

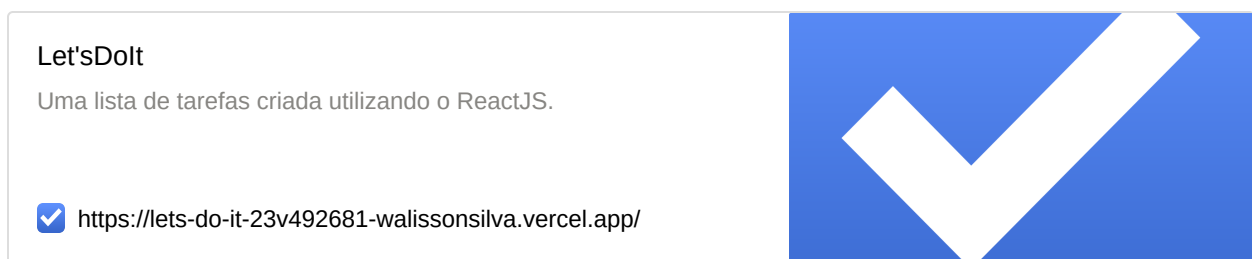
# Challenge Web - Fevereiro

Fala, pessoal! Sejam muito bem-vindos ao 1º Challenge de Web disponibilizado no Let's Code Pass! Você deve estar acompanhando as Short Classes de Web e, agora, é hora de colocar seus novos conhecimentos em prática! Vamos nessa?! 🚀

Antes de tudo, é válido destacar que por se tratar de um desafio, é essencial que você troque uma ideia com outras pessoas, caso sinta dificuldades, ou mesmo possa utilizar do nosso servidor do Discord para enviar suas dúvidas. Ah, claro, você também pode ficar de olho nas dúvidas de outras pessoas para ajudá-las. Tudo isso vai te ajudar a avançar cada vez mais rumo ao seu objetivo!

## Conhecendo o Challenge

Indo direto ao ponto... **Seu desafio será desenvolver uma aplicação para lista de tarefas, chamada Let'sDolt!** Caso deseje observar o resultado esperado por meio de uma aplicação de exemplo, você pode consultar o link abaixo:



A aplicação acima foi desenvolvida, utilizando ReactJS. No entanto, você também pode desenvolvê-la utilizando apenas o JavaScript. O visual da aplicação deve ser desenvolvido utilizando HTML e CSS. As cores utilizadas no projeto são elencadas abaixo:

- Azul: `#203047`
- Amarelo: `#f5d01f`
- Cinza (background): `#e7e7e7`

- Branco: #FFFFFF
- Preto: #000000

Além disso, a fonte utilizada é a Poppins (você pode fazer o download dela por meio [deste link](#)); porém, você pode optar por outra fonte de sua preferência.

## Requisitos Funcionais

Fique atento aos requisitos funcionais da aplicação:

1. **Adicionar Tarefa:** a funcionalidade mais importante da aplicação é aquela de adicionar novas tarefas. Ao digitar o nome da tarefa no input e clicar sobre o botão “Adicionar”, a tarefa digitada deverá ser adicionada em uma lista abaixo, conforme apresentado na aplicação de exemplo.
2. **Alterar status da tarefa:** cada tarefa adicionada deverá conter um checkbox ao lado. Ao clicar nesse checkbox, a tarefa deverá ser marcada como concluída ou não concluída, a depender do estado do checkbox (*checked* ou *unchecked*).
3. **Remover tarefa:** por fim, cada tarefa adicionada também deverá conter um botão para removê-la. Ao clicar nesse botão, a respectiva tarefa deverá ser removida da listagem.



Embora esses sejam os únicos requisitos funcionais, você pode ficar à vontade para adicionar outras funcionalidades, caso queira. Por exemplo, editar as tarefas adicionadas pode ser desejável para o usuário da sua aplicação. 😊

## Aprofundando no Challenge

Caso você queira ir além nesse challenge, listamos abaixo alguns outros recursos que você adicionar à sua aplicação:

1. **Ícone de lixeira:** Um detalhe bem simples que pode dar uma “cara” mais profissional à sua aplicação seria a adição de um ícone de lixeira. Para isso, caso esteja utilizando o ReactJS, você pode utilizar a biblioteca do [react-icons](#), por exemplo.

2. Adicionar toast: Já viu aqueles toasts que são utilizados em aplicações front-end? São, basicamente, elementos que aparecem (geralmente) no canto superior direito da tela para te trazer algumas informações importantes sobre alterações que foram efetuadas na aplicação. Você poderia experimentar exibi-los para informar que uma tarefa foi adicionada ou removida do sistema. Para isso, vai a dica da utilização da biblioteca [react-toastify](#).
3. Adicionar um modal de confirmação: Ações que são consideradas críticas em um sistema, de um modo geral, irão solicitar uma confirmação antes de serem realizadas. Note que na aplicação de exemplo, ao clicar no botão de remover a tarefa, ela será removida imediatamente. Mas, e se o usuário clicar no botão acidentalmente? Para isso, é importante que haja alguma confirmação. Você pode utilizar a própria caixa de diálogo de confirmação do JavaScript (clique [nesse link](#) para ver detalhes) ou você pode criar um modal de confirmação, utilizando a biblioteca [react-modal](#), por exemplo.