

---

## Estudo de Caso Geral: Sistema de Gestão de Funcionários de uma Startup de Tecnologia

A startup deseja um sistema simples no navegador para gerenciar os funcionários cadastrados. Os dados devem ser manipulados via JavaScript e exibidos dinamicamente em uma tabela HTML. Os usuários poderão **cadastrar**, **excluir**, **alterar**, **listar**, e gerar **relatórios** com **streams JS**.

---

## Exercício 1 – Cadastro e Listagem de Funcionários

**Conceitos:** Formulário HTML, Classe, Atributos, Construtor, Métodos, Eventos

### Objetivo:

Crie uma interface HTML com um formulário para cadastrar funcionários com os seguintes campos:

- **Nome** (input text)
- **Idade** (input number)
- **Cargo** (input text)
- **Salário** (input number)

Armazene os funcionários em um array de objetos da classe **Funcionario**, que deve ter os atributos e métodos adequados.

### Requisitos:

- Ao clicar em "Cadastrar", adicione o funcionário à lista.
- Exiba os dados em uma **<table>**.
- Crie a classe **Funcionario** com construtor, métodos de acesso (get/set) e um método **toString()**.

---

## Exercício 2 – Excluir e Alterar Funcionários

**Conceitos:** Eventos, Métodos, Manipulação de DOM

### Objetivo:

Permita ao usuário **excluir** e **editar** um funcionário da tabela.

### Requisitos:

- Inclua um botão "Excluir" ao lado de cada funcionário na tabela.
- Ao clicar em "Editar", os dados devem ser carregados no formulário para edição.
- Atualize os dados após edição.

Use métodos da classe `Funcionario` para atualizar os atributos. Utilize eventos `onclick` com funções anônimas.

---

## Exercício 3 – Funções Anônimas e Expressões Lambda

**Conceitos:** Funções anônimas, Funções lambda, Eventos

### Objetivo:

Refatore os eventos de cadastro, exclusão e edição para utilizar:

- **Funções anônimas** no evento de clique dos botões.
- **Funções lambda** para manipular o array de funcionários (ex: para buscar um funcionário pelo ID ou nome).

### Requisitos:

- Reescreva as funções de manipulação (busca, remoção, atualização) usando arrow functions.
- Mostre como substituir uma função anônima tradicional por uma arrow function.

---

## Exercício 4 – Relatório com Métodos de Array (map, filter, reduce)

Conceitos: Streams JS (`map`, `filter`, `reduce`)

### Objetivo:

Crie relatórios dinâmicos usando métodos de array JS.

### Relatórios obrigatórios:

- Listar todos os funcionários com salário maior que R\$ 5000.
- Mostrar a média salarial dos funcionários.
- Listar apenas os cargos únicos (sem repetição).
- Criar uma lista de nomes em maiúsculo.

### Dica:

Use `filter`, `map`, `reduce`, e `new Set()` para manipulação e geração de relatórios.

---

### Observação:

- O sistema deve ser versionado e compartilhado no GitHub
- Deve-se utilizar container Docker para o desenvolvimento