

# Webdev - Módulo 3

## Roteiro de aula

---

### Aula 9 - Hard: Rebobinando



#### Tópicos da aula:

- Revisão de React
- Deploy (render.com)
- Ajustes finais



#### Objetivos da aula:

1. Relembrar react-router-dom
2. Relembrar Contexto
3. Relembrar props
4. Relembrar virtual dom
5. Realizar o deploy da aplicação react (render.com)
6. Finalizar possíveis ajustes no projeto final



#### Atividades da Aula

- Livecoding 1: Revisão React, react-router-dom, Contexto, Props, Virtual DOM
- Materiais de apoio
  - ◆ [React router DOM](#)
  - ◆ [Props](#)
- Livecoding 2: Deploy, render.com
  - ◆ [Passo a passo do live coding](#)



#### Momento Aprendizagem em Pares

Esse momento é dedicado para vocês desenvolverem suas demandas e entregas para o curso.

- Utilize esse tempo da maneira que preferir, mas atente-se às aulas que você deve realizar as entregas.

- ◆ **Dica:** Nas Propostas dos projetos, vocês encontram uma sugestão de organização para a realização das atividades.
- ◆ **Lembre-se:** O momento de Aprendizagem em Pares é justamente para fazer trocas e aprender em comunidade, aproveite seus colegas e se desenvolvam juntos!

→ **Entregas:**

- ◆ Projeto individual: aula 5 - HARD
- ◆ Projeto em grupo: aula 10 - HARD
- ◆ Apresentação do projeto: aula 10 - HARD



## Revisão da aula

- O **React Router DOM** permite que você implemente o roteamento dinâmico em um aplicativo da web. Ao contrário da arquitetura de roteamento tradicional na qual o roteamento é tratado em uma configuração fora de um aplicativo em execução, o React Router DOM **facilita o roteamento baseado em componentes de acordo com as necessidades do aplicativo e da plataforma.**
- No React, o termo **props** significa propriedades, que se refere às propriedades de um objeto. Isso ocorre porque todos os elementos criados no React são objetos JavaScript. Como resultado, a passagem de dados para um componente é feita criando propriedades e valores para o objeto.
- O React cria uma **árvore de objetos** personalizados representando uma parte do DOM. Por exemplo, em vez de criar um elemento DIV real contendo um elemento UL, ele cria um objeto React.div que contém um objeto React.ul. Ele pode manipular esses objetos muito rapidamente sem realmente tocar no DOM real ou passar pela API do DOM. Então, quando renderiza um componente, ele usa esse DOM virtual para descobrir o que precisa fazer com o DOM real para fazer com que as duas árvores coincidam.



## Para ajudar



### Links interessantes:

- [Deploying hyper63 on Render.com](#)
- [Virtual DOM e Objetos Internos – React \(reactjs.org\)](#)
- [React Props \(w3schools.com\)](#)

- [Context – React \(reactjs.org\)](https://reactjs.org/docs/context.html)