

MIT em Desenvolvimento Full Stack

Front-end Jamstack com Gatsby

Agenda

Aula 1: Introdução ao Jamstack e Gatsby.

- Apresentações.
- Introdução ao Jamstack.
- Introdução ao Gatsby.



Apresentações



Armênio Cardoso

Iniciei minha carreira profissional em **1986** e desde **1990** procuro conciliar o trabalho em Desenvolvimento de Software com o de Professor.

Particpei em diversos projetos, atuando com modelagem, arquitetura e programação. Fui desenvolvedor Pascal, Clipper, Visual Basic, C/C++, Java, JavaScript.

Trabalhei como tradutor / revisor de livros técnicos e professor em várias instituições.

Em **2002** vim para o Infnet onde dou aulas nos cursos de Graduação, Pós-Graduação e de Extensão.

Desde **2012** trabalho como Consultor de TI e Tech Leader na DASA - Diagnósticos da América S.A., empresa da área de medicina diagnóstica.



<http://www.linkedin.com/in/armeniocardoso>

MIT em
Desenvolvimento
Full Stack

```
graph TD; A[MIT em Desenvolvimento Full Stack] --> B[Bloco: Front-end com Jamstack]; B --> C[Disciplina: Front-end Jamstack com Gatsby]; B --> D[Disciplina: Front-end com Next.JS];
```

Bloco: Front-end
com Jamstack

Disciplina: Front-end
Jamstack com
Gatsby

Disciplina: Front-end
com Next.JS

Introdução ao Jamstack



JavaScript



API



Markup

Jamstack é uma abordagem arquitetural para a **construção de aplicações web** que separa completamente a camada de front-end da camada de back-end, priorizando a entrega via CDN e o uso de APIs e serviços externos.

Jamstack é simplesmente uma tentativa de dar um nome a um conjunto de práticas arquitetônicas amplamente utilizadas.

Em vez de renderizar páginas dinamicamente no servidor a cada requisição (como PHP tradicional ou JSP), o **Jamstack** entrega HTML pré-construído, geralmente gerado em build time, e usa JavaScript + APIs para dar interatividade e dinamismo.

Características Jamstack	Descrição
Separação de front e back	<p>O front-end é independente, gerado como arquivos estáticos (HTML/CSS/JS) e hospedado em CDN.</p> <p>O back-end é acessado via APIs (REST, GraphQL, serverless functions).</p>
Pré-renderização	<p>Conteúdo é gerado de forma estática (SSG) ou semi-sincronizada (ISR, DSG), garantindo velocidade e SEO.</p>
Entrega via CDN	<p>O front-end é entregue diretamente por uma rede de distribuição de conteúdo, melhorando latência global e resiliência.</p>
Arquitetura orientada a serviços	<p>Funcionalidades dinâmicas (autenticação, e-commerce, busca, etc.) são obtidas via serviços especializados, muitas vezes “headless”.</p>

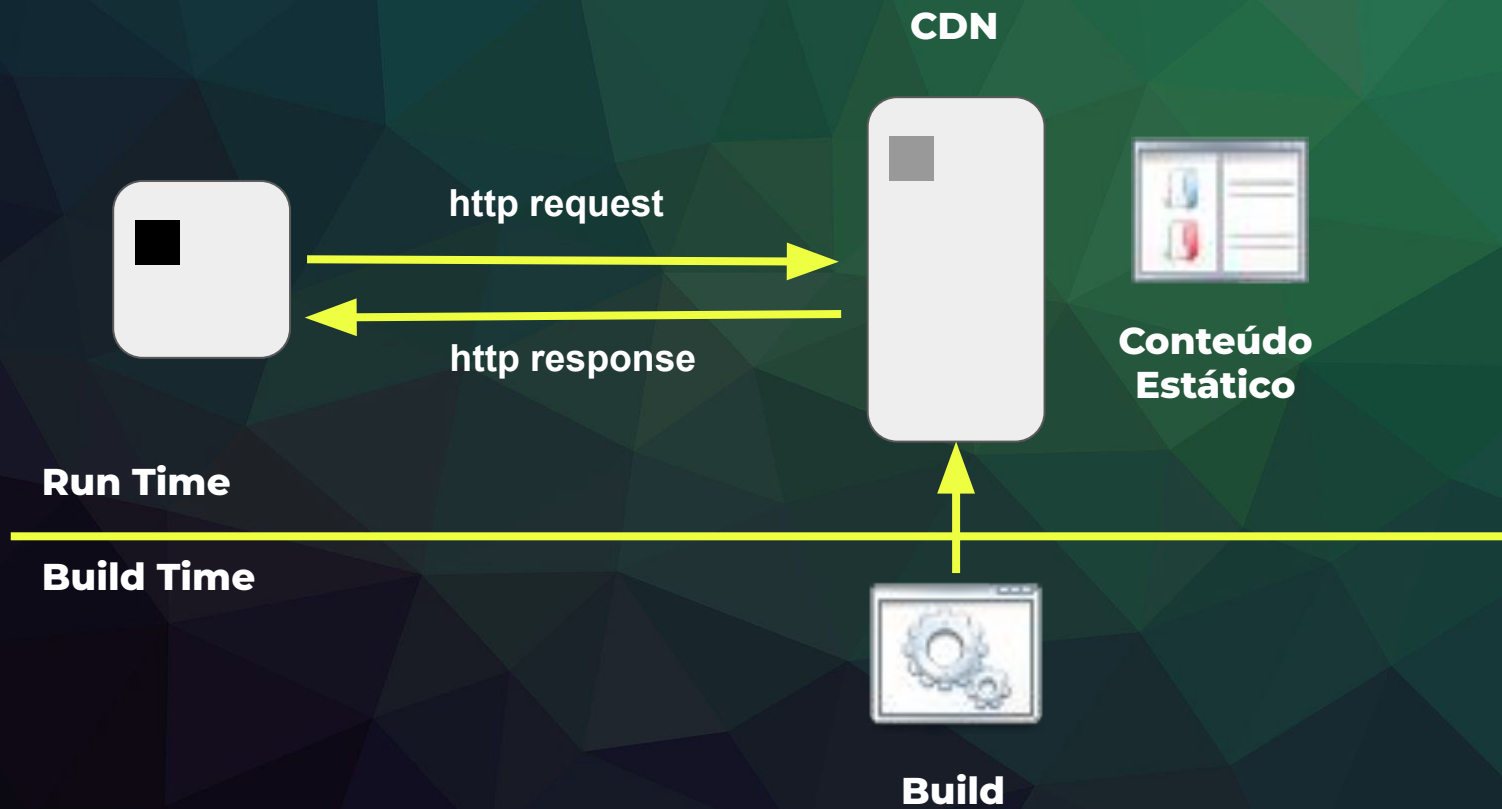
Pré-renderização

A marcação no **Jamstack** é entregue ao usuário por um modelo diferente.

A abordagem **Jamstack** é pré-construir toda a marcação antecipadamente e servi-la diretamente ao navegador a partir de um CDN.

Este processo envolve uma ferramenta de construção, um gerador de site estático como **Gatsby**, onde o conteúdo e os modelos são combinados em HTML, os arquivos de origem são **transpilados** ou compilados em JavaScript e CSS é executado por meio de pré-processadores ou pós-processadores.

Pré-renderização



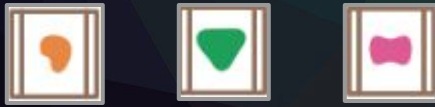


Content Delivery Network é uma rede de distribuição que fornece conteúdo de forma rápida, independentemente da localização.

Distribui um mesmo conteúdo em diversos servidores, para estar disponível o mais próximo usuário.

Funciona como um cache de arquivos em diferentes datacenters dispersos em todo o mundo.

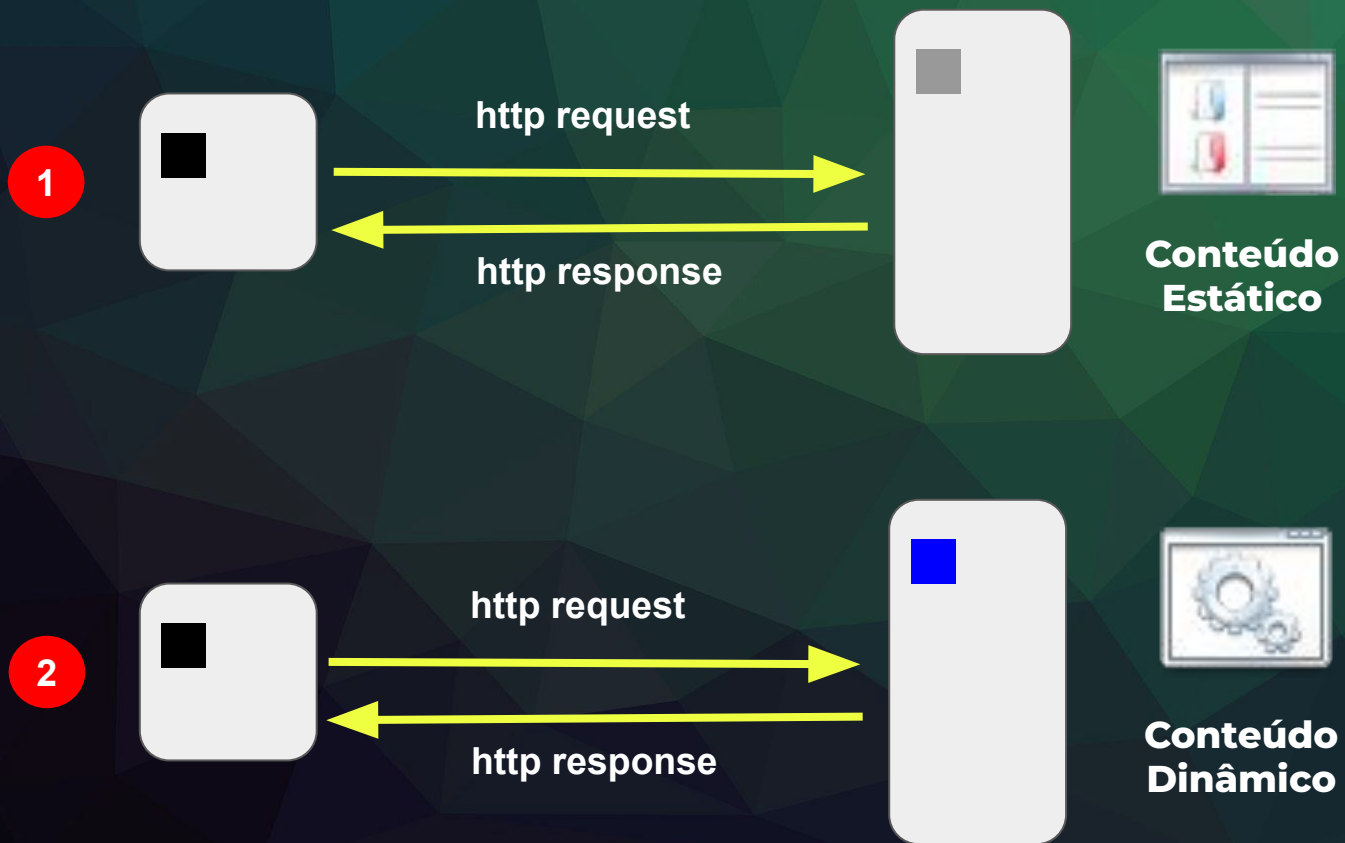
Arquitetura orientada a serviços distribui responsabilidades / funcionalidades por vários módulos que configuram cada serviço.



A implantação e escalabilidade é facilitada pela modularidade.

Os serviços são distribuídos entre os servidores, replicando quando necessário.

Conteúdo Estático X Dinâmico



Conteúdo Estático X Dinâmico

PHP Hypertext
Preprocessor - scripting
language



JSP JavaServer Pages



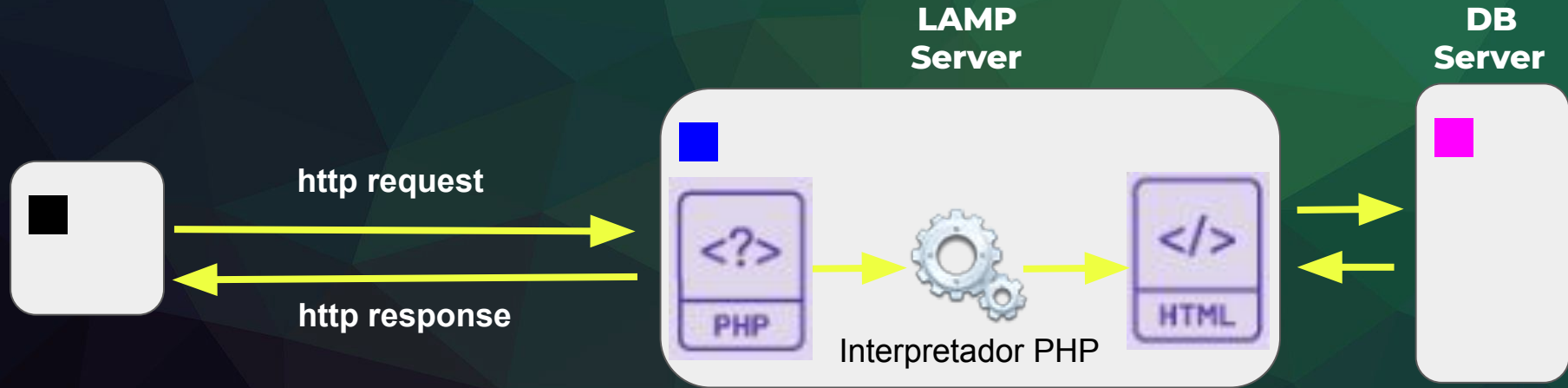
ASP Active Server Pages



CFML ColdFusion Markup
Language



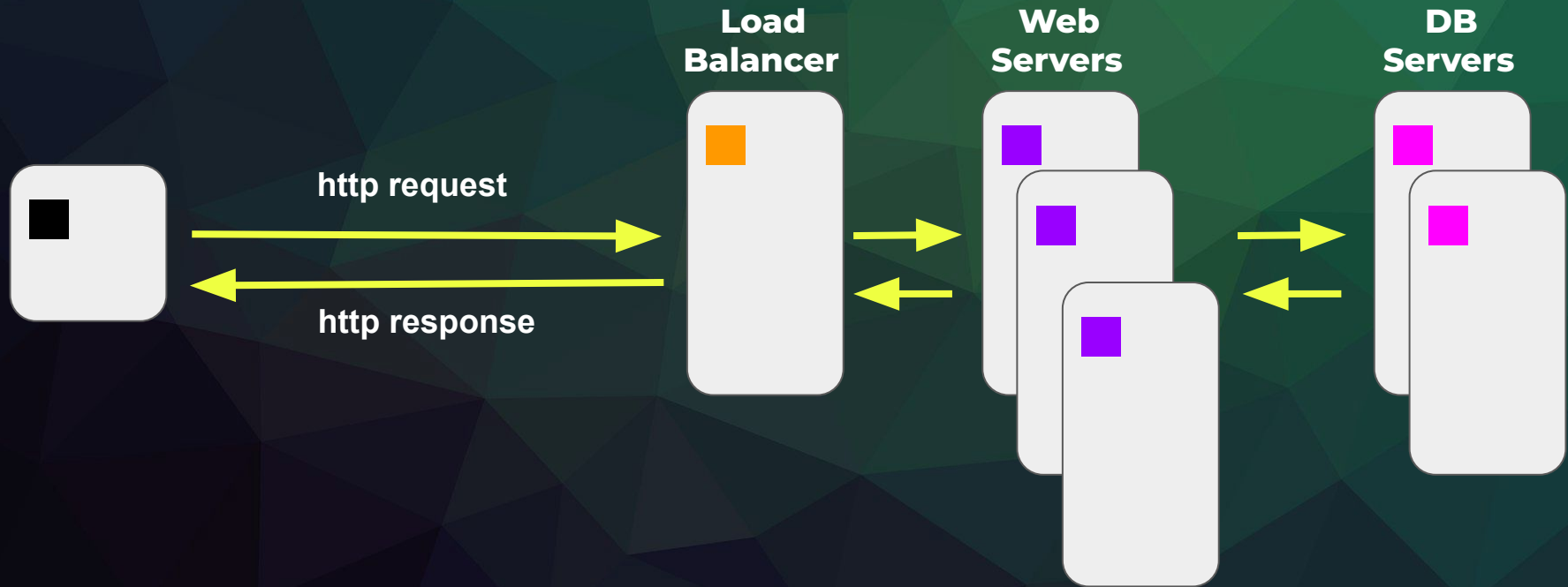
Conteúdo Dinâmico: LAMP



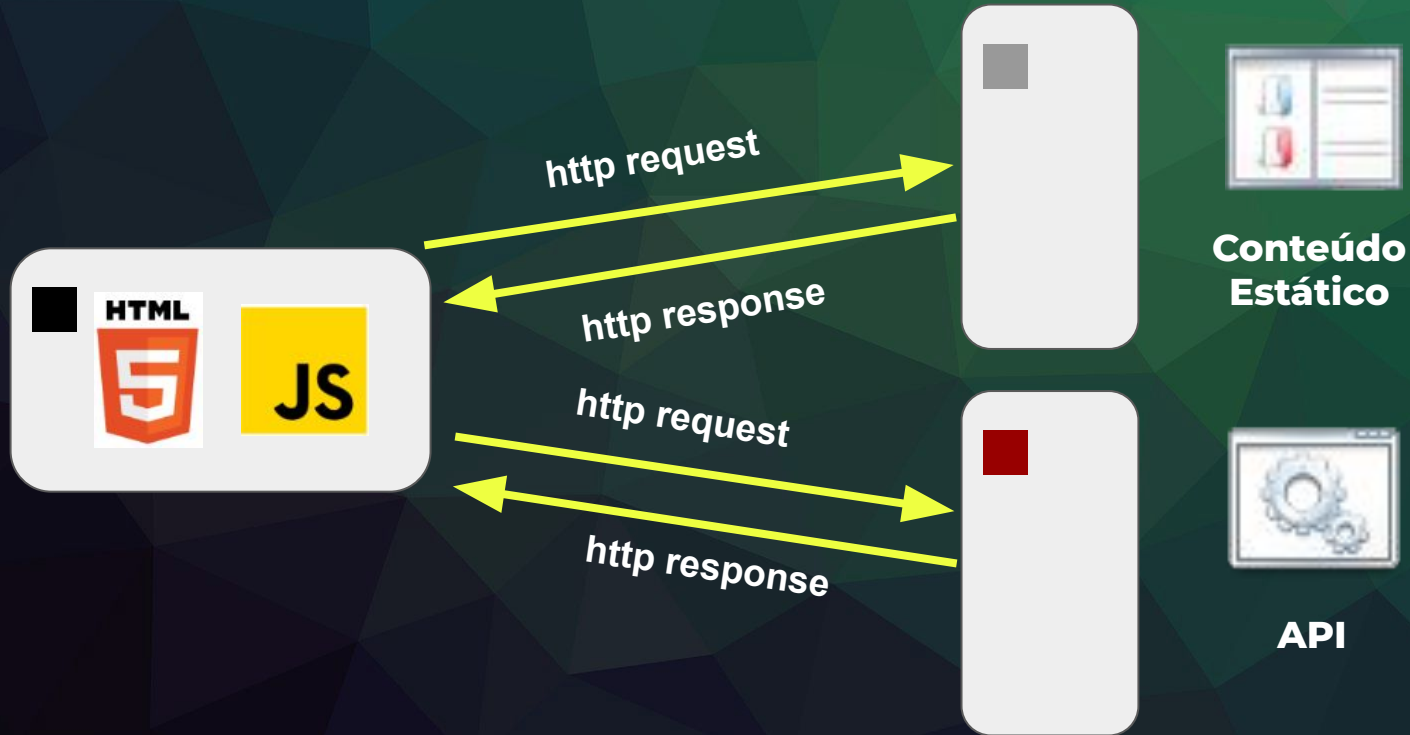
LAMP - Linux, Apache, MySQL, PHP



Conteúdo Dinâmico: Infraestrutura



Conteúdo Estático



Introdução ao Gatsby



Gatsby é um gerador de sites estáticos de código aberto **Jamstack**, construído sobre **Node.js** usando **React** e **GraphQL**.

HTML

CSS

JavaScript

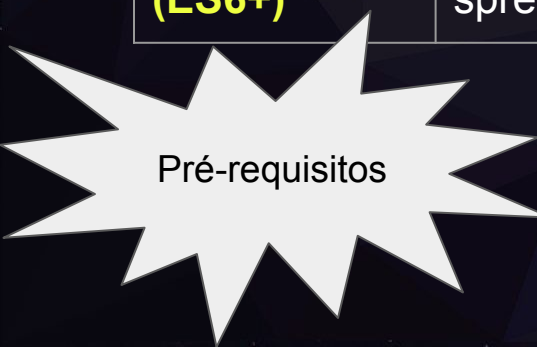
React

GraphQL

Static Site
Generator
SSG

	Pré-requisitos
Web	HTML5, CSS3, JS moderno
React	JSX, componentes, hooks básicos
Ambiente	Node.js + NPM/Yarn, terminal, Git
Gatsby	CLI instalada, GraphQL básico

	Competências
HTML5	Estrutura de páginas, semântica, tags básicas e meta tags (SEO).
CSS3	Seletores, flexbox/grid, responsividade, uso de classes, noções de pré-processadores ou Tailwind (desejável).
JavaScript (ES6+)	Import/export, arrow functions, fetch, async/await, promises, spread/rest, objetos, arrays, módulos.

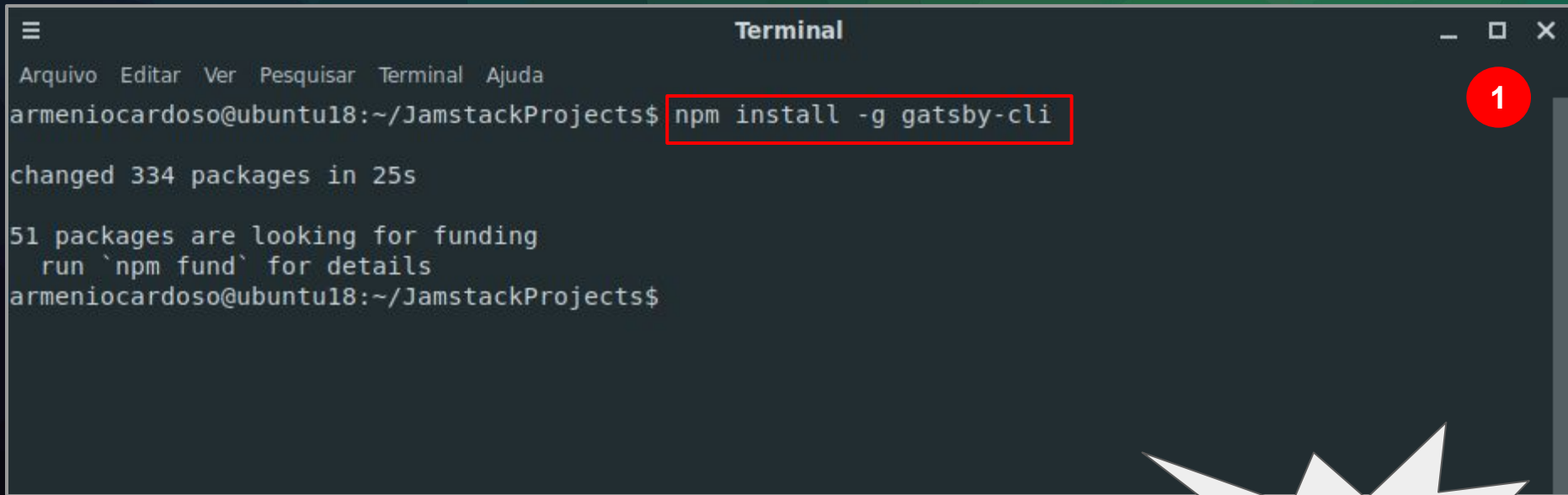


Pré-requisitos

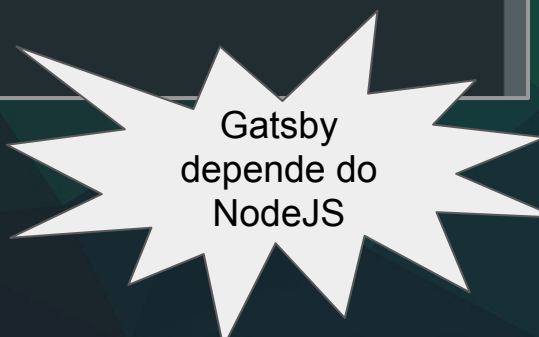
React	Competências
JSX	Sintaxe básica, diferenças entre HTML e JSX.
Componentes	Funções componentes, props, composição.
Estado e Efeitos	useState, useEffect.
Rotas e Navegação	Compreensão de navegação SPA e links.
Modularização	Organização de componentes em pastas e import/export.



Pré-requisitos

A terminal window titled "Terminal" with a menu bar containing "Arquivo", "Editar", "Ver", "Pesquisar", "Terminal", and "Ajuda". The terminal shows the command "npm install -g gatsby-cli" being executed, which results in "changed 334 packages in 25s" and a message about funding. The command "npm install -g gatsby-cli" is highlighted with a red box. A red circle with the number "1" is in the top right corner of the terminal window.

```
Terminal
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
armeniocardoso@ubuntu18:~/JamstackProjects$ npm install -g gatsby-cli
changed 334 packages in 25s
51 packages are looking for funding
  run `npm fund` for details
armeniocardoso@ubuntu18:~/JamstackProjects$
```

A white starburst-shaped callout box with a black border, containing text about Gatsby's dependency on NodeJS.

Gatsby
depende do
NodeJS

```
Terminal
Arquivo Editar Ver Pesquisar Terminal Ajuda
armeniocardoso@ubuntu18:~/JamstackProjects$ gatsby new
create-gatsby version 3.12.1
```

2

[Welcome to Gatsby!](#)

This command will generate a new Gatsby site for you in `/home/armeniocardoso/GatsbyProjects` with the setup you select. Let's answer some questions:

What would you like to call your site?

✓ · Hello Gatsby

What would you like to name the folder where your site will be created?

✓ GatsbyProjects/ gatsby_1

✓ Will you be using JavaScript or TypeScript?

· JavaScript

✓ Will you be using a CMS?

· No (or I'll add it later)

✓ Would you like to install a styling system?

· No (or I'll add it later)

Thanks! Here's what we'll now do:

✧ Create a new Gatsby site in the folder gatsby_1

✓ Created site from template

✓ Installed Gatsby

✓ Installed plugins

✓ Created site in jamstack_004

🎉 Your new Gatsby site Estudos de Jamstack 004 has been successfully created

at /home/armeniocardoso/JamstackProjects/jamstack_004.

You can now view **estudos-de-jamstack-004** in the browser.

`http://localhost:8000/`

View GraphQL, an in-browser IDE, to explore your site's data and schema

`http://localhost:8000/___graphql`

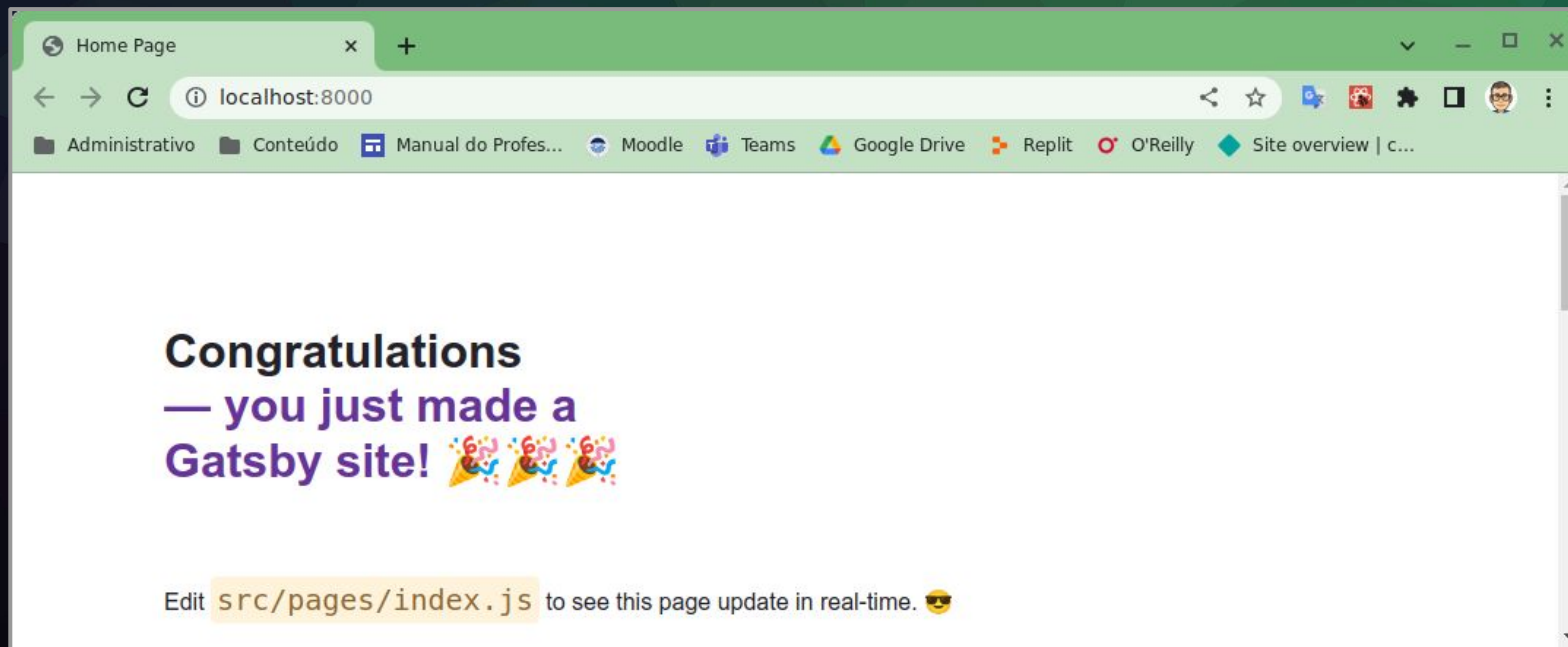
Note that the development build is not optimized.

To create a production build, use `gatsby build`

success Building development bundle - 30.589s

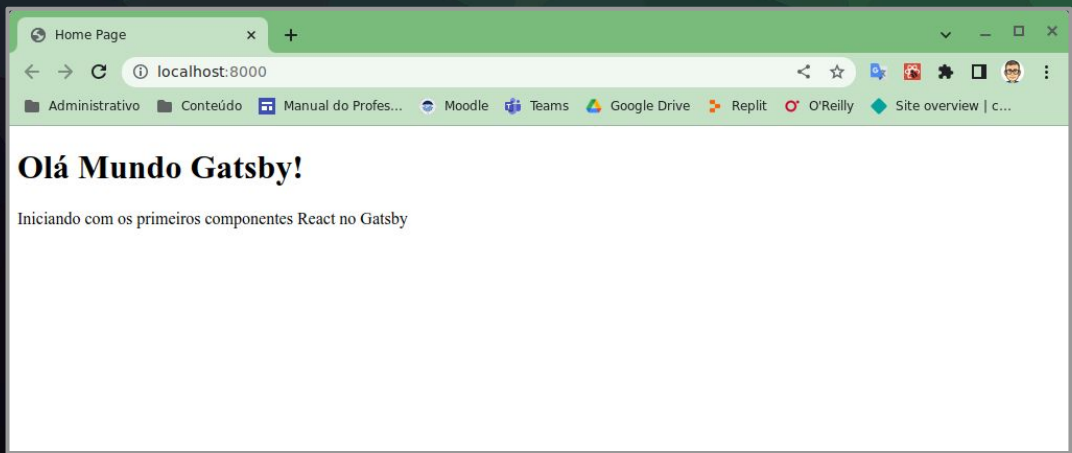
success Writing page-data.json and slice-data.json files to public directory - 0.222s - 3/4 18.00/s

3



```
JS index.js M x
src > pages > JS index.js > ...
126 const IndexPage = () => {
127   return (
128     <main>
129       <h1>Olá Mundo Gatsby!</h1>
130       <p>Iniciando com os primeiros componentes React no Gatsby</p>
131     </main>
132   )
133 }
134
135 export default IndexPage
136
137 export const Head = () => <title>Home Page</title>
```

4



JS index.js M

JS about.js U X

src > pages > JS about.js > ...

```
1  import * as React from 'react'
2
3  const AboutPage = () => {
4    return (
5      <main>
6        <h1>Sobre</h1>
7        <p>Estamos desenvolvendo estudos de Jamstack com Gatsby</p>
8      </main>
9    )
10 }
11
12 export const Head = () => <title>Sobre</title>
13
14 export default AboutPage;
```

5

Sobre

localhost:8000/about/

Administrativo Conteúdo Manual do Profes... Moodle Teams Google Drive Replit O'Reilly Site overview | c...

Sobre

Estamos desenvolvendo estudos de Jamstack com Gatsby

Critério	Gatsby	React (puro ou CRA / Vite)
Abordagem	Framework opinativo baseado em React, voltado para Static Site Generation (SSG) e Jamstack.	Biblioteca para construir interfaces. Maior liberdade, mas você precisa decidir e configurar tudo.
Performance inicial	Excelente: gera HTML estático, rápido no primeiro carregamento (melhor para SEO).	Precisa de renderização dinâmica (SPA). Carregamento inicial mais lento em sites grandes.
Integração de dados	Possui um data layer unificado com GraphQL , integrando CMS, Markdown, APIs, etc.	Você decide como consumir dados (fetch, Axios, GraphQL). Maior flexibilidade, mas sem camada integrada.
Ecosistema de plugins	Rico em plugins para otimização de imagens, SEO, CMS headless, deploy, etc.	Não possui plugins nativos, depende de libs externas e configuração manual.
Escalabilidade	Muito bom para sites de conteúdo (blogs, documentação, landing pages). Build pode ficar lento em sites gigantes.	Escalável para qualquer tipo de app (dashboards, e-commerce dinâmico, SPAs complexas).
Manutenção	Estrutura mais padronizada, menos código repetitivo.	Estrutura livre, risco de cada projeto seguir um padrão diferente.
Casos ideais	Blogs, sites institucionais, documentação, e-commerces estáticos, landing pages com SEO forte.	Aplicações dinâmicas: dashboards, apps com muita interação em tempo real, sistemas corporativos.