

# ReactJS

## Redux

### 1 Exercícios

Considere uma instituição de ensino que possui o seguinte funcionamento.

#### Departamento de Vestibular

1. Pessoas interessadas em se matricular realizam um vestibular. Para tal, elas informam seu nome e cpf.
2. Quando uma pessoa faz o vestibular, ela tem uma nota final atribuída, que varia de 0 a 10. Cada pessoa tem 70% de chance de tirar uma nota entre 6 e 10.
3. A instituição tem um departamento para armazenamento do histórico de vestibular. Cada entrada no histórico tem nome, cpf e nota final do candidato a que se refere.

#### Departamento de matrícula

1. O departamento de matrícula permite que alunos aprovados no vestibular se matriculem. Ao se matricular, um aluno informa apenas o seu cpf.
2. Alunos somente podem ser matriculados caso tenham sido aprovados no vestibular.
3. O departamento de matrícula armazena um histórico de todos os alunos que fazem matrícula. Cada entrada no histórico possui o cpf de um aluno e o seu status. Alunos que tentam se matricular e que não foram aprovados no vestibular, têm status "NM", de não matriculado. Alunos que tentam se matricular e que foram aprovados no vestibular, têm o status "M", de matriculado.

Escreva uma função de teste que oferece as seguintes opções.

1. Realizar vestibular. Esta funcionalidade captura nome e cpf de um candidato e, a seguir, simula a realização da prova do vestibular, armazenando seus dados e um valor gerado - neste momento - aleatoriamente variando no intervalo real  $[0, 10]$ . Lembre-se de seguir a distribuição de probabilidade estipulada.
2. Realizar matrícula. Esta funcionalidade captura o cpf de um candidato e, a seguir, tenta fazer a sua matrícula. A sua matrícula somente ocorre caso ele tenha sido considerado aprovado no vestibular. Aprovados são aqueles que têm nota maior ou igual a seis.
3. Visualizar meu status. Esta função captura o cpf de um candidato e exibe seu status.
4. Visualizar lista de aprovados. Esta função exibe o cpf de todos os alunos aprovados no vestibular.
0. Sair do sistema.

Faça a implementação utilizando a biblioteca Redux.

### ***Referências***

**React - A JavaScript library for building user interfaces.** 2021. Disponível em <<https://reactjs.org/>>. Acesso em agosto de 2021.

**Redux - A predictable state container for JavaScript apps. | Redux.** 2021. Disponível em <<https://redux.js.org>>. Acesso em outubro de 2021.