

Debugando React

Labenu_



O que vamos ver hoje?

- O que é Debug?
- Processo de Debug
- Últimos recursos



Motivação

Labenu_



Bugs

- Programar é instruir o computador a executar tarefas
- Ele segue **exatamente** as instruções que são dadas
- Quando damos **instruções erradas**, o computador vai segui-las cegamente. E é aí que surge o bug...
- Não existe código sem bugs, existe código com bugs que **ainda** não foram encontrados



O que é Debug?

- Processo de encontrar e descobrir os motivos dos erros para poder corrigi-lo
- Existem várias formas e ferramentas para isso
- É uma habilidade **muito importante** para todo dev
- **Faz parte do dia-a-dia!**



Processo de Debug

Labenu_



O que fazer quando dá um erro? 🙄

- Passo 0: **não se desesperar!**
- Erros são comuns e fazem parte da vida de todo dev
- Podemos encarar bugs como um **jogo** ou um **mistério** a ser resolvido
- Com calma e clareza do que está acontecendo, costuma ser mais tranquilo de achar o problema



O que fazer quando dá um erro? 🙄

- A partir daí, vamos **quebrar o problema** em 4 partes:
 1. Clareando o problema
 2. Examinando suas suposições
 3. Analisando o código passo a passo
 4. Testar possíveis soluções



1. Clareando o Problema



- A primeira coisa a fazer é identificar claramente o problema que você está tentando resolver
- Comece fazendo duas perguntas:
 - O que você esperava que seu código fizesse?
 - O que aconteceu no lugar?



1. Clareando o Problema

- Com a pergunta "**o que aconteceu no lugar?**", vamos invariavelmente cair em duas situações:
 - O código não executa → **exceções**
 - O código executa mas **não funciona** do jeito que eu estava esperando



Exceções

- Uma **exceção** é um **erro que impede o código de continuar executando**
- É quando a tela "explode"
- Bom sinal, pois temos uma sinalização clara **do que aconteceu e onde**
- Se você não conhece a exceção, **pesquise!**



2. Examinando suas suposições

- Quando você escreveu o código, você esperava que ele funcionasse (risos)
- Para isso, você fez uma série de **suposições** (Ex: eu supus que uma variável seria uma String)
- Especialmente, **pense nas coisas mais óbvias**, que você normalmente não para pra pensar



2. Examinando suas suposições

- Não existe nenhum **erro de digitação** no código?
- Está usando a função, estado, prop, etc. correta(o)?
- Você mudou alguma coisa que achou que não era relacionada ao problema?
- Você esperada que uma variável tivesse um valor que ela não tem?



3. Analisando o código passo a passo



- Agora que você já tem uma visão melhor sobre o problema e já descartou algumas hipóteses, é hora de procurar mais a fundo **aonde** está o erro
- Comece de onde você sabe ou imagina que o erro aconteceu e vá "**voltando**" no código até achá-lo
- Nesse passo, você tem um grande aliado: o **console**!



Console

- Mostra mensagens relevantes do que acontece → logs
- A principal função é prover informações para debug
- Níveis de log:
 - **Info/logs**: sem uma relevância específica
 - **Warnings**: coisas que podem originar problemas
 - **Erros**: problemas que não podem ser ignorados



Console



Elements Network Performance Console Sources Application Memory >> 3

top Filter All levels

Navigated to <http://localhost:3000/>

DevTools failed to load SourceMap: Could not load content for <chrome-extension://gighmmpiobklfepjocnamgkbbiglidom/include.preload.js.map>: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME

DevTools failed to load SourceMap: Could not load content for <chrome-extension://hdokiejnpimakedhajhdlcegeplioahd/sourcemaps/onloadwff.js.map>: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME

[HMR] Waiting for update signal from WDS... [log.js:24](#)

Uncaught Error: Cannot find module './componetns/Post' [App.css?4433:45](#)

- at webpackMissingModule ([App.css?4433:45](#))
- at Module../src/App.js ([App.css?4433:45](#))
- at __webpack_require__ ([bootstrap:784](#))
- at fn ([bootstrap:150](#))
- at Module../src/index.js ([index.css?f3f6:45](#))
- at __webpack_require__ ([bootstrap:784](#))
- at fn ([bootstrap:150](#))
- at Object.1 (serviceWorker.js:141)
- at __webpack_require__ ([bootstrap:784](#))
- at checkDeferredModules ([bootstrap:45](#))
- at Array.webpackJsonpCallback [as push] ([bootstrap:32](#))
- at [main.chunk.js:1](#)

DevTools failed to load SourceMap: Could not load content for <chrome-extension://gighmmpiobklfepjocnamgkbbiglidom/include.postload.js.map>: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME

./src/App.js [index.js:1](#)

Line 31:25: 'post' is not defined no-undef

Search for the keywords to learn more about each error.

./src/App.js [index.js:1](#)

Module not found: Can't resolve './componetns/Post' in '/Users/pedroarvas/future4/materiais/manipulacao-arrays/src'

DevTools failed to load SourceMap: Could not load content for <chrome-extension://gighmmpiobklfepjocnamgkbbiglidom/include.preload.js.map>: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME

DevTools failed to load SourceMap: Could not load content for <chrome-extension://gighmmpiobklfepjocnamgkbbiglidom/include.postload.js.map>: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME



Usando o Console



- Podemos usar o **console.log** para imprimir uma informação que queremos no meio do código
- **Muito útil** para debugar e analisar o seu código
- Podemos usá-lo para entender melhor o código
 - Por onde ele está passando
 - Quais dados as variáveis têm



4. Testar possíveis soluções

- Quando você estiver investigando, **não tenha medo** de testar soluções!
- Modifique o código e **veja se algo acontece**, então tente **entender** as novas condições
- Mas atenção: **faça testes conscientes!** Tente entender o que está acontecendo e fazer testes que te dão mais informações



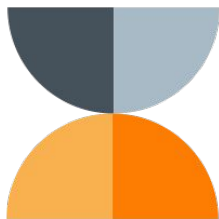
4. Testar possíveis soluções

- É muito importante **entender** o que está acontecendo e quais foram as **soluções encontradas**
- Arrumar bugs e encontrar soluções é uma das **melhores formas de aprender** mais e evitar que os mesmos erros ocorram no futuro



Pausa para relaxar 🤔

5 min



- Bugs? Mantenha a calma e debugue!
 1. Entenda o problema
 2. Examine suposições
 3. Analise o código passo a passo
 4. Teste possíveis soluções
- Use o `console.log` ele é seu amiguinho





Exercício 1

- Temos uma aplicação que:
 - Cria posts
 - Dá like no post e troca o texto do botão
 - Deleta Post
- Mas ela não está funcionando!
- Identifique os bugs e os conserte

Hoje é dia de aula daora

Hoje é dia de aula daoraaa!

"Frase muito profunda" - Lispector, Clarice

Bom dia família! Vim desejar uma boa semana



Pausa para relaxar 🥱

10 min



Mais Recursos

Labenu_



Já tentei tudo, e agora? 🤯

- Depois de entender o problema e tentar as soluções mais comuns, o que fazer?
- Às vezes, o bug **não é óbvio** e é muito difícil encontrá-lo
- Ainda assim, sempre é possível resolver!



Remover Código

- Uma saída válida em casos extremos é ir **removendo partes do código** até que o erro pare de acontecer
- O ideal é ir **comentando** pequenas partes do código e observar o comportamento
- Quando o erro para, sabemos que a última parte removida é a causadora do erro!



Apagar e reinstalar dependências

- É possível que por algum motivo, os arquivos das **bibliotecas** que usamos (**node_modules**) tenham sido instalados com algum problema
- Frequentemente isso pode ser resolvido apagando a pasta **node_modules** e reinstalando as dependências com `npm install`



Use o Google

- É muito difícil cairmos em bugs inéditos
- Quando caímos em um bug, **provavelmente existe algo sobre esse bug no Google**
- Para sabermos isso, basta **copiar a mensagem de erro** e colar no buscador
- Acreditem: **todos** os devs usam essa técnica...



Junior dev watching senior dev copying the error message, pasting on Google, and opening the first StackOverflow link



Buscar Ajuda 🧑🏻 🤝 🧑🏻

- Quando já tentou de tudo, não exite em pedir ajuda para um colega
- Tenha certeza de que coletou a maior quantidade de informações possível sobre o problema
- Explique o que você sabe detalhadamente → muitas vezes você entenderá a solução só de explicar!



E se ainda não funcionar? 🖥️🔨



Como melhorar?

Labenu_



Conheça as Ferramentas

- Existem muitas ferramentas úteis para debug
- Falamos superficialmente de algumas, mas elas tem muitas **funcionalidades extras**
- **Estude** e **pesquise** sobre elas! Isso pode facilitar muito a sua vida :)



Pratique

- **Não tenha medo de bugs**
- Faça um esforço para resolver problemas e **entender** as soluções
- Criar bugs **faz parte** de programar: aceite isso!
- **Ajude os colegas** → mais experiência traz mais agilidade



Dúvidas? 🧐

Labenu_





Obrigado(a)!