

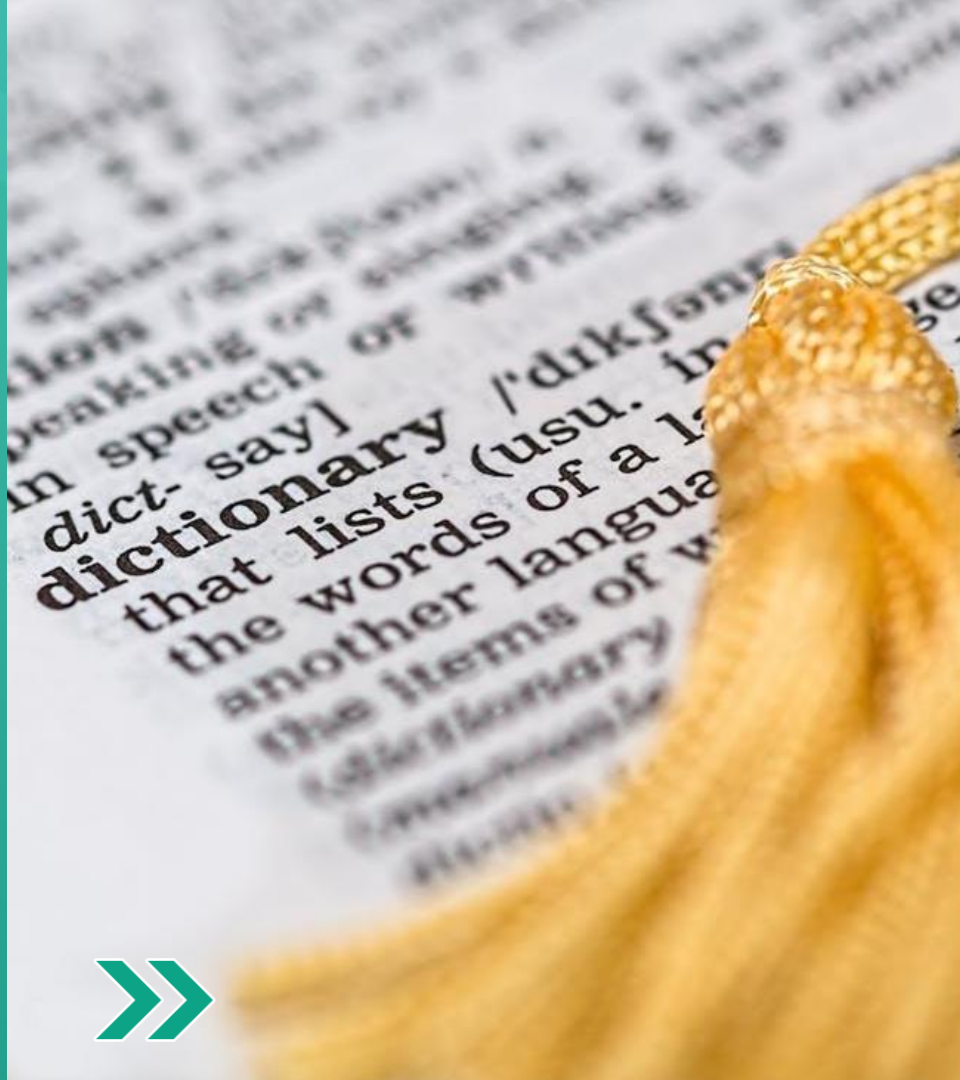


escola  
britânica de  
artes criativas  
& tecnologia

# Profissão: Engenheiro Front-End



# GLOSSÁRIO



# React Testing Library



Dica: para encontrar rapidamente a palavra que procura aperte o comando CTRL+F e digite o termo que deseja achar.

- **Conheça os tipos de testes**
- **Escreva os primeiros testes**
- **Explore testes no React**
- **Faça testes com React e Redux**
- **Conheça o Mock Service Worker**



# Conheça os tipos de testes



# Conheça os tipos de testes

## Glossário

### Jest

O nome "Jest" foi escolhido para o *framework* de testes JavaScript por razões que podem ser interpretadas de várias maneiras. Uma explicação possível é que o nome "Jest" pode ser visto como uma combinação de "test (teste)" e "jest (brincadeira)" para enfatizar sua finalidade de maneira divertida e eficiente. Entretanto, o nome também pode ter sido escolhido com base na preferência e na criatividade dos desenvolvedores envolvidos.



# Escreva os primeiros testes



# Escreva os primeiros testes

## Glossário

### **calculadora.spec.js**

Arquivo de teste que segue uma convenção de nomenclatura comum em muitos frameworks de testes JavaScript. A extensão ".spec.js" indica que esse arquivo contém especificações (ou seja, testes) para o código da calculadora. É usado para testar o comportamento e a precisão das funcionalidades da calculadora, garantindo que ela realize corretamente as operações matemáticas esperadas e trate possíveis cenários de borda.

### **Expect**

Função utilizada para criar expectativas (*assertions*) em testes. Ela é a base para escrever afirmações que verificam se o comportamento esperado do código está de acordo com o resultado real obtido durante a execução do teste.



# Escreva os primeiros testes

## Glossário

### **Plugin Jest**

Plugin do Jest que fornece uma maneira de escrever testes de unidade para o código do seu projeto. Ele ajuda a identificar e corrigir erros no código antes que ele seja liberado para produção.

### **Suíte de teste (conjunto de testes)**

É um agrupamento lógico de casos de teste relacionados que têm um objetivo em comum. Ela é uma coleção de testes que verifica o comportamento de uma determinada funcionalidade, componente ou módulo do *software*. As suítes de teste são usadas para organizar e estruturar os casos de teste em grupos coesos, o que torna a execução e a manutenção dos testes mais eficientes.





# Explore testes no React



# Explore testes no React

## Glossário

### ● Coverage (Cobertura de testes)

Métrica usada para avaliar a extensão em que o código-fonte de um *software* é exercitado pelos testes automatizados.

### ● FireEvent

Funcionalidade da Testing Library que permite simular interações do usuário com os elementos renderizados durante os testes. Essas interações incluem cliques, digitação de texto, envio de formulários, mudança de valores de *input* e outras ações que um usuário real faria na *interface*.



# Explore testes no React

## Glossário

### **jest-dom**

Biblioteca de expectativas personalizadas (matchers) desenvolvida para uso com o Jest e outras bibliotecas de testes JavaScript. Ela é parte do ecossistema da Testing Library, que visa facilitar a escrita de testes mais legíveis e centrados no comportamento do usuário para aplicações web. Ela adiciona vários matchers personalizados ao Jest, permitindo que você escreva testes que verificam interações com a interface do usuário de maneira mais expressiva e semelhante à forma como um usuário real interagiria com o aplicativo.

### **Render**

Método da Testing Library que é uma função utilizada para renderizar componentes React em um ambiente de teste. Quando você está escrevendo testes para componentes React, o método "render" é responsável por criar uma representação virtual do componente em memória e montá-lo em um ambiente de teste simulado. Isso permite que você interaja e verifique o comportamento do componente como se ele estivesse sendo exibido em um navegador real, mas tudo acontece dentro do contexto do ambiente de teste.



# Explore testes no React

## Glossário

### **Screen**

Funcionalidade do React Testing Library que é um objeto que contém um conjunto de funções úteis para interagir com componentes renderizados e selecionar elementos na árvore de componentes do React durante os testes. É uma abstração fornecida pela Testing Library para facilitar a seleção de elementos e a realização de verificações nos testes. Ela é uma maneira conveniente de acessar os elementos renderizados sem a necessidade de importar funções específicas de diferentes módulos da Testing Library.



# Faça testes com React e Redux



# Faça testes com React e Redux

## Glossário

### **getByText**

Função da Testing Library usada para selecionar elementos com base em seu texto visível durante os testes automatizados. Ela procura e seleciona o primeiro elemento encontrado na árvore de componentes React que contém o texto especificado. Se nenhum elemento contendo o texto for encontrado, ela lançará um erro, indicando que o elemento não foi localizado na tela durante o teste.

### **preloadedState**

Argumento opcional que pode ser passado para a função `createStore`, responsável por criar a "store" centralizada que contém o estado da aplicação. A "store" é uma parte essencial do padrão de gerenciamento de estado do Redux. É utilizado para fornecer um estado inicial à "store" quando ela é criada. Em outras palavras, ele define o estado inicial da aplicação antes de qualquer ação ser despachada ou redutor ser aplicado.



# Conheça o Mock Service Worker



# Conheça o Mock Service Worker

Glossário

## Mock Service Worker (MSW)

Biblioteca de desenvolvimento que permite criar mocks (simulações) de serviços de rede em testes automatizados ou ambientes de desenvolvimento local.

Acesse o site <https://mswjs.io/> para obter mais informações.





# Bons estudos!

