

Sessão 07 (02/02/2026) – Módulo 02 – Fundamentos JavaScript – Aula 04 –

```
// =====  
// LISTA DE EXERCÍCIOS - MANIPULAÇÃO DE DADOS COMPLEXOS  
// Sessão 8 - High Order Functions, Imutabilidade, Spread, Rest e Destructuring  
// =====
```

```
// =====  
// TEMA 2: map() - 3 EXERCÍCIOS  
// =====
```

```
// Exercício 2.1: Criar lista de nomes completos  
// Use map para criar um array com nomes completos (nome + apelido)
```

```
console.log("Exercício 2.1: Nomes completos");  
const pessoas = [  
  { nome: "João", apelido: "Silva" },  
  { nome: "Maria", apelido: "Santos" },  
  { nome: "Pedro", apelido: "Costa" }  
];
```

```
// Resultado esperado: ["João Silva", "Maria Santos", "Pedro Costa"]
```

```
// Exercício 2.2: Aplicar IVA aos preços  
// Use map para adicionar 23% de IVA a cada preço  
// Retorne objetos com precoSemIVA e precoComIVA
```

```
const precosSemIVA = [100, 250, 50, 1200];
```

```
// Resultado: array de objetos com ambos os preços
```

```
// Exercício 2.3: Formatar datas  
// Use map para formatar as datas no formato "DD/MM/AAAA"
```

```
console.log("Exercício 2.3: Formatar datas");  
const datas = [
```

```
{ dia: 15, mes: 3, ano: 2024 },  
{ dia: 1, mes: 12, ano: 2023 },  
{ dia: 25, mes: 6, ano: 2024 }  
];
```

```
// Resultado: ["15/03/2024", "01/12/2023", "25/06/2024"]
```

```
// =====  
// TEMA 3: filter() - 2 EXERCÍCIOS  
// =====
```

```
// Exercício 3.2: Filtrar strings longas  
// Filtre apenas as palavras com mais de 5 caracteres
```

```
const palavras = ["JavaScript", "HTML", "CSS", "Python", "Java", "C"];
```

```
// Exercício 3.3: Filtrar transações por tipo e valor  
// Filtre transações do tipo "débito" com valor maior que 100€
```

```
const transacoes = [  
  { tipo: "crédito", valor: 500, descricao: "Salário" },  
  { tipo: "débito", valor: 150, descricao: "Compras" },  
  { tipo: "débito", valor: 50, descricao: "Combustível" },  
  { tipo: "débito", valor: 200, descricao: "Renda" }  
];
```

```
// =====  
// TEMA 4: find(), some(), every() - 3 EXERCÍCIOS  
// =====
```

```
// Exercício 4.1: Encontrar utilizador por username  
// Use find para encontrar o utilizador com username "admin"
```

```
const utilizadores = [  
  { id: 1, username: "joao123", role: "user" },
```

```
{ id: 2, username: "admin", role: "admin" },
{ id: 3, username: "maria456", role: "user" }
];

// Exercício 4.2: Verificar se há produtos esgotados
// Use some para verificar se existe algum produto com stock === 0

console.log("Exercício 4.2: Verificar produtos esgotados");
const estoque = [
  { produto: "Camisa", stock: 10 },
  { produto: "Calça", stock: 5 },
  { produto: "Sapatos", stock: 0 }
];

// Exercício 4.3: Verificar se todos passaram
// Use every para verificar se todos os alunos têm média >= 9.5

const alunos = [
  { nome: "Ana", media: 15.5 },
  { nome: "Bruno", media: 12.0 },
  { nome: "Carla", media: 18.5 },
  { nome: "Daniel", media: 14.0 }
];

// =====
// TEMA 5: reduce() - 3 EXERCÍCIOS
// =====

console.log("--- TEMA 5: reduce() ---\n");

// Exercício 5.1: Calcular média de idades
// Use reduce para calcular a média de idade dos utilizadores

console.log("Exercício 5.1: Média de idades");
const usuarios = [
  { nome: "João", idade: 25 },
  { nome: "Maria", idade: 30 },
```

```
{ nome: "Pedro", idade: 28 },  
{ nome: "Ana", idade: 22 }  
];
```

```
// Exercício 5.2: Contar ocorrências de cada fruta  
// Use reduce para contar quantas vezes cada fruta aparece
```

```
const frutas = ["maçã", "banana", "maçã", "laranja", "banana", "maçã"];
```

```
// Resultado esperado: { maçã: 3, banana: 2, laranja: 1 }
```

```
// Exercício 5.3: Concatenar nomes com vírgulas  
// Use reduce para juntar todos os nomes em uma única string  
// separada por vírgulas
```

```
const nomes = ["Ana", "Bruno", "Carla", "Daniel"];
```

```
// Resultado: "Ana, Bruno, Carla, Daniel"
```

```
// =====  
// TEMA 6: Spread Operator (...) - 3 EXERCÍCIOS  
// =====
```

```
// Exercício 6.1: Juntar dois arrays de números  
// Use spread para criar um novo array com todos os números  
// dos dois arrays
```

```
const numeros1 = [1, 2, 3];  
const numeros2 = [4, 5, 6];
```

```
// Exercício 6.2: Adicionar propriedades ao objeto  
// Use spread para criar uma cópia do produto e adicionar  
// as propriedades: emPromocao: true e desconto: 15
```

```
const produto = {  
  id: 1,
```

```
    nome: "Notebook",  
    preco: 1200  
};
```

```
// Exercício 6.3: Inserir elemento no início do array  
// Use spread para adicionar "TypeScript" no início do array  
// sem modificar o original
```

```
const linguagens = ["JavaScript", "Python", "Java"];
```

```
// =====  
// TEMA 7: Destructuring - 3 EXERCÍCIOS  
// =====
```

```
// Exercício 7.1: Extrair coordenadas  
// Use destructuring para extrair latitude e longitude
```

```
const localizacao = {  
    cidade: "Lisboa",  
    pais: "Portugal",  
    coordenadas: {  
        latitude: 38.7223,  
        longitude: -9.1393  
    }  
};
```

```
// Extraia latitude e longitude usando destructuring
```

```
// Exercício 7.2: Trocar valores de variáveis  
// Use destructuring para trocar os valores de a e b  
// sem usar variável auxiliar
```

```
let a = 10;  
let b = 20;
```

```
// Exercício 7.3: Extrair primeiro e último elemento
// Use destructuring com rest para extrair o primeiro
// elemento e todos os outros
```

```
const numeros = [10, 20, 30, 40, 50];
```

```
// Extraia: primeiro e resto
```

```
// =====
// TEMA 8: Encadeamento (Chaining) - 3 EXERCÍCIOS
// =====
```

```
// Exercício 8.1: Processar notas
// Encadeie métodos para:
// 1. Filtrar notas >= 10
// 2. Calcular a média das notas aprovadas
```

```
const notas = [8, 12, 15, 7, 18, 9, 16, 11];
```

```
// (use filter e reduce encadeados)
```

```
// Exercício 8.2: Obter emails de utilizadores ativos
// Encadeie métodos para:
// 1. Filtrar apenas utilizadores ativos
// 2. Extrair apenas os emails
// 3. Juntar em string separada por ponto e vírgula
```

```
const users = [
  { nome: "João", email: "joao@mail.com", ativo: true },
  { nome: "Maria", email: "maria@mail.com", ativo: false },
  { nome: "Pedro", email: "pedro@mail.com", ativo: true }
];
```

```
// (filter → map → join)
```

```
// Exercício 8.3: Produtos mais vendidos
// Encadeie métodos para:
```

```
// 1. Filtrar produtos com vendas > 50
// 2. Ordenar por vendas (maior para menor)
// 3. Pegar apenas os 2 primeiros
// 4. Extrair apenas os nomes
```

```
const vendidos = [
  { nome: "Produto A", vendas: 45 },
  { nome: "Produto B", vendas: 120 },
  { nome: "Produto C", vendas: 80 },
  { nome: "Produto D", vendas: 30 }
];
```

```
// (filter → sort → slice → map)
```