

## Bootcamp: Desenvolvedor React

## Desafio

Módulo 3	React II
----------	----------

## Objetivos

Exercitar os seguintes conceitos trabalhados no Módulo:

- ✓ Criar aplicações React em TypeScript.
- ✓ Uso do Material-UI e React Router.
- ✓ Gestão de estado.
- ✓ Acesso ao Back End com autenticação.
- ✓ Criação de *hooks*.
- ✓ Otimização de aplicações.

## Enunciado

Aprimore a aplicação construída no Trabalho Prático adicionando duas funcionalidades:

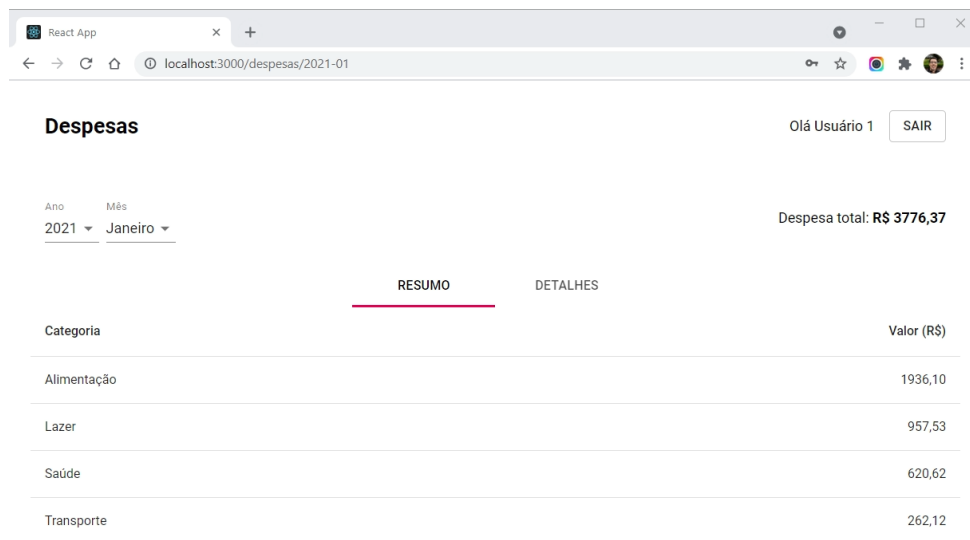
1. Acesso com autenticação. Exiba uma tela de login, se ainda não estiver autenticado, e a opção de sair, se já estiver autenticado.
2. Opção de escolher entre a visualização de todas as despesas do mês (como já foi feito) e a exibição do total de despesas por categoria, ordenadas do maior valor para o menor (nova funcionalidade).

Os dados exibidos serão novamente buscados de um Back End fornecido pelo professor, que **será postado no fórum de avisos**, com instruções de execução. Além do *endpoint* `/despesas`, o Back End agora exige autenticação, e, para isso, existem os seguintes *endpoints*:

POST <code>/sessao/criar</code>	Cria uma sessão. Deve ser enviado, no corpo da requisição, um objeto JSON com campos <b>e-mail</b> e <b>senha</b> . O Back End está preparado para aceitar o e-mail <b>usuario@email.com</b> e senha <b>1234</b> .
POST <code>/sessao/finalizar</code>	Finaliza a sessão.
GET <code>/sessao/usuario</code>	Responde com um objeto JSON com os campos <b>e-mail</b> e <b>nome</b> do usuário logado. Se não houver sessão, responde com <i>status</i> 401.

Note que as demais chamadas feitas sem autenticação também resultam em respostas com *status* de erro 401.

Na imagem a seguir, temos um exemplo da interface da aplicação após fazer login, mostrando a nova visualização de resumo das despesas, onde temos o valor total gasto por categoria.



Além de implementar as duas funcionalidades descritas, você deve cumprir mais duas exigências:

1. Crie um *hook* customizado que agrupe toda a lógica de **buscar as despesas** de um mês, de calcular a **despesa total** e de calcular a **despesa total de cada categoria** existente. Seu *hook* deve retornar um objeto contendo essas três informações.
2. Otimize o código, garantindo que os cálculos das **despesas total** e **por categoria** sejam executados apenas quando necessário.

Adicionalmente, você pode usar a API Context, ou mesmo o *hook* `useReducer` se desejar, mas não é obrigatório.

## Atividades

Os alunos deverão desempenhar as seguintes atividades:

1. Adicione autenticação na aplicação.
2. Adicione a funcionalidade de exibir despesas totais por categoria.
3. Garanta que as exigências especificadas acima estejam atendidas.
4. Com base no código fonte da sua aplicação e nos dados exibidos por ela, responda ao questionário de entrega do Trabalho Prático (a aplicação não é entregue, basta responder o questionário).

## Dicas

- Não se esqueça da opção **credentials** ao fazer a chamada ao Back End usando *fetch*.
- Fique atento ao *status* da resposta das requisições HTTP. Lembre-se que *fetch* não resulta em falha se o *status* é diferente de 200. Você precisa verificá-lo.