



inaC Merge pull request #9 from tryber/enhancement/add-resolution-resizin...



on 2 Mar

5

[View code](#)

README.md



# Boas vindas ao repositório do projeto Lista de Tarefas!

Você já usa o GitHub diariamente para desenvolver os exercícios, certo? Agora, para desenvolver os projetos, você deverá seguir as instruções a seguir. Fique atento a cada passo e, se tiver qualquer dúvida, nos envie por *Slack*! #vqv 🚀

Aqui você vai encontrar os detalhes de como estruturar o desenvolvimento do seu projeto a partir deste repositório, utilizando uma branch específica e um *Pull Request* para colocar seus códigos.

## O que deverá ser desenvolvido

Neste projeto, você implementará uma lista de tarefas. Ou seja, uma aplicação que permita a pessoa que usa adicionar tarefas, marcá-las como concluídas e limpar todas ou apenas aquelas que foram concluídas.

Abaixo você pode ver um exemplo de uma lista de tarefas. Utilize este exemplo como referência. Você poderá estilizar seu projeto da forma que desejar desde que todos os requisitos sejam cumpridos. Use sua imaginação!

## Minha lista de tarefas

Clique duas vezes em um item para marcá-lo como completo

Adicionar

×

↑

↓

New recording

Limpar Completos

Limpar Lista

Salvar Lista

## Entregáveis

Para entregar o seu projeto você deverá criar um *Pull Request* para este repositório no **GitHub**.

Este *Pull Request* deverá conter, necessariamente, os arquivos `index.html`, `style.css` e `script.js`, que conterão seu código **HTML**, **CSS** e **JavaScript**, respectivamente. ⚠ É importante que seus arquivos tenham exatamente estes nomes! ⚠

Você pode adicionar outros arquivos se julgar necessário.

## Requisitos do projeto

A seguir, estão listados todos os requisitos do projeto. Leia-os atentamente e siga à risca o que for pedido. Em particular, atente-se para os nomes de classes e ids que alguns elementos de seu projeto devem possuir. **Não troque ids por classes ou vice-versa**. O não cumprimento de um requisito, total ou parcialmente, impactará em sua avaliação.

Os requisitos do seu projeto são avaliados automaticamente, sendo utilizada a resolução `1366 x 768` (1366 pixels de largura por 768 pixels de altura). Logo, recomenda-se desenvolver seu projeto usando a mesma resolução, via instalação [deste plugin](#) do `Chrome` para facilitar a configuração dessa resolução.

Você tem liberdade para adicionar novos comportamentos ao seu projeto, seja na forma de aperfeiçoamentos em requisitos propostos ou novas funcionalidades, **desde que tais comportamentos adicionais não conflitem com os requisitos propostos**. Em outras palavras, você pode fazer mais do que foi pedido, mas nunca menos. Contudo, tenha em mente que **nada além do que foi pedido nos requisitos será avaliado**. Esta é uma oportunidade de você exercitar sua criatividade e experimentar os conhecimentos adquiridos.

**1 - Sua lista deve possuir o título "Minha Lista de Tarefas" em uma tag `<header>` . Esse título não varia de acordo com o conteúdo da lista. *Não confunda esse título com o título da página, que deve aparecer dentro da tag `head` .***

**2 - Abaixo do título deve haver um pequeno e discreto parágrafo com `id="funcionamento"` e com o texto "Clique duas vezes em um item para marcá-lo como completo" explicando o funcionamento da lista para o usuário.**

**3 - Deve haver um `input` com o `id="texto-tarefa"` onde o usuário poderá digitar o nome do item que deseja adicionar à lista.**

**4 - Deve haver uma lista ordenada de tarefas com o `id="lista-tarefas"` .**

**5 - Deve haver um botão com `id="criar-tarefa"` e ao clicar nesse botão, um novo item deverá ser criado ao final da lista e o texto do input deve ser limpo.**

**6 - Os itens da lista de tarefas devem ser ordenados por ordem de criação.**

**7 - Ao passar o mouse sobre qualquer botão o ícone do mouse deve mudar para uma mãozinha apontando ele.**

**8 - Ao clicar em um item da lista, altere a cor de fundo do item para cinza `rgb(128,128,128)`.**

**9 - Ao clicar duas vezes em um item, ele deverá ser riscado, indicando que foi completo. Deve ser possível desfazer essa ação**

**clicando novamente duas vezes no item.**

Pontos importantes sobre este requisito:

- \* Crie uma classe CSS com o nome "completed" e defina a propriedade "text-decoration" com o valor "line-through".

- \* Utilize a classe CSS "completed" para adicionar o efeito de letra tachada (riscada) as tarefas finalizadas.

**10 - Deve haver um botão com `id="apaga-tudo"` que quando clicado deve apagar todos os itens da lista.**

**11 - Deve haver um botão `id="remover-finalizados"` que quando clicado remove somente os elementos finalizados da sua lista.**

## BÔNUS

**1 - Adicione um botão com `id="salvar-tarefas"` que salve o conteúdo da lista. Se você fechar e reabrir a página, a lista deve continuar no estado em que estava. Dica: Leia sobre *Local Storage* na seção de recursos adicionais.**

**2 - Como sua lista é ordenada, o que acontece se você esquecer de um item? Você teria que apagar a lista e começar tudo de novo, pois só é possível inserir um item no final, certo? Adicione dois botões, um com `id="mover-cima"` e outro com `id="mover-baixo"`, que permitam mover o item selecionado para cima ou para baixo na lista de tarefas.**

Pontos importantes sobre este requisito bônus:

- \* Antes de começar a desenvolver essa funcionalidade, pare e pense. O que significa mover um item de uma lista para cima ou para baixo no `**_DOM_**`? Você já possui todas as habilidades necessárias para fazer isso.

- \* Habitue-se a pensar nos casos especiais ao construir programas. O que acontece se o usuário tentar mover o primeiro item para cima ou o último para baixo?

3 - Adicione um botão com `id="remover-selecionado"` que, quando clicado, remove o item selecionado.

---

## Instruções para entregar seu projeto:

---

### ANTES DE COMEÇAR A DESENVOLVER:

1. Clone o repositório

- `git clone https://github.com/tryber/sd-03-block5-project-todo-list.git` .

2. Crie uma branch a partir da branch `master`

- Verifique que você está na branch `master`
  - Exemplo: `git branch`
- Se não estiver, mude para a branch `master`
  - Exemplo: `git checkout master`
- Agora, crie uma branch onde você vai guardar os `commits` do seu projeto
  - Você deve criar uma branch no seguinte formato: `nome-de-usuario-nome-do-projeto`
  - Exemplo: `git checkout -b exemplo-todo-list-project`

3. Adicione as mudanças necessárias para iniciar a implementação do seu projeto ao `stage` do Git e faça um `commit`

- Verifique que as mudanças ainda não estão no `stage`
  - Exemplo: `git status` (devem aparecer listados os arquivos modificados em vermelho)
- Adicione o novo arquivo ao `stage` do Git
  - Exemplo:
    - `git add .` (adicionando todas as mudanças - *que estavam em vermelho* - ao `stage` do Git)
    - `git status` (devem aparecer listados os arquivos modificados em verde)
- Faça o `commit` inicial
  - Exemplo:
    - `git commit -m 'iniciando o projeto. VAMOS COM TUDO :rocket:'` (fazendo o primeiro `commit`)
    - `git status` (deve aparecer uma mensagem tipo *nothing to commit*)



4. Adicione a sua branch com o novo `commit` ao repositório remoto

- Usando o exemplo anterior: `git push -u origin exemplo-todo-list-project`

## 5. Crie um novo `Pull Request` (PR)

- Vá até a página de *Pull Requests* do [repositório no GitHub](#)
- Clique no botão verde "New pull request"
- Clique na caixa de seleção "Compare" e escolha a sua branch **com atenção**
- Clique no botão verde "Create pull request"
- Adicione uma descrição para o *Pull Request* e clique no botão verde "Create pull request"
- **Não se preocupe em preencher mais nada por enquanto!**
- Volte até a [página de Pull Requests do repositório](#) e confira que o seu *Pull Request* está criado

## DURANTE O DESENVOLVIMENTO

-  **LEMBRE-SE DE CRIAR TODOS OS ARQUIVOS DENTRO DA PASTA COM O SEU NOME** 
- Faça `commits` das alterações que você fizer no código regularmente
- Lembre-se de sempre após um (ou alguns) `commits` atualizar o repositório remoto
- Os comandos que você utilizará com mais frequência são:
  - i. `git status` (para verificar o que está em vermelho - fora do stage - e o que está em verde - no stage)
  - ii. `git add` (para adicionar arquivos ao stage do Git)
  - iii. `git commit` (para criar um commit com os arquivos que estão no stage do Git)
  - iv. `git push -u nome-da-branch` (para enviar o commit para o repositório remoto na primeira vez que fizer o `push` de uma nova branch)
  - v. `git push` (para enviar o commit para o repositório remoto após o passo anterior)

## DEPOIS DE TERMINAR O DESENVOLVIMENTO

Para "entregar" seu projeto, siga os passos a seguir:

- Vá até a página **DO SEU Pull Request**, adicione a label de "code-review" e marque seus colegas
  - No menu à direita, clique no *link* "**Labels**" e escolha a *label* **code-review**

- No menu à direita, clique no *link* "**Assignees**" e escolha **o seu usuário**
- No menu à direita, clique no *link* "**Reviewers**" e digite `students`, selecione o time `tryber/students-sd-03`

Se ainda houver alguma dúvida sobre como entregar seu projeto, [aqui tem um video explicativo](#).

## REVISANDO UM PULL REQUEST



À medida que você e os outros estudantes forem entregando os projetos, vocês serão alertados **via Slack** para também fazer a revisão dos *Pull Requests* dos seus colegas. Fiquem atentos às mensagens do "*Pull Reminders*" no *Slack*!

Use o material que você já viu sobre [Code Review](#) para te ajudar a revisar os projetos que chegaram para você.

### Releases

No releases published

[Create a new release](#)

### Packages

No packages published

[Publish your first package](#)

### Contributors 3



inaC



jeanpsv Jean Paulo Silva Vasconcelos



LeandroLM