

# Exercícios Práticos em Dart — Etapa 4

Nesta etapa vamos trabalhar com **listas (vetores)** e **listas de listas (matrizes)** em Dart. Os exercícios ajudam a praticar entrada de dados em sequência, iteração com `for`, e manipulação de estruturas de dados.

## ✓ Instruções para Execução

Execute os blocos em um arquivo Dart com a função `main()` ou use um terminal com `dart run`:

```
dart run exercicios.dart
```

## \* Etapa 4 — Vetores e Matrizes

### ♦ Exercício 1: Armazenar e Exibir Notas com Situação

```
List<int> notas = [];  
  
for (int i = 0; i < 5; i++) {  
    stdout.write("Digite a nota ${i + 1}: ");  
    notas.add(int.parse(stdin.readLineSync(!)));  
}  
  
for (int i = 0; i < notas.length; i++) {  
    String situacao = notas[i] >= 7 ? "Aprovado" : "Reprovado";  
    print("Nota ${i + 1}: ${notas[i]} - \${situacao}");  
}
```

### ♦ Exercício 2: Maior Valor de um Vetor

```
List<int> numeros = [];  
  
for (int i = 0; i < 10; i++) {  
    stdout.write("Digite o número ${i + 1}: ");  
    numeros.add(int.parse(stdin.readLineSync(!)));  
}  
  
int maior = numeros.reduce((a, b) => a > b ? a : b);  
print("Maior número digitado: \${maior}");
```

### ◆ Exercício 3: Contar Pares e Ímpares

```
List<int> valores = [];
int pares = 0, impares = 0;

for (int i = 0; i < 8; i++) {
    stdout.write("Número ${i + 1}: ");
    int valor = int.parse(stdin.readLineSync());
    valores.add(valor);
    if (valor % 2 == 0) pares++; else impares++;
}

print("Pares: \${pares} | Ímpares: \${impares}");
```

### ◆ Exercício 4: Média de 4 Notas

```
List<double> notas = [];
double soma = 0;

for (int i = 0; i < 4; i++) {
    stdout.write("Nota ${i + 1}: ");
    double nota = double.parse(stdin.readLineSync());
    notas.add(nota);
    soma += nota;
}

double media = soma / notas.length;
print("Média: ${media.toStringAsFixed(1)} - ${media >= 6 ? 'Aprovado' : 'Reprovado'}");
```

### ◆ Exercício 5: Contar Valores Positivos

```
List<int> numeros = [];
int positivos = 0;

for (int i = 0; i < 6; i++) {
    stdout.write("Número ${i + 1}: ");
    int valor = int.parse(stdin.readLineSync());
    numeros.add(valor);
    if (valor > 0) positivos++;
}

print("Quantidade de números positivos: \${positivos}");
```

### ♦ Exercício 6: Verificar se há Negativos

```
List<int> entradas = [];
bool temNegativo = false;

for (int i = 0; i < 8; i++) {
    stdout.write("Digite o número ${i + 1}: ");
    int valor = int.parse(stdin.readLineSync());
    entradas.add(valor);
    if (valor < 0) temNegativo = true;
}

print(temNegativo ? "Há número negativo." : "Nenhum número negativo foi digitado.");
```

### ♦ Exercício 7: Buscar um Valor

```
List<int> dados = [];

for (int i = 0; i < 10; i++) {
    stdout.write("Número ${i + 1}: ");
    dados.add(int.parse(stdin.readLineSync()));
}

stdout.write("Digite um número para buscar: ");
int buscado = int.parse(stdin.readLineSync());

print(dados.contains(buscado) ? "Número encontrado." : "Número não encontrado.");
```

### ♦ Exercício 8: Contar Valores Acima da Média

```
List<double> valores = [];
double soma = 0;

for (int i = 0; i < 5; i++) {
    stdout.write("Valor ${i + 1}: ");
    double valor = double.parse(stdin.readLineSync());
    valores.add(valor);
    soma += valor;
}

double media = soma / valores.length;
int acima = valores.where((v) => v > media).length;
```

```
print("\$acima valores estão acima da média de $  
{media.toStringAsFixed(2)}.");
```

#### ♦ Exercício 9: Verificar Ordem Crescente

```
List<int> ordem = [];  
bool crescente = true;  
  
for (int i = 0; i < 6; i++) {  
    stdout.write("Número ${i + 1}: ");  
    int valor = int.parse(stdin.readLineSync());  
    ordem.add(valor);  
    if (i > 0 && valor < ordem[i - 1]) crescente = false;  
}  
  
print(crescente ? "Ordem crescente." : "Não está em ordem crescente.");
```

#### ♦ Exercício 10: Exibir em Ordem Inversa

```
List<int> numeros = [];  
  
for (int i = 0; i < 7; i++) {  
    stdout.write("Número ${i + 1}: ");  
    numeros.add(int.parse(stdin.readLineSync()));  
}  
  
print("Ordem inversa:");  
for (int i = numeros.length - 1; i >= 0; i--) {  
    print(numeros[i]);  
}
```