# **PLANO DE AULA**

Aula 17 | Tempo estimado: 1 hora e 30 minutos | Web Starter

Tipo da atividade: Offline

Ferramenta(s): Computador, Git, nodeJS e VS Code

Conteúdos

* React, servidores e portas;

Objetivos

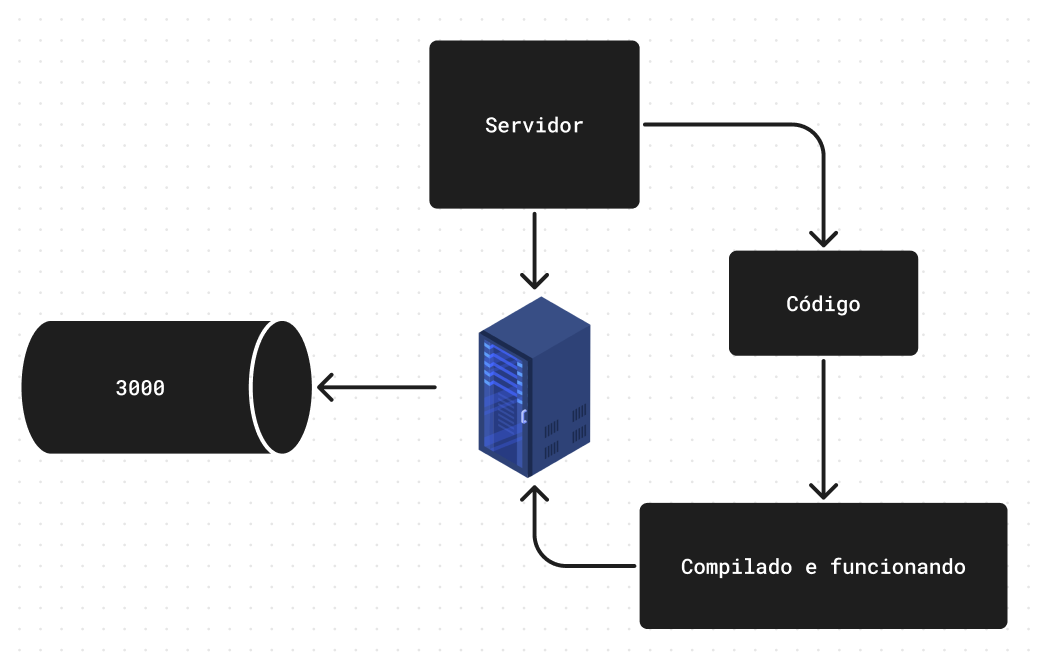
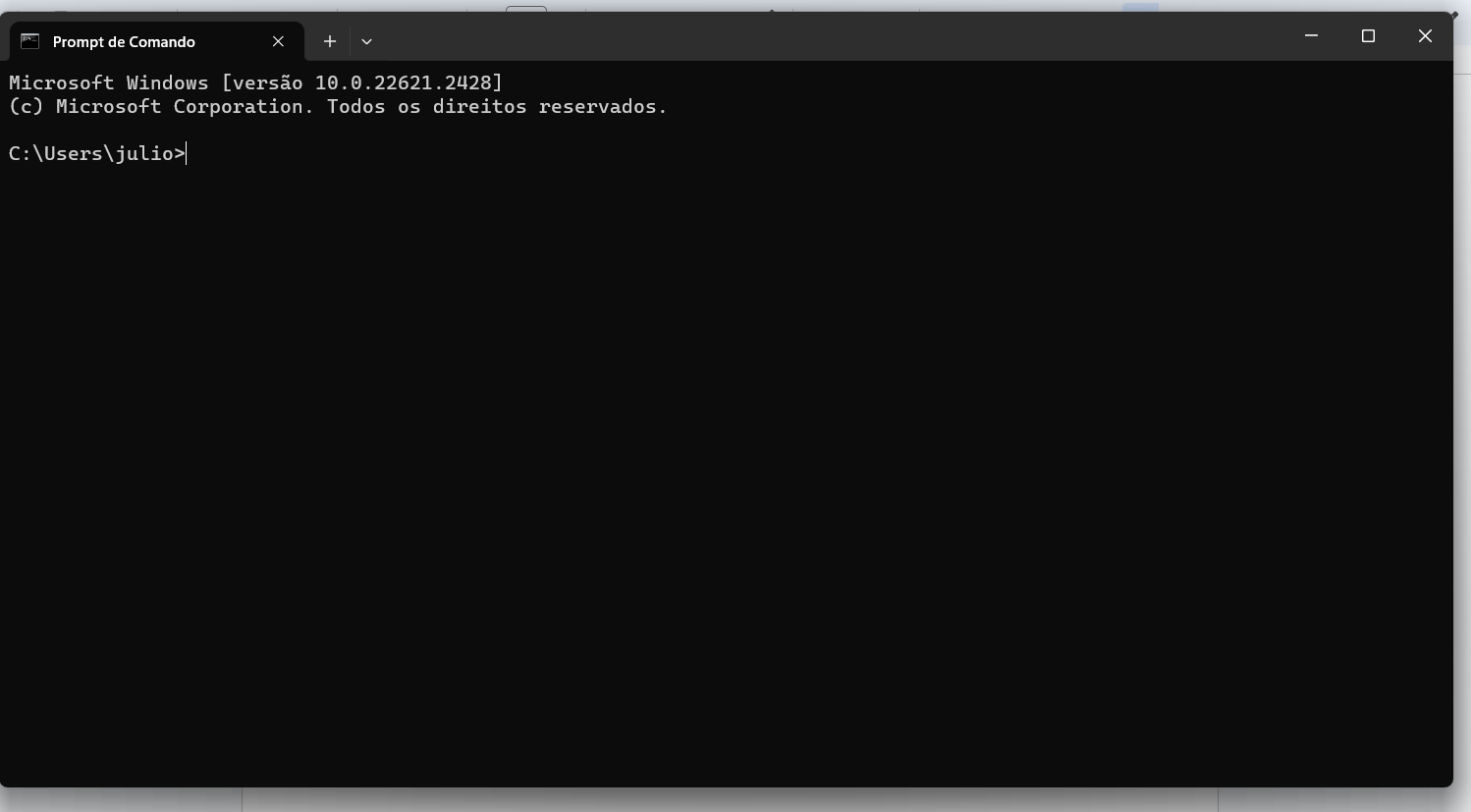
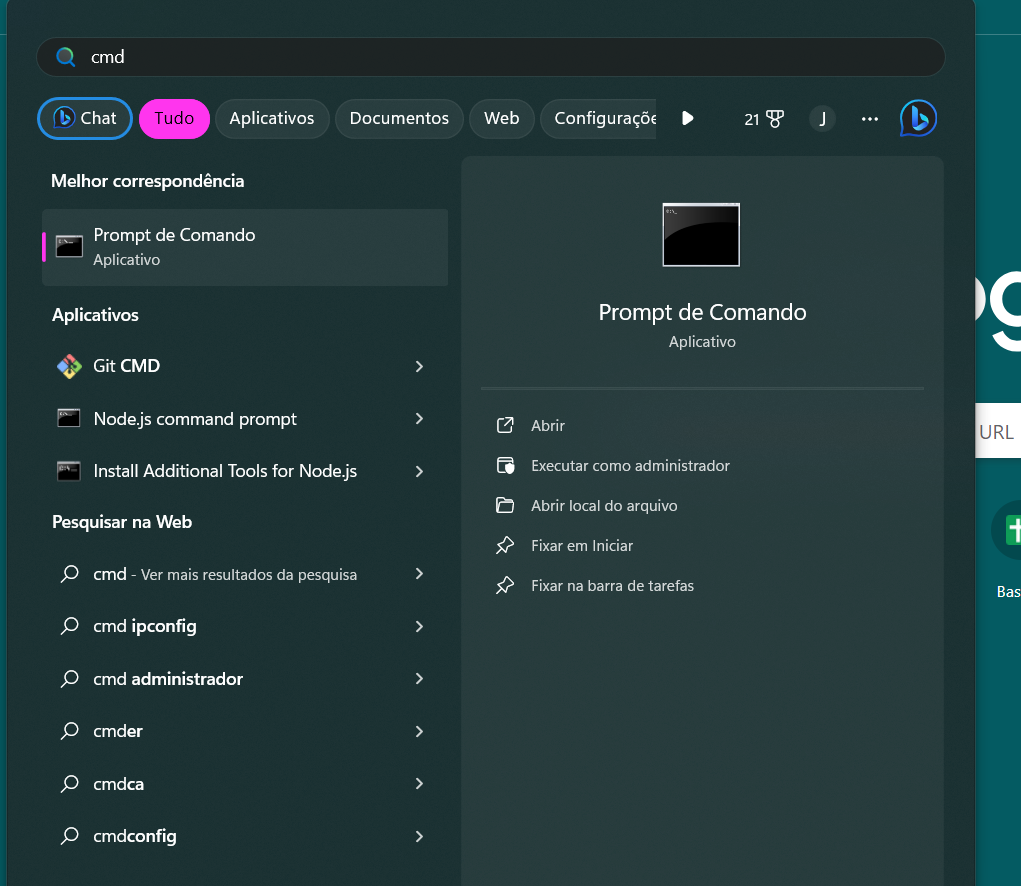
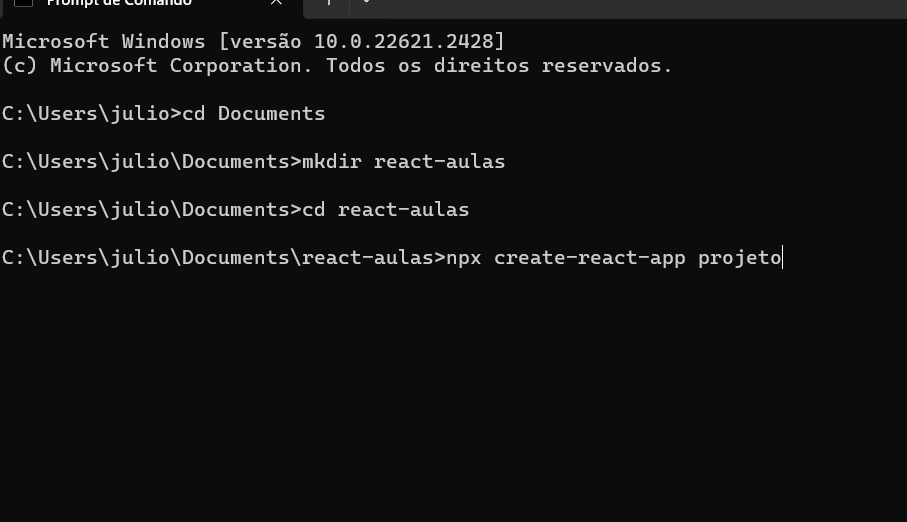
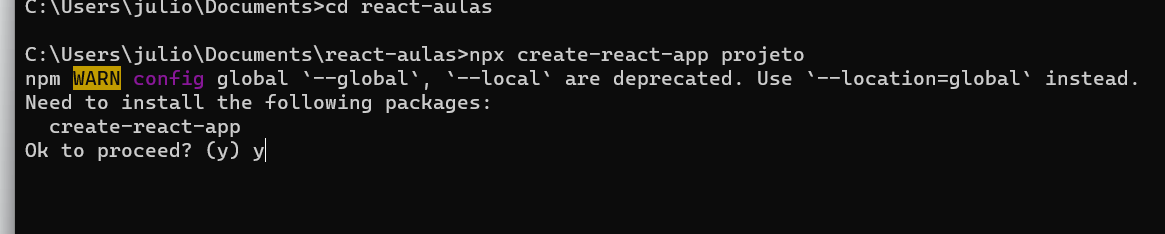
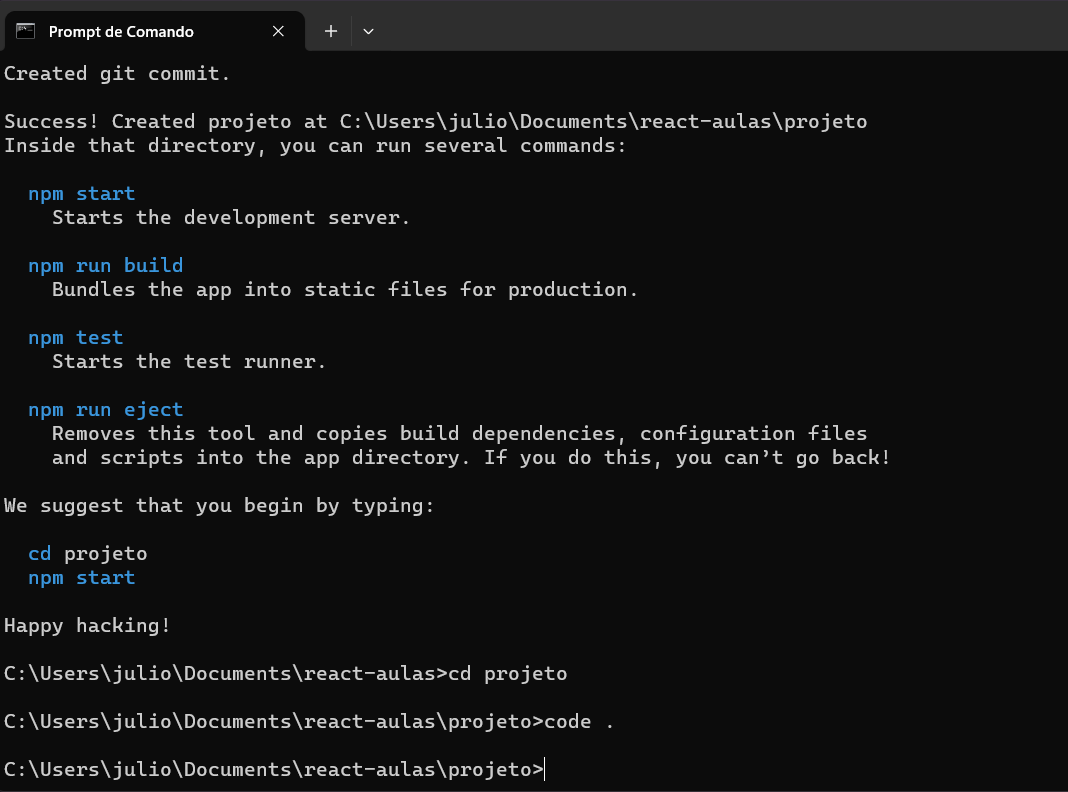
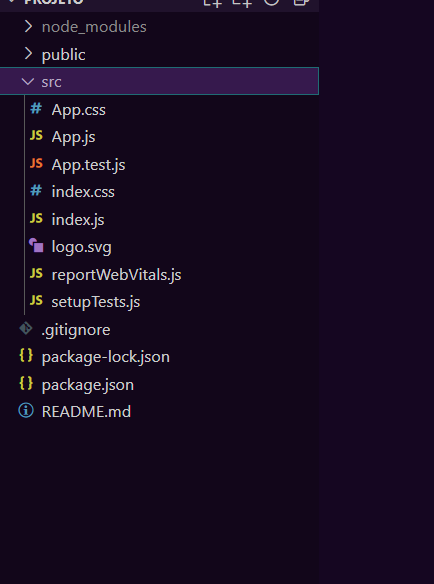
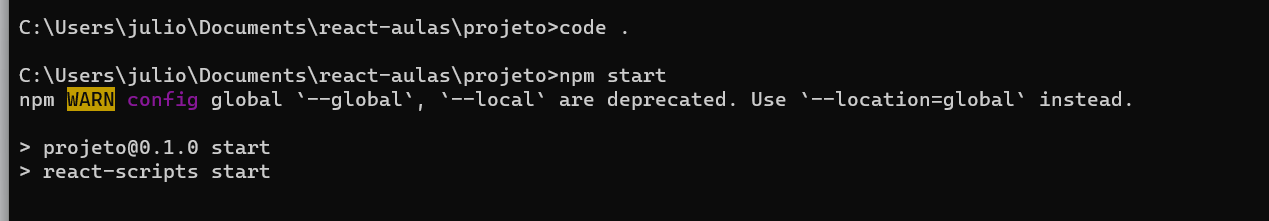
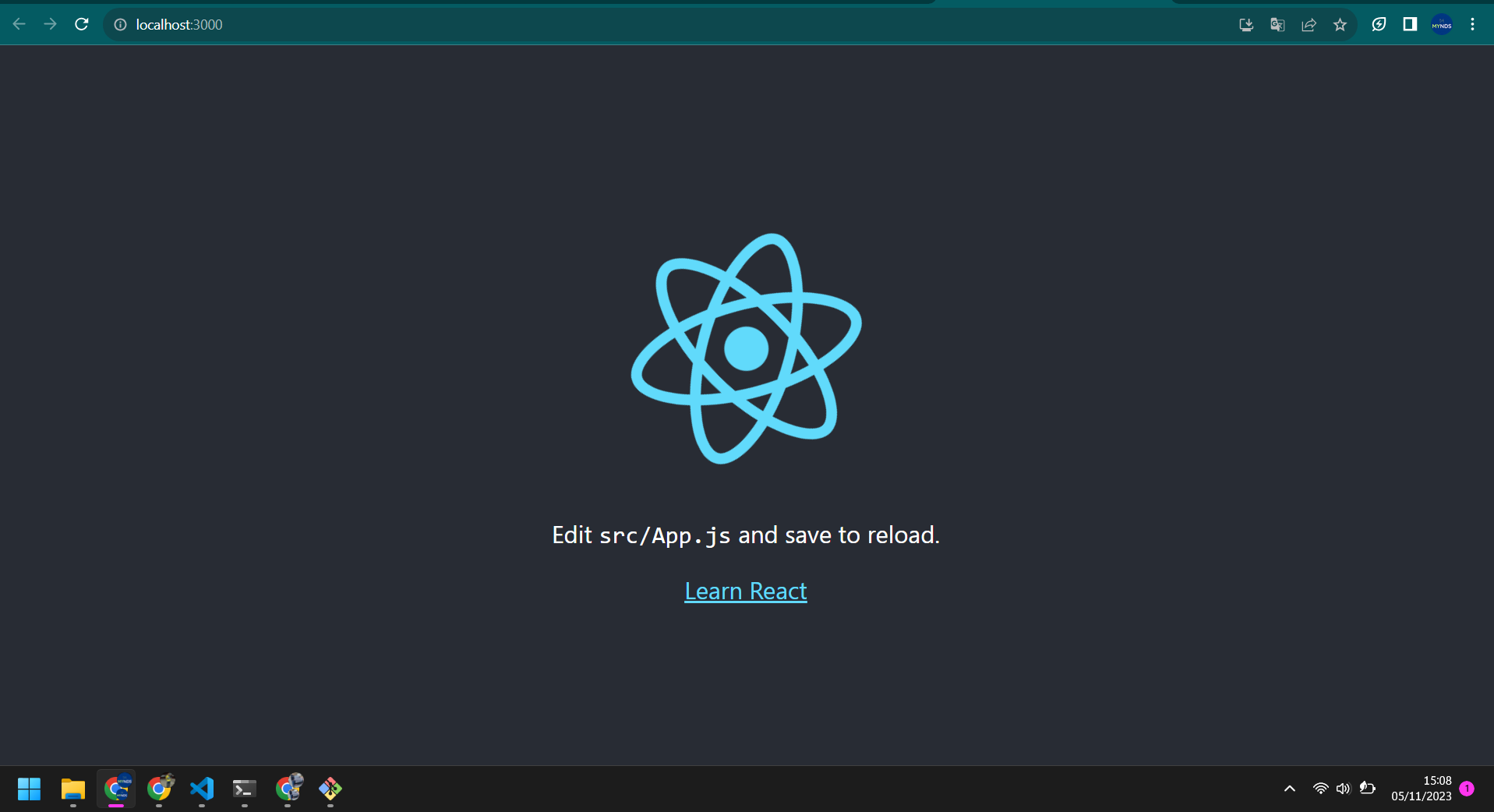
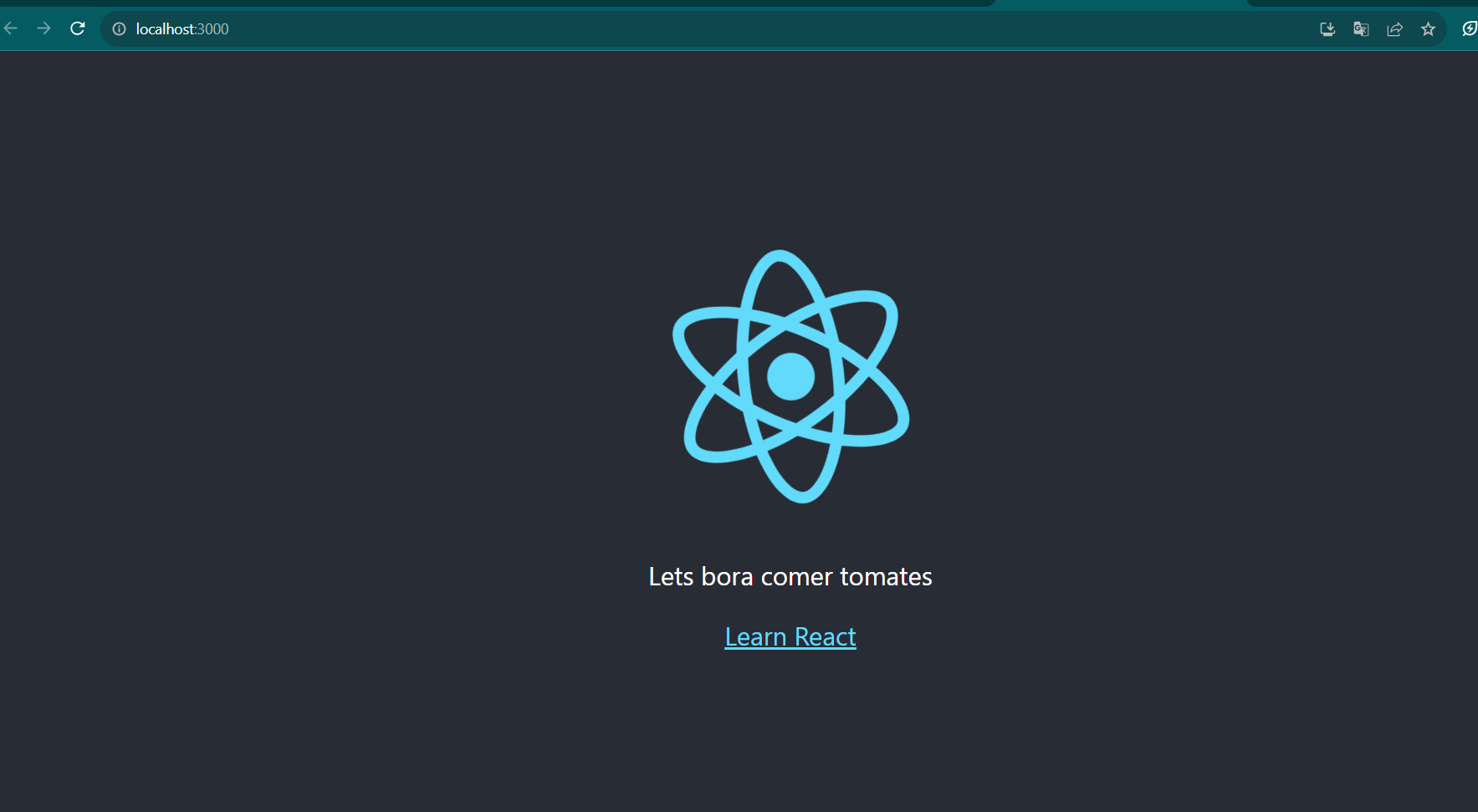
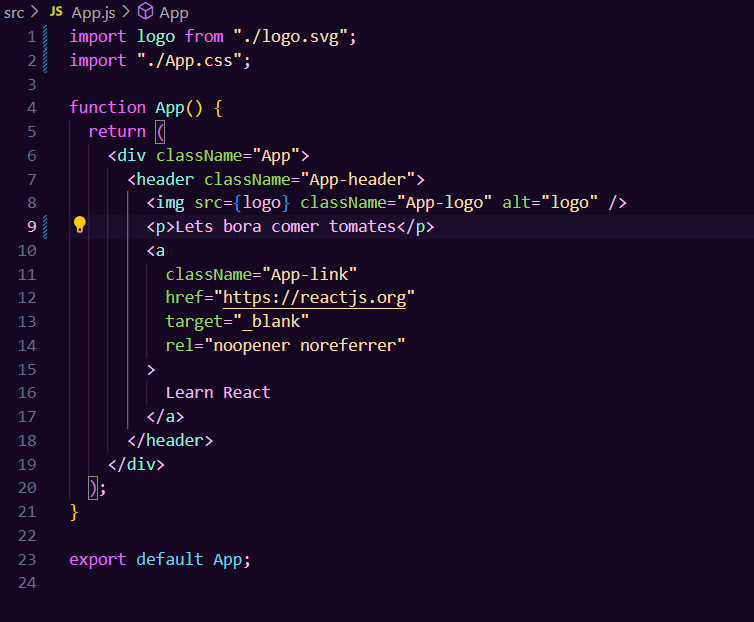
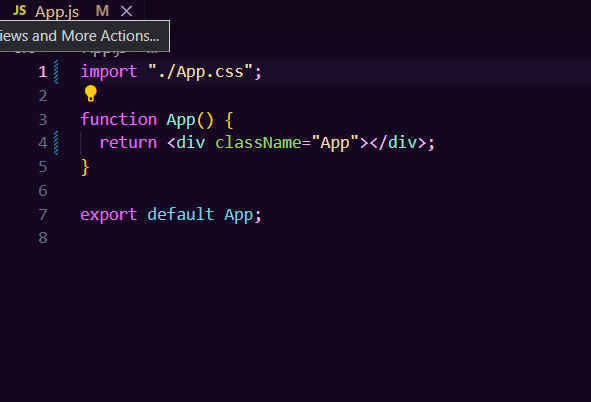
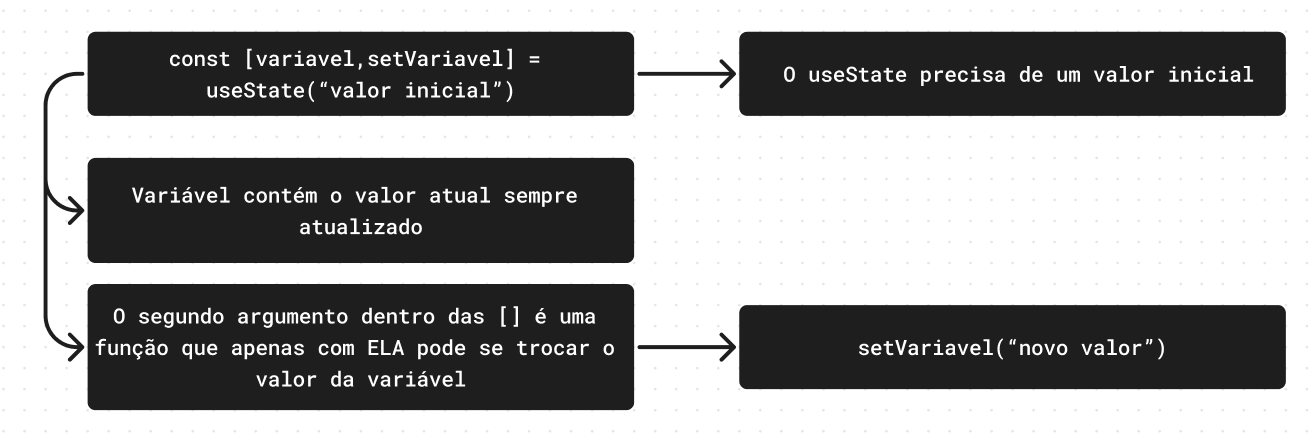
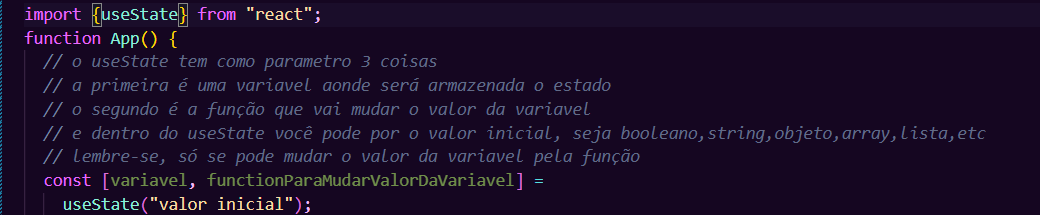
* Ensinar sobre React;
* Aprender o que é um servidor;
* Configurar um projeto.

Estratégias e atividades

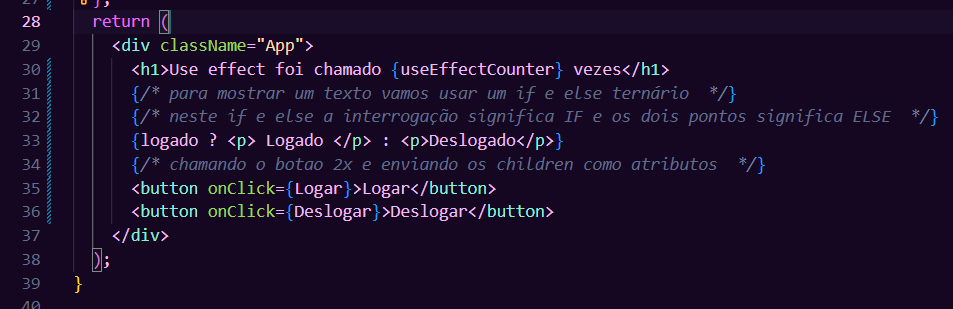
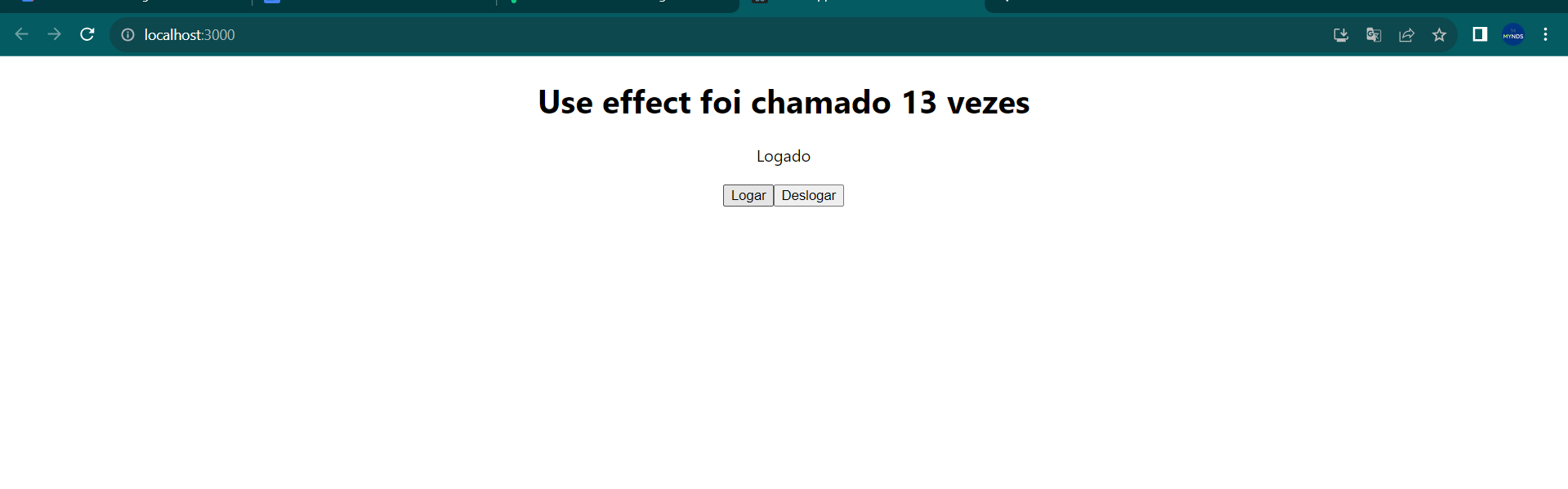
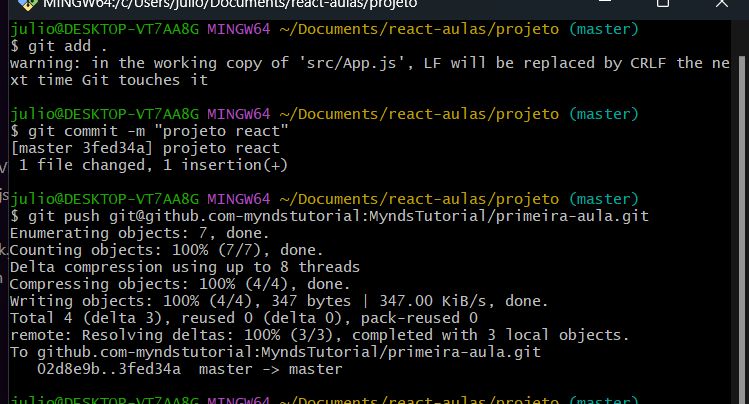
* Criar um App simples usando React.

1. **Hoje vamos conhecer o que é React.**

O React foi criado pelo Facebook para solucionar um problema enorme que eles estavam tendo: Atualizar as notificações em tempo real. Fazendo com que o projeto renderize modificações na tela automaticamente sempre que o usuário fizer algo.

1. O React precisa de algo chamado *servidor* para poder funcionar. O que um servidor faz é pegar um código e executá-lo, transformando ele em binário e jogando em uma porta do computador. A padrão do React é 3000:  
     
   
2. Precisamos instalar o nodeJS para poder rodar aplicativos React no computador. O nodeJS é um *build runner*, ou seja, um compilador JavaScript para o computador rodar. Mas não se engane, ele é escrito em C++, ou seja, várias linguagens interagem entre si programando o sistema: <https://nodejs.org/en/download>
3. Vamos começar então criando um projeto React. Clique no ícone do Windows no teclado e digite CMD:  
   
4. Vamos entrar dentro de *documentos* usando o comando *cd (change directory)* e o comando *mkdir* para criar uma pasta chamada *react-aulas*, e ao final dar o comando *npx create-react-app nome-do-projeto*:  
     
   
5. Agora, com o projeto criado, vamos entrar nele (vou fazer pelo CMD mas pode ser como você preferir):  
   
6. Repare que o arquivo HTML está no *public*, porém o nosso App está apenas no *src*. Então, é neles que iremos mexer:  
   
7. Para iniciar o projeto, nós precisamos no CMD, dentro da pasta do projeto, digitar: *npm start*  
   
8. Repare que abriu um projeto na porta 3000:  
   
9. Este projeto está em App.js. Repare que se eu editar o código ele renderiza automaticamente na tela:   
   
10. Vamos limpar o App.js:  
    
11. O React nada mais é que um conjunto de funções de JavaScript, que podem retornar códigos HTML, fazendo com que a tela se atualize mudando o retorno das funções.
12. *useState* é a ferramenta do React para manter o conteúdo na tela sem renderizar o valor. Imagine que, conforme você clica, você quer mudar a função mas não quer mudar o conteúdo dela. Apenas o valor, para utilizar o *useState* precisamos primeiro de tudo importar ele.  
    
13. Dica do *useState*, coloque o nome da função com setNomeDaVariavel:   
    
14. Agora vamos aprender sobre o *useEffect*. Ele tem 2 argumentos, o primeiro é uma função *()=>{ fazAlgumaCoisa() }*, e o segundo é um array com todas as variáveis que vão ser chamadas [].
15. Mas o que o *useEffect* faz? Ele é um loop de código, uma função que pode ser chamada sempre que o valor da variável que está dentro dos parentes for mudado. O padrão é sempre renderizar quando o aplicativo é carregado, ele só terá mais chamadas se a variável dos parênteses mudar:



1. Vamos agora usar as variáveis na tela. O código HTML pode chamar coisas JavaScript quando se coloca entre chaves:  
   
2. E o resultado é um App que loga e desloga:  
   
3. Para finalizar, vamos subir o código no Github.  
   

Recursos

* Git Chrome e VS Code.

Observação

* Sempre que você quiser testar algum exemplo em JavaScript, você pode digitar um código direto no navegador como por exemplo:  
  

Tarefas

* Responder as perguntas. O que é React e quais suas vantagens. Montar uma apresentação e enviar para o e-mail [myndstechschool@gmail.com](mailto:myndstechschool@gmail.com) ou via Discord do seu professor.