Alerta AMARELO: (temp entre 30-35 E umidade > 80) OU variação > 15°C
  - \* Alerta VERDE: condições normais
  - Retorna código de alerta (0=verde, 1=amarelo, 2=vermelho)
6. `calcularEstatisticasAvancadas()`
- Retorna array com: [temperatura média geral, maior temperatura, menor temperatura, desvio padrão simples]
  - Deve usar loops implícitos (cálculos manuais em arrays)
7. `compararCidades(int cidade1, int cidade2)`
- Compara múltiplos parâmetros entre duas cidades
  - Retorna String indicando qual é "mais quente", "mais úmida", "mais confortável"
8. `gerarRelatorioDetalhado()` (método principal de saída)
- Gera tabela formatada com todas as análises
  - Usa `printf` com formatação colunar

#### **Exemplo de Saída Esperada:**

```
=====
| SISTEMA DE ANÁLISE METEOROLÓGICA INTELIGENTE
=====
=====

ANÁLISE DETALHADA POR CIDADE:

CIDADE | T.MAX | T.MIN | T.MÉD | UMIID% | CLASSIFICAÇÃO | ALERTA
-----
1 | 32.5°C| 22.1°C| 29.4°C| 73.3 | QUENTE MODERADO | AMARELO
2 | 28.3°C| 18.7°C| 25.4°C| 67.7 | CONFORTAVEL | VERDE
3 | 35.8°C| 24.9°C| 32.6°C| 78.3 | MUITO QUENTE | VERMELHO
4 | 30.2°C| 20.5°C| 27.3°C| 70.7 | QUENTE LEVE | VERDE
5 | 25.7°C| 15.3°C| 22.6°C| 64.3 | CONFORTAVEL | VERDE
-----


ESTATÍSTICAS GERAIS:
• Temperatura média geral: 27.3°C
• Cidade mais quente: #3 (35.8°C)
• Cidade mais fria: #5 (15.3°C)
• Maior amplitude térmica: #3 (10.9°C)
• Índice de calor mais alto: #3 (38.2°C)

ANÁLISE COMPARATIVA:
• Cidade #3 é significativamente mais quente que #5 (+10.1°C)
• Cidade #1 tem maior umidade que #5 (+9.0%)
• Para conforto térmico, recomenda-se visitar cidade #2
=====
```

Arquivo: saída-meteorologica.txt

**Dicas para Implementação:**

1. Comece definindo constantes para limites e classificações
2. Implemente métodos auxiliares simples primeiro
3. Use métodos de debug para imprimir arrays intermediários
4. Teste cada método isoladamente com valores conhecidos
5. Documente as fórmulas matemáticas usadas nos cálculos

**Entrega Esperada:**

1. `AnaliseMeteorologica.java` - Código completo e bem comentado
2. `RelatorioTecnico.txt` - Documentação explicando:
  - Fórmulas matemáticas utilizadas
  - Lógica das classificações
  - Decisões de design importantes
  - Desafios encontrados e soluções
3. Compactar os arquivos em: `nomeAluno.zip`

Copiar o arquivo para a pasta:

Publica\Pedro\_TI\TI44\Proj-Meteor\nomeAluno\