Aprenda com quem faz

React III

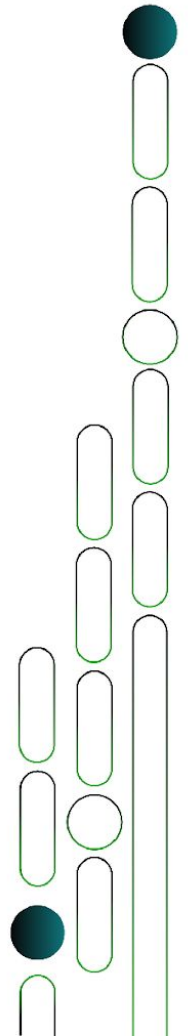
Capítulo 1. Introdução

Aula 1.1. Plano de ensino, expectativas e objetivos

Prof. Rodrigo Borba

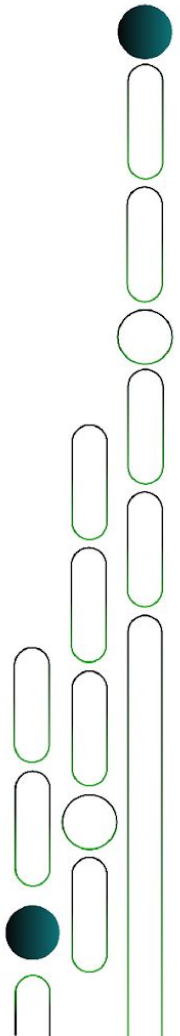
Nesta aula

- Introdução:
  - Apresentação.
  - Expectativas e objetivos.
  - Plano de ensino.



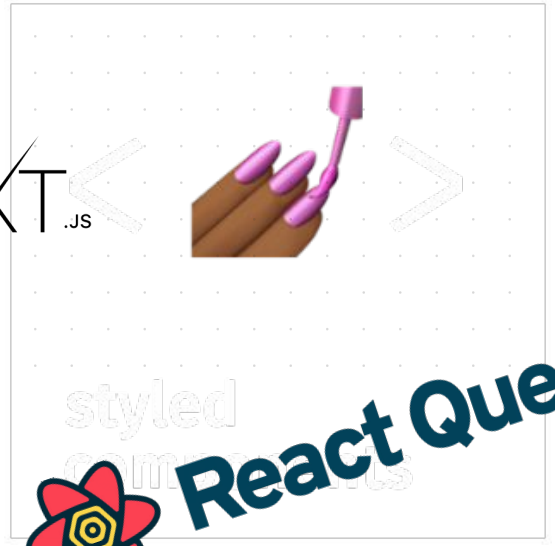
Apresentação

- Rodrigo Borba.
- Desenvolvedor Frontend Web.
- Blockchain ❤️.



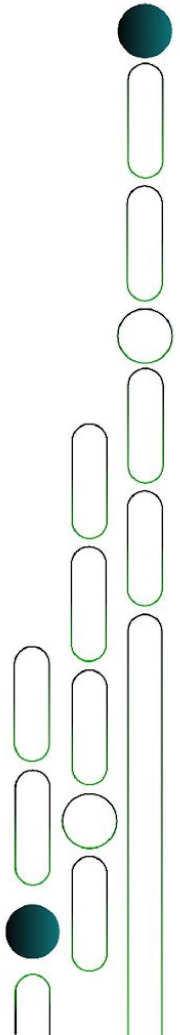
Plano de ensino

- Styled componentes.
- Bibliotecas de Data Fetching.
- Estratégias de renderização (SPA, SSR, SSG).
- Next.Js.
- Testando aplicações React.
- Deploy de aplicações.



Conclusão

- ☑️ Apresentação.
- ☑️ Expectativas e objetivos.
- ☑️ Plano de ensino.



Próxima aula

- Styled Componentes:
  - Motivação para seu uso.
  - Instalação.

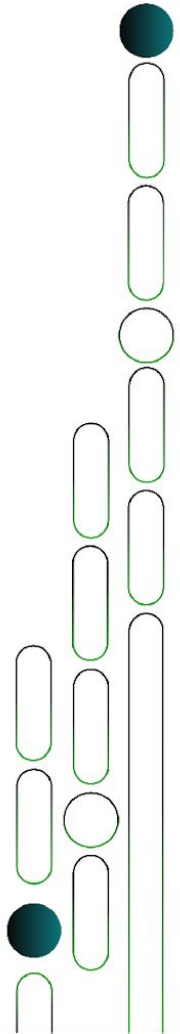
Aprenda com quem faz

React III

Capítulo 2. Styled Components

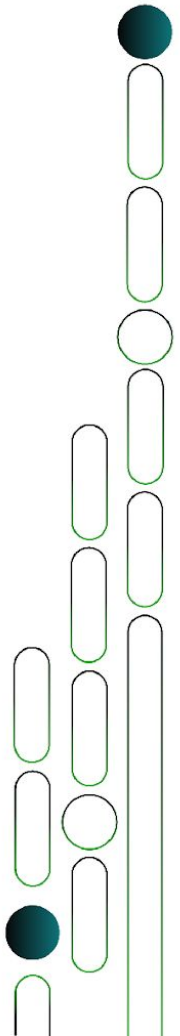
Aula 2.1. Motivação para o uso do Styled Components e Instalação

Prof. Rodrigo Borba



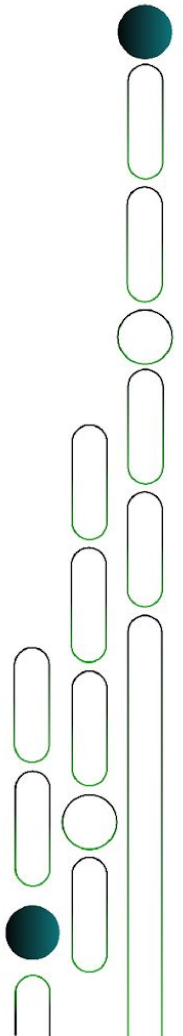
Nesta aula

- Quais problemas o Styled Components resolve.
- Instalação.



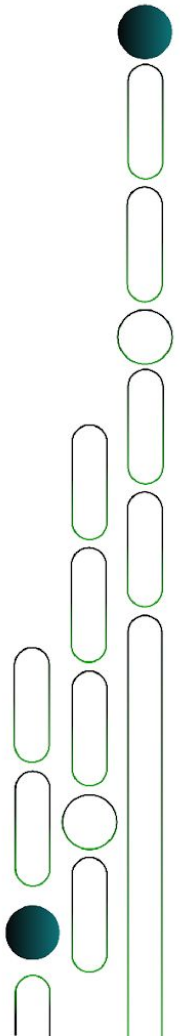
Quais problemas o Styled Components resolve

- *"styled-components is the result of wondering how we could enhance CSS for styling React component systems" — Documentação oficial.*



Quais problemas o Styled Components resolve

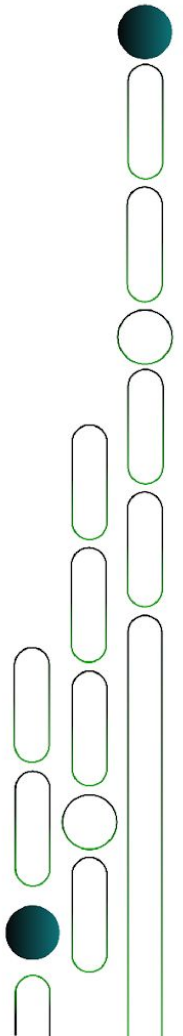
- Apenas o CSS crítico.
- Sem bugs por conta de nomes de classes.
- Fácil remoção de CSS.
- Estilização dinâmica de maneira fácil.
- Manutenção fácil.
- Inserção automática de prefixo de Vendor.



Instalação

NPM
npm install --save styled-components

YARN
yarn add styled-components



Próxima aula

- Styled Componentes:
  - Sintaxe.
  - Estilos dinâmicos baseados em propriedades.

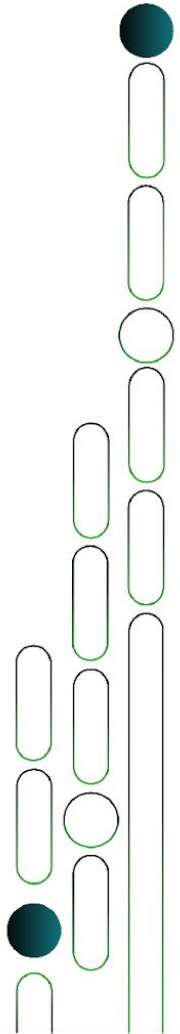
Aprenda com quem faz

React III

Capítulo 2. Styled Components

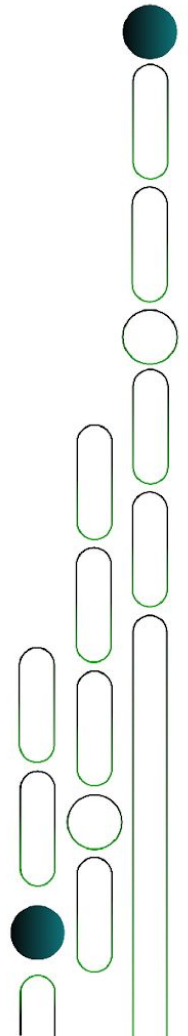
Aula 2.2. Estilos dinâmicos baseados em propriedades

Prof. Rodrigo Borba



Nesta aula

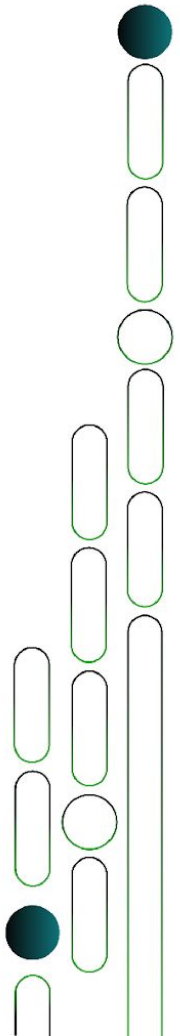
- Sintaxe.
- Estilos dinâmicos.



Sintaxe

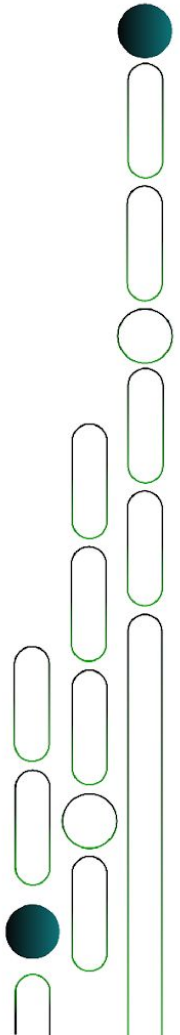
Template String ou Template Literal ES6

```
const nome = "Rodrigo"
console.log(`Meu nome é ${nome}`);

// Meu nome é Rodrigo
```



Sintaxe

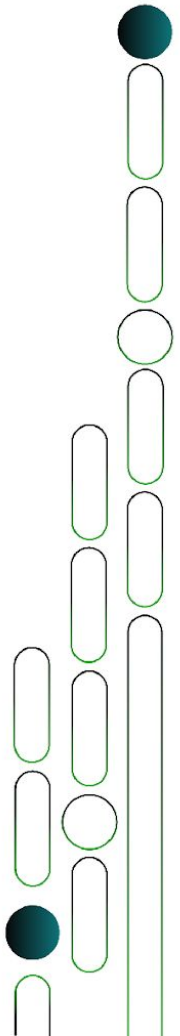
```
styled.div`...`;
styled.button`...`;
styled.section`...`;
Etc...
```



Sintaxe

```
styled.div`
  padding: 4em;
  background: red;
`;
```

```
styled.div`
  padding: 4em;
  background: ${(props) => /* props... */};
`;
```

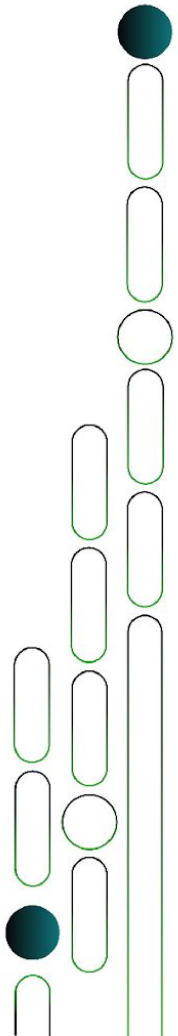


Estilos dinâmicos

Dentro de template strings conseguimos acessar objetos e aplicar lógicas

```
const wrapper = styled.div`
  padding: 4em;
  background: ${(props) => props.customColor ?
props.customColor : red};
`;
```

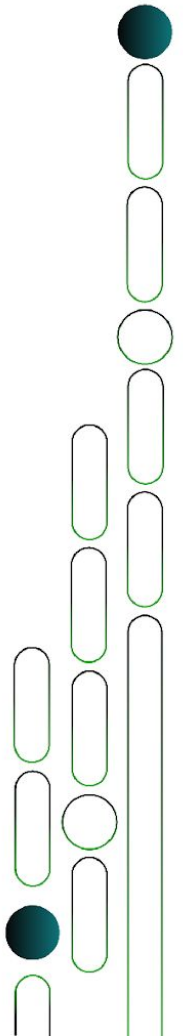
```
<Wrapper customColor=“blue"></Wrapper>
```



Tema

Também conseguimos definir temas

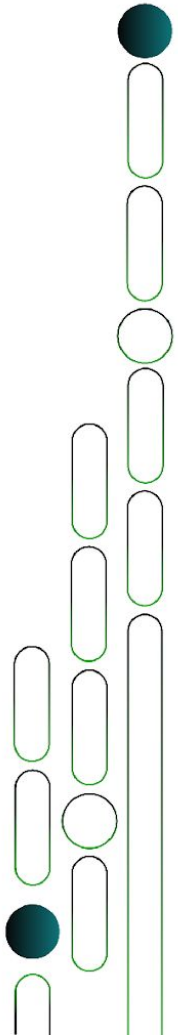
```
const theme = {
  main: "blue"
};

const Button = styled.button`
  color: ${props => props.theme.main};
  border: 2px solid ${props => props.theme.main};
`;
```



Tema

Para o tema ser aplicado é preciso encapsular os Styled Components dentro do ThemeProvider

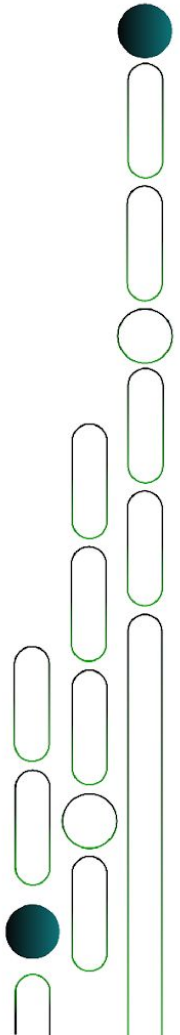
```
render(
  <div>
    <Button>Normal</Button> <<<<<< Não tem acesso as variáveis de tema

    <ThemeProvider theme={theme}> <<<<<< Tem acesso as variáveis
      <Button>Themed</Button>
    </ThemeProvider>
  </div>
);
```



Tema

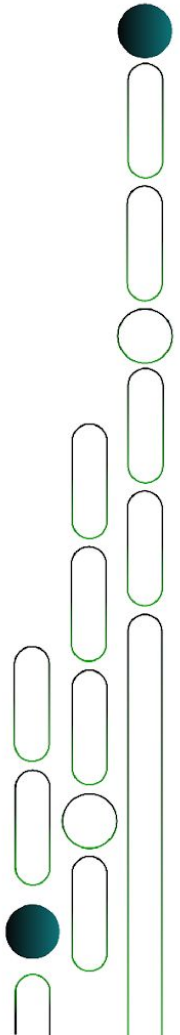
Funções também podem ser passadas no tema.

```
const theme = {
  fg: "palevioletred",
  bg: "white"
};
```



Próxima aula

- Estendendo estilos e aprendendo animações.



Aprenda com quem faz

React III

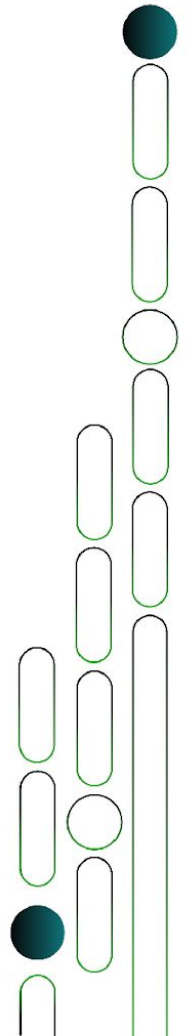
Capítulo 2. Styled Components

Aula 2.3. Estendendo estilos e aprendendo animações

Prof. Rodrigo Borba

Nesta aula

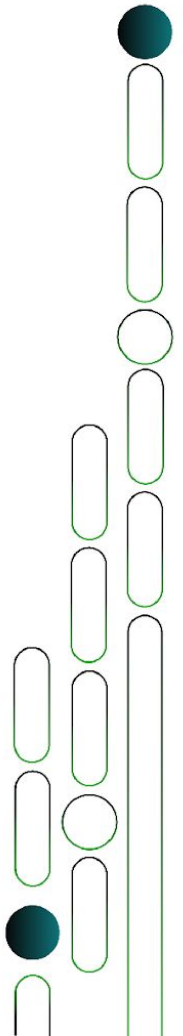
- Estendendo estilos.
- Animações.



Estendendo Estilos

Styled Componentes podem ser estendidos facilmente.

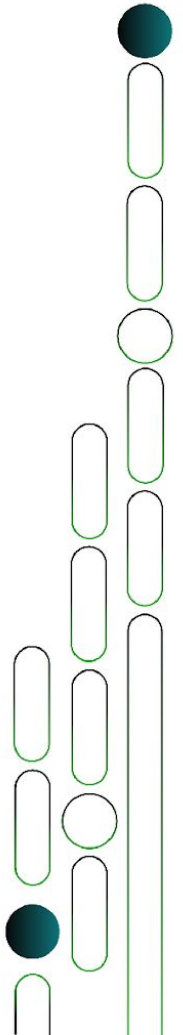
```
const StyledButtonBase = ...

const StyledButtonExtended = styled(StyledButtonBase)`
....
`
```



Animações

Keyframe helper deve ser exportado para que uma única instância seja usada na aplicação.

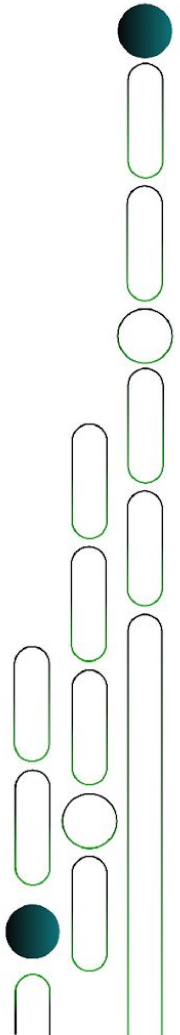
```
const rotateKeyFrame = keyframes`
  from {
    transform: rotate(0deg);
  }

  to {
    transform: rotate(360deg);
  }
`;
```



Animações

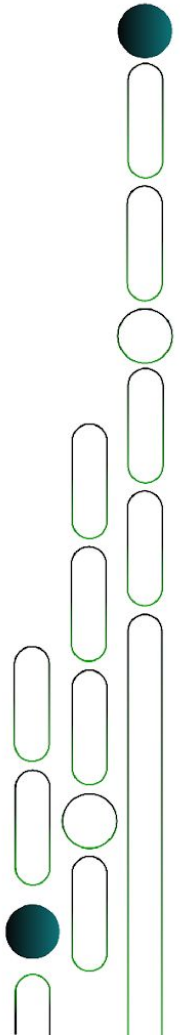
E então:

```
const Rotate = styled.div`
  animation: ${rotateKeyFrame} 2s linear infinite;
`;
```



Próxima aula

- Na prática!



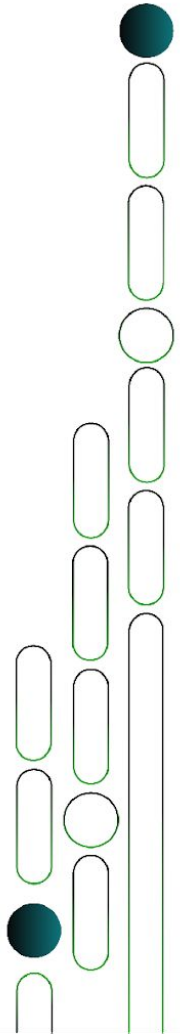
Aprenda com quem faz

React III

Capítulo 2. Styled Components

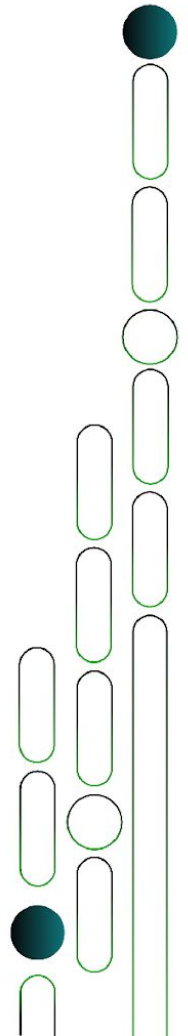
Aula 2.4. Na prática!

Prof. Rodrigo Borba

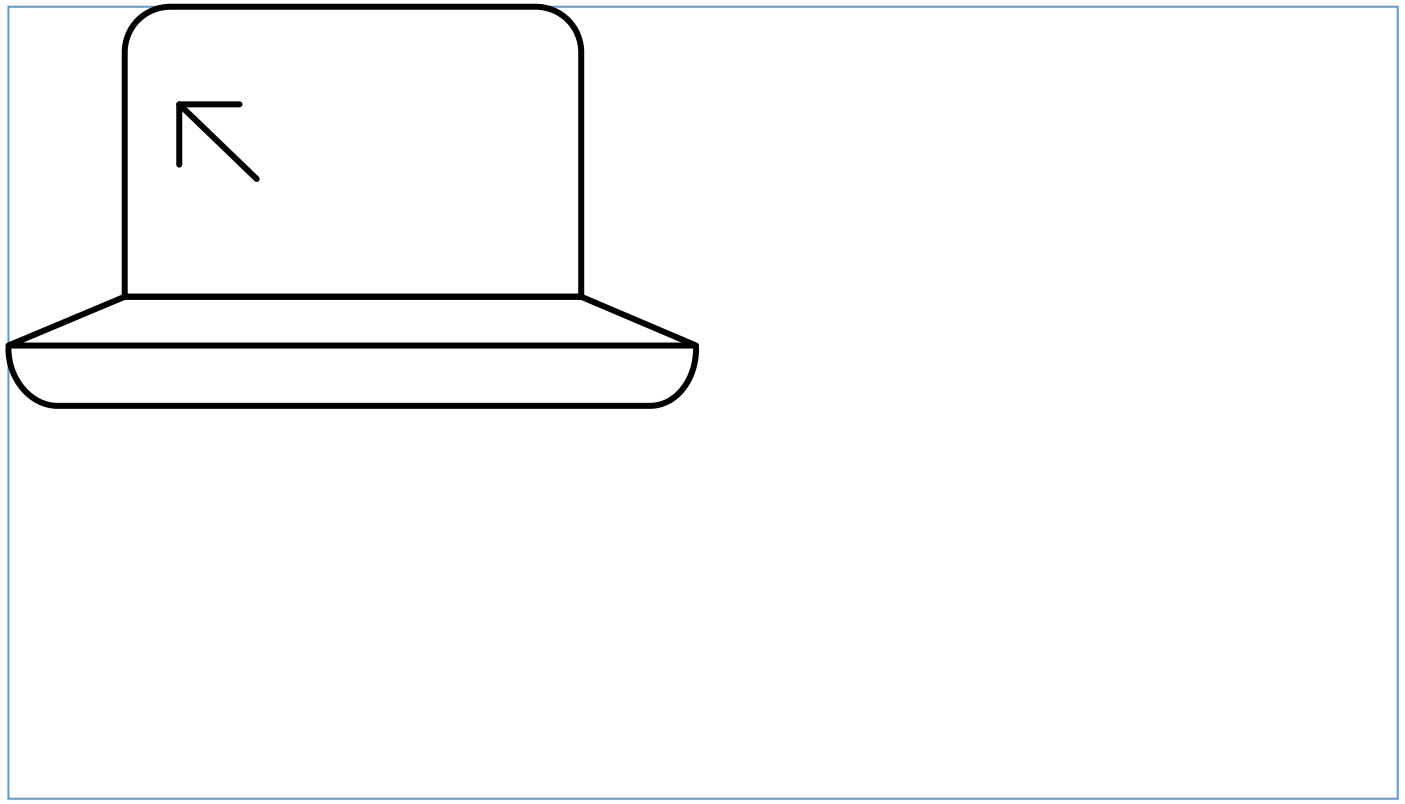


Nesta aula

- Styled Components na prática.

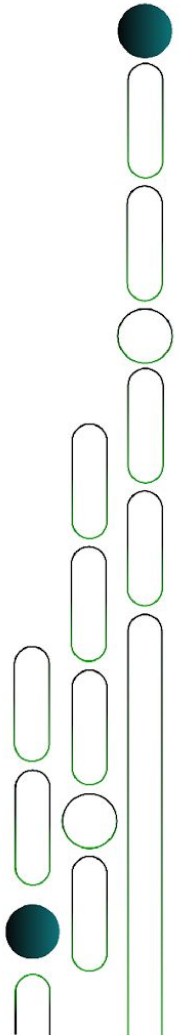


Acompanhe o professor



Próxima aula

- Bibliotecas de *Data Fetching*:
  - Introdução.



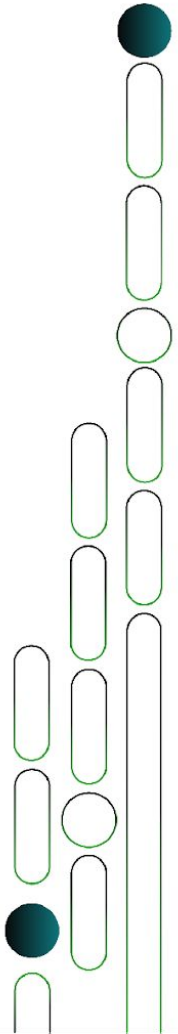
Aprenda com quem faz

React III

Capítulo 3. Bibliotecas de *Data Fetching*

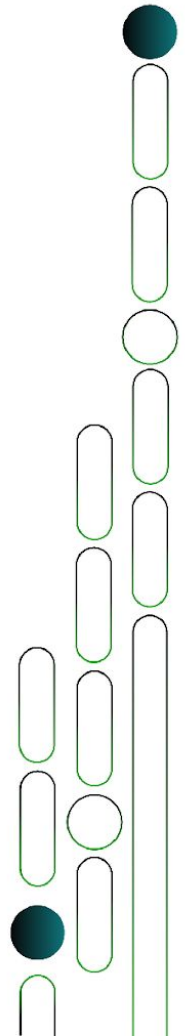
Aula 3.1. Introdução

Prof. Rodrigo Borba



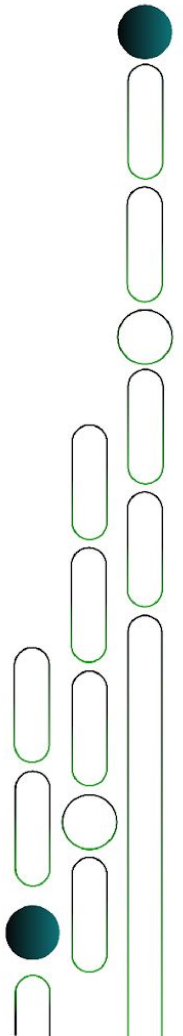
Nesta aula

- Introdução a bibliotecas de Datafetching.



Introdução a bibliotecas de Data fetching

- Data fetch rápido, leve e “reusável”.
- Cache e prevenção de requests repetidos.
- Experiência em tempo real.
- Agnóstico quanto ao protocolo.
- Suporte a Server Side Rendering, Incremental Static Regeneration e Static Site Generation.
- Pronto para o TypeScript.
- Suporte ao React Native.



Introdução a bibliotecas de Data fetching

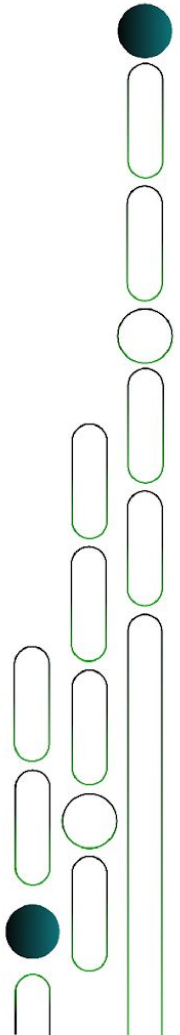
“React puro”

```
const [data, setData] = useState()
const [error, setError] = useState()

 useEffect(() => {
   fetch("www.someSite")
    .then((data) => setData(data))
    .catch(err => setError(err));
 }, [])
```

Introdução a bibliotecas de Data fetching

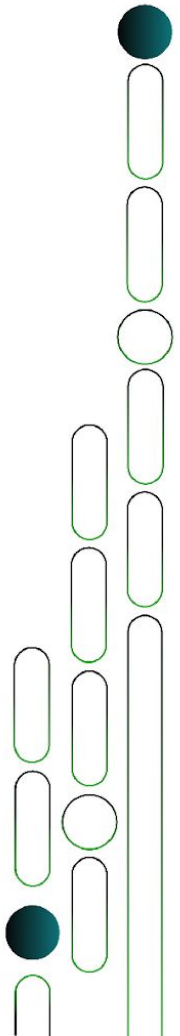
Pode ser substituído, por exemplo, por:

```
const { data, loading, error } = useSWR('/api/data/algum_id',
fetcher)

const { isLoading, error, data } =
useQuery('/api/data/algum_id', fetcher)
```



Próxima aula

- SWR x React Query.



Aprenda com quem faz

React III

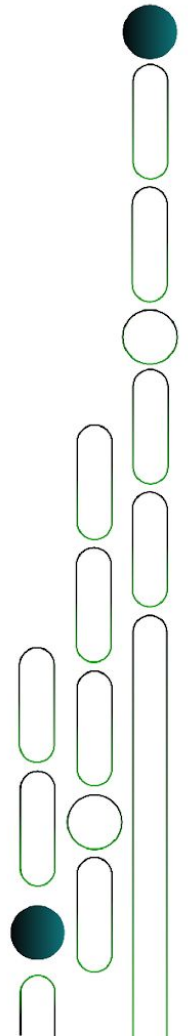
Capítulo 3. Bibliotecas de *Data Fetching*

Aula 3.2. SWR x React Query

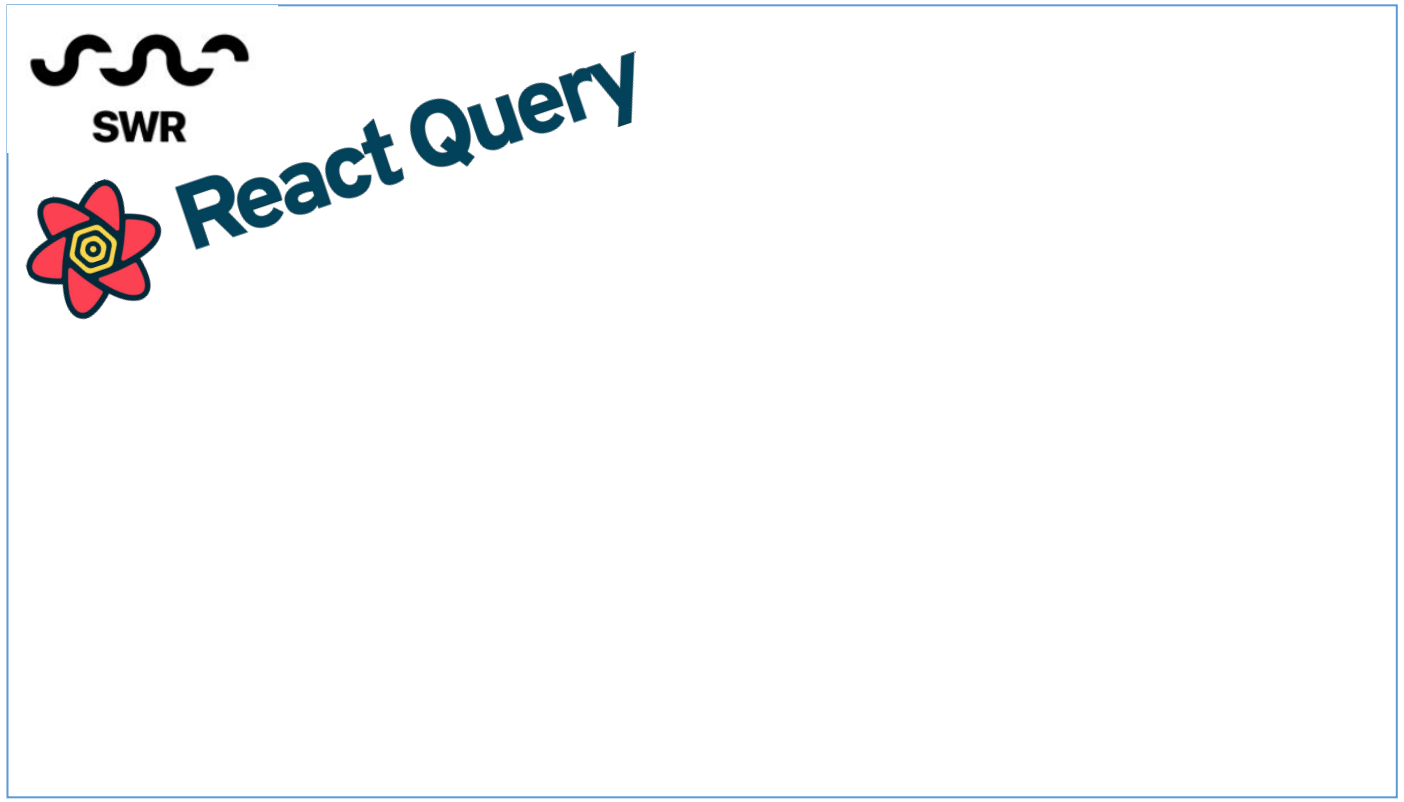
Prof. Rodrigo Borba

Nesta aula

- SWR x React Query.



SWR x React Query



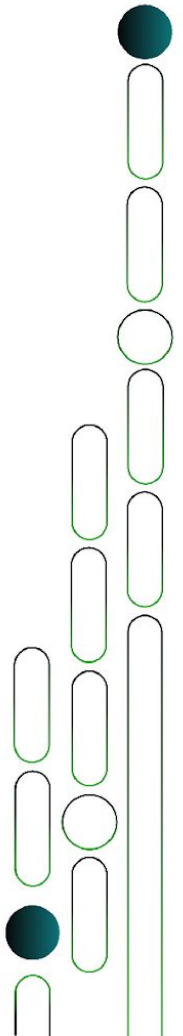
SWR

“Stale while Revalidate”

Exibe primeiramente os dados do Cache (antigos, possivelmente defasados).

Busca os dados atualizados.

Em posse dos dados atualizados, passa a exibir eles no lugar dos antigos.



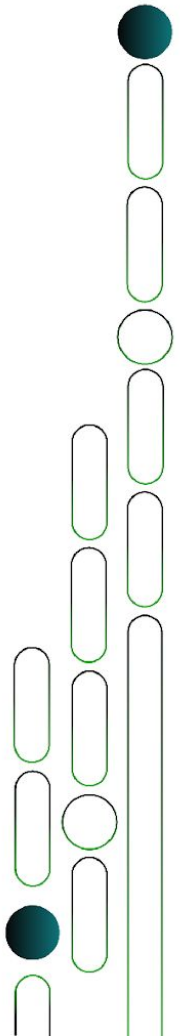
SWR

1) Define um fetcher (pode ser reusado em diferentes contextos):

```
const fetcher = (...args) => fetch(...args).then(res => res.json())
```

2) Com o hook useSWR, passamos o ID (geralmente o PATH para uma API):

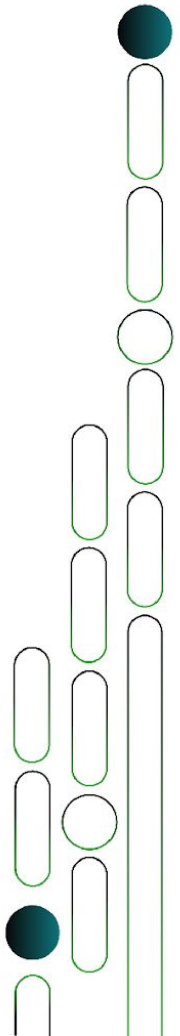
```
const { data, loading, error } = useSWR('/api/data/algum_id', fetcher)
```



SWR

Também podemos passar parâmetros na chamada:

```
useSWR(key, fetcher, {
   revalidateIfStale: true,
   revalidateOnFocus: true,
   revalidateOnReconnect: true,
   refreshInterval: 1000
})
```

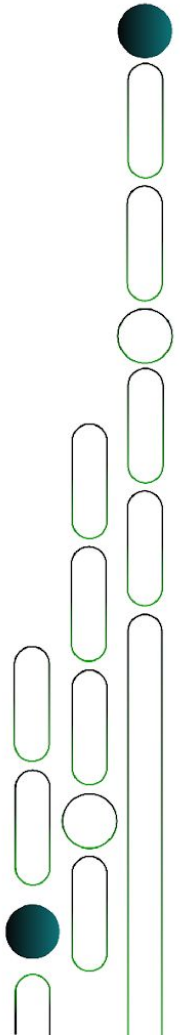


React Query

Mesmo princípio do SWR, stale-while-revalidate.

- É necessário envolver a aplicação no QueryClientProvider:

```
import { QueryClient, QueryClientProvider } from
"react-query";
const queryClient = new QueryClient();

ReactDOM.render(
  <QueryClientProvider client={queryClient}>
    <App />
  </QueryClientProvider>,
  document.getElementById('root')
);
```



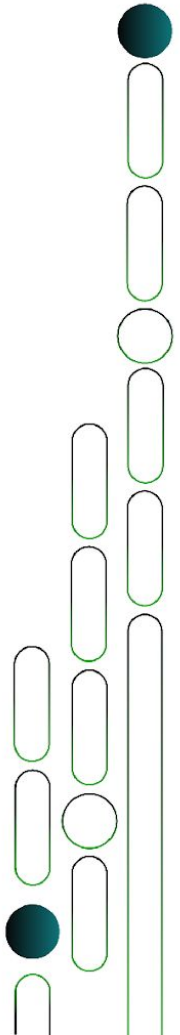
React Query

1) Define o fetcher:

```
const fetcher = (...args) => fetch(...args).then(res =>
res.json())
```

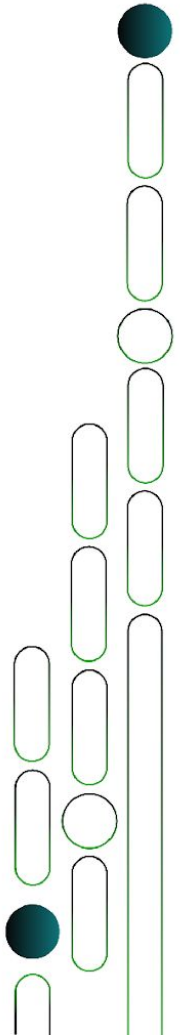
2) O usamos na chamada:

```
const component = (_ => {
   const { isLoading, error, data } = useQuery('queryId', fetcher)
   ....
}
```



Próxima aula

- Na prática.



Aprenda com quem faz

React III

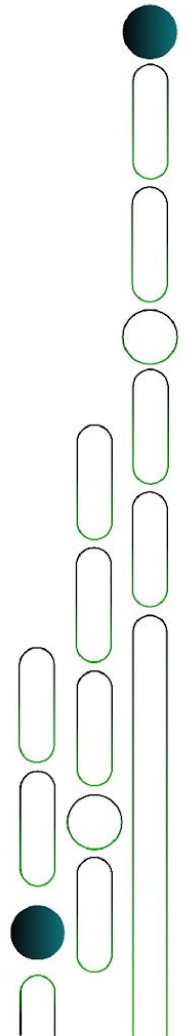
Capítulo 3. Bibliotecas de Data Fetching

Aula 3.3. Na prática!

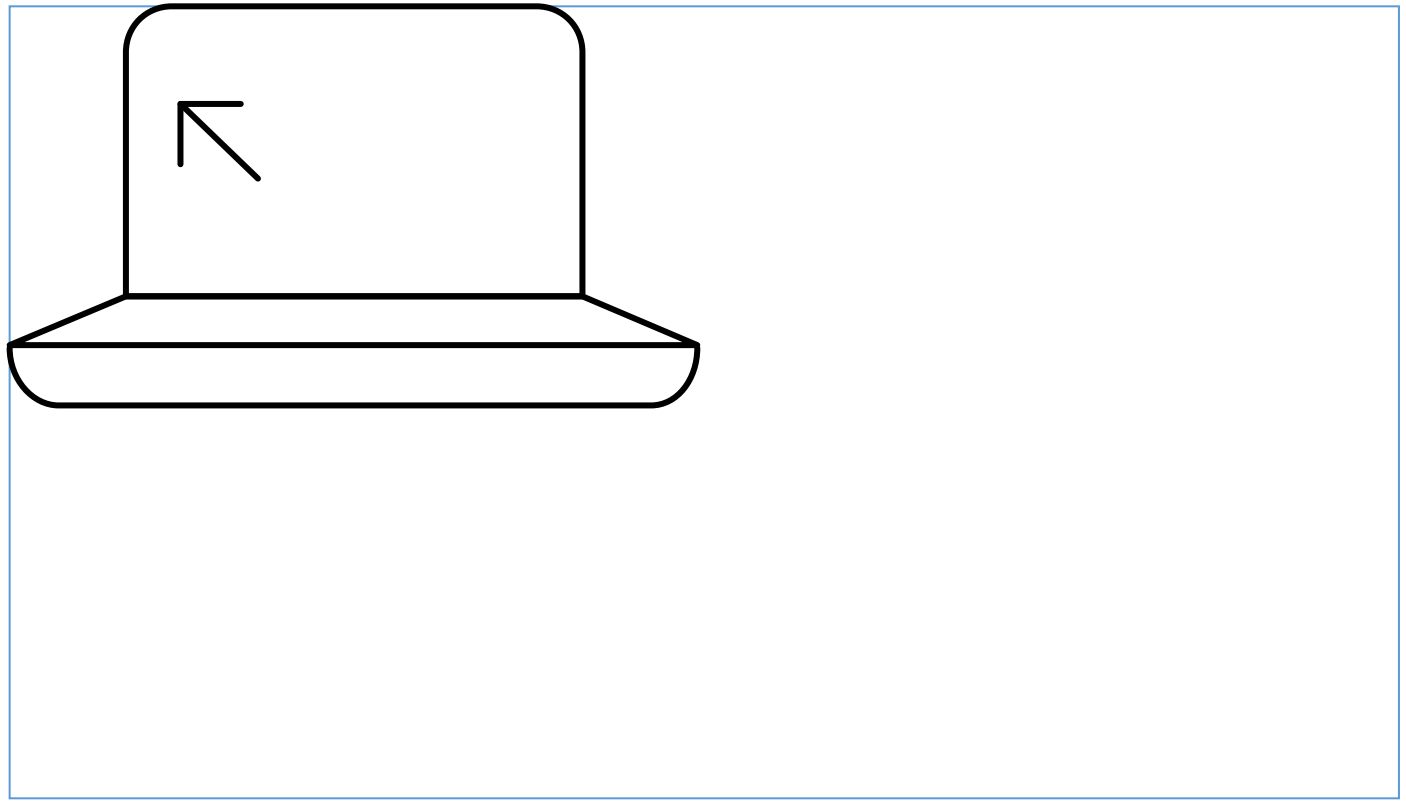
Prof. Rodrigo Borba

Nesta aula

- SWR e React Query na prática.

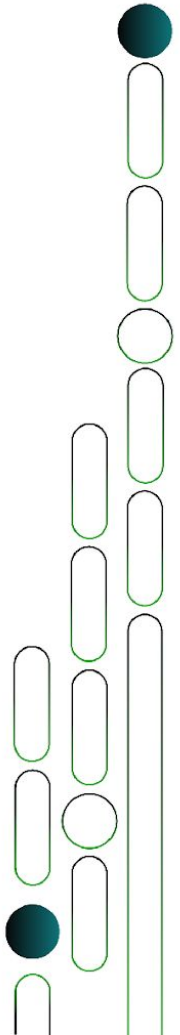


Acompanhe o professor



Próxima aula

- Estratégias de Renderização:
  - Introdução a estratégias de renderização.



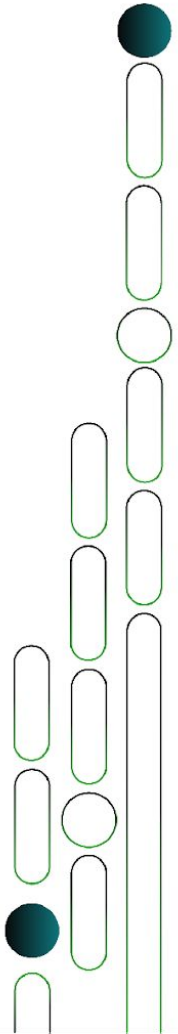
Aprenda com quem faz

React III

Capítulo 4. Estratégias de Renderização

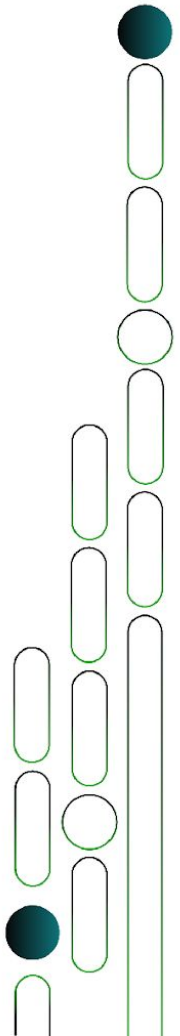
Aula 4.1. Introdução a estratégias de renderização

Prof. Rodrigo Borba



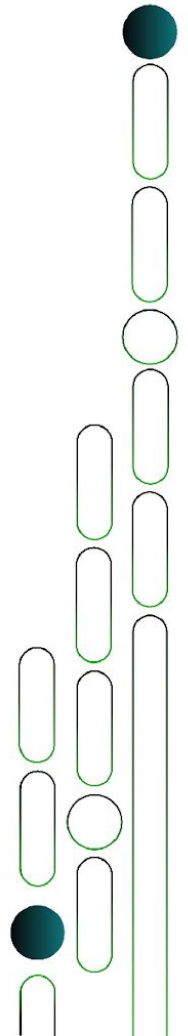
Nesta aula

- Introdução.
- CSR - Client Side Rendering.
- SSR - Server Side Rendering.
- SSG - Static Site Generation.
- ISR - Incremental Static Regeneration.
- SPA - Single Page Application.



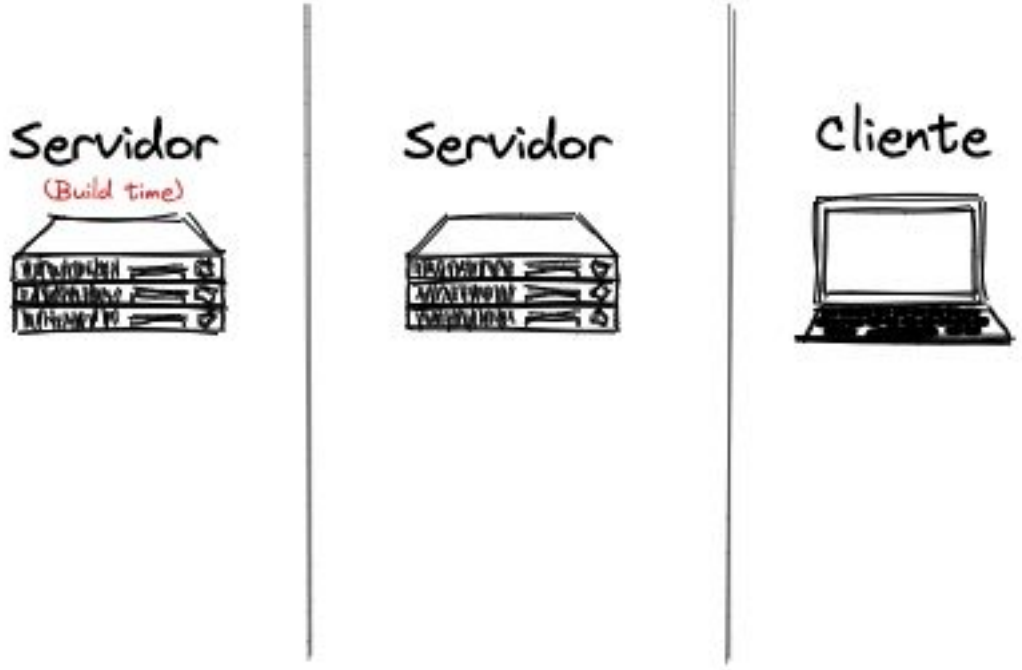
Introdução

Momentos em que o Javascript responsável por gerar a página HTML pode ser executado.

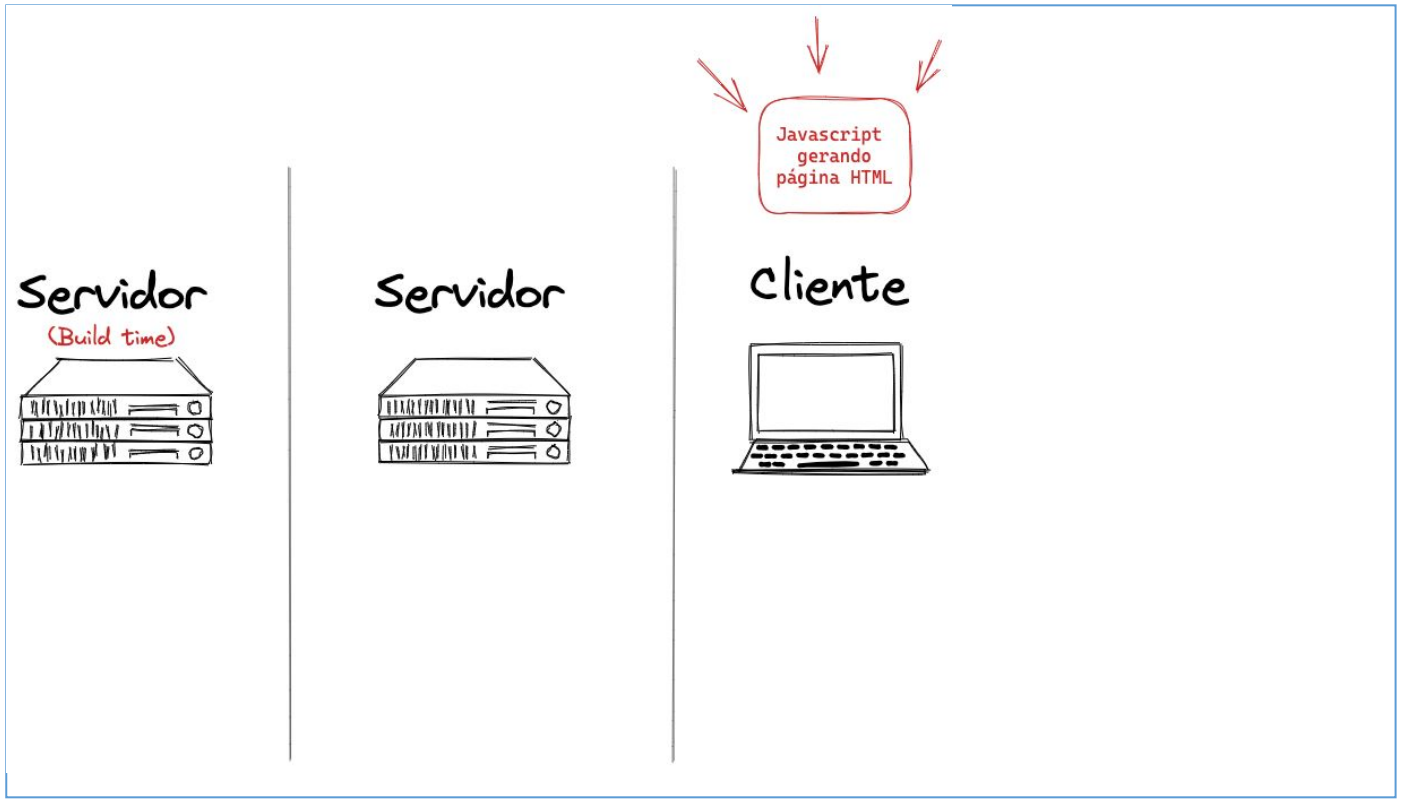


Introdução

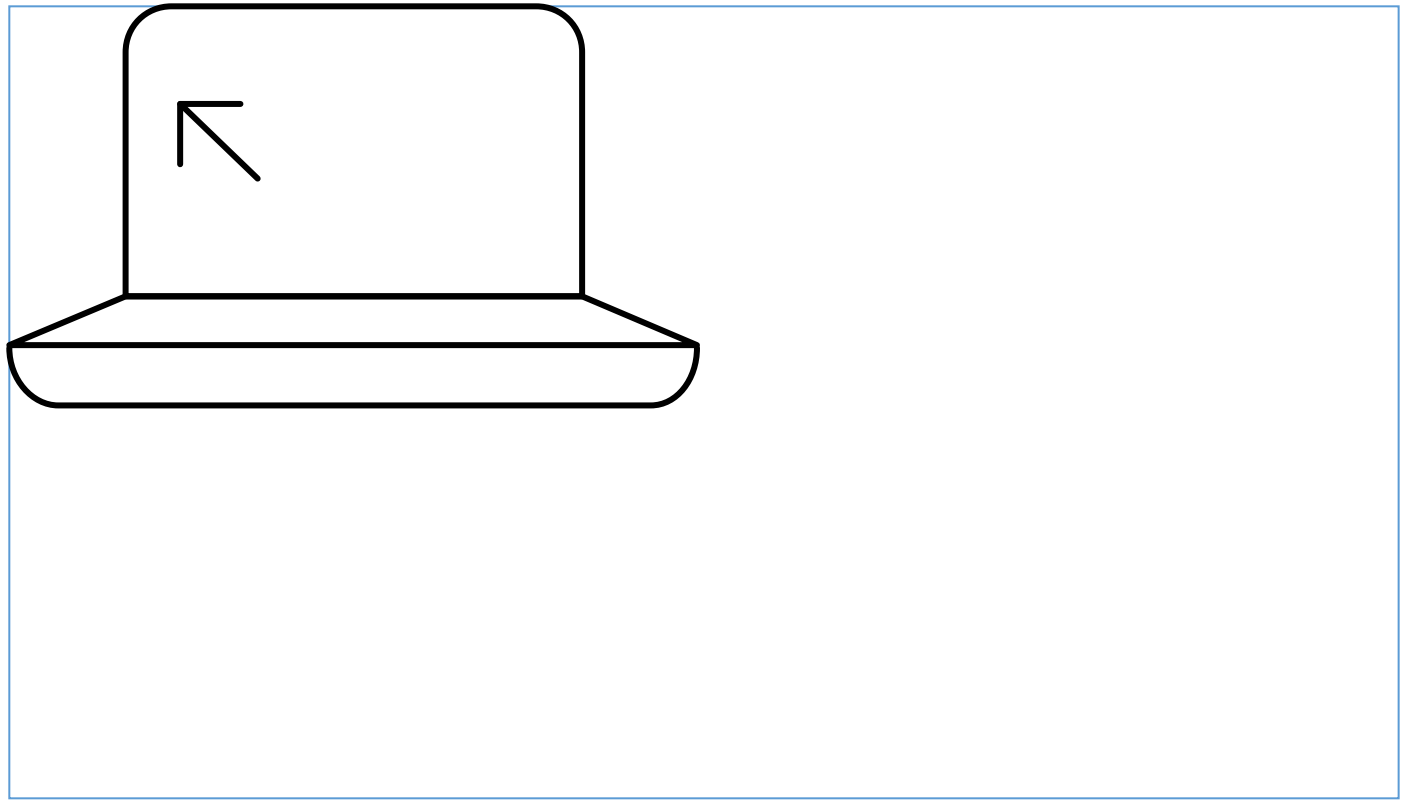
Momentos em que o Javascript responsável por gerar a página HTML pode ser executado.



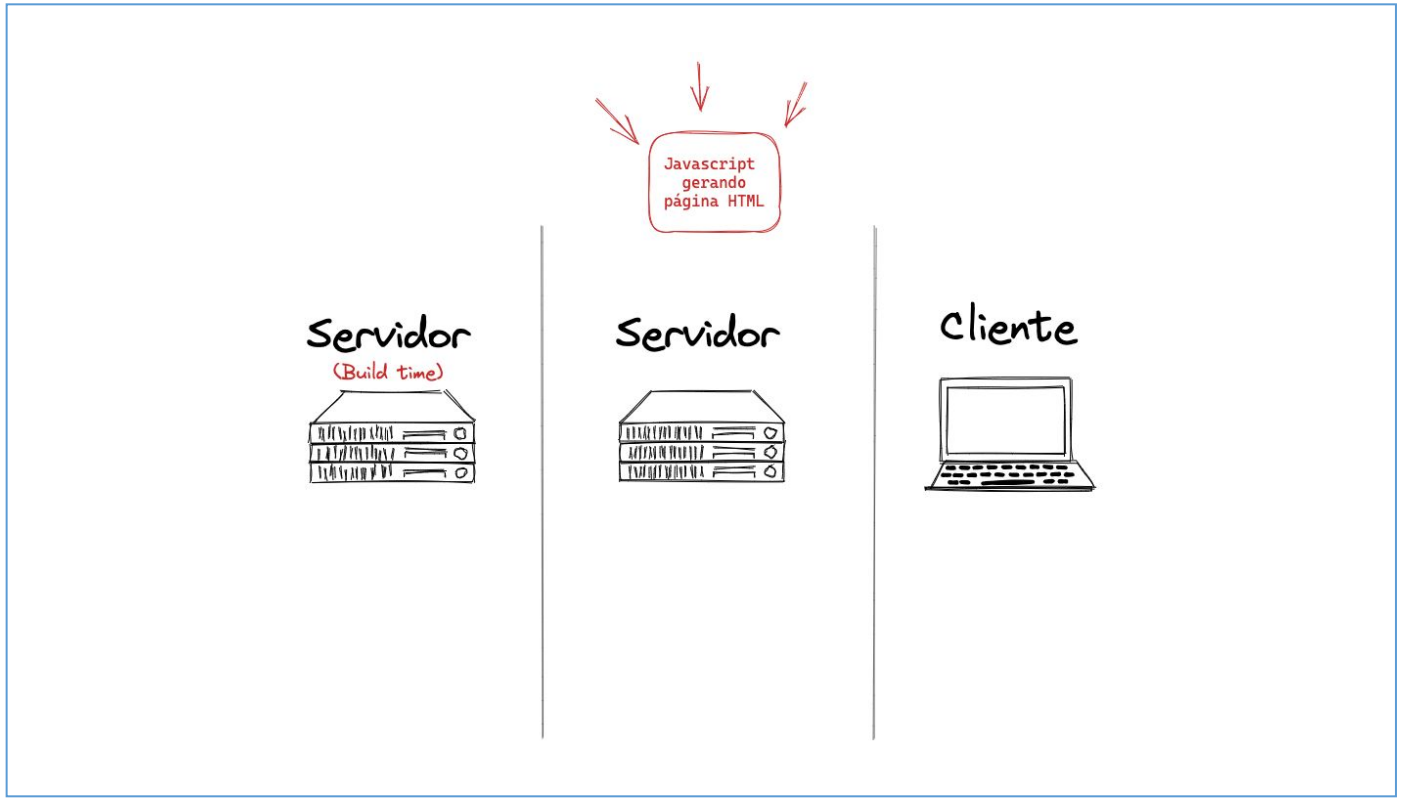
CSR – Client Side Rendering



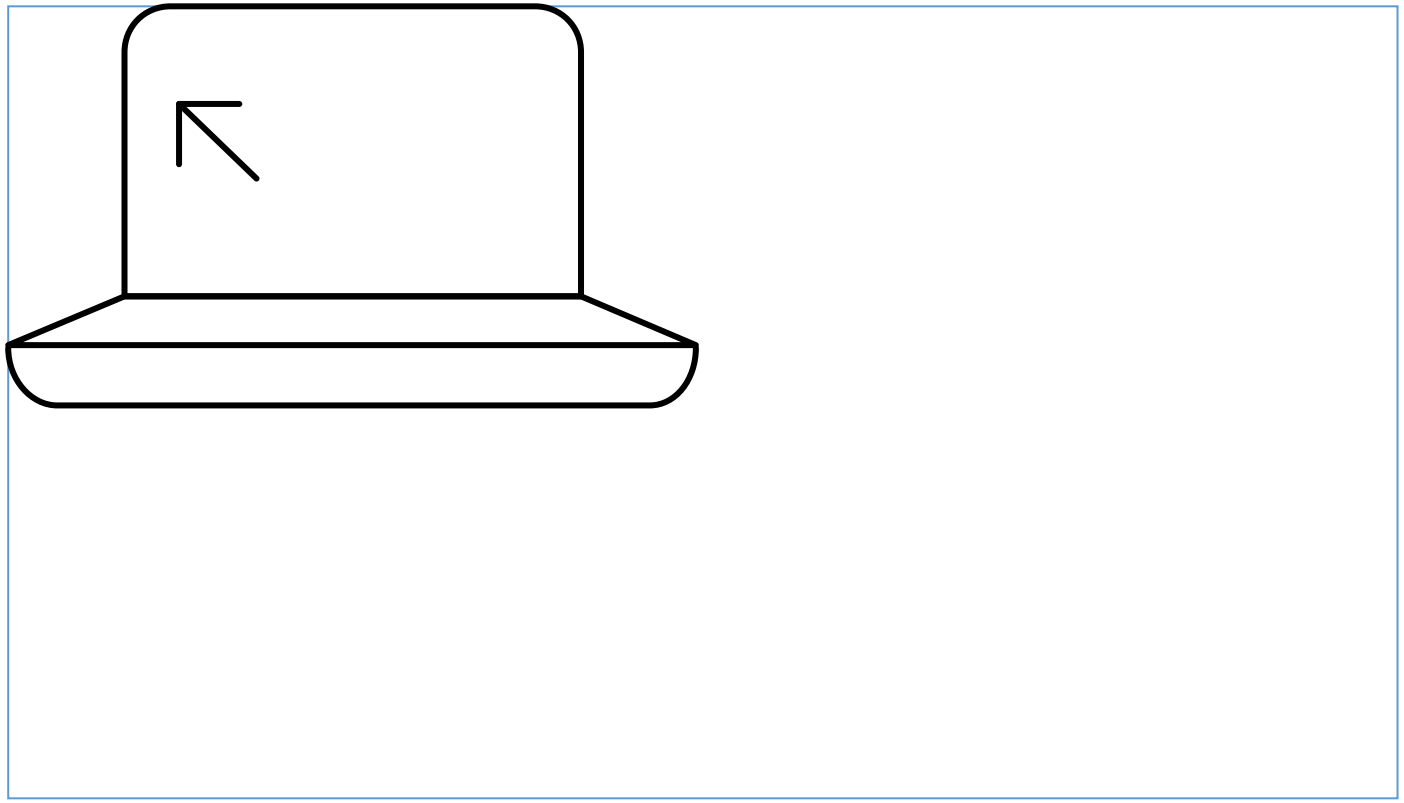
Acompanhe o professor



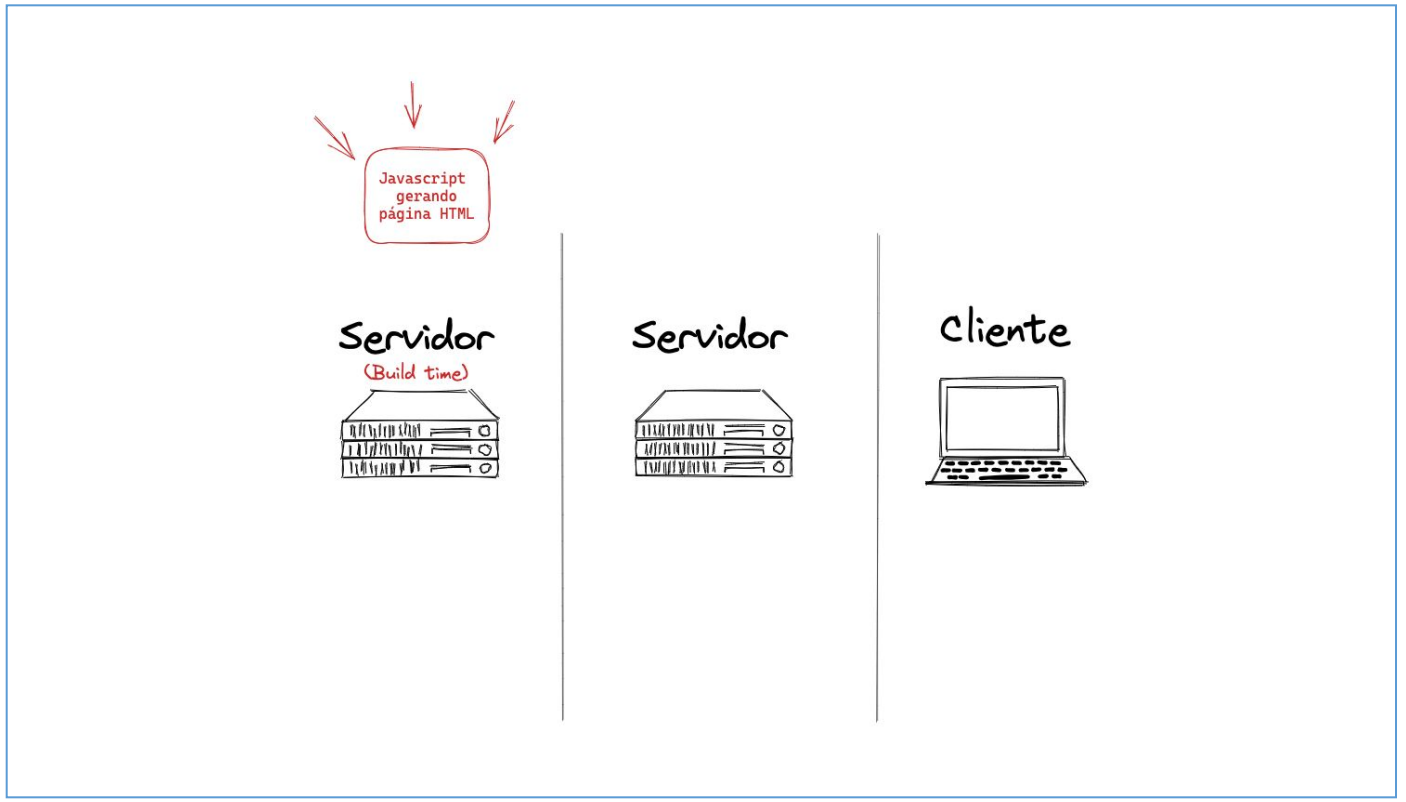
SSR – Server Side Rendering



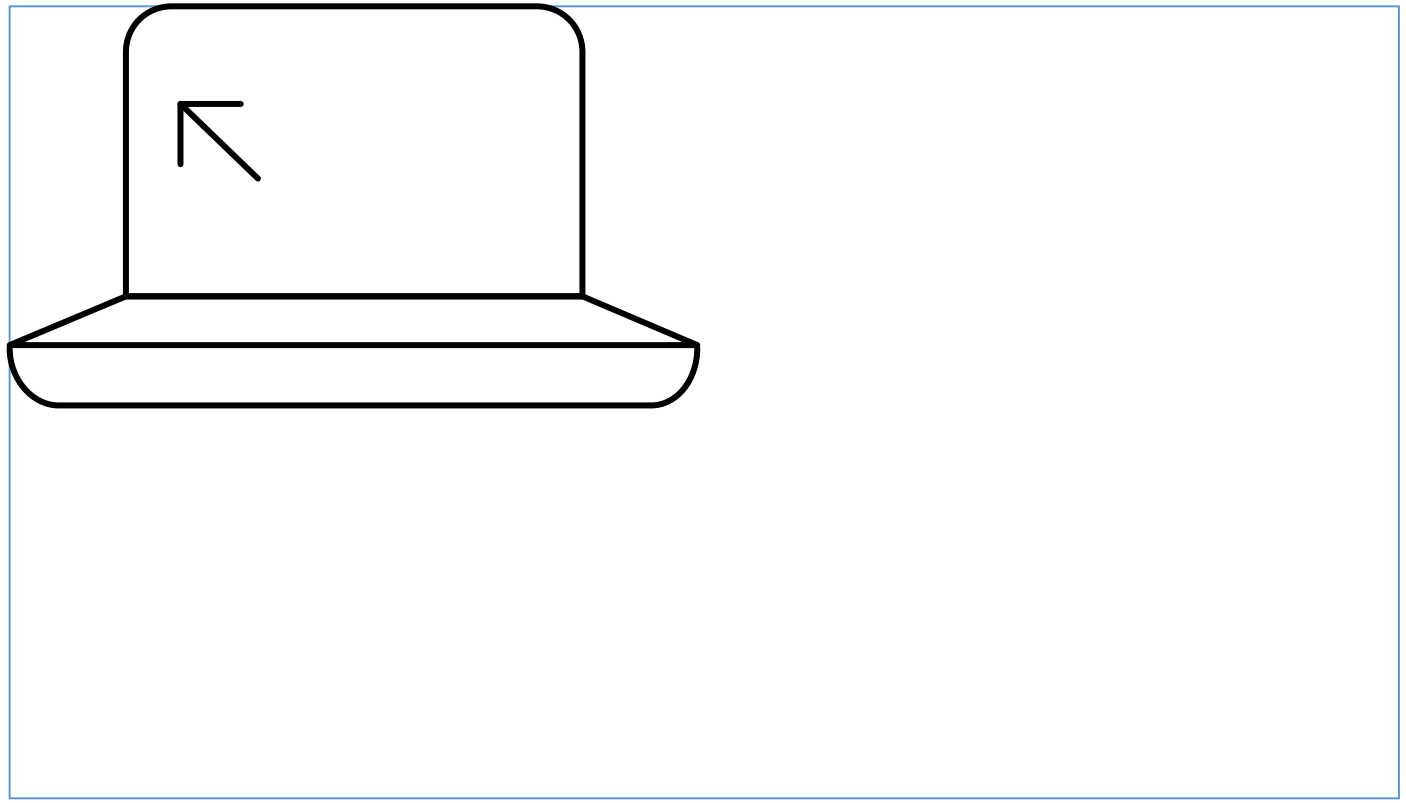
SSR – Server Side Rendering



SSG – Static Site Generation

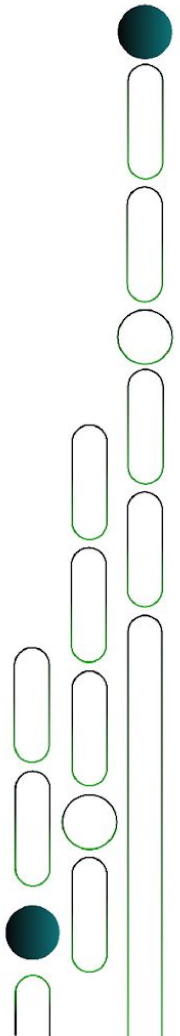


Acompanhe o professor



Próxima aula

- Client Side Rendering (CSR).



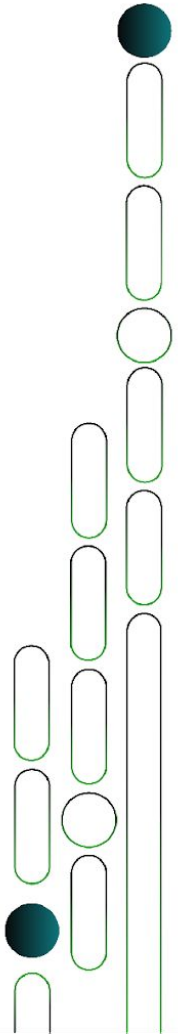
XPe
Aprenda com quem faz

React III

Capítulo 4. Estratégias de Renderização

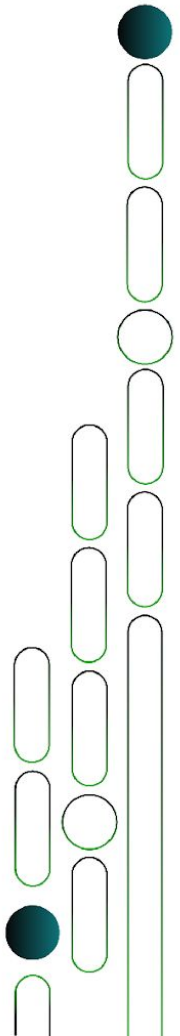
Aula 4.2. Client Side Rendering (CSR)

Prof. Rodrigo Borba

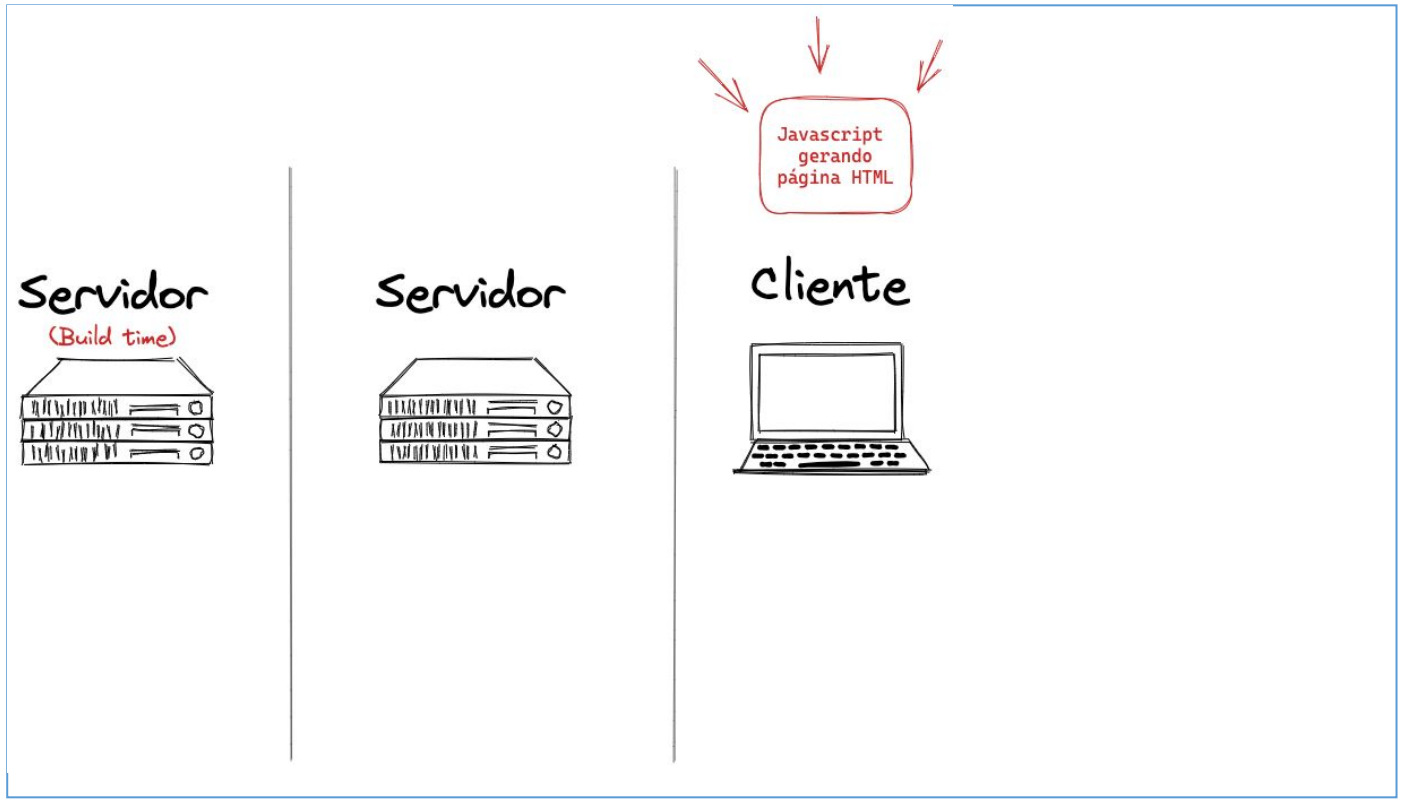


Nesta aula

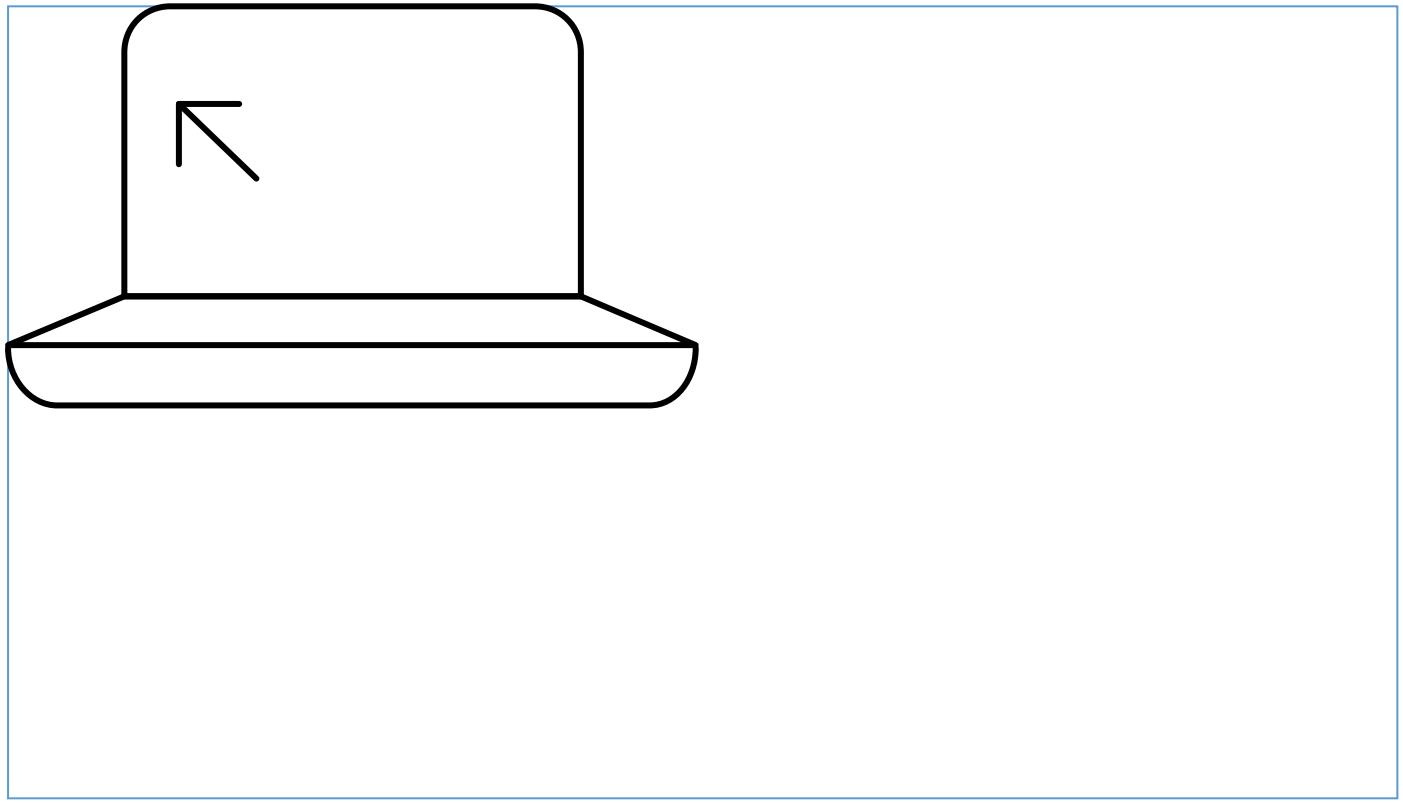
- Client Side Rendering.



CSR – Client Side Rendering

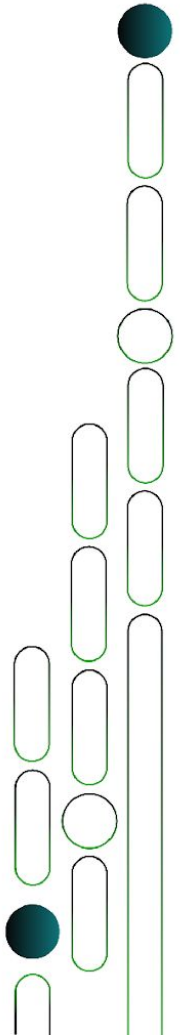


Acompanhe o professor



Próxima aula

- Server Side Rendering (SSR).



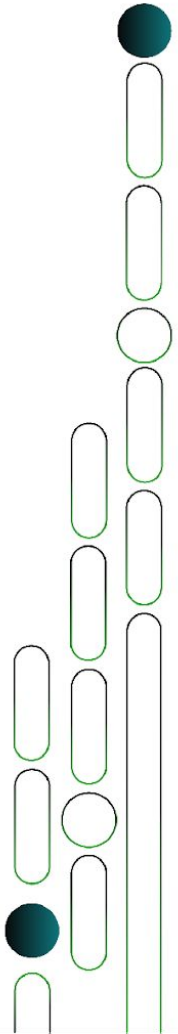
XPe
Aprenda com quem faz

React III

Capítulo 4. Estratégias de Renderização

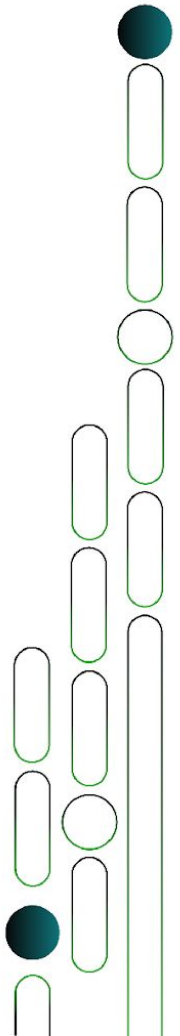
Aula 4.3. Server Side Rendering (SSR)

Prof. Rodrigo Borba

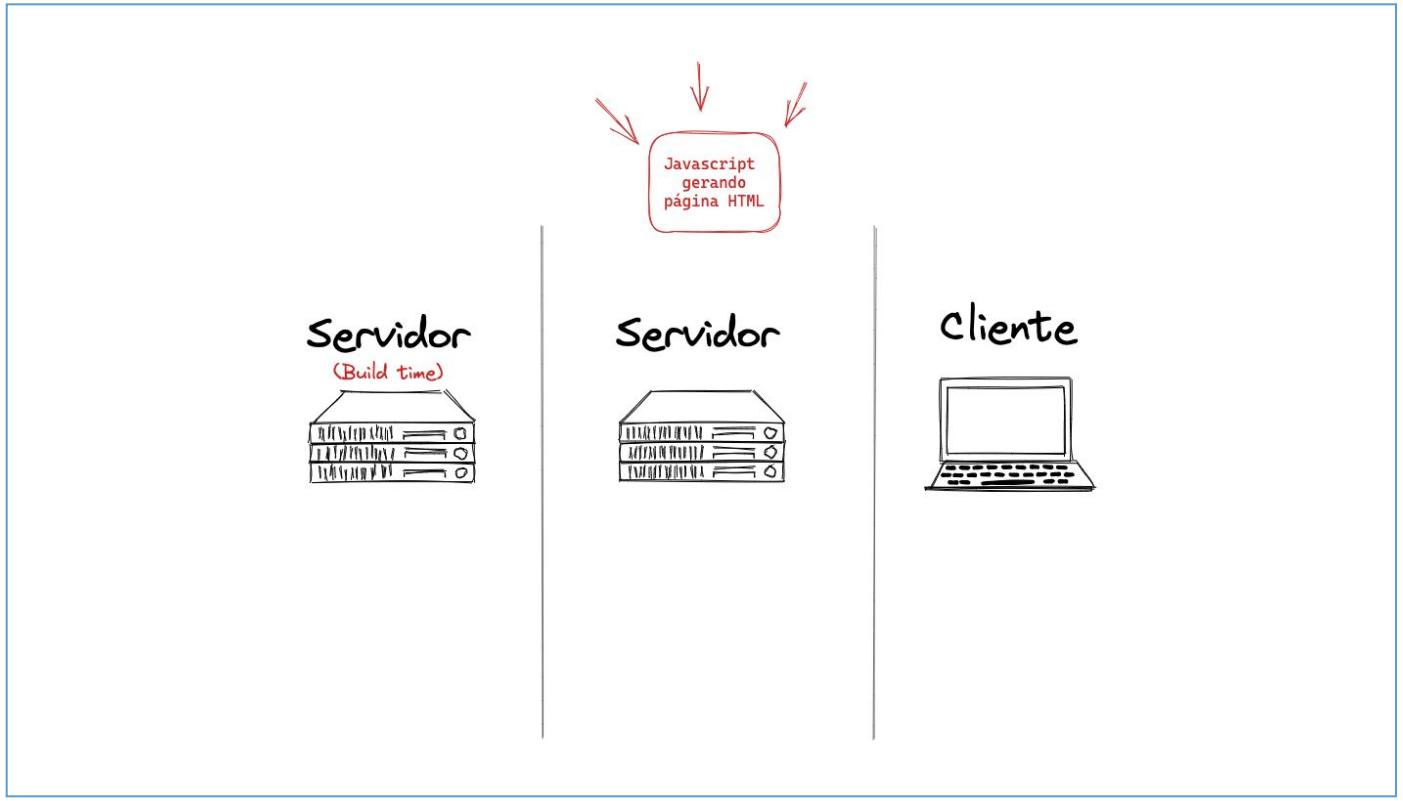


Nesta aula

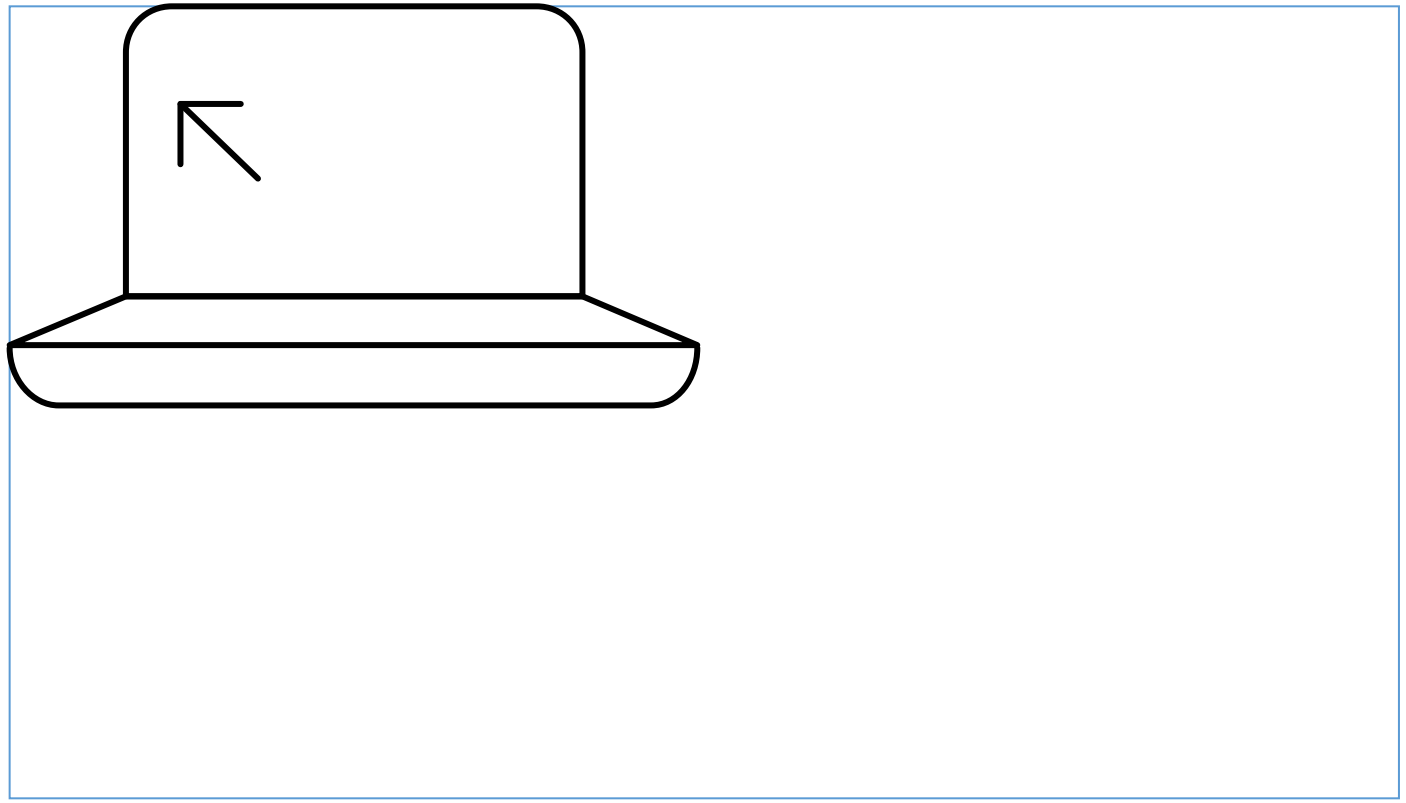
- Server Side Rendering.



SSR – Server Side Rendering

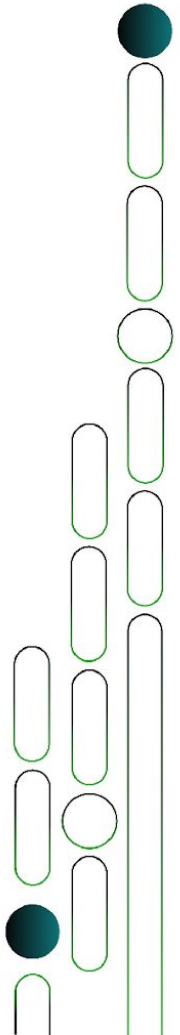


Acompanhe o professor



Próxima aula

- Static Site Generation (SSG).



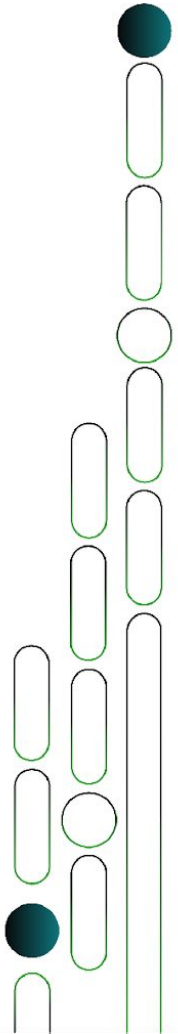
XPe
Aprenda com quem faz

React III

Capítulo 4. Estratégias de Renderização

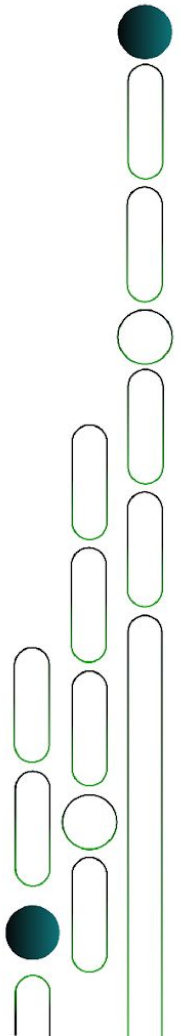
Aula 4.4. Static Site Generation (SSG)

Prof. Rodrigo Borba

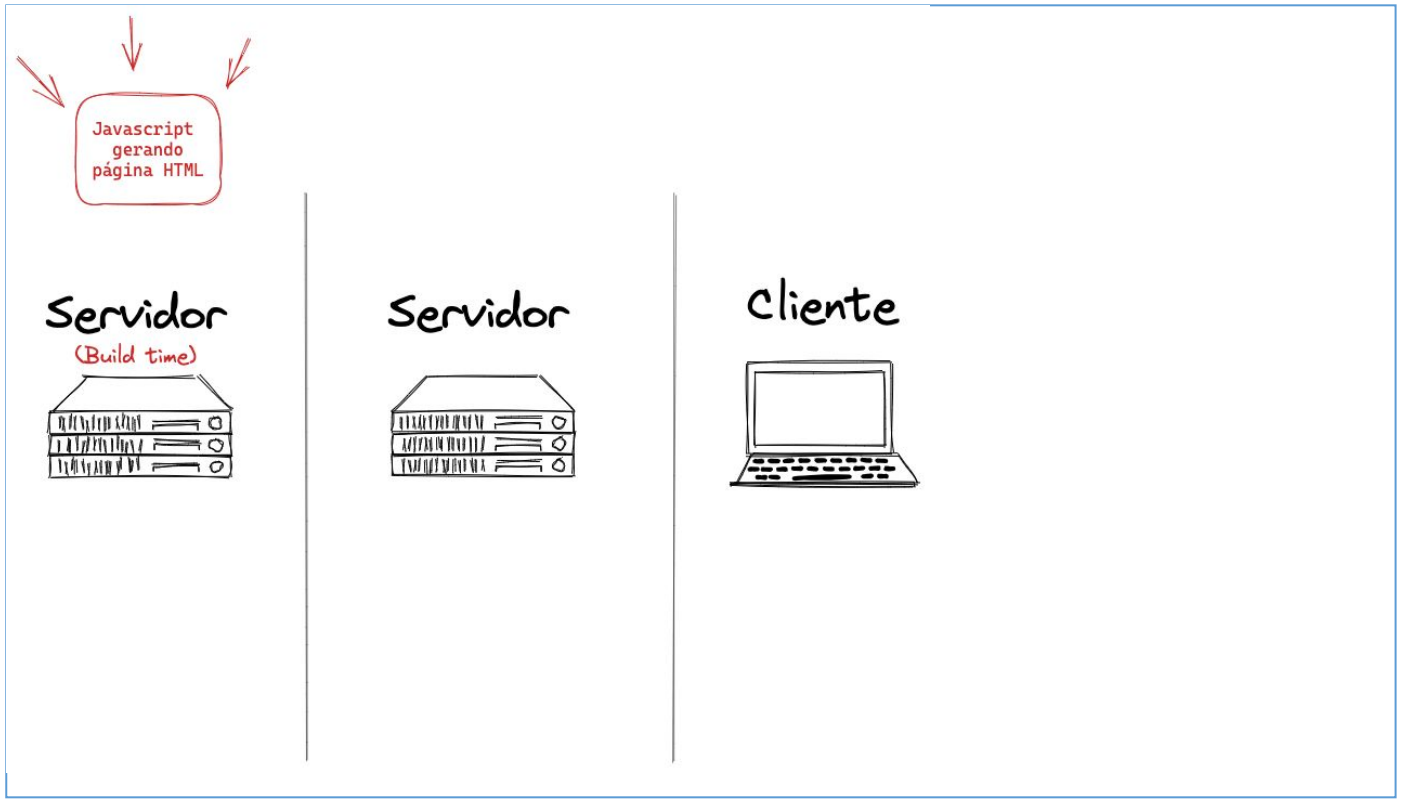


Nesta aula

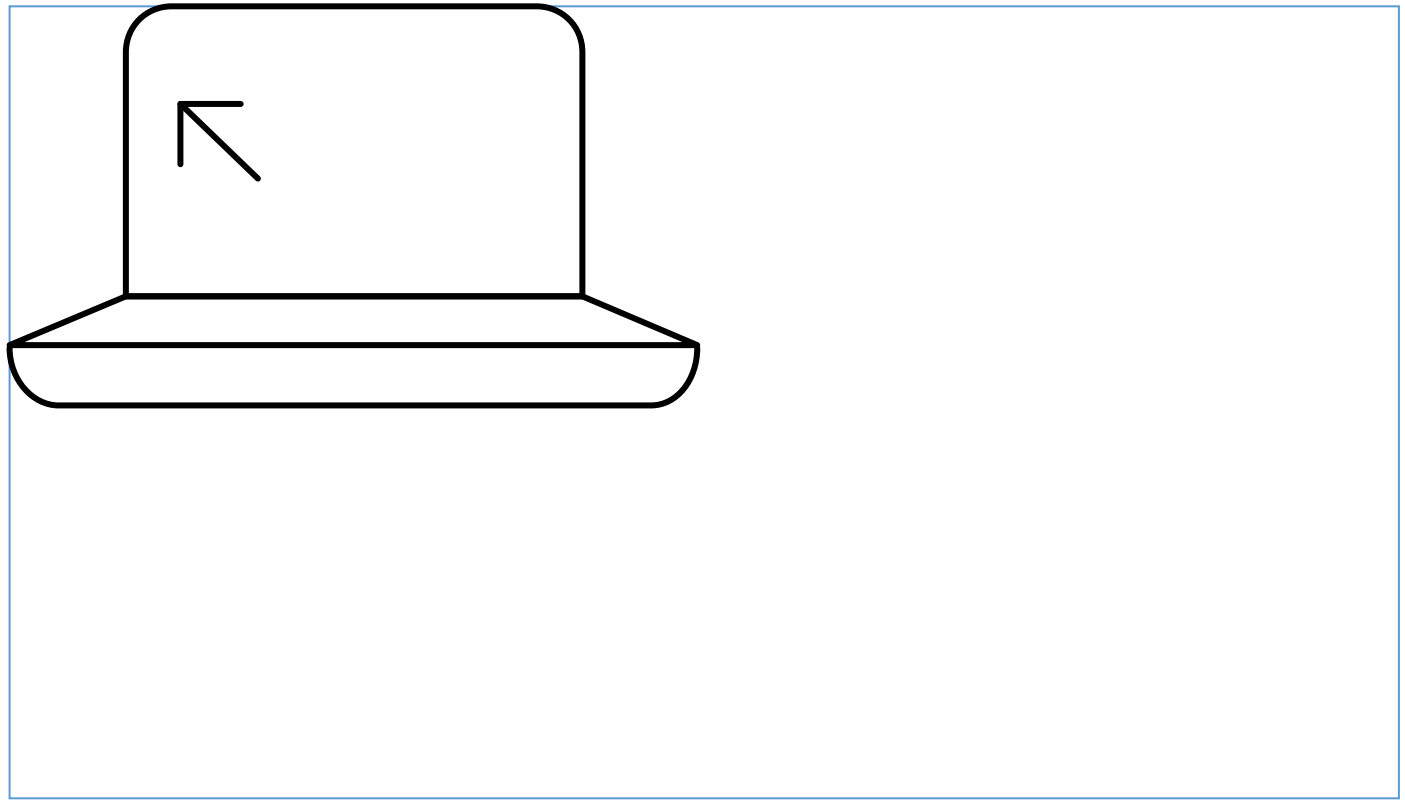
- Static Site Generation.



SSG – Static Site Generation

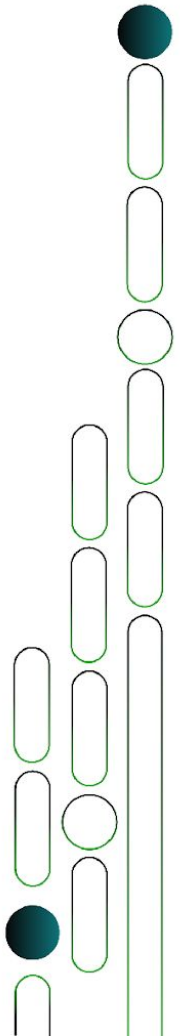


Acompanhe o professor



Próxima aula

- Single Page Application (SPA).



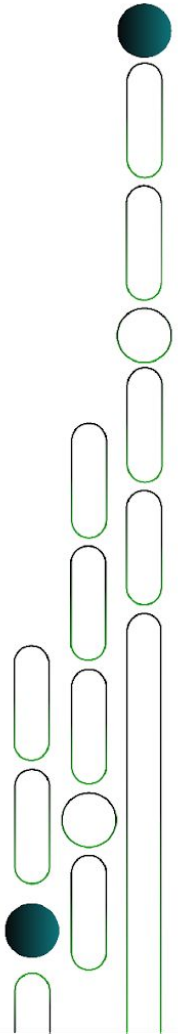
XPe
Aprenda com quem faz

React III

Capítulo 4. Estratégias de Renderização

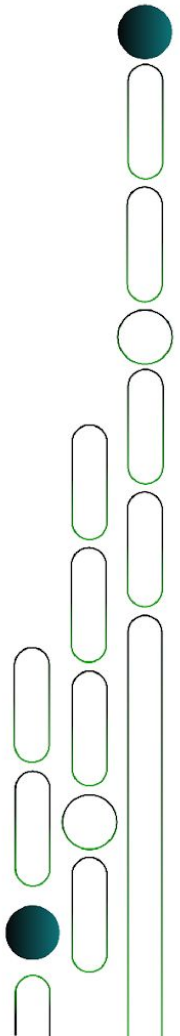
Aula 4.5. Single Page Application (SPA)

Prof. Rodrigo Borba



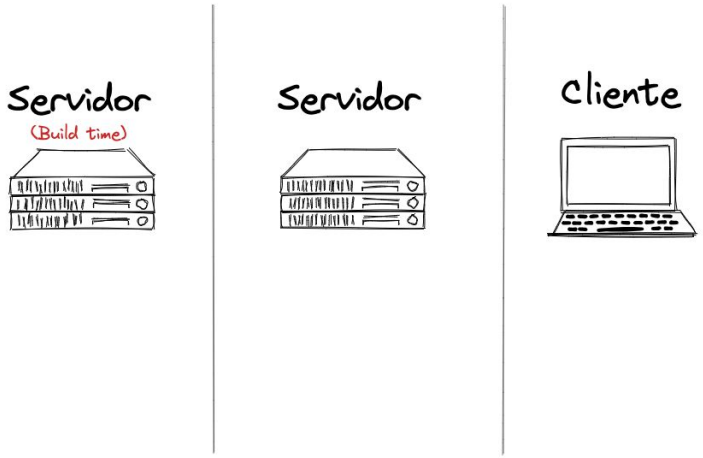
Nesta aula

- Single Page Application.

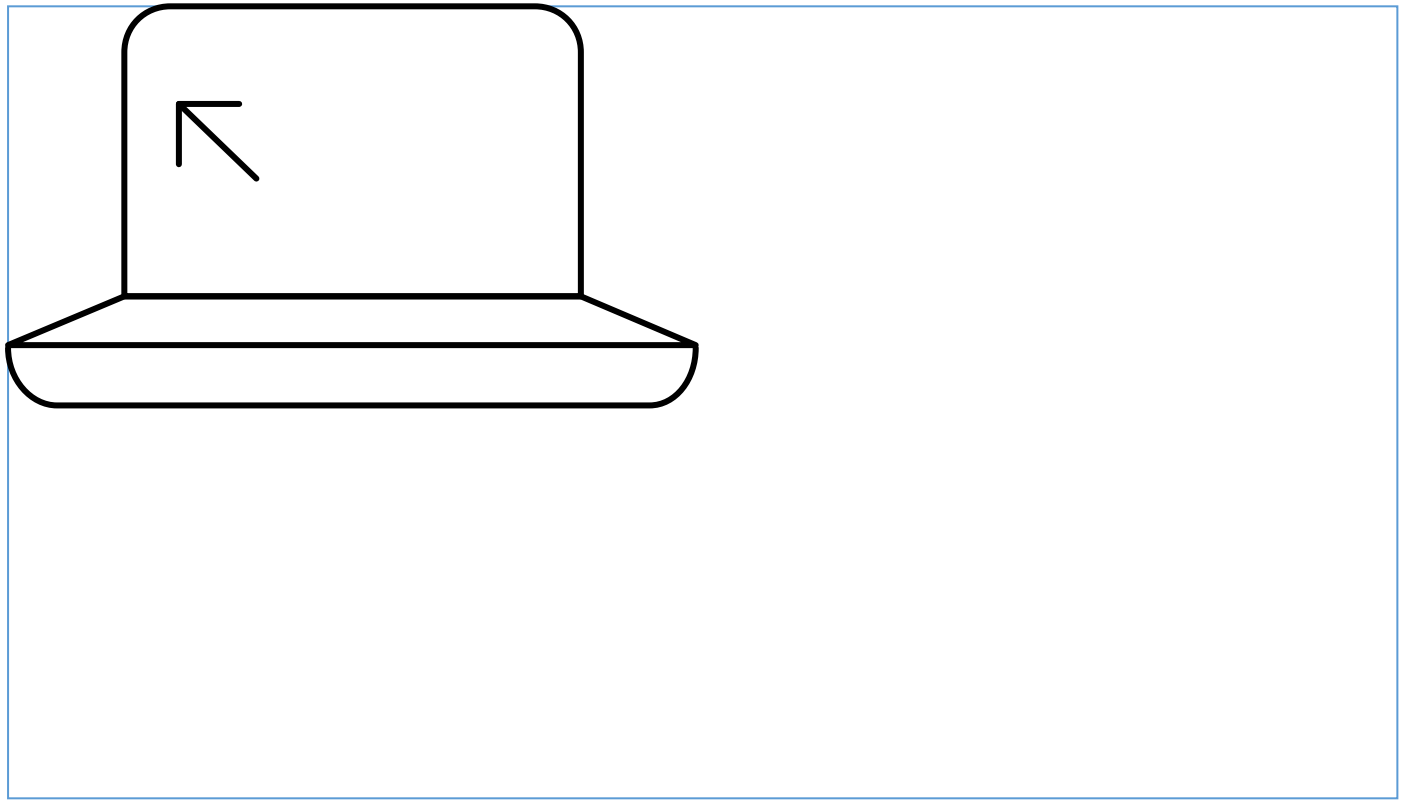


SPA – Single Page Application

O que é?

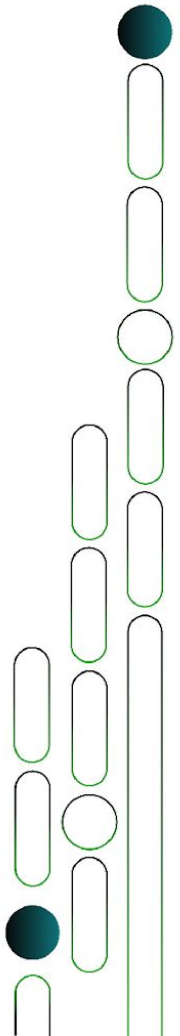


Acompanhe o professor



Próxima aula

- Next I:
  - Introdução.



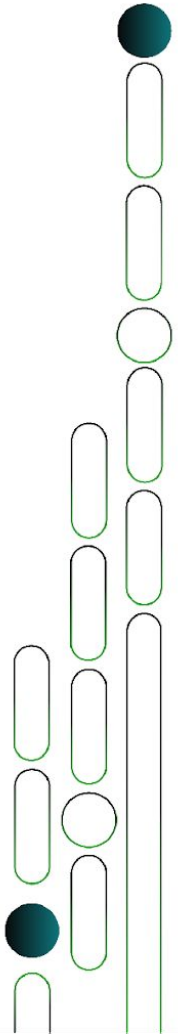
Aprenda com quem faz

React III

Capítulo 5. Next I

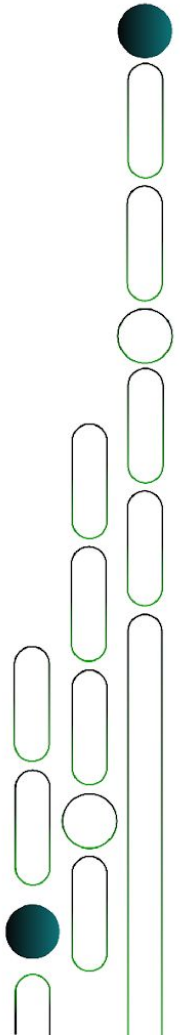
Aula 5.1. Introdução

Prof. Rodrigo Borba



Nesta aula

- Next.js.



Next.js

Next.js

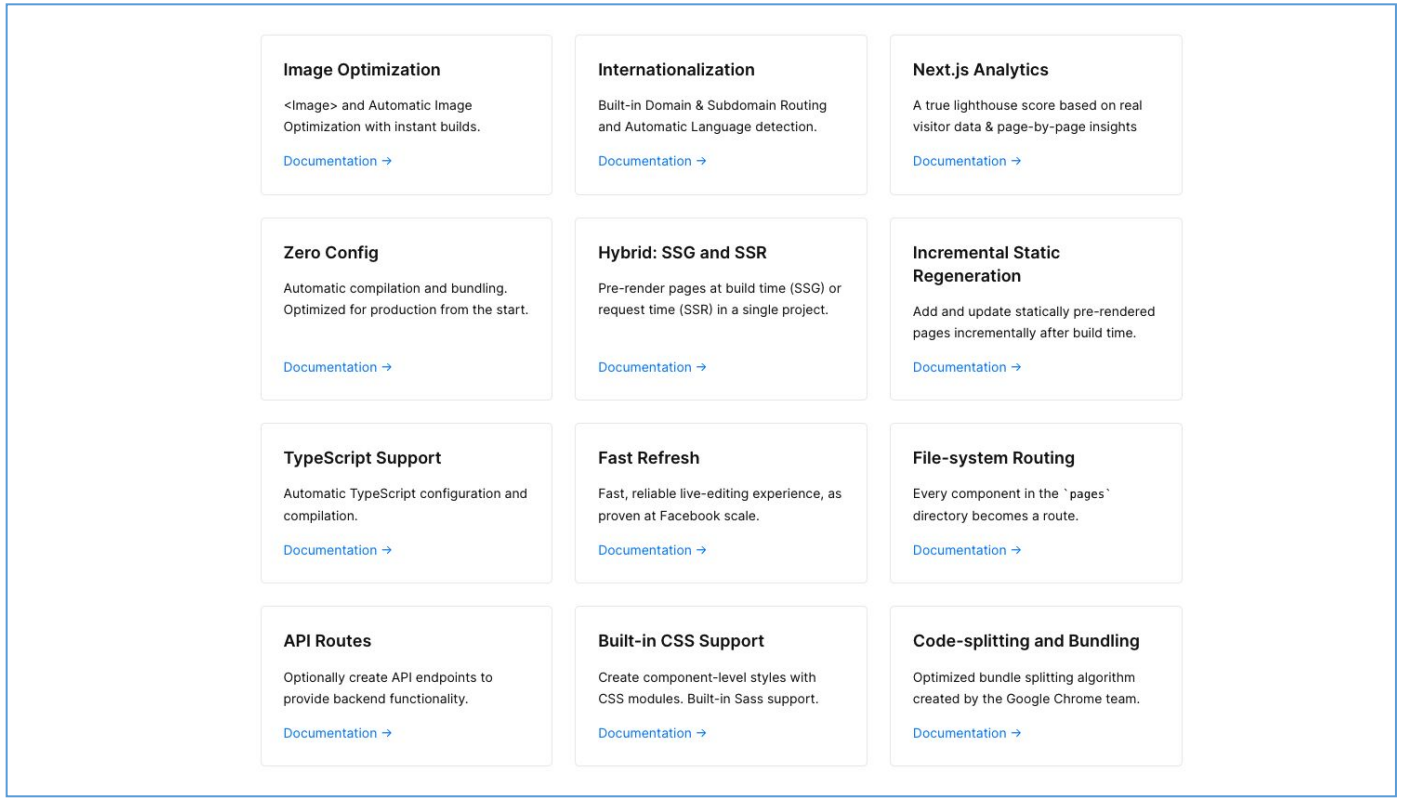


Image Optimization

<Image> and Automatic Image Optimization with instant builds.

Documentation →

Internationalization

Built-in Domain & Subdomain Routing and Automatic Language detection.

Documentation →

Next.js Analytics

A true lighthouse score based on real visitor data & page-by-page insights

Documentation →

Zero Config

Automatic compilation and bundling. Optimized for production from the start.

Documentation →

Hybrid: SSG and SSR

Pre-render pages at build time (SSG) or request time (SSR) in a single project.

Documentation →

Incremental Static Regeneration

Add and update statically pre-rendered pages incrementally after build time.

Documentation →

TypeScript Support

Automatic TypeScript configuration and compilation.

Documentation →

Fast Refresh

Fast, reliable live-editing experience, as proven at Facebook scale.

Documentation →

File-system Routing

Every component in the `pages` directory becomes a route.

Documentation →

API Routes

Optionally create API endpoints to provide backend functionality.

Documentation →

Built-in CSS Support

Create component-level styles with CSS modules. Built-in Sass support.

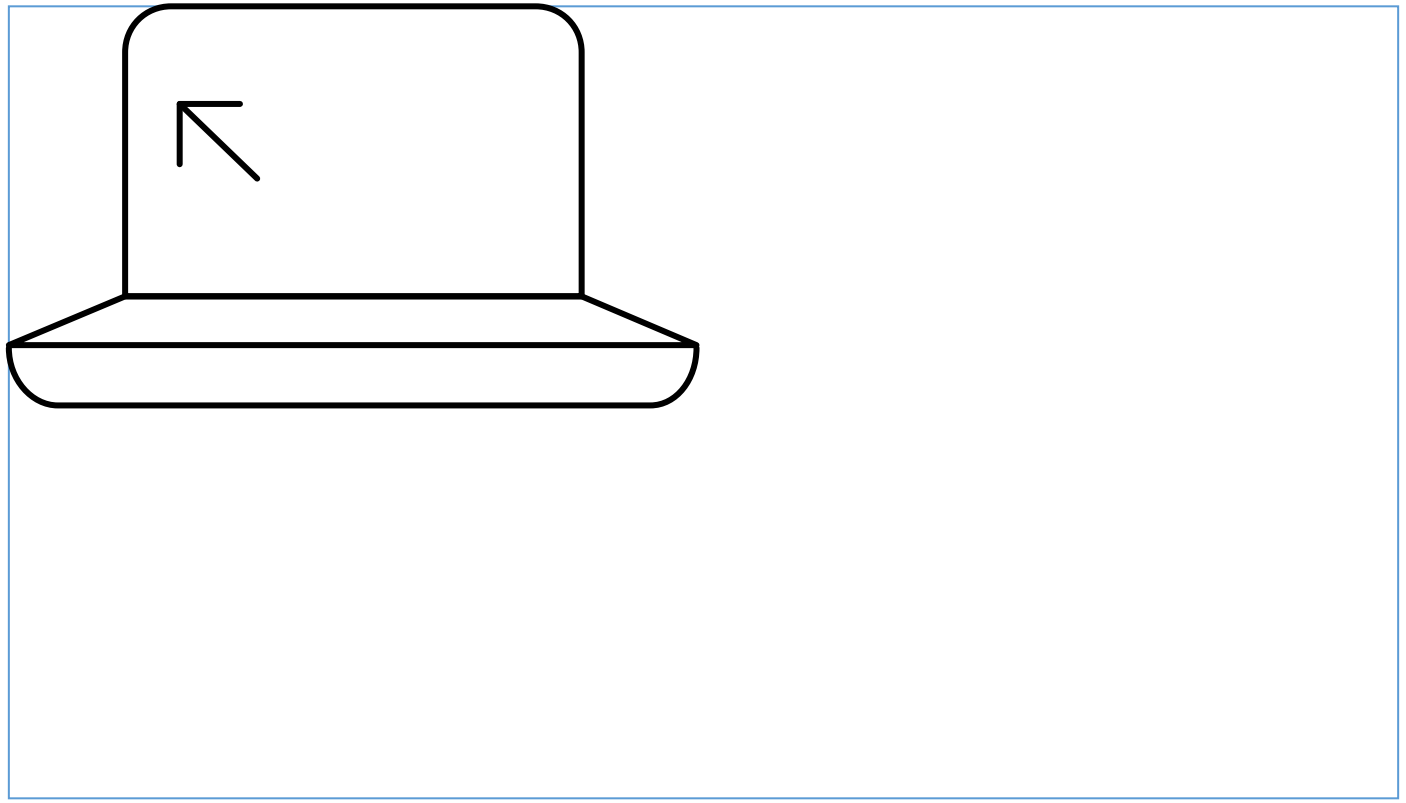
Documentation →

Code-splitting and Bundling

Optimized bundle splitting algorithm created by the Google Chrome team.

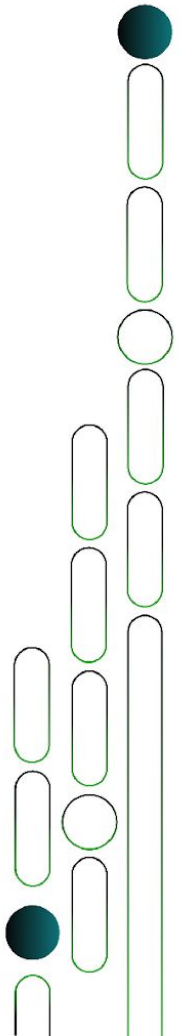
Documentation →

Acompanhe o professor



Próxima aula

- Instalação e Setup.



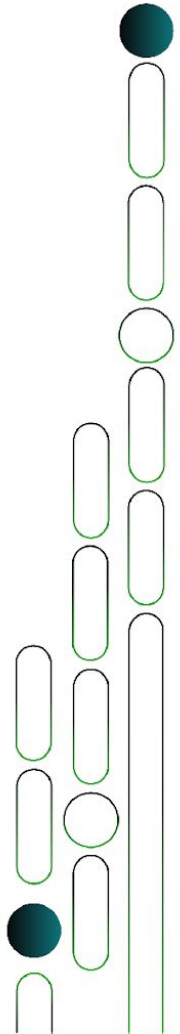
XPe
Aprenda com quem faz

React III

Capítulo 5. Next I

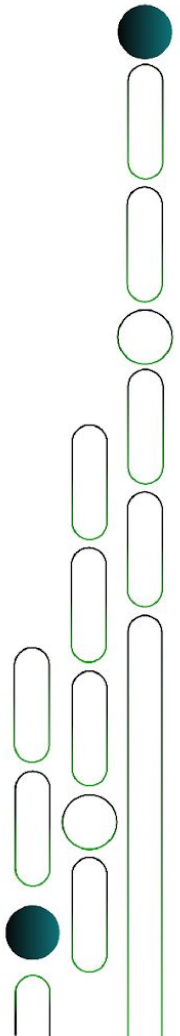
Aula 5.2. Instalação e Setup

Prof. Rodrigo Borba



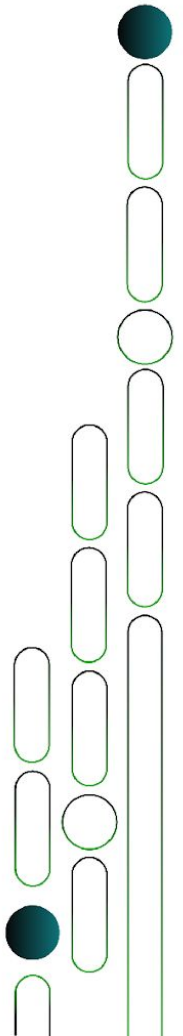
Nesta aula

- Setup e instalação.

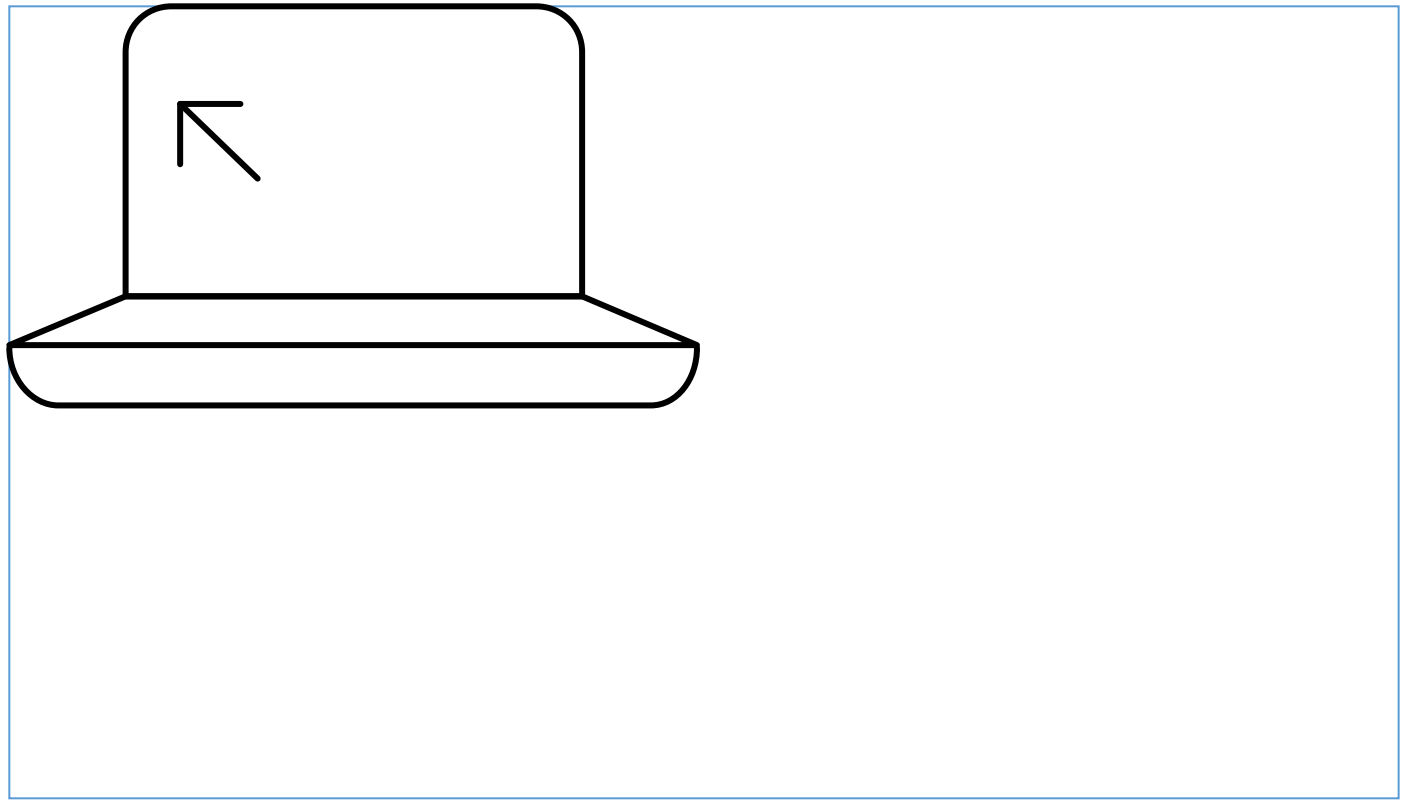


Next.js – Setup e instalação

Requisitos:
Node.js 12.22.0 ou superior
Mac, Windows ou Linux

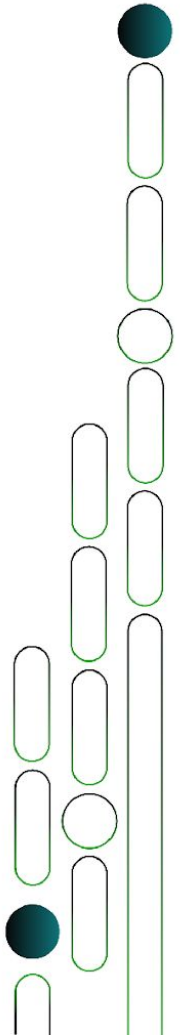


Acompanhe o professor



Próxima aula

- Rotas (pages).



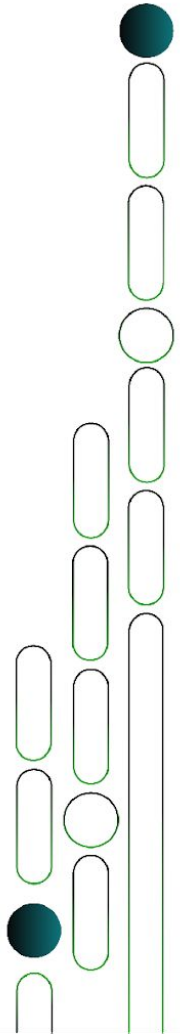
Aprenda com quem faz

React III

Capítulo 5. Next I

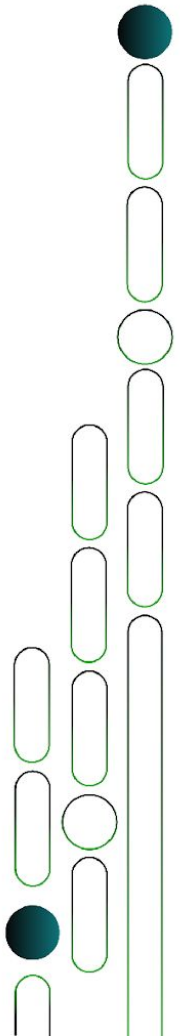
Aula 5.3. Rotas (pages)

Prof. Rodrigo Borba



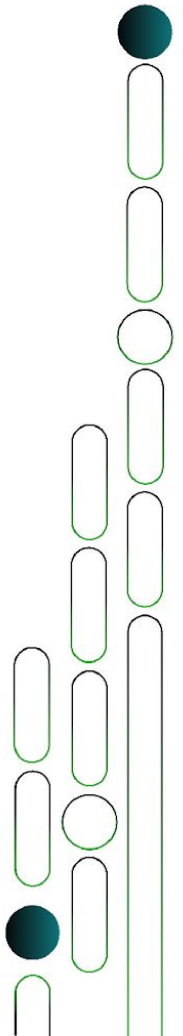
Nesta aula

- Roteamento em forma de Sistema de Arquivos (Pages).
- Rotas dinâmicas.



Next.js – Rotas

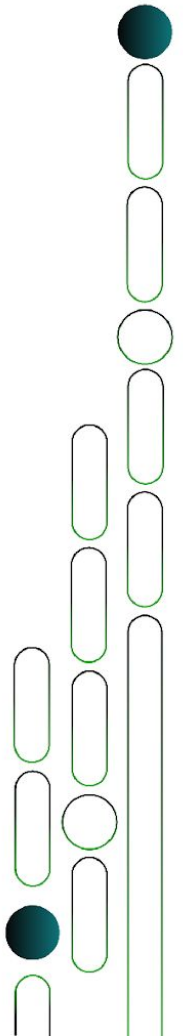
src/pages/products/index.jsx > localhost/products/
src/pages/index.jsx > localhost/
src/pages/about.jsx > localhost/about



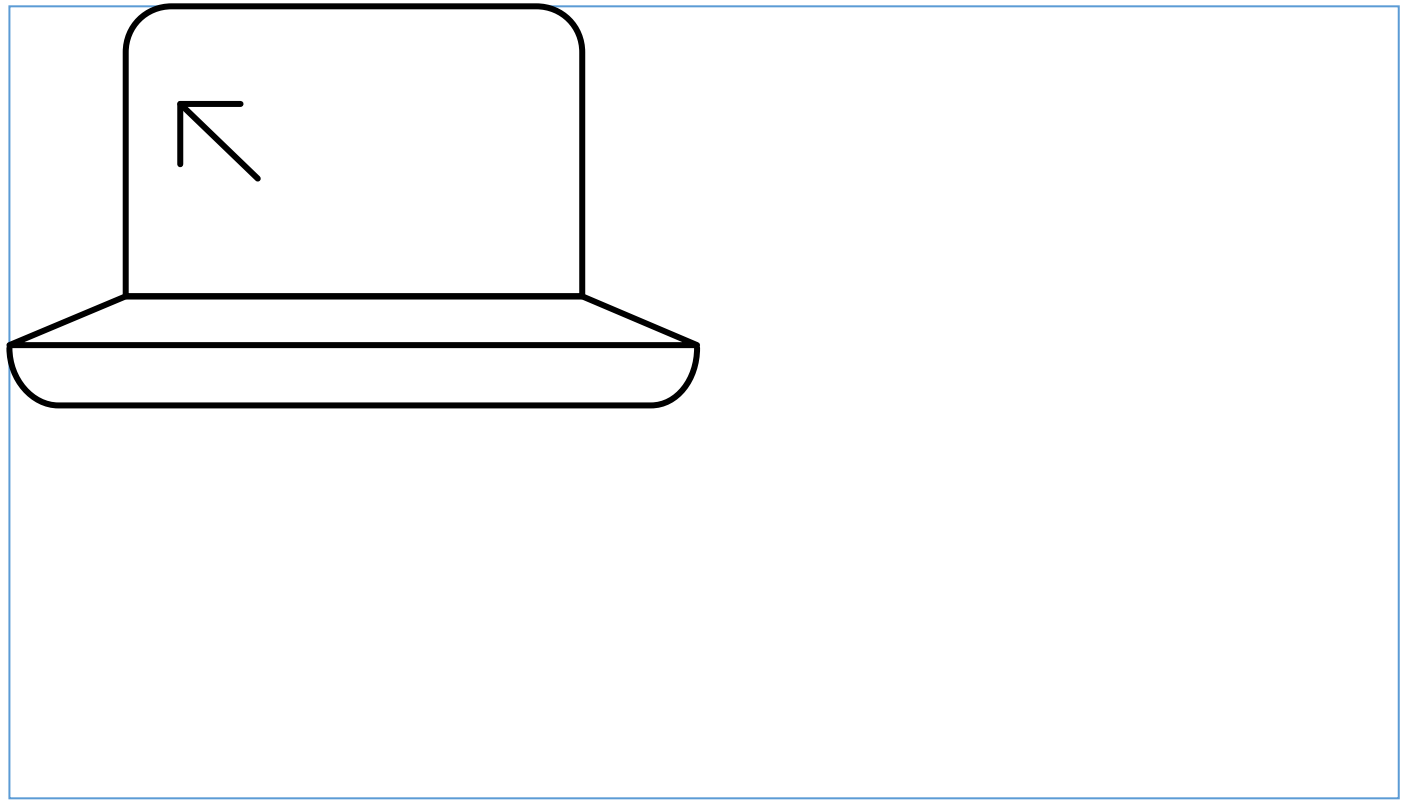
Next.js – Rotas Dinâmicas

src/pages/products/[id].jsx > localhost/products/1
localhost/products/2
localhost/products/3
Etc...

```
import { useRouter } from 'next/router'

const Product = () => {
  const router = useRouter()
  const { id } = router.query

  return <p>Product ID: {id}</p>
}

export default Product
```

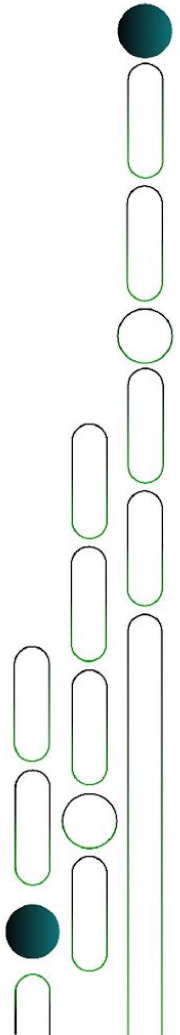


Acompanhe o professor



Próxima aula

- Pre-rendering (Parte 1).



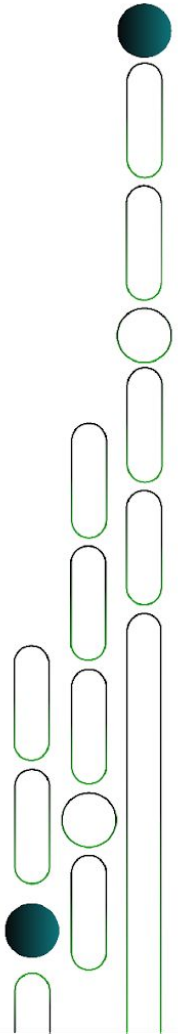
Aprenda com quem faz

React III

Capítulo 5. Next I

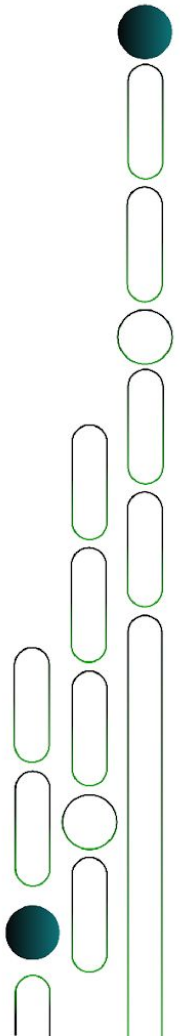
Aula 5.4.1. Pre-rendering (Parte 1)

Prof. Rodrigo Borba

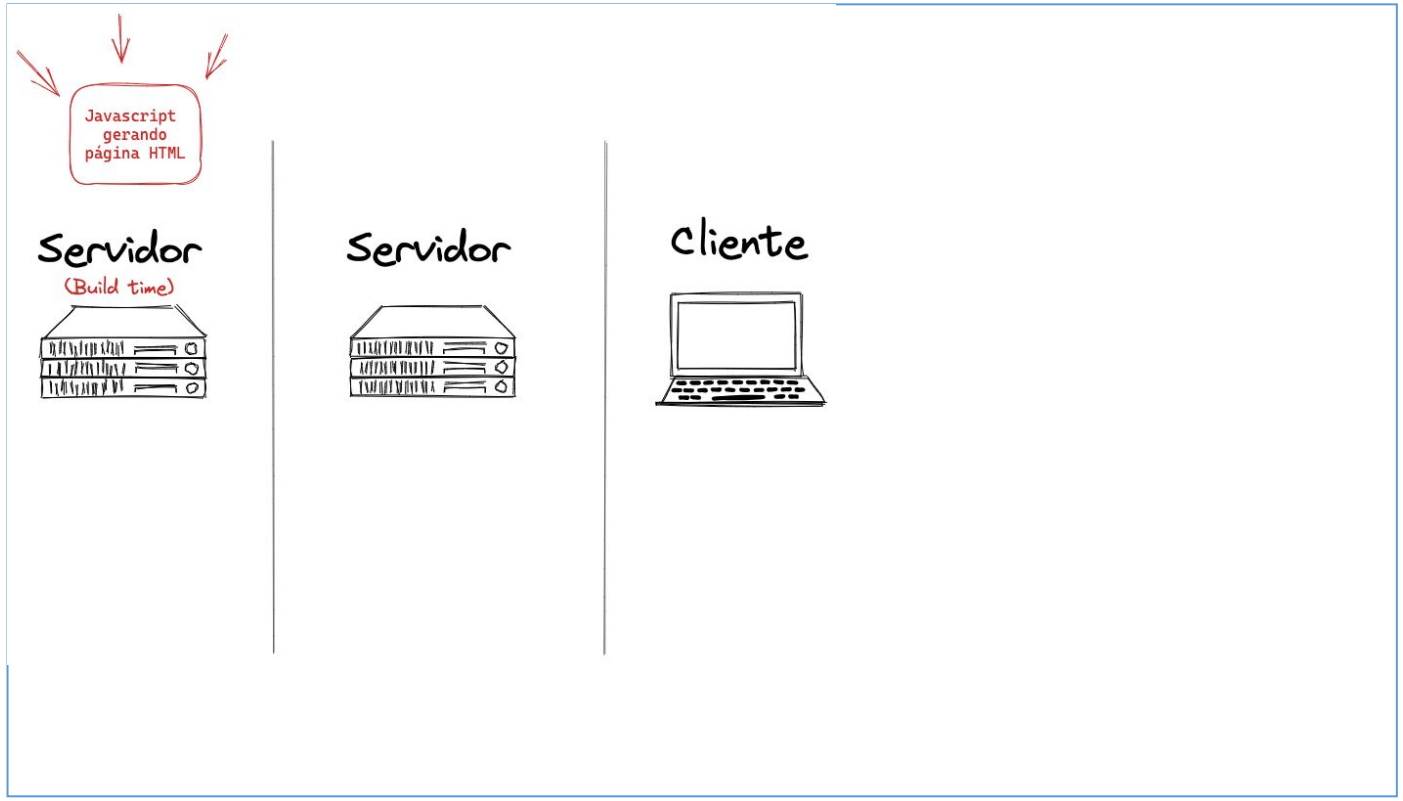


Nesta aula

- Static Generation.
- Server-side rendering.

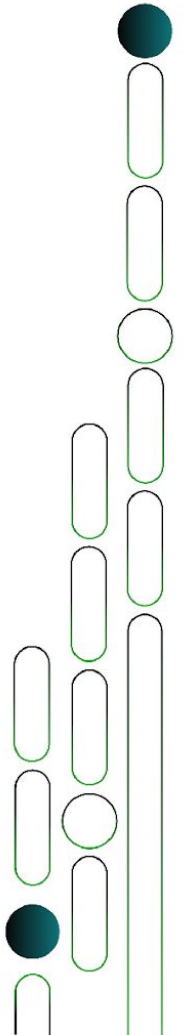


Static Generation



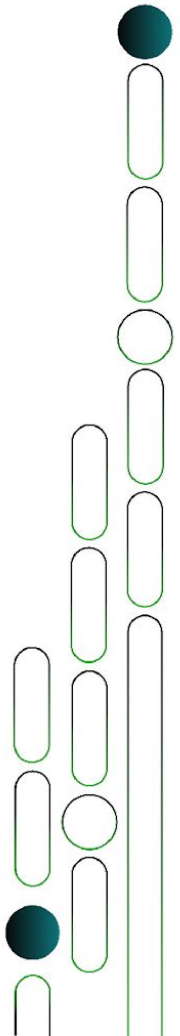
Static Generation

```
function About() {
  return <div>About</div>
}

export default About
```



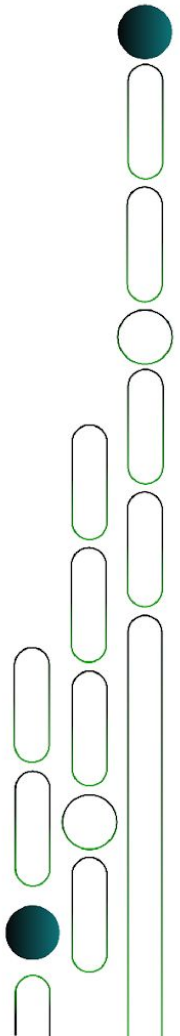
Static Generation

```
export async function getStaticProps() {
  // Chama API externa para buscar a versão
  const res = await fetch('https://.../version)
  const versionObject = await res.json()

  // Ao retornar { props: { posts } }, o componente
  // about recebe versionObject como prop
  return {
    props: {
      versionObject,
    },
  }
}
```

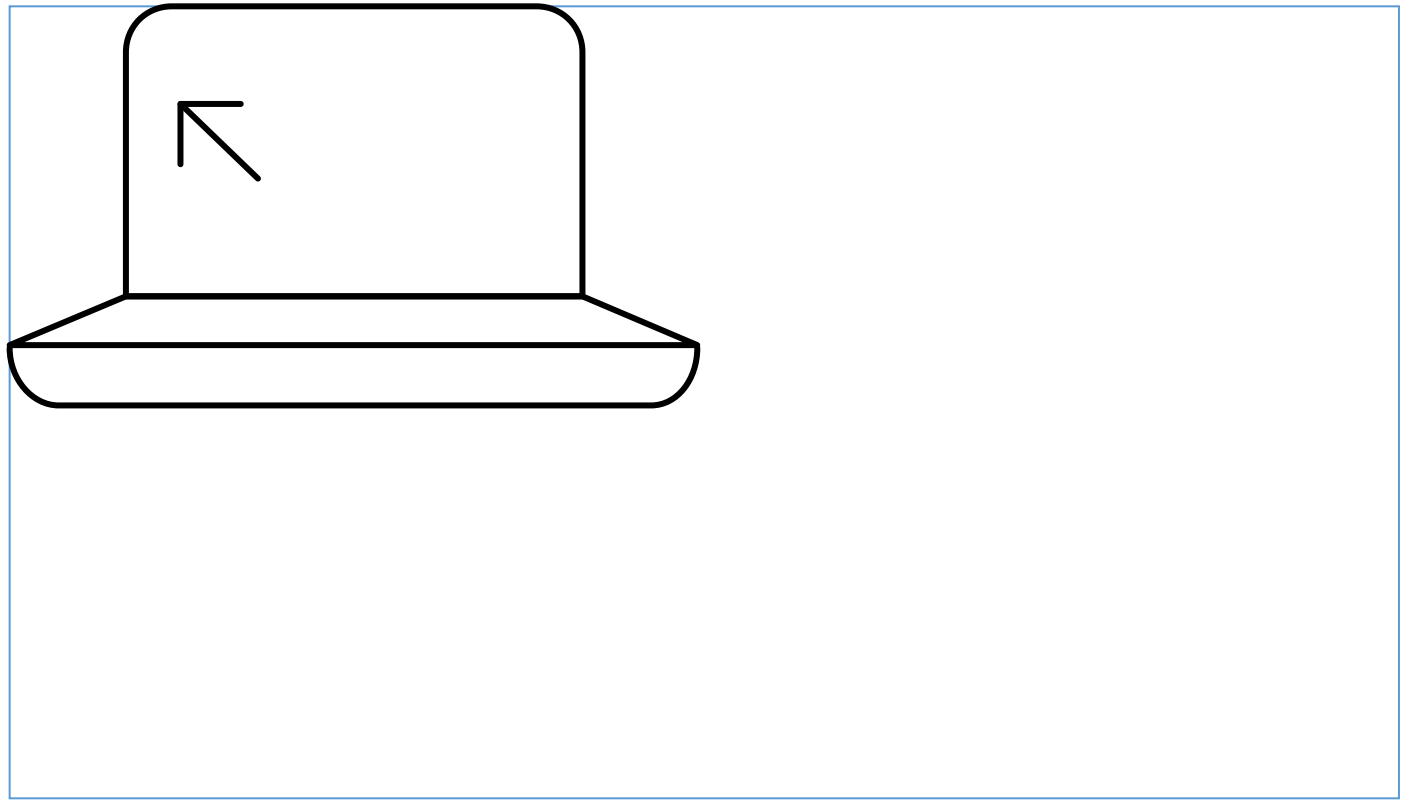


Static Generation

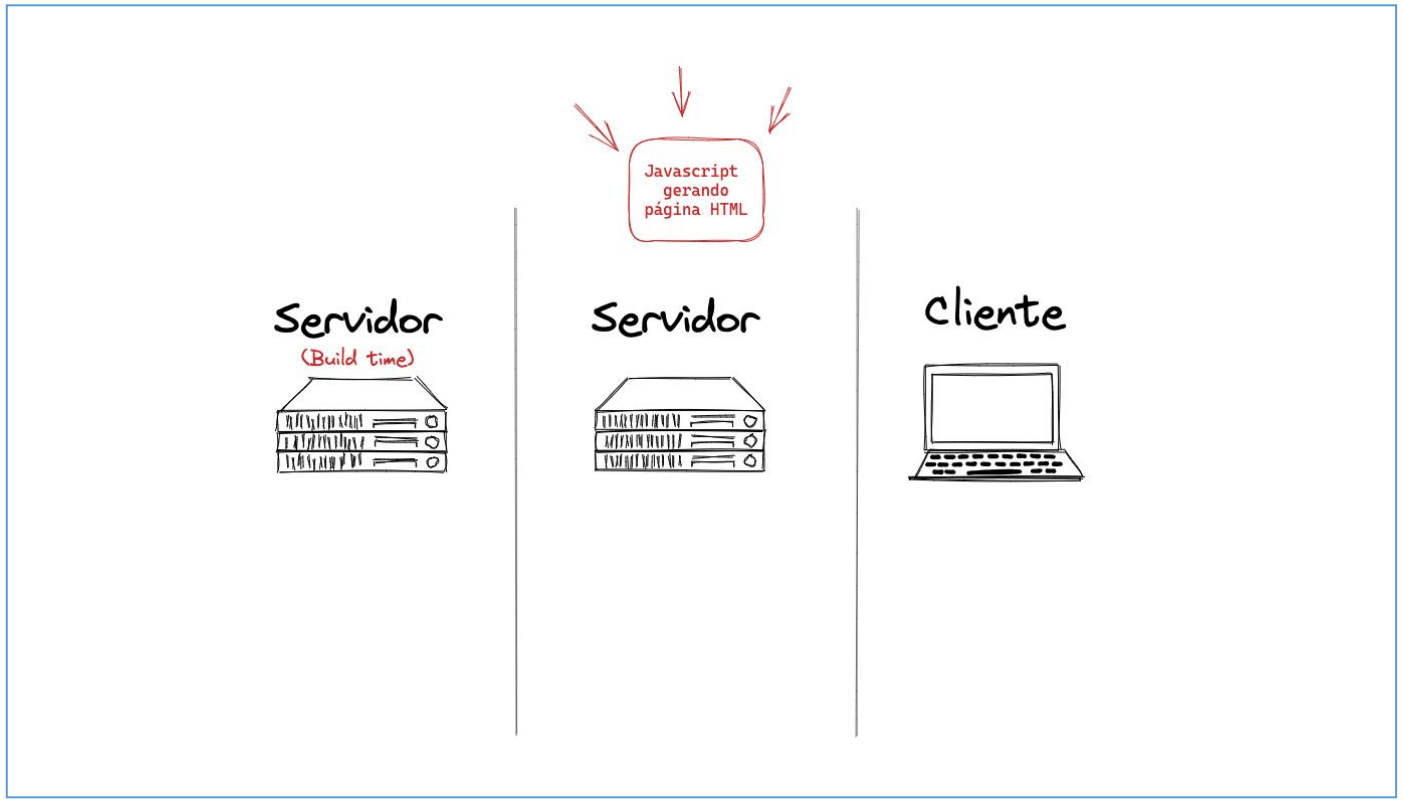
```
function About({ versionObject }) {
  return <div>About version: {versionObject}</div>
}

export default About
```



Acompanhe o professor

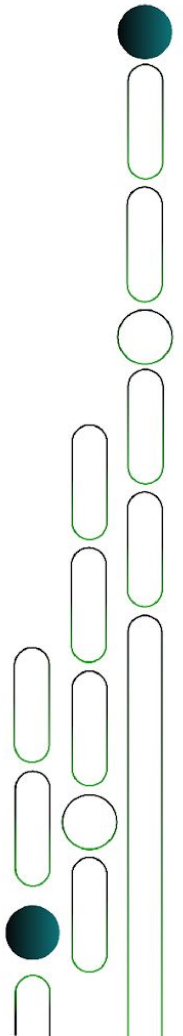


Server-side Rendering



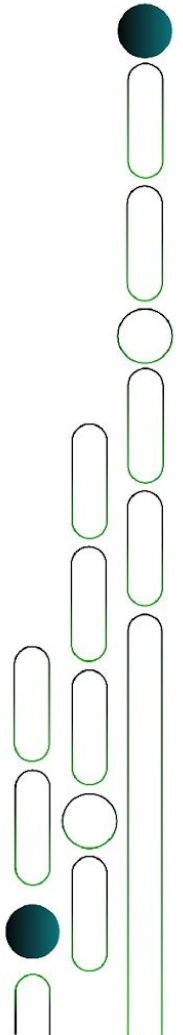
Server-side Rendering

```
// Essa função é chamada a cada request
export async function getServerSideProps() {
  // Busca informações da API
  const res = await fetch(`https://.../data`)
  const data = await res.json()

  // Passa as informações via props
  return { props: { data } }
}
```

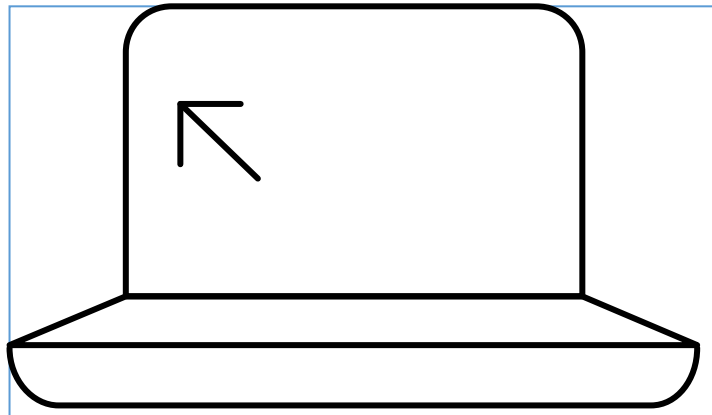


Server-side Rendering

```
function Page({ data }) {
  // Renderiza data...
}
```

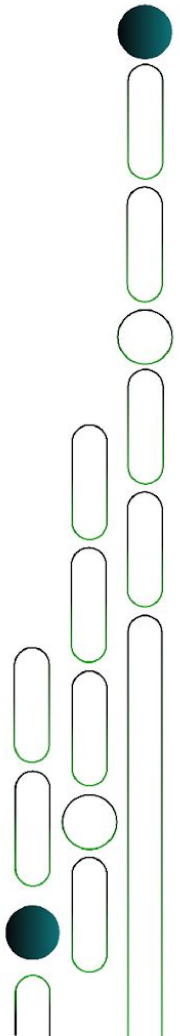


Acompanhe o professor



Próxima aula

- Pre-rendering (Parte 2).



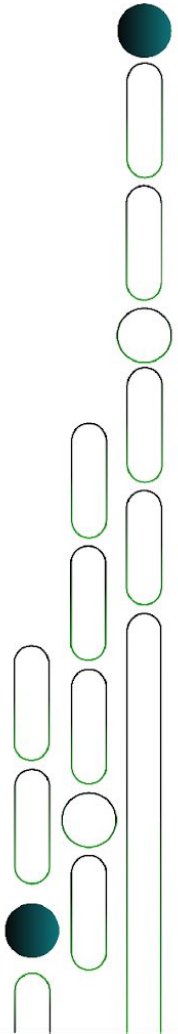
Aprenda com quem faz

React III

Capítulo 5. Next I

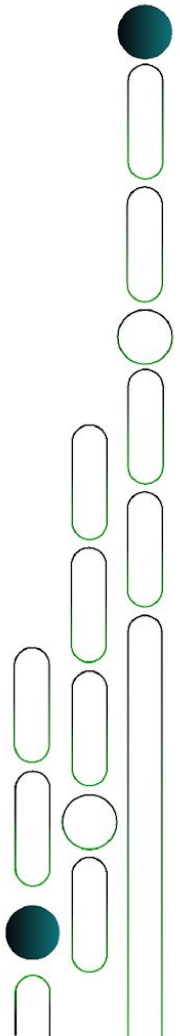
Aula 5.4.2. Pre-rendering (Parte 2)

Prof. Rodrigo Borba

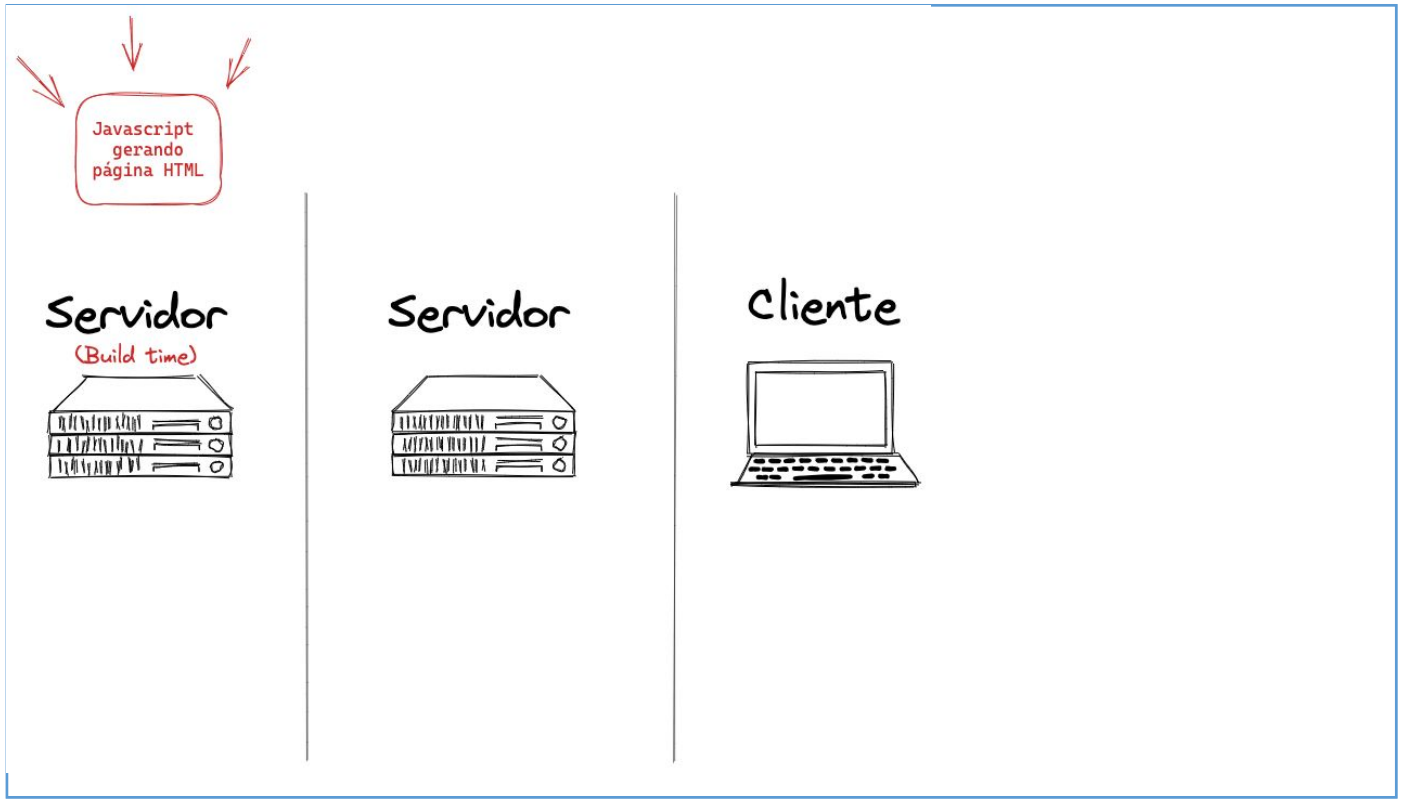


Nesta aula

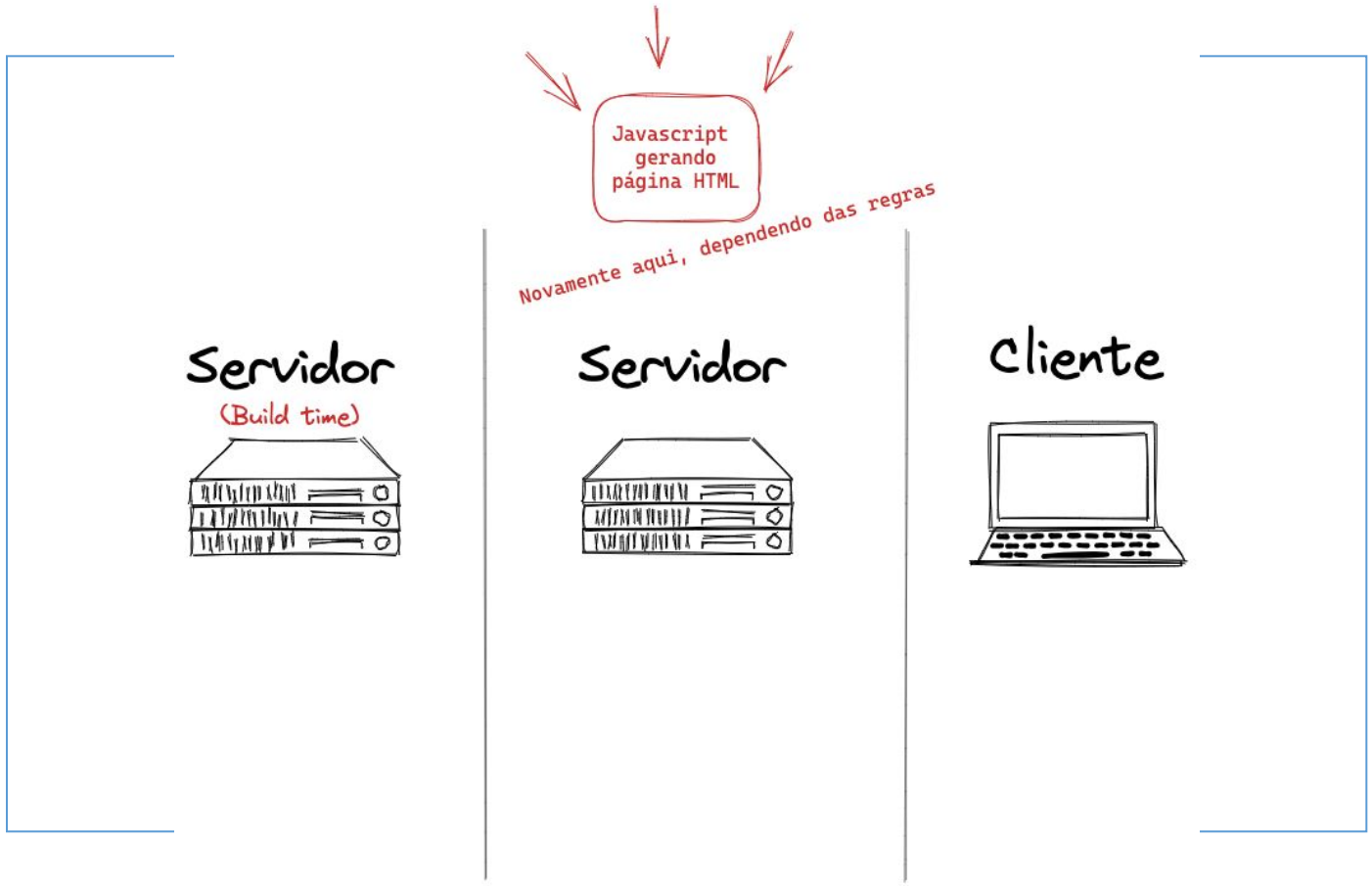
- Incremental Static Regeneration.
- Static Paths.



Incremental Static Regeneration

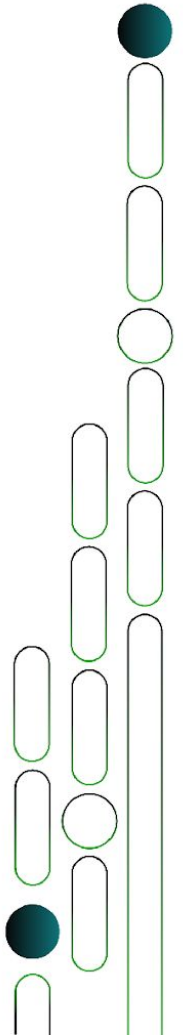


Incremental Static Regeneration



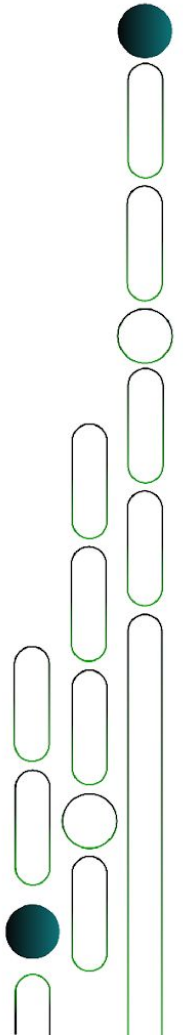
Incremental Static Regeneration

```
// Essa função é chamada em tempo de Build no lado do servidor.
export async function getStaticProps() {
  const res = await fetch('https://.../products')
  const data = await res.json()

  return {
    props: {
      data,
    },
    // Next.js irá tentar gerar novamente a página quando um novo request chegar, porém apenas se o tempo desde o último request for superior a 10 segundos
    revalidate: 10, // Em segundos
  }
}
```

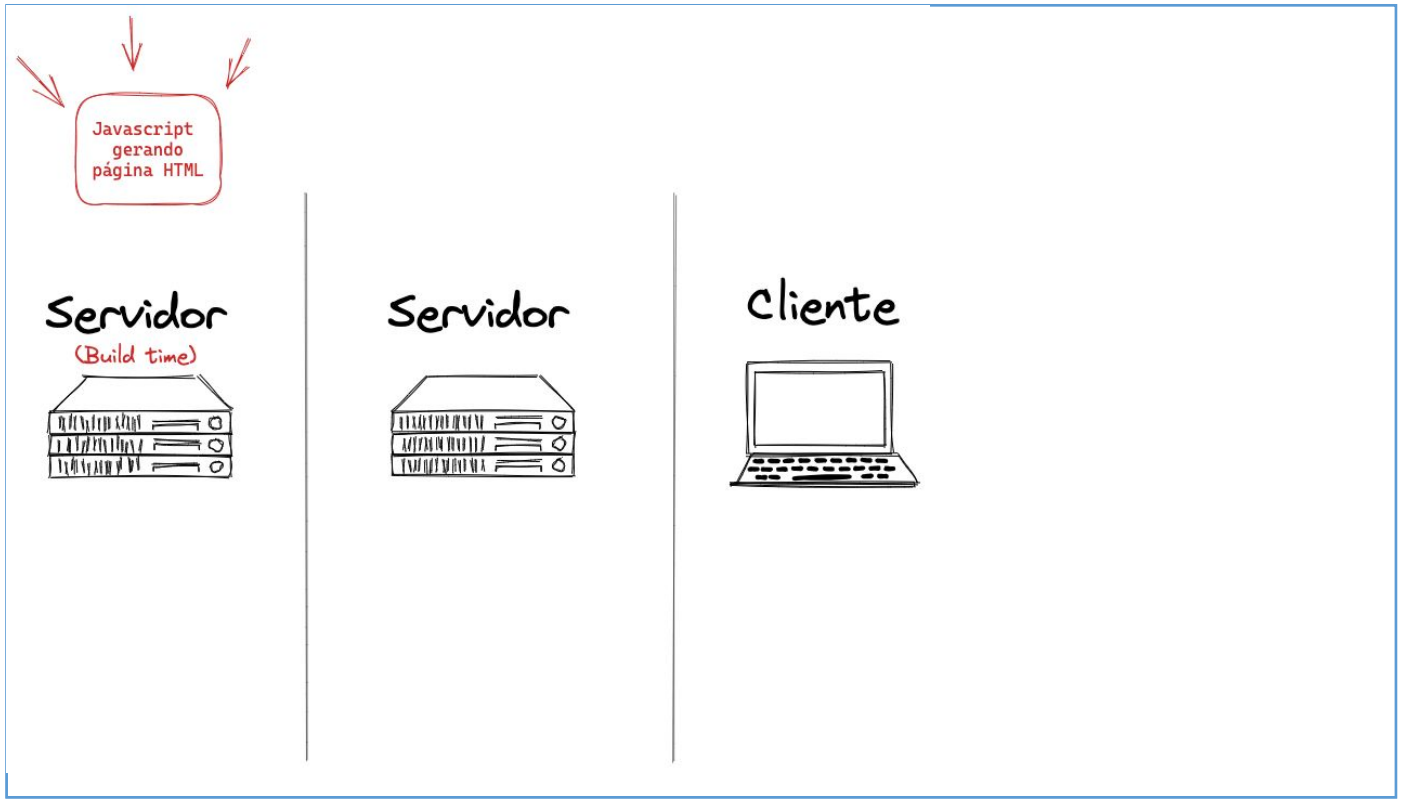


Incremental Static Regeneration

```
function Page({ data }) {
  // Renderiza data...
}
```

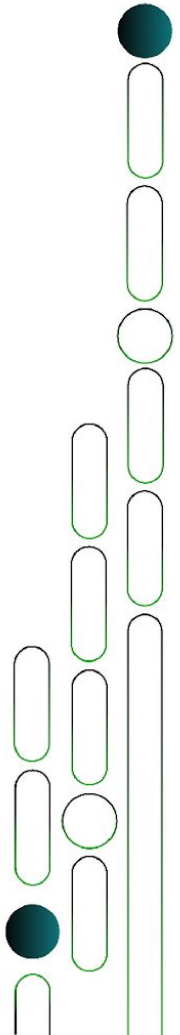


Static Paths



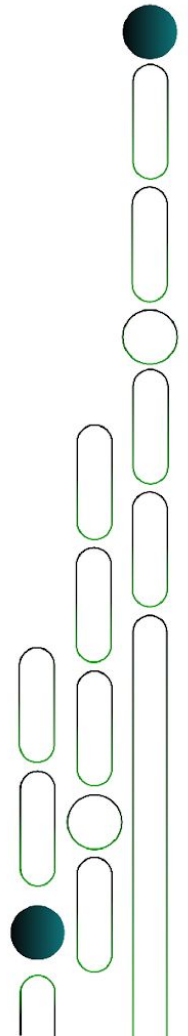
Static Paths

```
// Função chamada em tempo de build
export async function getStaticPaths() {
  // Chamamos a API externa
  const res = await fetch('https://.../products?top=10')
  const products = await res.json()

  // Gera a lista com os IDs que usaremos para gerar as páginas estáticas
  const paths = products.map((product) => ({
    params: { id: product.id },
  }))

  // { fallback: false } significa que outras rotas (além das retornadas no getStaticPaths devem retornar 404
  return { paths, fallback: false }
}
```

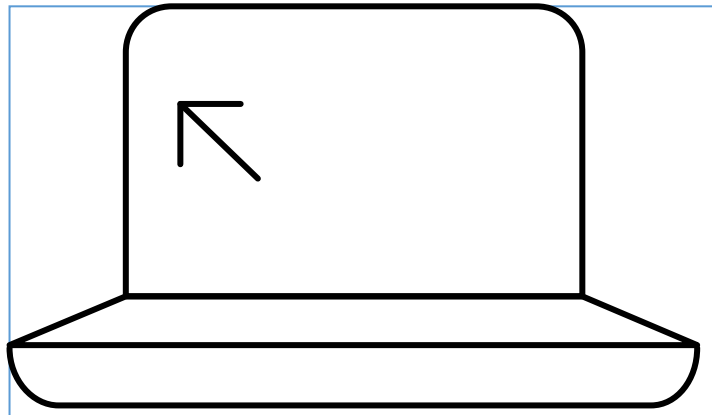


Static Paths

```
// Essa função é chamada em tempo de Build no lado do servidor. Por conta do getStaticPaths, será chamada UMA VEZ para cada PATH descrito no array “paths”.
export async function getStaticProps({params}) {
  const res = await fetch(`https://.../products/${params.id}`)
  const data = await res.json()

  return {
    props: {
      data,
    }
}
```

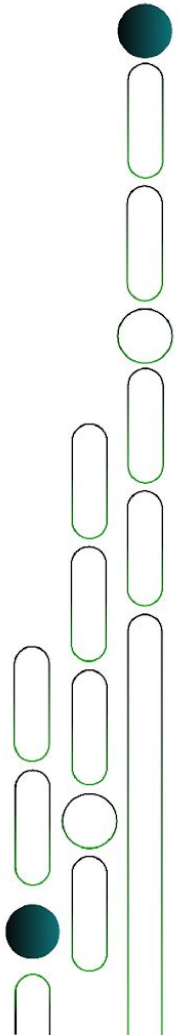


Acompanhe o professor



Próxima aula

- Next II.
  - Optimização de imagens / Internacionalização / Environment Variables.



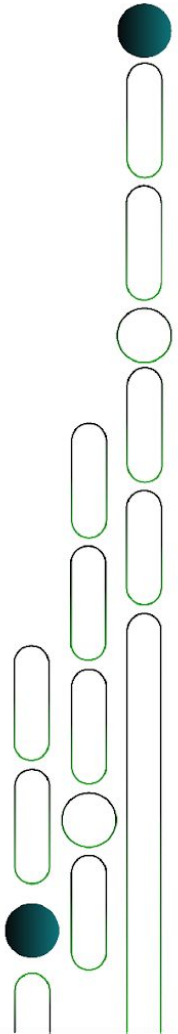
XPe
Aprenda com quem faz

React II

Capítulo 6. Next II

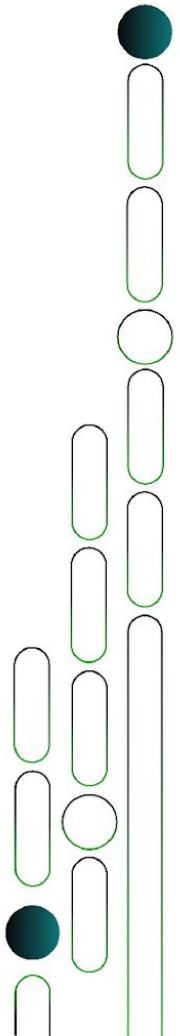
Aula 6.1. Optimização de imagens / Internacionalização / Environment Variables

Prof. Rodrigo Borba



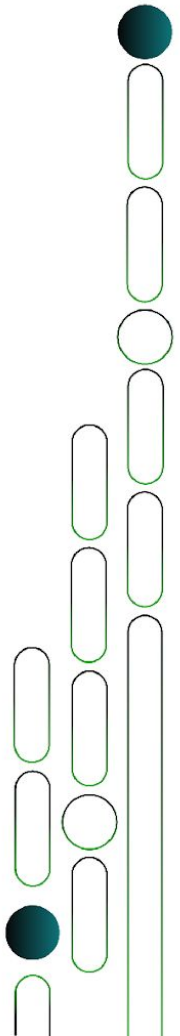
Nesta aula

- Next/image.
- Internationalized Routing.
- Environment Variables.



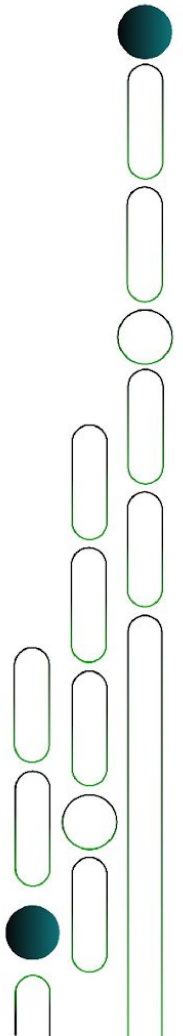
Next/image

```
<img>++

import Image from 'next/image'

<Image
    src={profilePic}
     alt="Picture of the author"
    />
```

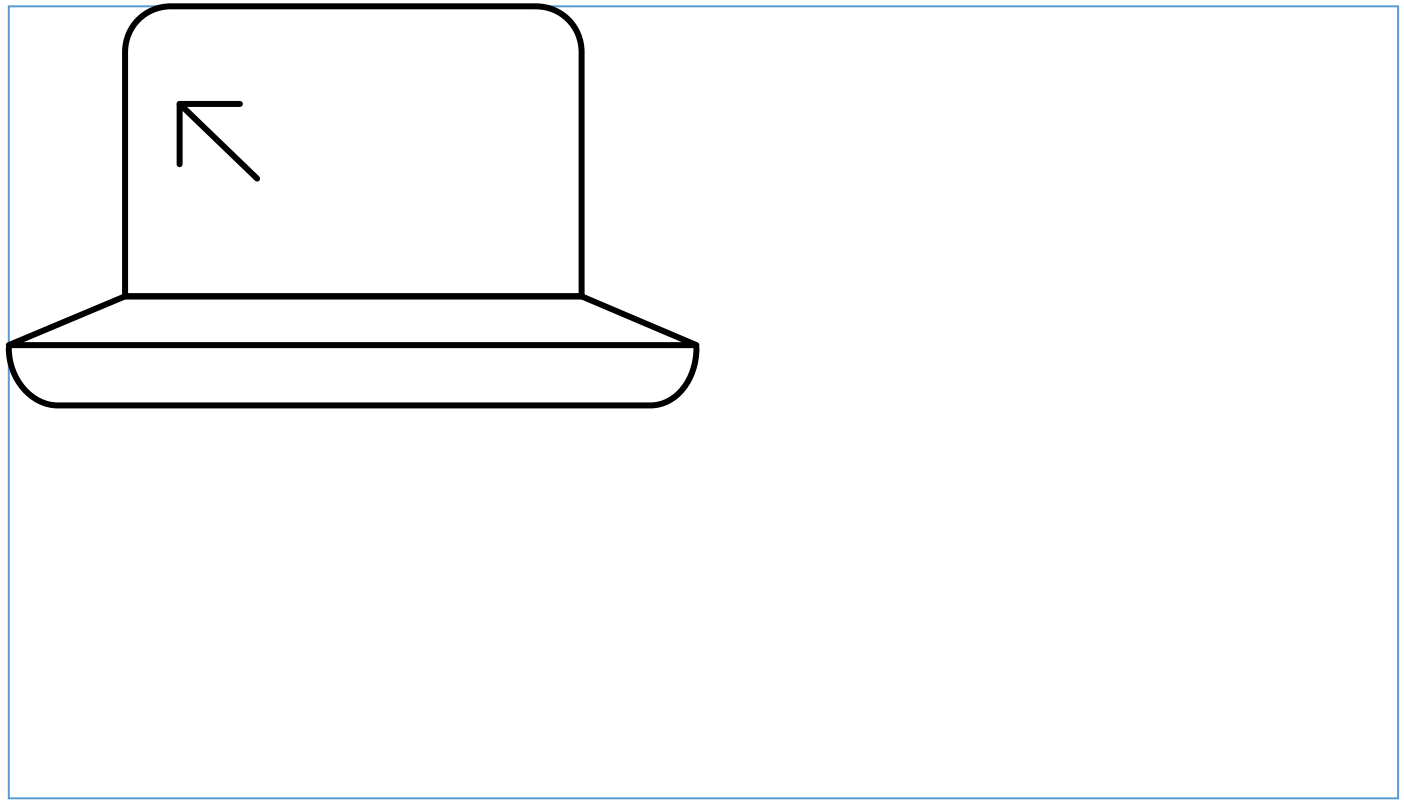


Vantagens

- Performance melhorada.
- Estabilidade visual.
- Carregamentos mais rápidos das páginas.
- Flexibilidade de assets.



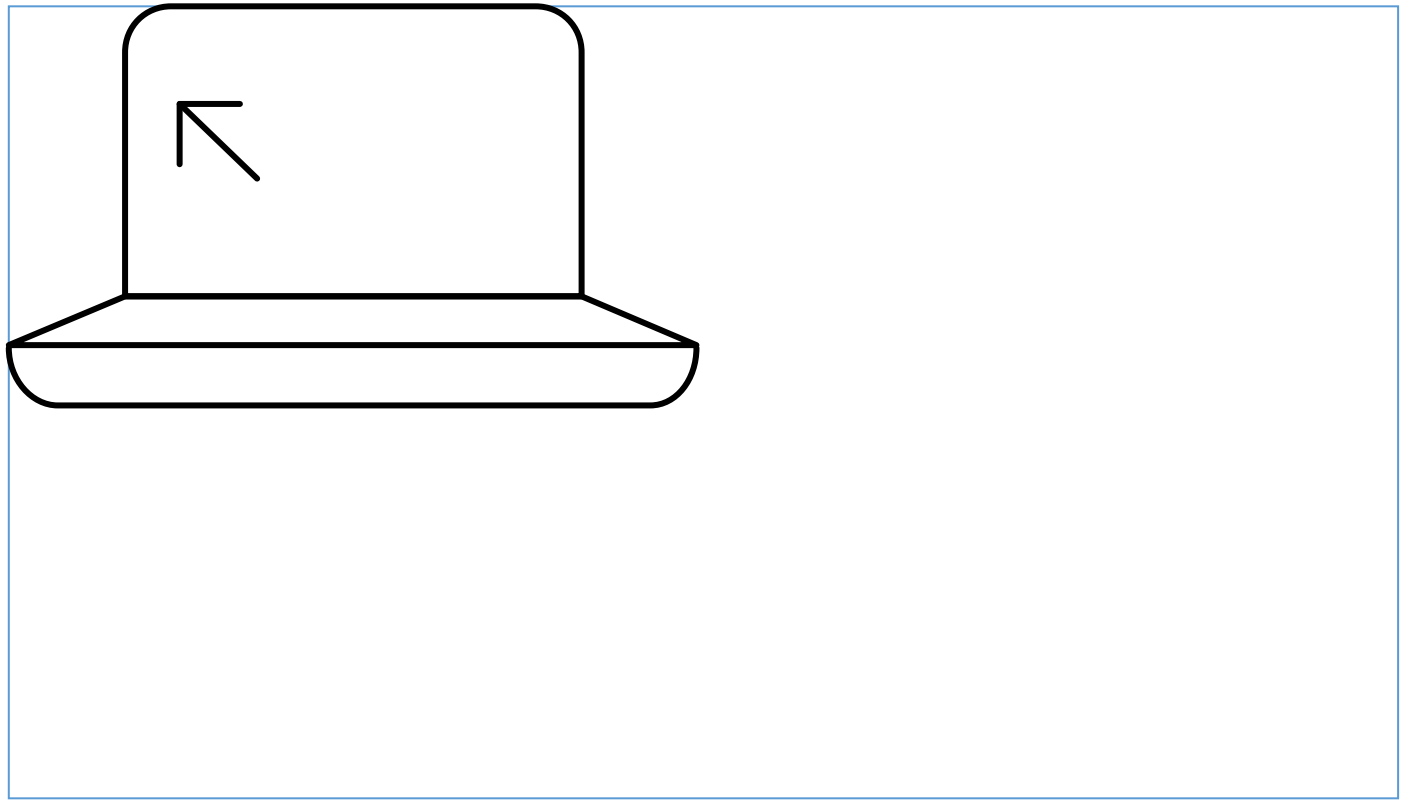
Acompanhe o professor



Internacionalização

- **i18n**

Acompanhe o professor



Environment Variables

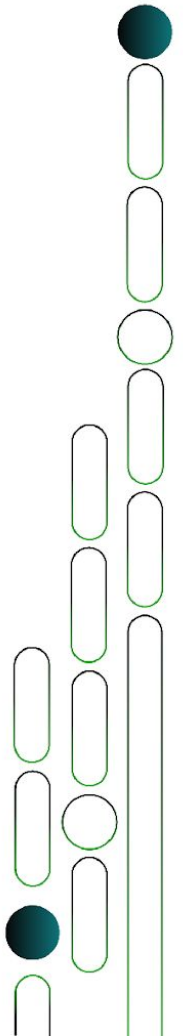
.env.local

- Server only

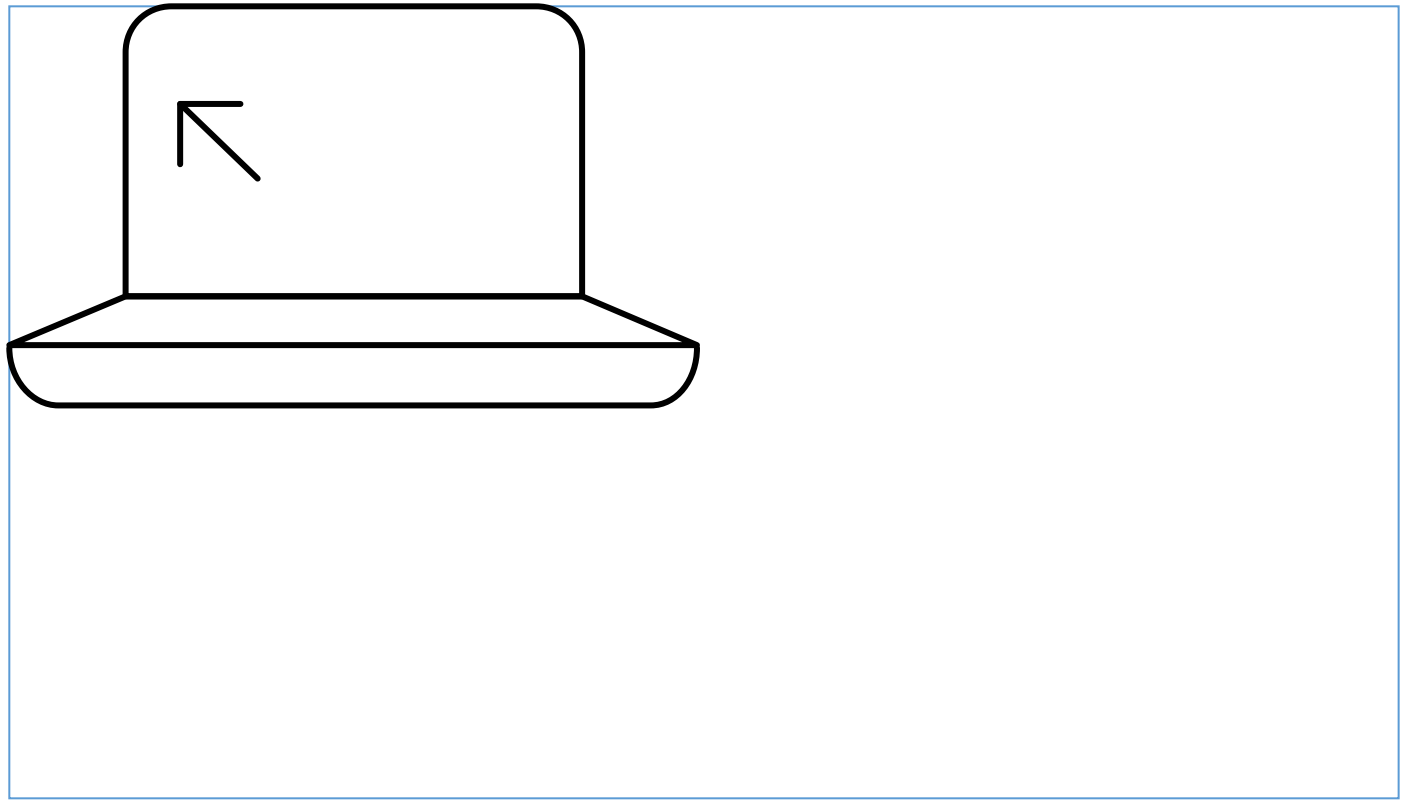
DB_HOST=localhost
DB_USER=myuser
DB_PASS=mypassword

- **Browser**

NEXT_PUBLIC_{{VAR_NAME}}

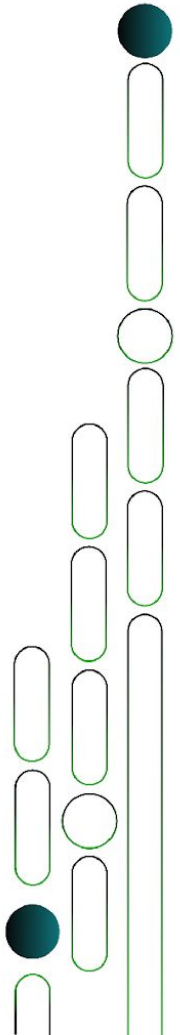


Acompanhe o professor



Próxima aula

- Font Optimization / Static File Serving / Fast refresh.



Aprenda com quem faz

React II

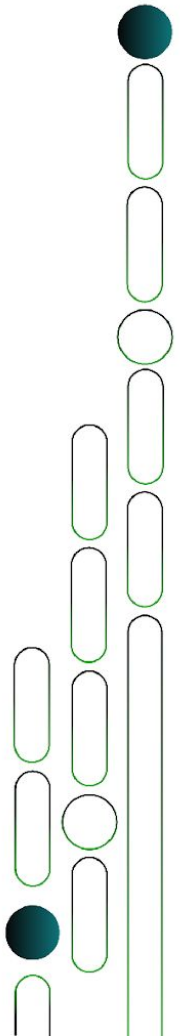
Capítulo 6. Next II

Aula 6.2. Font Optimization / Static File Serving / Fast refresh

Prof. Rodrigo Borba

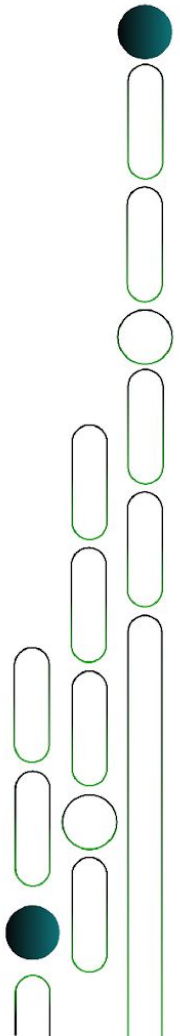
Nesta aula

- Font Optimisation.
- Static File Serving.
- Fast Refresh.



Font Optimization

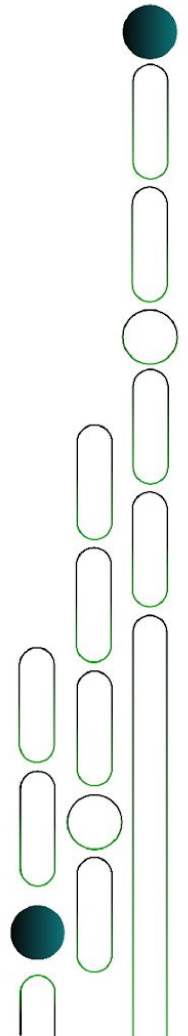
```
import Head from 'next/head'

export default function IndexPage() {
  return (
   <div>
     <Head> <<< Override aqui!
       <link
            href="https://fonts.googleapis.com/css2?family=Inter&display=optional"
         rel="stylesheet"
       />
     </Head>
     <p>Hello world!</p>
   </div>
  )
}
```



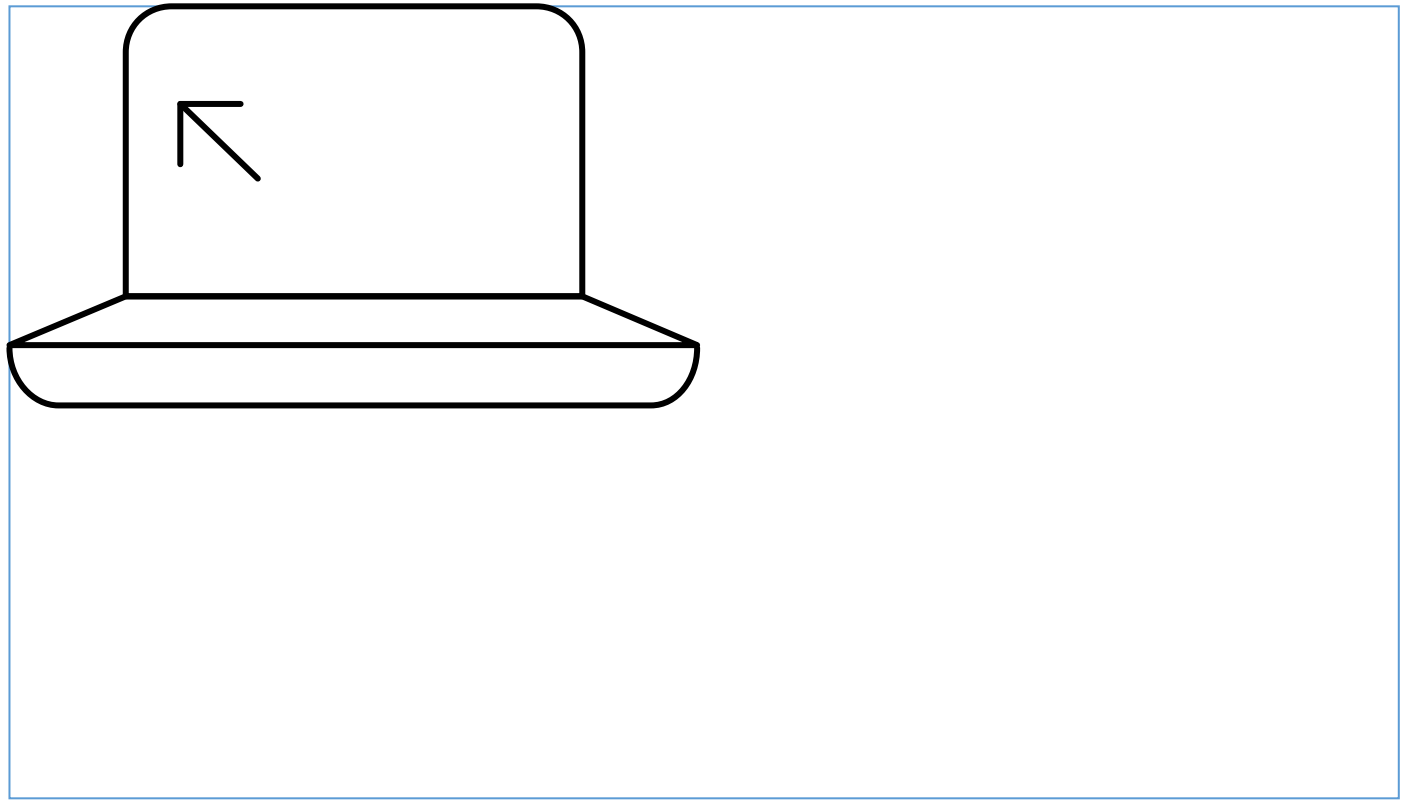
Static File Serving

```
import Image from 'next/image'

function Avatar() {
  return <Image src="/me.png" alt="me" width="64" height="64" />
}

export default Avatar
```

/me.png está disponível em /public/

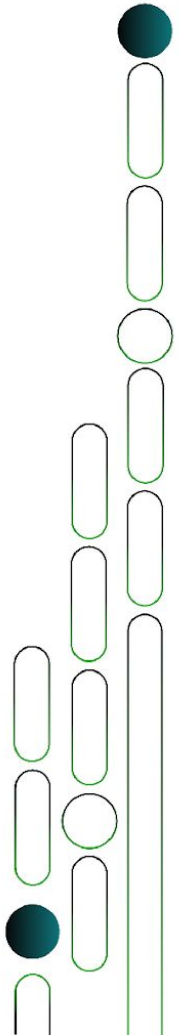


Acompanhe o professor



Próxima aula

- Testando aplicações React I:
  - Introdução a conceitos básicos de teste.



Aprenda com quem faz

React II

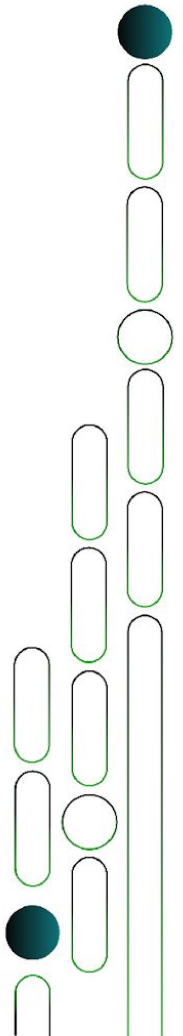
Capítulo 7. Testando aplicações React I

Aula 7.1. Introdução a conceitos básicos de teste

Prof. Rodrigo Borba

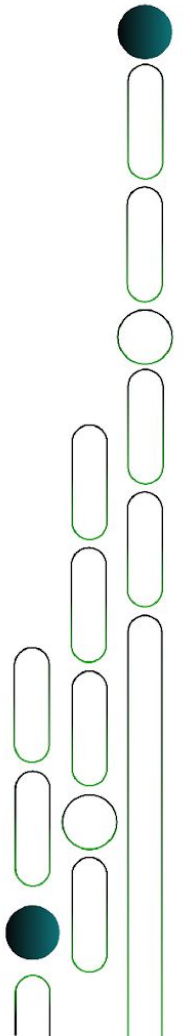
Nesta aula

- Como desenvolvedor, por que criar testes?
- Custo de bugs.
- Testando aplicações WEB.



Como Dev, Por que testar

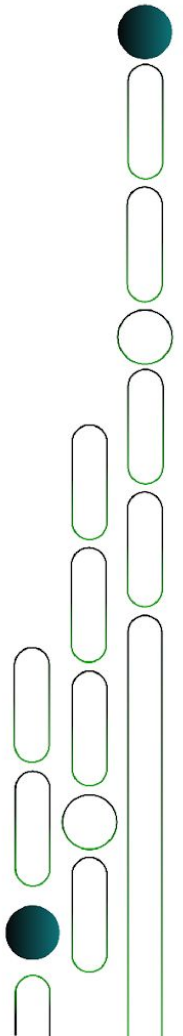
- Documentação.
- Confiança para refatorar.
- Erros percebidos mais cedo.
- Código desacoplado por consequência.



Custo dos Bugs

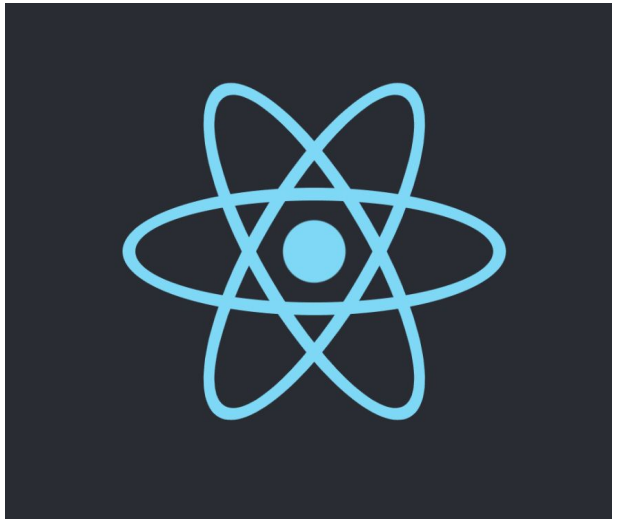
Testando aplicações Web

- Testes de regressão.
- Testes de cross-browser.
- Testes de performance.
- Testes funcionais.

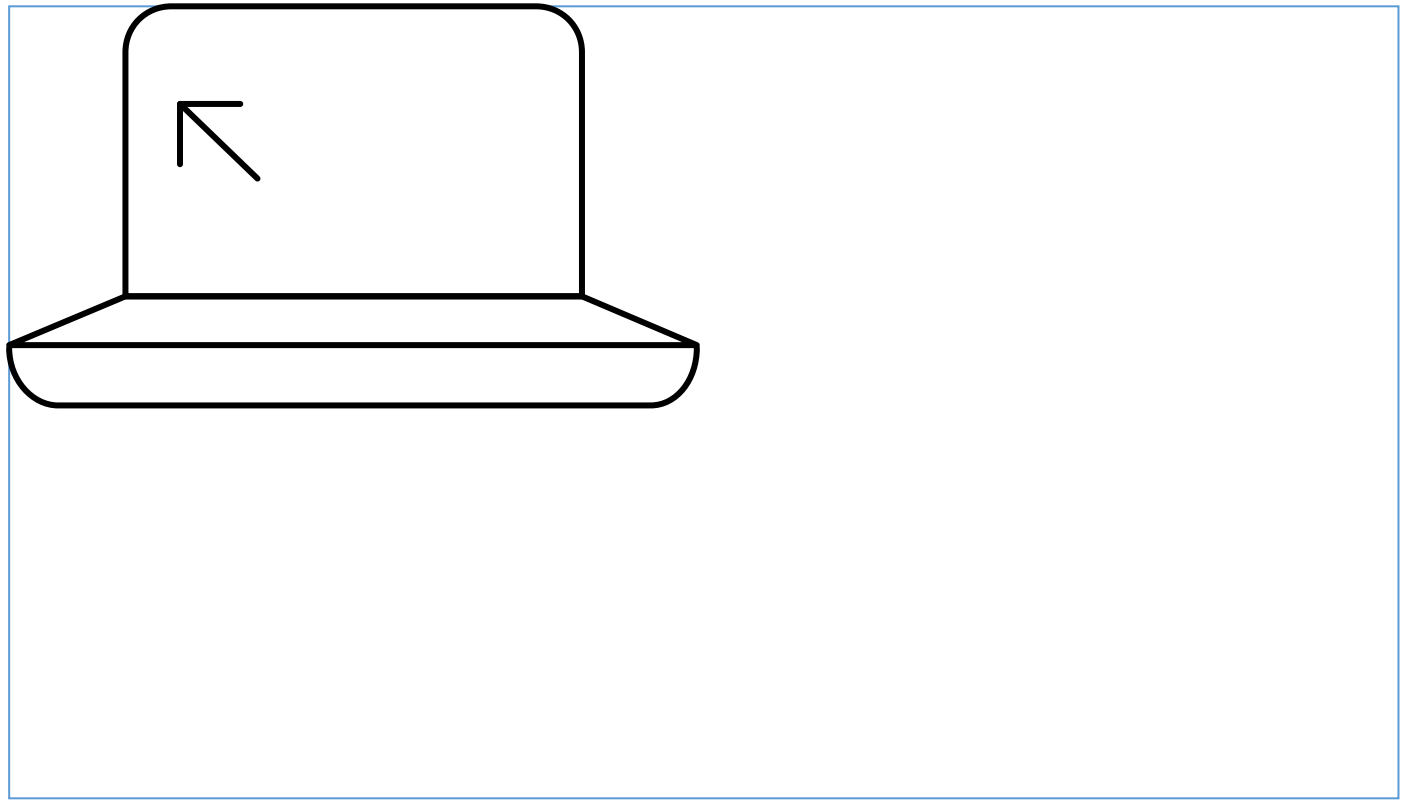


Testando aplicações Web

- TDD.
- Jest.
- react-testing-library.
- React-hook-testing-library.
- Cypress.

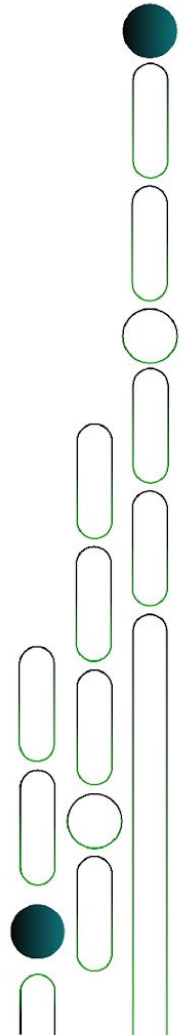


Acompanhe o professor



Próxima aula

- Jest.



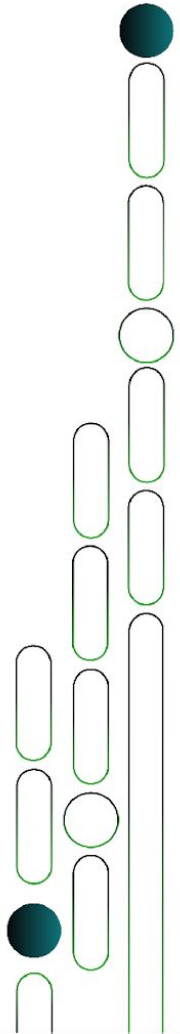
XPe
Aprenda com quem faz

React II

Capítulo 7. Testando aplicações React I

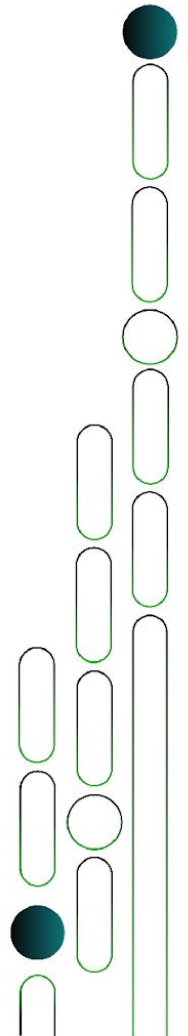
Aula 7.2. Jest

Prof. Rodrigo Borba



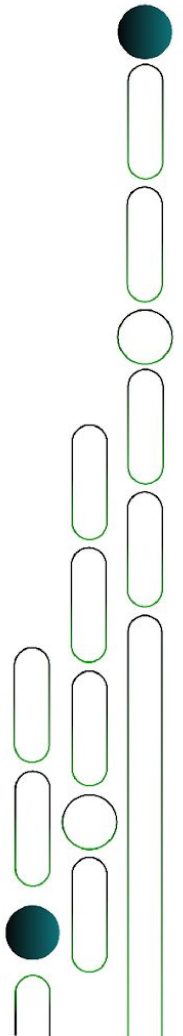
Nesta aula

- Overview Jest.
- Instalação.
- Testando aplicações WEB.



Overview

- Nenhuma configuração necessária.
- Testes Snapshot.
- Cobertura de código.
- Facilidade em mocking.

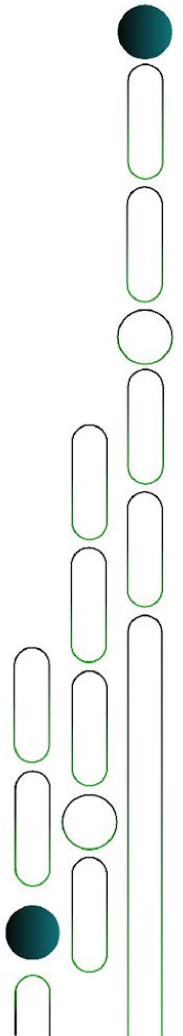


Instalação

yarn add --dev jest ou npm install --save-dev jest

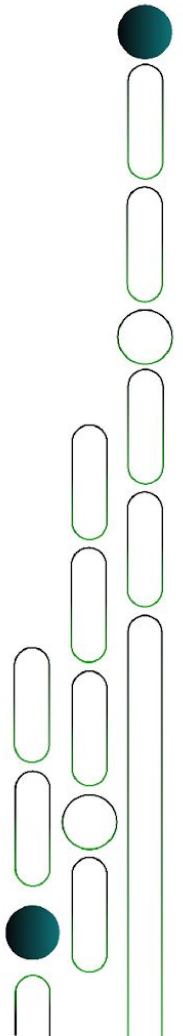
No package.js

```
{
 "scripts": {
   "test": "jest"
 }
}
```

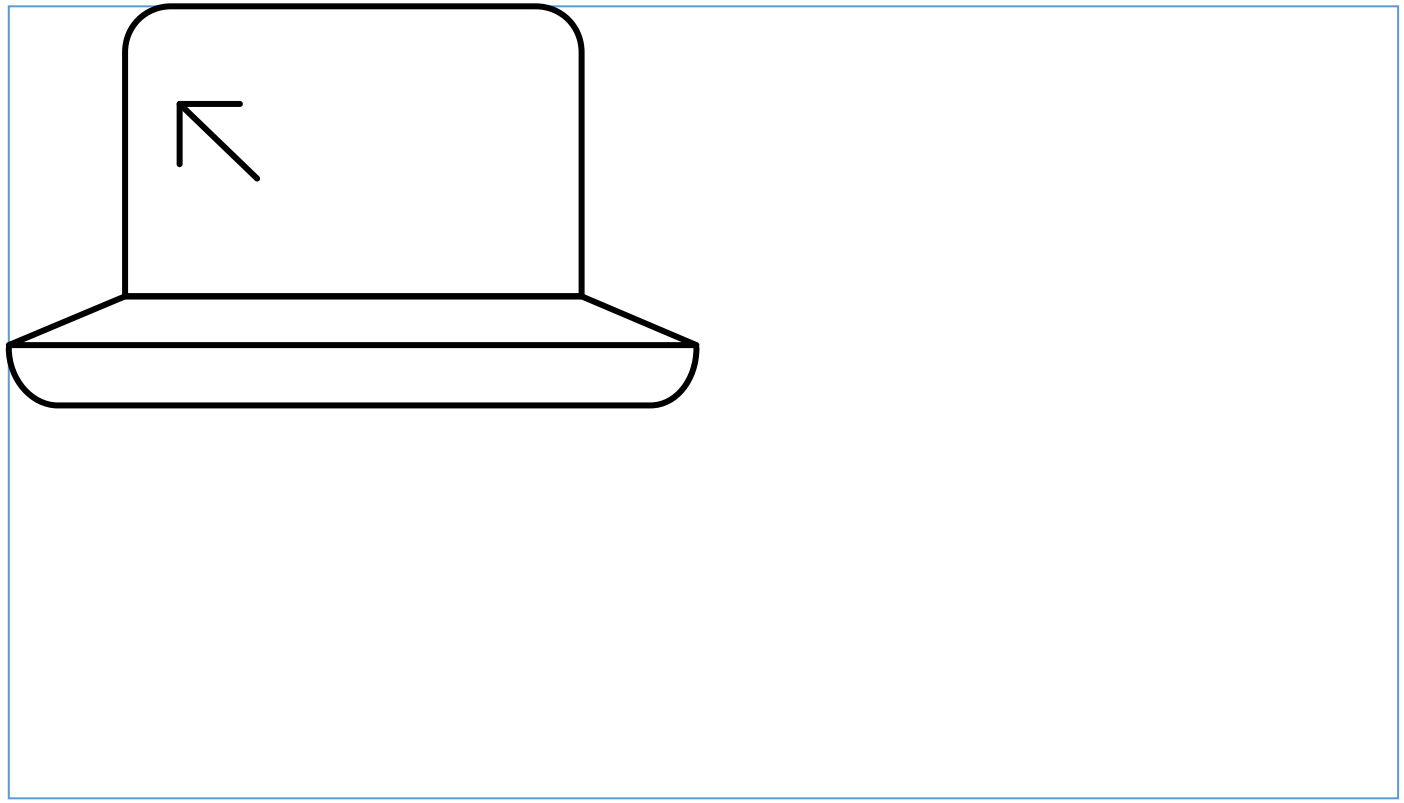


Testando aplicações WEB

- Syntax.
- Matchers:
    - Exceções.
- Before / After.

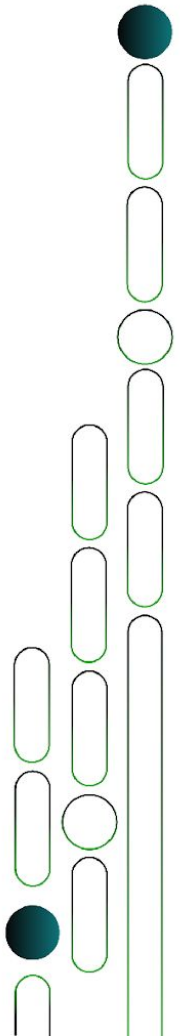


Acompanhe o professor



Próxima aula

- Test Driven Development.



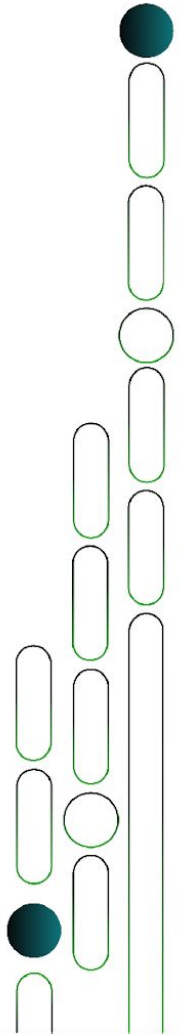
Aprenda com quem faz

React II

Capítulo 7. Testando aplicações React I

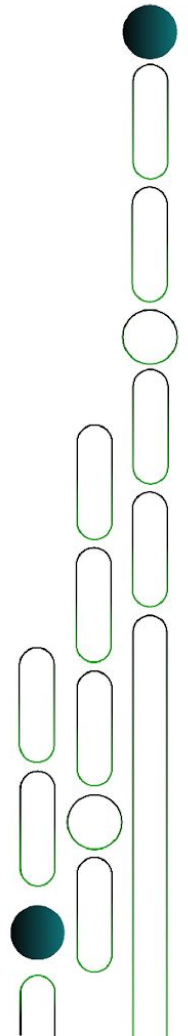
Aula 7.3. Test Driven Development

Prof. Rodrigo Borba

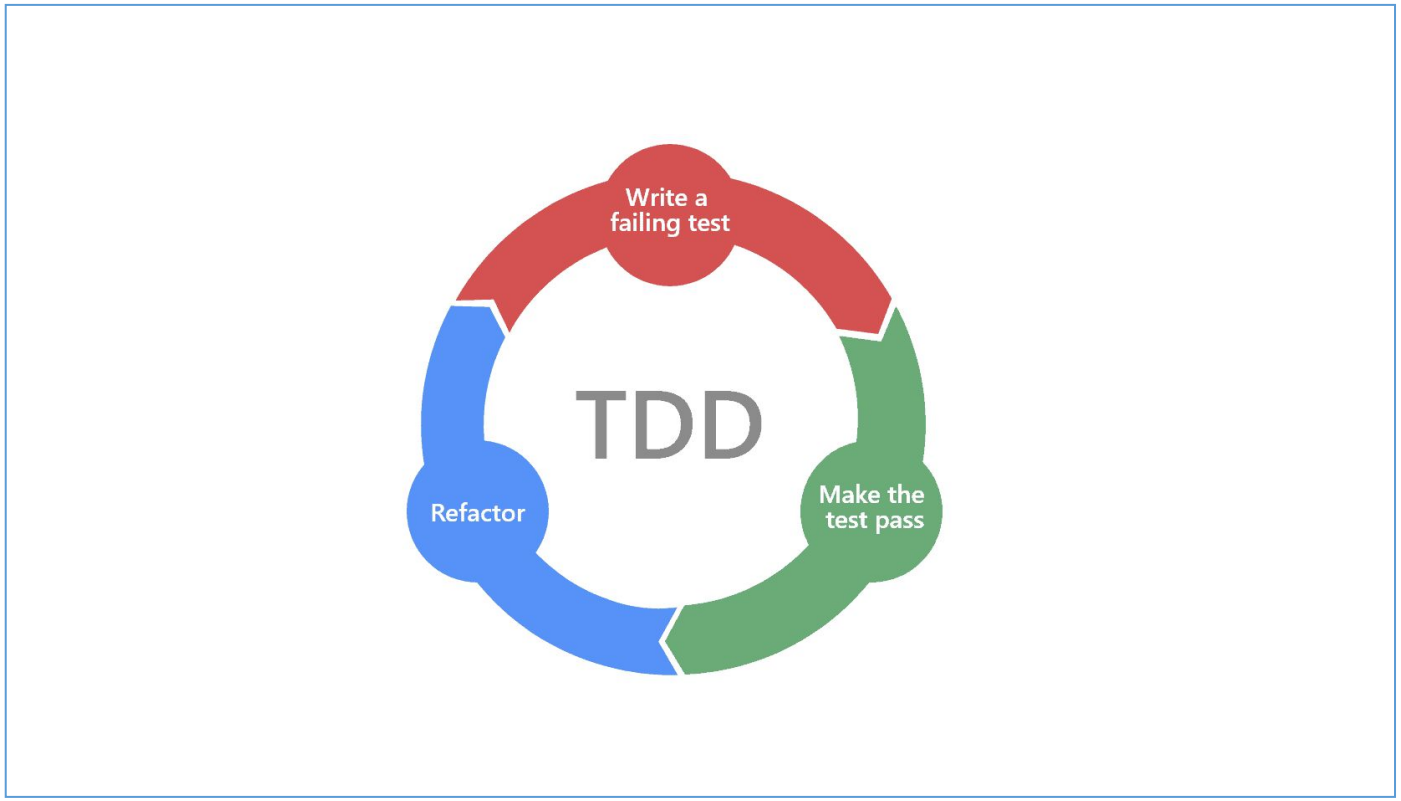


Nesta aula

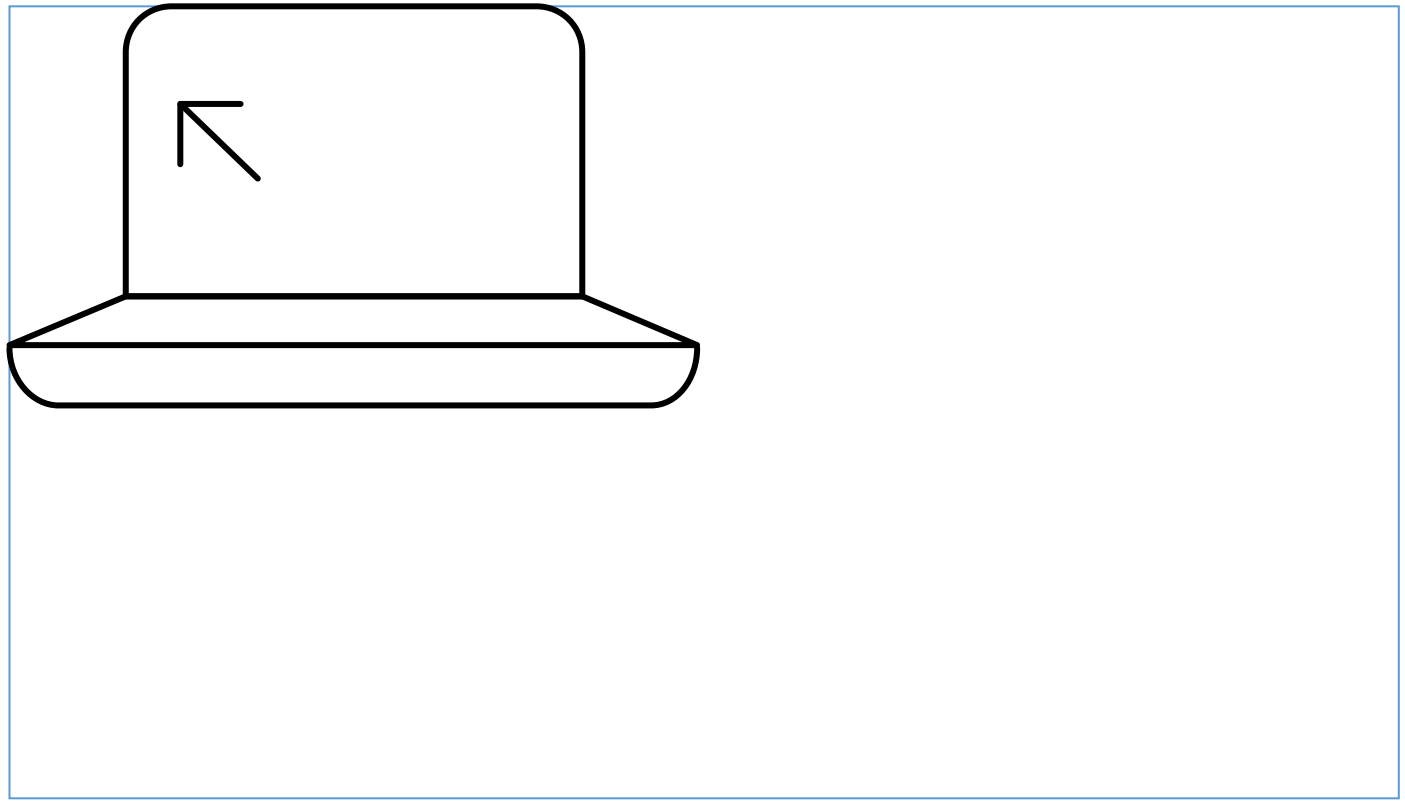
- TDD.



TDD

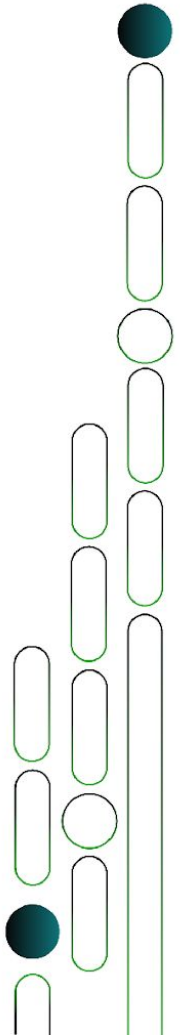


Acompanhe o professor



Próxima aula

- Testando aplicações React II:
  - React Testing Library.



Aprenda com quem faz

React II

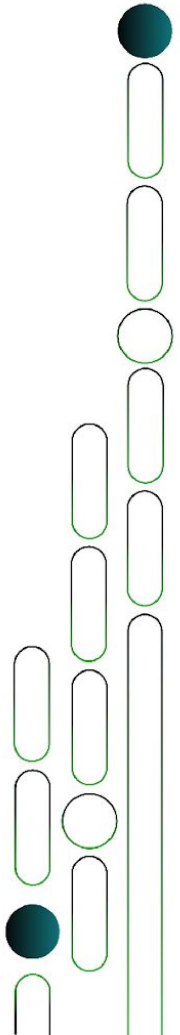
Capítulo 8. Testando aplicações React II

Aula 8.1. React Testing Library

Prof. Rodrigo Borba

Nesta aula

- Overview.
- Instalação.
- Queries.



Overview

- Não é um test-runner (como o Jest).
- Não é para ser usado em um test runner específico.
- Simplifica (e MUITO) a interação com componentes.

Instalação

XPe

- npm install --save-dev @testing-library/react
- yarn add @testing-library/react -D
- No Next:
  - npx create-next-app@latest --example with-jest with-jest-app
  - (https://nextjs.org/docs/testing#jest-and-react-testing-library)

Uso

- Screen.
- FireEvent.

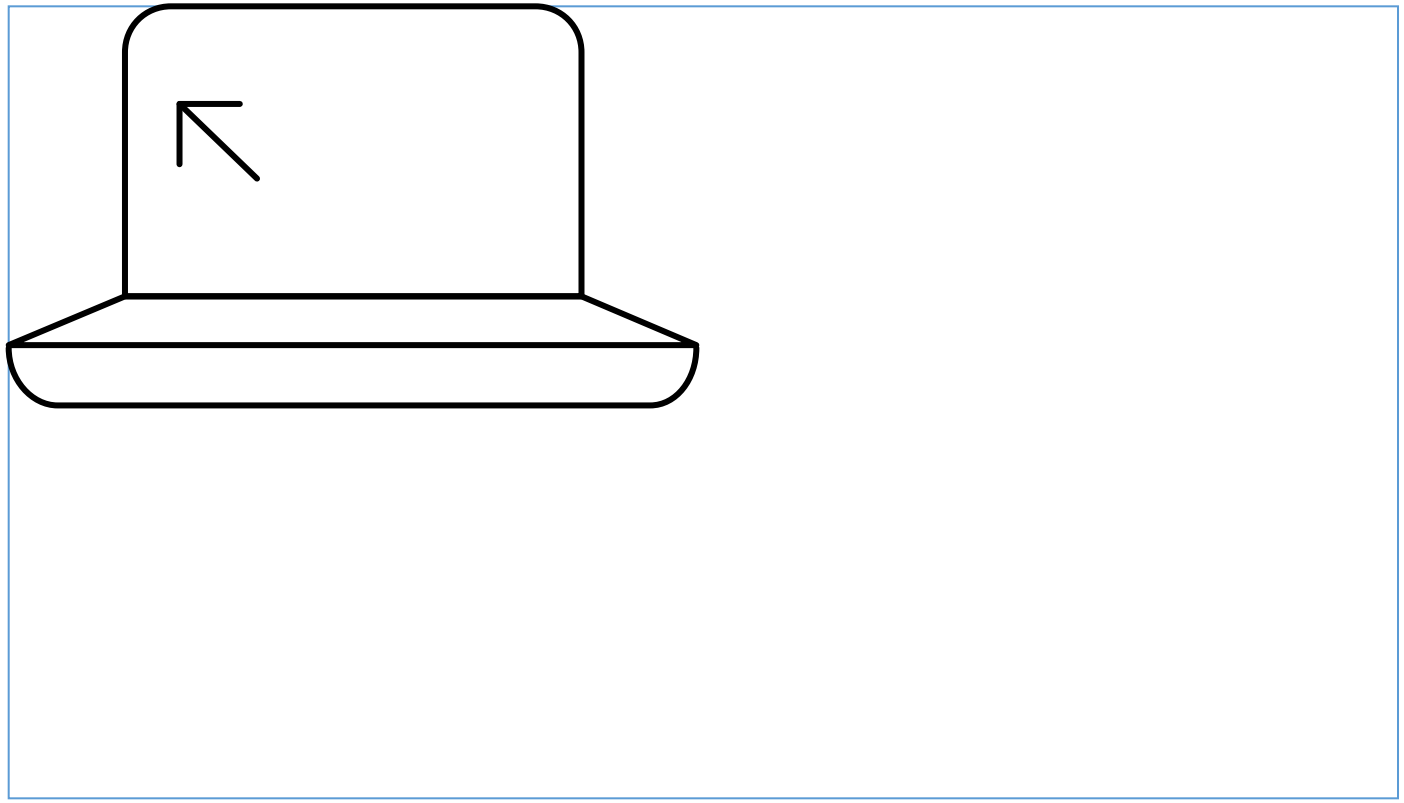
Queries

Type of Query	0 Matches	1 Match	>1 Matches	Retry (Async/Await)
Single Element				
getBy...	Throw error	Return element	Throw error	No
queryBy...	Return null	Return element	Throw error	No
findBy...	Throw error	Return element	Throw error	Yes
Multiple Elements				
getAllBy...	Throw error	Return array	Return array	No
queryAllBy...	Return []	Return array	Return array	No
findAllBy...	Throw error	Return array	Return array	Yes

Prioridades

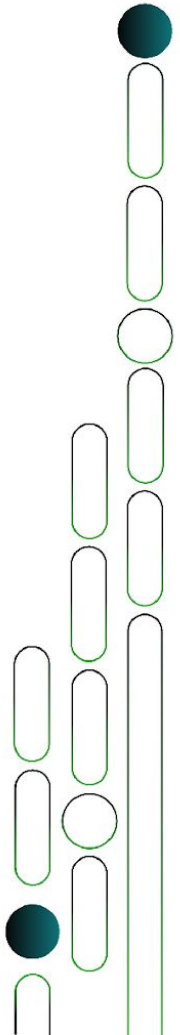
1. getByRole
2. getByLabelText
3. getByPlaceholderText
4. getByText
5. getByDisplayValue

Acompanhe o professor



Próxima aula

- React Hooks Testing Library.



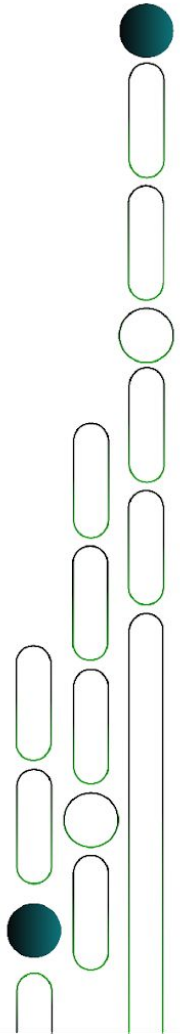
Aprenda com quem faz

React II

Capítulo 8. Testando aplicações React II

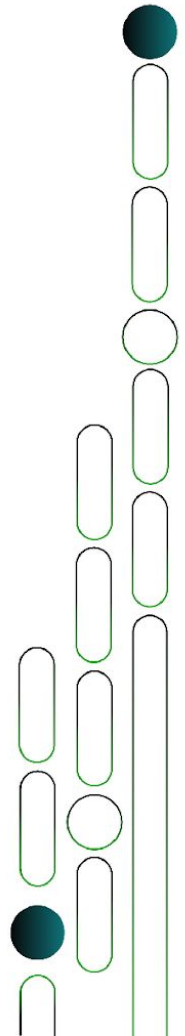
Aula 8.2. React Hooks Testing Library

Prof. Rodrigo Borba



Nesta aula

- Overview.
- Instação.
- Quando usar e não usar.



Instalação

- npm install --save-dev @testing-library/react-hooks
- yarn add @testing-library/react-hooks -D

Overview

- Simplifica o teste de hooks sem associação com componentes.

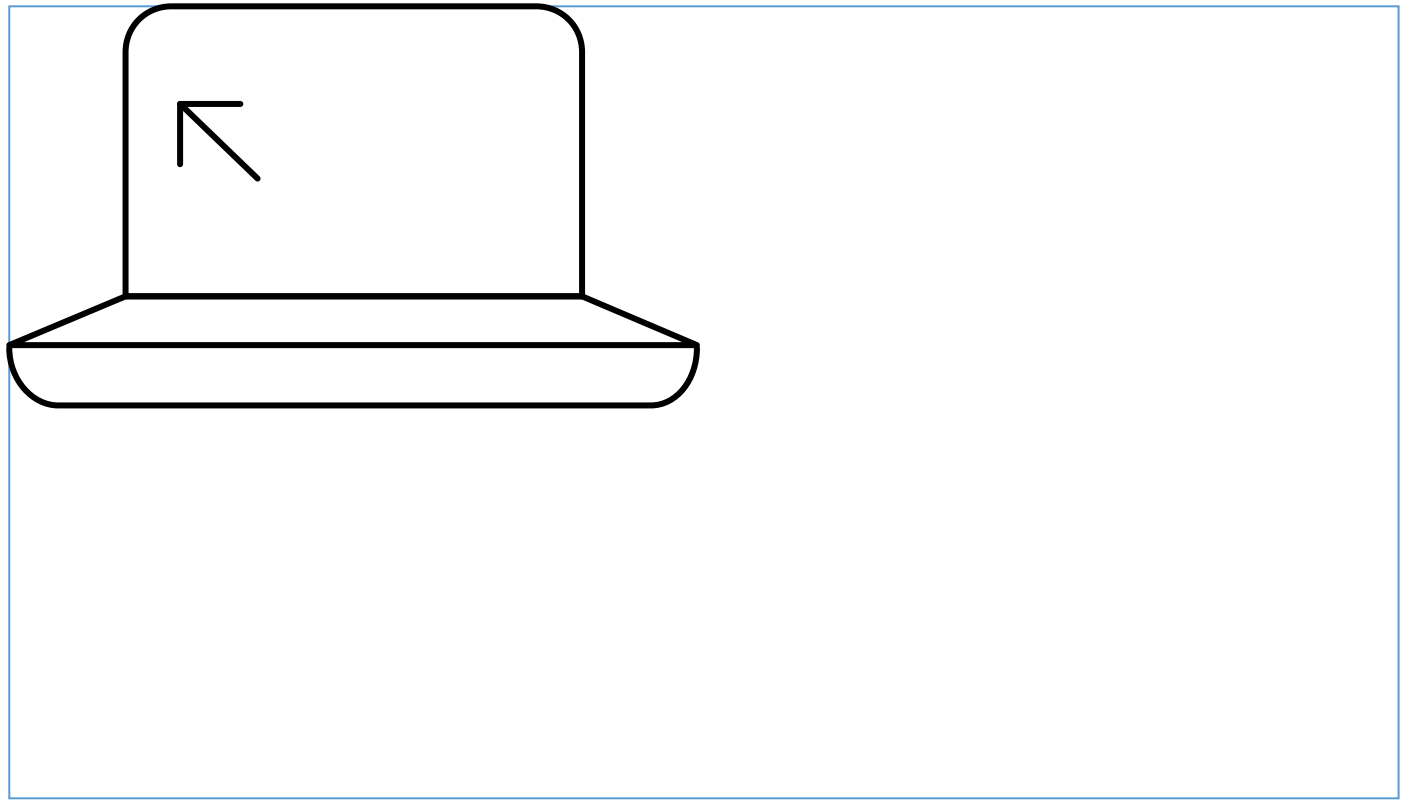
Quando usar e não usar

- Usar:
  - Quando seu hook não for necessariamente ligado a um componente específico.
  - Quando seu hook for difícil de ser testado através de componentes.

Quando usar e não usar

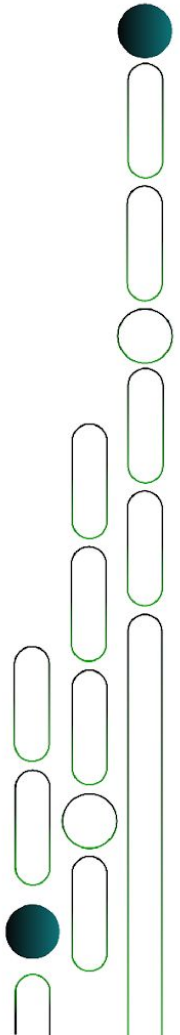
- Não usar:
  - Seu hook é fortemente acoplado a um componente.
  - Seu hook é facilmente testável através de componentes.
- Regra de ouro: **Use se realmente precisar.**

Acompanhe o professor



Próxima aula

- Testando aplicações React III:
  - Cypress.



Aprenda com quem faz

React II

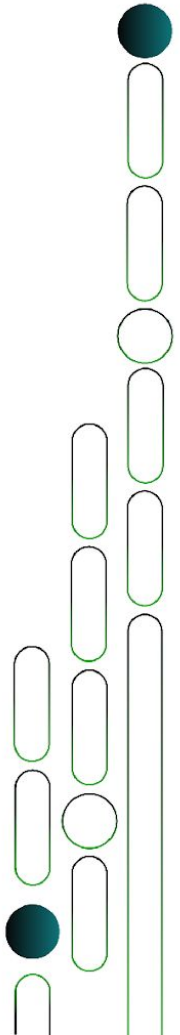
Capítulo 9. Testando aplicações React III

Aula 9.1. Cypress

Prof. Rodrigo Borba

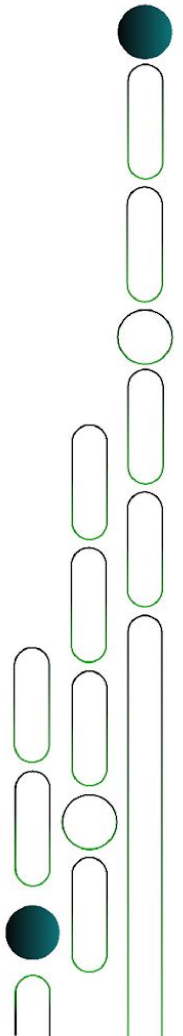
Nesta aula

- Overview.
- Instalação.
- Prática.



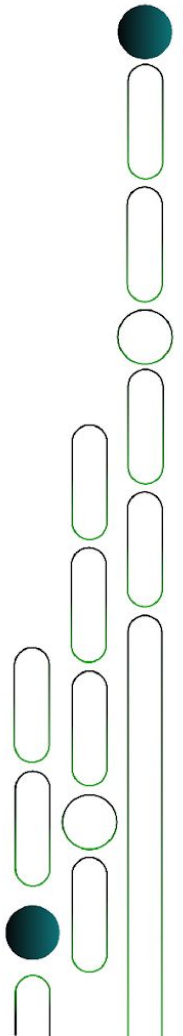
Overview

- Testes de integração em browser real.
- Linha do tempo das interações.
- Passos do testes podem ser gravados direto na UI.
- Várias opções de browsers.
- Sintaxe Mocha e Chai.

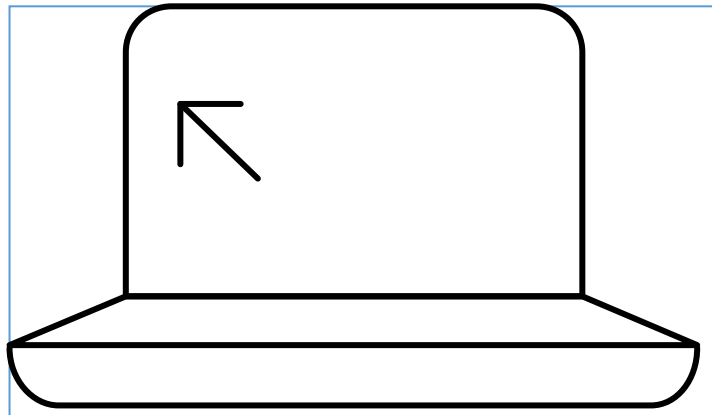


Instalação

- npm install cypress —save-dev
- yarn add cypress -D
- No next:
  - npx create-next-app@latest --example with-cypress with-
- No package:
  - Adicionar script: "cypress": "cypress open"

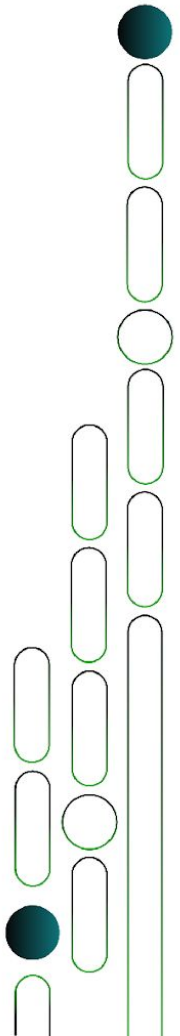


Acompanhe o professor



Próxima aula

- Deploy Application:
  - Plataformas de Deploy / Deploy na Vercel.



Aprenda com quem faz

React II

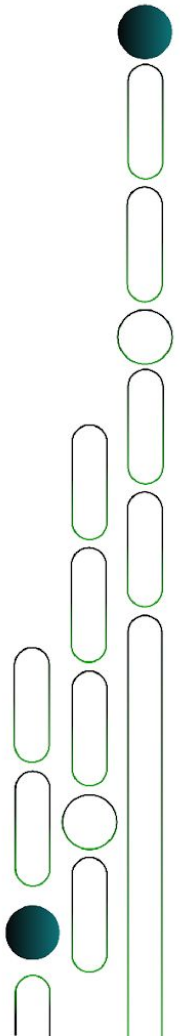
Capítulo 10. Deploy Application

Aula 10.1. Plataformas de Deploy / Deploy na Vercel

Prof. Rodrigo Borba

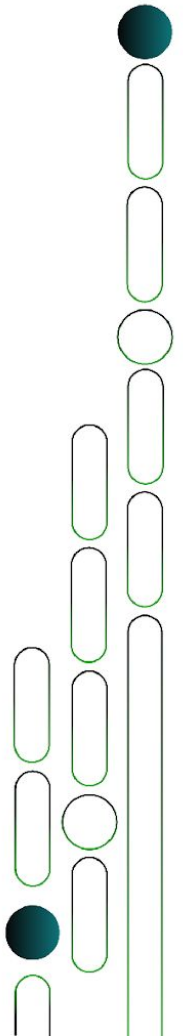
Nesta aula

- Plataformas para Deploy.
- Vercel.

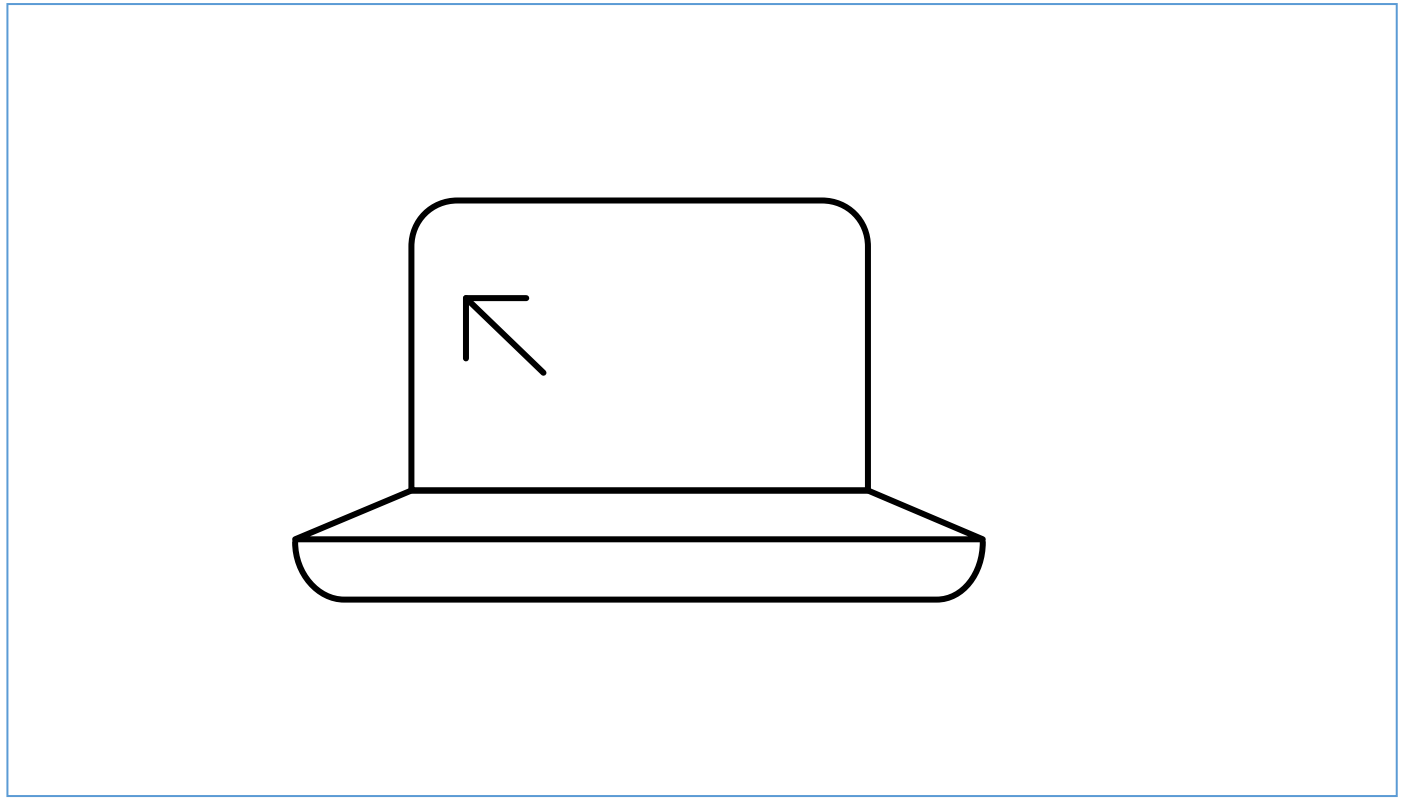


Overview

- Github Pages.
- Zeit Now.
- Heroku.
- Vercel.



Vercel



Conclusão