

# WebDev - Módulo 3

## Roteiro de aula

---

### Aula 3 - Hard: Beabá do JavaScript



#### Tópicos da aula:

- Destructuring
- CommonJS vs módulos ES6;
- Json Server
- Render.com (Hospedagem)



#### Objetivos da aula:

1. Relembrar conceitos de protocolos da web (HTTP)
2. Revisar conceitos de requisições
3. Reconhecer a vantagem de se empregar Fetch, no lugar de XMLHttpRequest
4. Empregar Fetch para realizar requisições web



#### Atividades da Aula

##### → Atividade 1: Desestruturando tudo

- ◆ *Em grupos de 3 a 4 pessoas vocês irão se reunir (apenas para discussão e apoio) e fazer individualmente a desestruturação de algumas variáveis que estão armazenadas como arrays e objetos, no arquivo disponibilizado no roteiro da aula. Se atentem aos requisitos do exercício. No arquivo também temos alguns comentários do que deve ser feito. De modo geral, vocês deverão:*

- Acessar o arquivo
- Acessar todos os itens do array alunos
- Acessar todos os itens do array disciplinas, exceto o terceiro
- Acessar todos os itens dos dois objetos disponíveis
- Se atentar caso haja necessidade de renomear algo

- ◆ [Arquivo com os arrays e objetos para a atividade](#)



#### 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 **destructuring (ou desestruturação)** é um recurso interessante que veio com o ES6. A expressão torna possível “desembalar” valores dos arrays ou propriedades de objetos em variáveis distintas. Tá, mas o que isso quer dizer? Imagine que você quer fazer um sanduíche. Você vai tirar tudo que existe na geladeira para fazer seu sanduíche? Não, você retira apenas os itens que gostaria de usar no seu sanduíche. A desestruturação segue a mesma lógica. **Podemos estar trabalhando com um array ou objeto, mas naquele momento só precisamos de alguns itens contidos neles. A desestruturação facilita a extração apenas do necessário.**
- Antes dos módulos EcmaScript existirem, os desenvolvedores utilizavam o CommonJS (o “require”) que foi mantido pela organização do Mozilla para padronizar o ecossistema javascript fora dos browsers. Após esse período, o CommonJS deixou de ser mantido. Os responsáveis pelo Node.js e npm assumiram as especificações dos módulos, possibilitando a criação de um grande ecossistema Javascript. Como resultado, o padrão de módulos **CommonJS + Node/npm** se espalhou por todo o ecossistema javascript, tanto no servidor, quanto no cliente, e está crescendo cada vez mais. O problema é que esse padrão não possui um comitê oficial ativo e já se tornou obsoleto. Com os módulos EcmaScript, teremos uma nova especificação mantida pelo comitê ativo do TC39 que vai tirar proveito das últimas funcionalidades do EcmaScript

para solucionar os problemas mais comuns da modularização no mundo Javascript.

- No mesmo sentido, o **JSON Server** não é diferente. O servidor JSON é um módulo de nó que você pode usar para criar serviços REST JSON de demonstração em um curto espaço de minutos. Tudo o que precisamos fazer é ter um arquivo JSON como dados de amostra. Vamos aprender mais sobre como instalar e configurar um servidor JSON em nosso sistema e ao mesmo tempo tentaremos criar uma API REST. Vamos aprender mais sobre isso.
- O **Render** é um novo provedor de hospedagem web e de aplicativos, descrito pelo seu CEO como "o melhor dos dois mundos", comparando fornecedores em nuvem de larga escala, como a AWS, e provedores PaaS (Plataforma como Serviço), como a Heroku. O Render **ainda está construindo sua suíte de serviços, mas já possui os requisitos fundamentais para rodar uma aplicação básica**. Através do painel de controle, um desenvolvedor pode decidir que tipo de serviço deseja criar, sendo ele público ou privado. O painel está conectado a um repositório do GitHub, que fará as publicações das atualizações para o serviço toda vez que uma alteração de código for realizada no repositório. Não há a necessidade de criar pipelines apartados para build e deployment.



## Para ajudar



### Links interessantes:

- [Criando Json Server E Hospedando - DEV Community](#) 🏠🏠
- [Documentação da Hospedagem | Render](#)
- [JSON Server \(json-server\) | DigitalOcean](#)