



Frontend II

Avançando o desenvolvimento do To-Do App - Parte 2/2

Na última aula, conseguimos capturar as tarefas de um usuário específico e as exibimos no console da aplicação. Na aula de hoje, precisamos finalizar a implementação de todos os requisitos essenciais do projeto. Para isso, você deverá desenvolver a estrutura que permite a renderização das tarefas obtidas.

Requisitos funcionais

- Dentro do script **tasks.js** continuaremos a criar uma função dedicada à renderização.
 - Esta função deve receber como parâmetro uma **lista**.
 - Em cada iteração, deve ser avaliado se a tarefa corresponde à lista de "tarefas-pendentes" ou "tarefas-terminadas".
 - Em cada conteúdo html deve ser inserido um item que respeite a estrutura dos seguintes modelos:

Tarefa terminada

```
<li class="tarefa">
  <div class="done"></div>
  <div class="descricao">
    <p class="nome">${tarefa.descricao}</p>
    <div>
      <button><i id="${tarefa.id}" class="fas
fa-undo-alt change"></i></button>
      <button><i id="${tarefa.id}" class="far
fa-trash-alt"></i></button>
    </div>
  </div>
</li>
```



Tarefa pendente

```
<li class="tarefa">
    <div class="not-done" id="${tarefa.id}"></div>
    <div class="descricao">
        <p class="nome">${tarefa.descricao}</p>
        <p class="timestamp"><i class="far
fa-calendar-alt"></i> ${tarefa.createdAt}</p>
    </div>
</li>
```

- Você deve desenvolver uma função dedicada a alterar o status das tarefas, de pendentes para terminadas e vice-versa.
- Você deve desenvolver uma função dedicada a apagar uma tarefa específica presente na lista das “tarefas-terminadas” (ao clicar na opção representada pelo ícone de lixeira)
- Após a execução das funções de renderização das tarefas, crie uma nova função que seja responsável por **fechar a sessão**. Ela deve remover o usuário logado e removê-lo da visualização. Use o botão correspondente na interface para implementar esta função.

Requisitos não funcionais

- ❖ Mantenha um código ordenado em blocos para poder identificar as diferentes funcionalidades.
- ❖ Chame a função de renderização sempre que julgar necessário.

[**DOCUMENTAÇÃO DA API**](#)