

Gerenciado de Tarefas

Objetivo: Desenvolver um sistema simples de gerenciamento de tarefas em JavaScript padrão, aplicando os conceitos solicitados e salvando as tarefas no `localStorage`.

Descrição

Você deve criar um programa em JavaScript que permita adicionar, listar, buscar e remover tarefas. O objetivo é aplicar os conceitos de `let`, `const`, `arrow functions`, `template literals`, `shorthand props`, `default params`, `métodos de string`, `métodos de array`, `destruturação`, `spread operator`, `rest params`, `promises`, e `async/await`.

Requisitos Obrigatórios

1. Configuração Inicial:

- Crie um arquivo HTML simples com um campo de entrada, botões para cada funcionalidade e uma área para exibir as tarefas.
- Inicialize um array de tarefas e salve-o no `localStorage`.

2. Estrutura da Tarefa:

- Cada tarefa deve ser um objeto com as seguintes propriedades:

```
{ id: number, title: string, description: string, completed: boolean }
```

- Use `shorthand props` para criar os objetos de tarefas.

3. Funções:

- **Adicionar Tarefa:**

- Crie uma função `addTask` que adiciona uma nova tarefa ao array. Use `default params` para definir valores padrão para `completed`.

- **Listar Tarefas:**

- Crie uma função `listTasks` que exibe todas as tarefas formatadas com `template literals`.

- **Buscar Tarefa por Título:**

- Crie uma função `findTaskByTitle` que retorna uma tarefa pelo título utilizando `find`.

- **Remover Tarefa:**

- Crie uma função `removeTask` que remove uma tarefa pelo ID utilizando `findIndex` e `splice`.

4. Manipulação de Strings:

- Adicione métodos para buscar tarefas cujo título comece (`startsWith`), termine (`endsWith`) ou inclua (`includes`) uma string específica.

5. Manipulação de Arrays:

- Use métodos como `map`, `reduce`, `fill`, `includes`, `Array.from`, e `Array.of` para manipular o array de tarefas.
 - Exemplo: Crie uma função que retorna todos os títulos das tarefas usando `map`.

6. Desestruturação e Operadores Spread/Rest:

- Use desestruturação para extrair propriedades de objetos e arrays.
- Use o operador spread para copiar e adicionar novas tarefas.
- Use parâmetros rest para funções que aceitam múltiplos argumentos.

7. Promessas e Async/Await:

- Simule chamadas a uma API para carregar e salvar tarefas utilizando `promises` e `async/await`.
- Crie funções `loadTasks` e `saveTasks` que simulam leitura e gravação de tarefas com atraso.