Projeto Avaliativo 1 - To-Do List

Peso: 40% da nota do módulo 1

DEVinHouse

Sumário

1. Introdução

Você está participando de um processo seletivo para ingressar em uma vaga de programador em uma grande empresa de TI. Uma das etapas do processo envolve a criação de uma aplicação web de lista de afazeres (“to-do list”).

2. Requisitos da Aplicação

A aplicação que deverá ser realizada individualmente, deve contemplar os seguintes requisitos:

- Um título na aba do navegador, para que o usuário encontre a sua aplicação no meio das várias abas que constantemente mantém abertas.
- Um cabeçalho dentro da página, para que o usuário saiba facilmente em que página se encontra e do que se trata o conteúdo.
- Um campo de texto para digitar o nome de uma nova atividade a ser adicionada à lista.
- Um botão para adicionar uma nova atividade à lista.
- Uma lista contendo as atividades já inseridas.
- Cada linha da lista deve conter: checkbox para o usuário marcar que aquela atividade já foi realizada; o texto que o usuário digitou ao cadastrar a atividade; botão para excluir a atividade da lista, caso desejado.
- Quando o usuário marcar uma tarefa como realizada, o texto daquela linha deve ser tachado (line-through).
- A lista deve ser salva no "localStorage" do navegador (incluindo quais itens já foram realizados), e deve ser carregada sempre que a página for reaberta.

3. Exemplo de aplicação

A aplicação deverá conter os requisitos apresentados anteriormente, sendo codificada em html, css, javascript e markdown (para o readme.md).
A imagem a seguir demonstra um exemplo do FrontEnd que deverá ser desenvolvido.

4. Entrega

O código desenvolvido deverá ser submetido no GitHub, e o link deverá ser disponibilizado na tarefa **Submeter aqui a URL do projeto 1**, presente na semana 4 do AVA até o dia 14/11/2021 às 23h55.
Entregas realizadas após a data limite sofrerão **decréscimo na nota** de avaliação, sendo considerado **80%** da nota para tarefas submetidas até o dia 21/11/2021 às 23h55 e **50%** para tarefas submetidas até o dia 28/11/2021 às 23h55. **Não serão** avaliados projetos submetidos após o dia 28/11/2021.
Importante: Será considerado como data de entrega a **última atualização** no repositório do projeto no GitHub. Lembre-se de não modificar o código até receber sua nota.

5. Critérios de Avaliação

A tabela abaixo apresenta os critérios que serão avaliados durante a correção do projeto. O mesmo possui variação de nota de 0 (zero) a 10 (dez) como nota mínima e máxima, e possui peso de 40% sobre a avaliação do módulo 1.

Serão desconsiderados e atribuída a nota 0 (zero) os projetos que apresentarem plágio de soluções encontradas na internet ou de outros colegas. Lembre-se: Você está livre para utilizar outras soluções como base, mas **não é permitida** a cópia.

Nº	Critério de Avaliação	0	0,15	0,3	0,5
1	O aluno desenvolveu uma página que apresenta o título? (peso 0,5)	O aluno não conseguiu apresentar título na página.	O aluno conseguiu apresentar título, mas que não representa o conteúdo.	O aluno conseguiu apresentar título que representa bem o conteúdo.	Além de o aluno conseguir apresentar título que representa bem o conteúdo, também adicionou um ícone de favorito.
2	O aluno desenvolveu uma página que apresenta o cabeçalho do conteúdo? (peso 0,5)	O aluno não conseguiu desenvolver uma página que apresenta um cabeçalho de conteúdo.	O aluno conseguiu desenvolver uma página que apresenta um cabeçalho, porém ele não representa o conteúdo ou não utilizou a tag correta.	O aluno conseguiu desenvolver uma página que apresenta um cabeçalho, que representa bem o conteúdo e utilizou a tag correta.	O aluno, além de desenvolver uma página que apresente cabeçalho, que representa bem o conteúdo e utiliza a tag correta, também posicionou bem o cabeçalho na página para uma melhor visualização do usuário.

Nº	Critério de Avaliação	0	0,3	0,7	1
3	O aluno desenvolveu um campo de texto para digitação de novas atividades? (peso 1)	O aluno não desenvolveu um campo de texto para digitação de novas atividades.	O aluno desenvolveu um campo de texto mal posicionado ou não utilizou a tag correta.	O aluno desenvolveu um campo de texto bem posicionado e utilizando a tag correta.	O aluno, além de inserir campo de texto bem posicionado e utilizar a tag correta, também inseriu um rótulo ou placeholder informando o usuário sobre do que se trata aquele campo, e as propriedades de seleção/identificação foram utilizadas corretamente.

4	O aluno desenvolveu um botão para inserção de novas atividades na lista? (peso 1)	O aluno não conseguiu apresentar o botão.	O aluno desenvolveu um botão, mas não desenvolveu o funcionamento do botão para inserir itens na lista.	O aluno desenvolveu o botão que insere o texto digitado na lista, posicionou bem o botão na página e inseriu texto informando o que o botão faz ao ser clicado.	Além de o aluno programar o botão para cadastrar a atividade na lista, posicioná-lo bem na página e informar corretamente o que o botão faz, o aluno também fez com que o clique no botão apague o texto digitado no campo textual, utilizou a tag e as propriedades de seleção/identificação corretamente.
5	O aluno desenvolveu uma lista que exibe as atividades cadastradas? (peso 1)	O aluno não desenvolveu uma lista que exibe as atividades cadastradas.	O aluno desenvolveu uma lista que exibe as atividades cadastradas, mas mal posicionada e com o conteúdo dos itens desorganizados.	O aluno desenvolveu uma lista, que exibe as atividades cadastradas, bem posicionada e com o conteúdo dos itens (checkbox, texto, botão) bem organizados de uma forma agradável e intuitiva.	O aluno, além de desenvolver uma lista que exibe as atividades cadastradas bem posicionada, e com conteúdo bem organizado, também utilizou a tag e as propriedades de seleção/identificação corretamente.
6	O aluno desenvolveu uma lista, sendo que os itens da lista apresentam "checkbox" para marcar quando uma atividade foi concluída? (peso 1)	O aluno não desenvolveu uma lista, sendo que os itens da lista apresentam "checkbox" para marcar quando uma atividade foi concluída	O aluno desenvolveu uma lista, sendo que os itens da lista apresentam "checkbox" para marcar quando uma atividade foi concluída, mas numa posição em que fica confuso de entender qual a atividade referente.	O aluno desenvolveu uma lista, sendo que os itens da lista apresentam "checkbox" para marcar quando uma atividade foi concluída, em uma posição de fácil detecção de qual atividade ele está relacionado.	Além de o aluno apresentar o checkbox numa posição que facilite a identificação de qual a atividade relacionada, ao marcar a caixinha a atividade relacionada é riscada/tachada.

7	O aluno desenvolveu uma lista, sendo que os itens da lista apresentam o texto que o usuário digitou no momento do cadastro da atividade? (peso 1)	O aluno não desenvolveu uma lista, sendo que os itens da lista apresentam o texto que o usuário digitou no momento do cadastro da atividade.	Aluno desenvolveu uma lista que apresenta algum texto em seus itens, mas não exatamente o que foi cadastrado.	Aluno desenvolveu uma lista que apresenta exatamente o texto digitado pelo usuário no momento do cadastro da tarefa.	Além de o aluno desenvolver uma lista que apresenta exatamente o texto digitado pelo usuário, também conseguiu estilizar o texto para ser facilmente legível e conter um bom espaçamento entre os outros elementos (checkbox e botão excluir) e as outras atividades da lista.
8	O aluno desenvolveu uma lista, em que os itens da lista apresentam um botão para excluir a atividade? (peso 1)	O aluno não desenvolveu uma lista, em que seus itens apresentem um botão para excluir a atividade.	O aluno desenvolveu uma lista, em que seus itens apresentam um botão, mas o botão não exclui a atividade relacionada ao ser clicado.	O aluno desenvolveu uma lista, em que os itens da lista apresentam um botão para excluir a atividade, em uma posição de fácil identificação com a atividade relacionada e com texto/ícone informando o que o botão faz ao ser clicado.	Além de o aluno conseguir programar o clique do botão para excluir a atividade correta da lista, posicionar bem o botão na página e informar corretamente o que ele faz, o aluno também programou para a página pedir uma confirmação do usuário antes de excluir o item, e utiliza a tag e as propriedades de seleção/identificação corretamente.

9	O aluno desenvolveu uma lista em que os afazeres ficam salvos no LocalStorage e é recarregada quando a página é fechada e reaberta? (peso 1)	O aluno não desenvolveu uma lista em que os afazeres ficam salvos no LocalStorage e é recarregada quando a página é fechada e reaberta	O aluno desenvolveu uma lista em que os afazeres ficam salvos no LocalStorage, mas não conseguiu carregar a mesma ao fechar/reabrir a página.	O aluno desenvolveu uma lista em que os afazeres ficam salvos no LocalStorage e é recarregada quando a página é fechada e reaberta.	Aluno conseguiu salvar a lista sempre no momento que a mesma sofreu qualquer alteração (inserção/exclusão de item), e conseguiu carregá-la corretamente ao reabrir a página.
10	O aluno desenvolveu uma página que apresenta um design agradável e intuitivo? (peso 1)	O aluno desenvolveu uma página que não apresenta um design agradável e intuitivo	O aluno desenvolveu uma página e inseriu alguns estilos, como tipo/tamanho/cor de fonte, largura/altura/margem de alguns elementos, mas todos inseridos diretamente no HTML.	O aluno desenvolveu uma página e inseriu alguns estilos, bem organizados no seu próprio arquivo, separado do HTML.	Além de o aluno conseguir apresentar estilos textuais e de posicionamento básico, num arquivo separado do HTML, também estilizou o fundo da página, os botões, a lista e economizou texto de botão quando não havia muito espaço.
11	O aluno desenvolveu um código JavaScript que está bem organizado e é facilmente legível, conforme as boas práticas propostas pelos grandes nomes do desenvolvimento de software? (peso 1)	Aluno não conseguiu inserir código JavaScript na página.	O código JS desenvolvido pelo aluno está bagunçado, com nomes de variáveis e funções não explicativas, sem indentação correta e misturado com o HTML.	O código JS desenvolvido pelo aluno está separado em um arquivo diferente do HTML, mas ainda apresenta algum problema de organização/legibilidade (indentação incorreta, nomes de variáveis/funções não explicativos)	O código JS desenvolvido pelo aluno está separado em um arquivo diferente do HTML, bem organizado, com nomes de funções e variáveis explicativos e indentação correta.

6. Plano de Projeto

Ao construir uma aplicação de *To-do List* salvando os dados no *LocalStorage* do navegador, o aluno estará colocando em prática os aprendizados em:

- HTML: principais tags como head, meta, title, body, div, h1, form, input, button, ul, li. Atributos de tags como class, id, type. Inclusão de arquivos de estilos (css) e de script (js) na página.
- CSS: estilizar a página, os botões, inputs, alterar atributos dos elementos da tela de acordo com a interação do usuário para uma melhor experiência do usuário (UX).
- Javascript: variáveis, arrays, funções, manipulação do DOM (eventos, elementos e seus atributos), manipular objetos (JSON), utilizar o localStorage.