

MIT em Desenvolvimento Full Stack

Front-end Jamstack com Gatsby

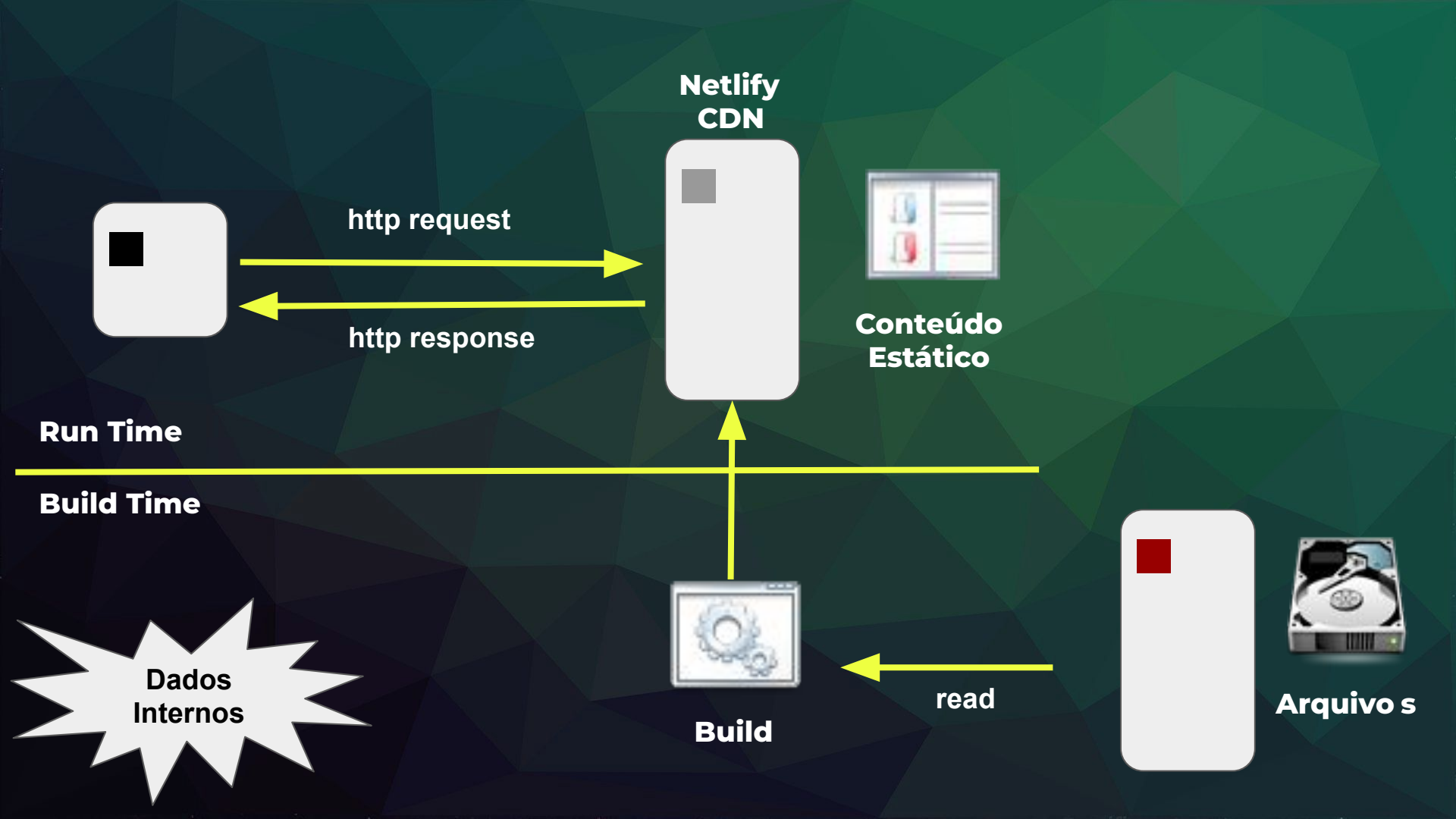
Agenda

Aula 5: Desenvolvimento de Rotas em Gatsby:

- Aprimorando as Consultas GraphQL.
- Processando Arquivos.
- Criando Arquivos MDX.
- Criando Rotas Dinâmicas.



Aprimorando as Consultas GraphQL



Your Filesystem

CMS

Private API

Database

source
plugins

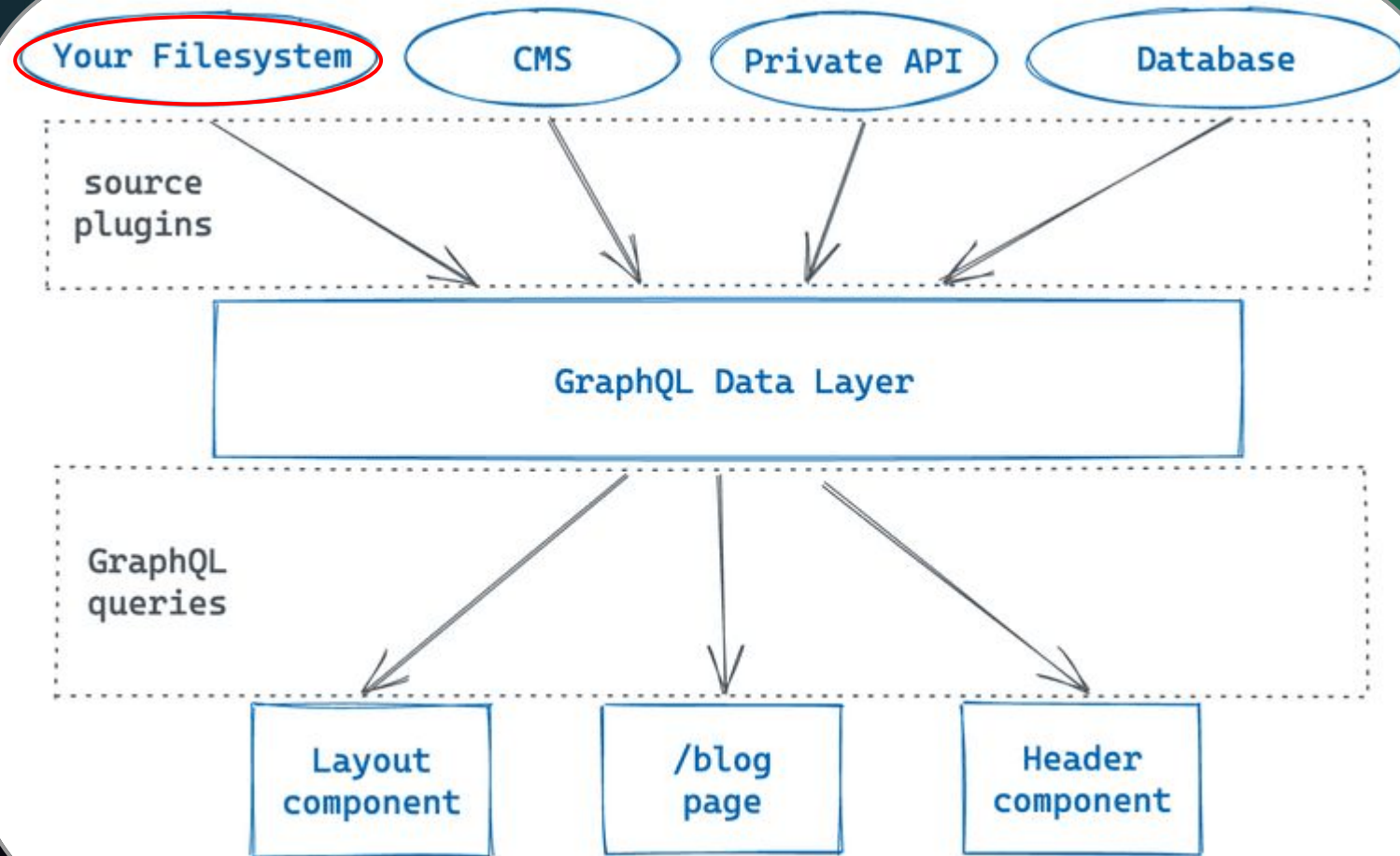
GraphQL Data Layer

GraphQL
queries

Layout
component

/blog
page

Header
component





Arquivo Editar Ver Pesquisar Terminal Ajuda

armeniocardoso@ubuntu18:~/GatsbyProjects\$ **npm init gatsby -- -y gatsby_5**

create-gatsby version 3.13.1

Welcome to Gatsby!

✓ Created site from template

✓ Installed Gatsby

✓ Installed plugins

✓ Created site in **gatsby_5**

🎉 Your new Gatsby site **gatsby_5** has been successfully created at **/home/armeniocardoso/GatsbyProjects/gatsby_5**. Start by going to the directory with

cd gatsby_5

Start the local development server with

npm run develop

See all commands at

<https://www.gatsbyjs.com/docs/reference/gatsby-cli/>

armeniocardoso@ubuntu18:~/GatsbyProjects\$

1

4

Aplicando Meta
Dados

```
JS layout.js x
src > components > JS layout.js > Layout
1  import React from "react"
2  import "./layout.css"
3  import { Link, useStaticQuery, graphql } from 'gatsby'
4  import Footer from "../footer"
5
6  export default function Layout({ children }) {
7
8    const data = useStaticQuery(graphql`
9      query {
10        site {
11          siteMetadata {
12            title
13          }
14        }
15      `)
16
17
18    return (
19      <main className="layout">
20        <div className="header">
21          <h3>{data.site.siteMetadata.title}</h3>
22          <nav className="topnav">
23            <Link to="/">Início</Link>
24          </nav>
25        </div>
```


Processando Arquivos

JS blog.js x

src > pages > JS blog.js > ...

```
1  import * as React from 'react'
2  import Layout from '../components/layout'
3
4  const BlogPage = () => {
5    return (
6      <Layout>
7        | <p>Esses são os últimos posts:</p>
8      </Layout>
9    )
10 }
11
12 export const Head = () => <title>Posts</title>
13
14 export default BlogPage
```

1

Criando um Blog

2

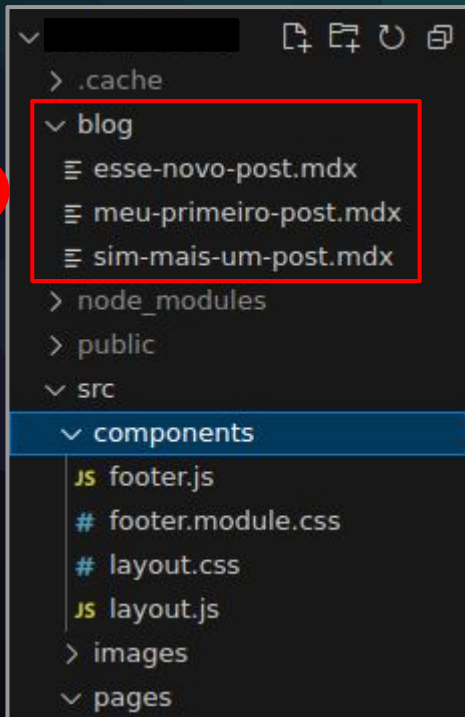
JS layout.js x

src > components > JS layout.js > ...

```
18   return (  
19     <main className="layout">  
20       <div className="header">  
21         <h3>{data.site.siteMetadata.title}</h3>  
22         <nav className="topnav">  
23           <Link to="/">Início</Link>  
24           <Link to="/blog">Blog</Link>  
25         </nav>  
26       </div>  
27       <div className="main">  
28         {children}  
29       </div>  
30       <Footer copyrightYear="2023" />  
31     </main >  
32   )  
33 }
```

Criando um Blog

3



```
npm install gatsby-source-filesystem
```

4

```
JS gatsby-config.js x
JS gatsby-config.js > ...
1  module.exports = {
2    siteMetadata: {
3      title: `Estudos do Jamstack com Gatsby`,
4    },
5    plugins: [
6      {
7        resolve: "gatsby-source-filesystem",
8        options: {
9          name: `blog`,
10         path: `${__dirname}/blog`,
11       },
12     ],
13   ],
14 }
```

5

Acessando um
diretório com arquivos

6

Gatsby - GraphQL

localhost:8000/___graphql?query=query%20MyQuery%20%7B%0A%20%20allFile%20%20...

Administrativo Conteúdo Ferramentas

```
1 query MyQuery {  
2   allFile {  
3     nodes {  
4       name  
5     }  
6   }  
7 }  
8
```

GraphQL

```
{  
  "data": {  
    "allFile": {  
      "nodes": [  
        {  
          "name": "esse-novo-post"  
        },  
        {  
          "name": "meu-primeiro-post"  
        },  
        {  
          "name": "sim-mais-um-post"  
        }  
      ]  
    }  
  },  
  "extensions": {}  
}
```

Acessando um diretório com arquivos

JS blog.js

src > pages > JS blog.js > ...

```
7   const data = useStaticQuery(graphql`  
8     query {  
9       allFile {  
10        nodes {  
11          name  
12        }  
13      }  
14    }  
15  `)
```

7

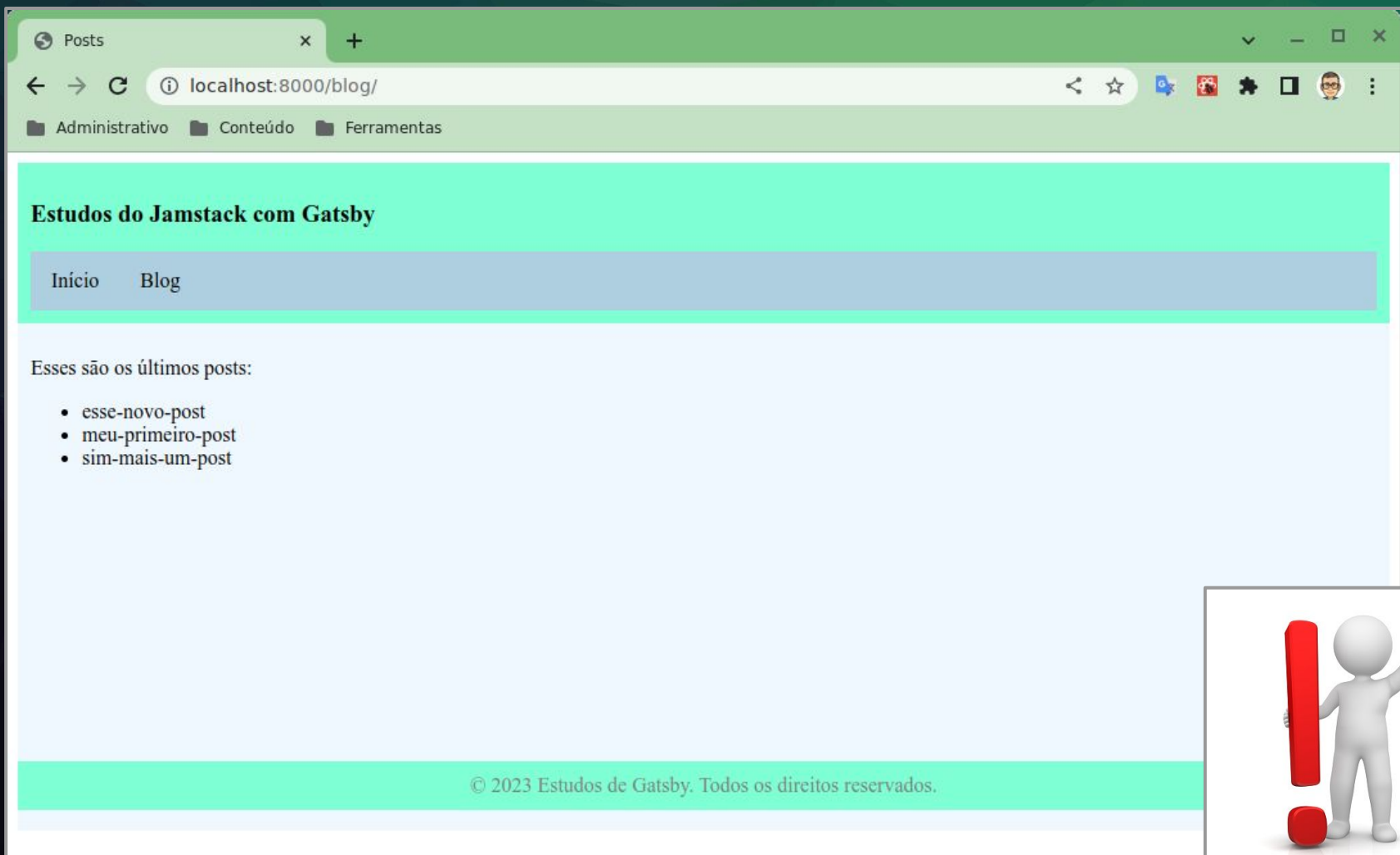
JS blog.js

src > pages > JS blog.js > ...

```
17  return (  
18    <Layout>  
19      <p>Esses são os últimos posts:</p>  
20      <ul>  
21        {  
22          data.allFile.nodes.map(node => (  
23            <li key={node.name}>  
24              {node.name}  
25            </li>  
26          ))  
27        }  
28      </ul>  
29    </Layout>  
30  )  
31 }
```

8

Acessando um
diretório com arquivos



Criando Arquivos MDX

O Que é MDX?

MDX é uma extensão que permite incluir JSX em documentos **Markdown**.

MDX torna possível incluir componentes React em suas postagens e páginas de um blog em Gatsby.

Markdown define uma sintaxe de texto simples para elementos HTML como h1, strong e além disso suporta HTML inline.

```
# Hello world!
```

```
You can use Markdown to create documents for [Gatsby](https://www.gatsbyjs.com/).
```

```
<figure class="chart">  
  <object data="chart.svg" type="image/svg+xml"></object>  
  <figcaption>MDX adoption has increased 120% since last year.</figcaption>  
</figure>
```

```
export const Figure = props => {  
  return (  
    <figure class="chart">  
      <object data={props.data} type="image/svg+xml"></object>  
      <figcaption>{props.caption}</figcaption>  
    </figure>  
  )  
}
```



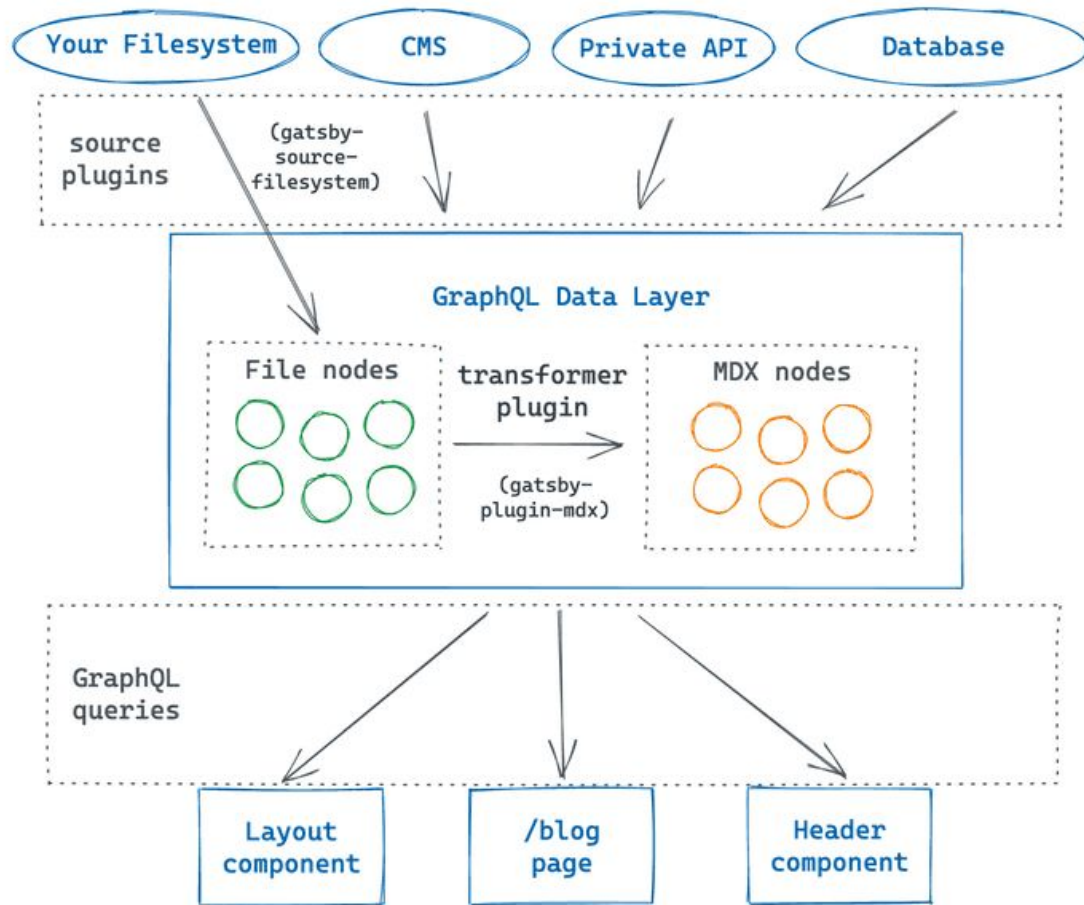
```
import { Figure } from "../components/Figure"
```

```
# Hello world!
```

You can use Markdown to create documents for [Gatsby](https://www.gatsbyjs.com/).

```
<Figure data="chart.svg" caption="MDX adoption has increased 120% since last year." />
```





Front-matter é um bloco de texto no início do arquivo MDX que é usado para definir configurações para o conteúdo que será escrito (como páginas ou postagens).

MDX

```
---  
title: Hello, world!  
slug: /hello-world  
date: 2019-01-29  
---  
  
# Hello, world!
```

GRAPHQL

```
query {  
  allMdx {  
    nodes {  
      frontmatter {  
        title  
        slug  
        date(formatString: "MMMM DD, YYYY")  
      }  
    }  
  }  
}
```


≡ meu-primeiro-post.mdx ×

blog > ≡ meu-primeiro-post.mdx

```
1 ---
2 title: "Meu Primeiro Post"
3 date: "2023-08-15"
4 slug: "meu-primeiro-post"
5 ---
```

Front-matter

7 Esse é o meu primeiro post: não é *bacana*?

9 Algumas linguagens de programação **favoritas**:

```
10
11 * JavaScript
12 * TypeScript
13 * JScript
```

≡ sim-mais-um-post.mdx ×

blog > ≡ sim-mais-um-post.mdx

```
1 ---
2 title: "Mais um Post"
3 date: "2023-10-15"
4 slug: "mais-um-post"
5 ---
```

7 Esse é mais um post.

≡ esse-novo-post.mdx ×

blog > ≡ esse-novo-post.mdx

```
1 ---
2 title: "Meu Outro Post"
3 date: "2023-09-15"
4 slug: "meu-outro-post"
5 ---
```

7 Esse é o meu outro post.

Criando arquivos MDX

4

```
npm install gatsby-plugin-mdx @mdx-js/react
```

5

```
JS gatsby-config.js x
JS gatsby-config.js > ...
1  module.exports = {
2    siteMetadata: {
3      title: `Estudos do Jamstack com Gatsby`,
4    },
5    plugins: [
6      {
7        resolve: "gatsby-source-filesystem",
8        options: {
9          name: `blog`,
10         path: `${__dirname}/blog`,
11       },
12     ],
13     "gatsby-plugin-mdx",
14   ],
15 }
```

Lendo arquivos MDX



6

+  GraphQL

```

▼ {
  "data": {
    "allMdx": {
      "nodes": [
        {
          "frontmatter": {
            "date": "October 15, 2023",
            "title": "Mais um Post"
          },
          "id": "67cf4d90-18f4-5cfc-8d28-5574ebbe5223",
          "excerpt": "Esse é mais um post."
        },
        {
          "frontmatter": {
            "date": "September 15, 2023",
            "title": "Meu Outro Post"
          },
          "id": "6eb06c98-c925-52c4-aa0d-794a3c876af1",

```

Lendo arquivos MDX

7

JS blog.js x

src > pages > JS blog.js > [e] default

```
1  import * as React from 'react'
2  import { useStaticQuery, graphql } from 'gatsby'
3  import Layout from '../components/layout'
4
5  const BlogPage = () => {
6
7    const data = useStaticQuery(graphql`
8      query {
9        allMdx(sort: { frontmatter: { date: DESC } }) {
10          nodes {
11            frontmatter {
12              date(formatString: "MMMM D, YYYY")
13              title
14            }
15            id
16            excerpt
17          }
18        }
19      }
20    `)
```

Lendo arquivos MDX

8

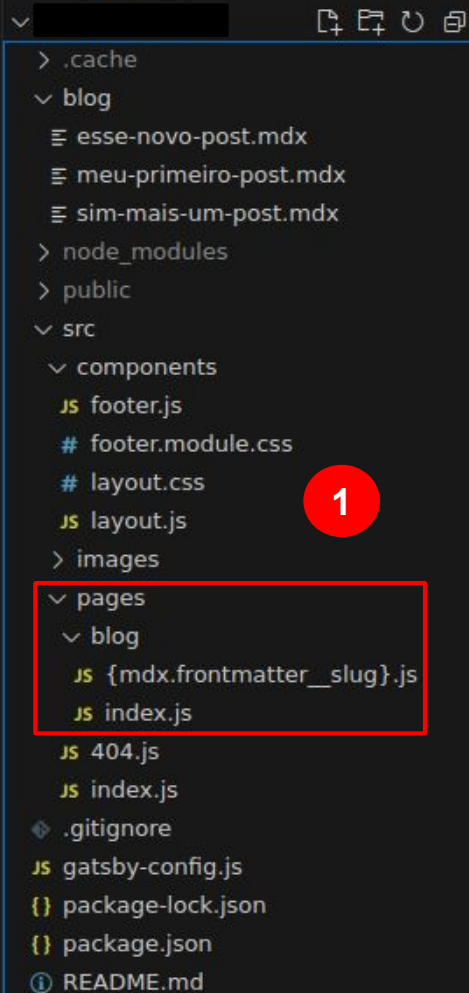
JS blog.js x

src > pages > JS blog.js > [e] default

```
22   return (  
23     <Layout>  
24       <p>Esses são os últimos posts:</p>  
25       {  
26         data.allMdx.nodes.map((node) => (  
27           <article key={node.id}>  
28             <h2>{node.frontmatter.title}</h2>  
29             <p>Posted: {node.frontmatter.date}</p>  
30             <p>{node.excerpt}</p>  
31           </article>  
32         ))  
33       }  
34     </Layout>  
35   )  
36 }
```

Lendo arquivos MDX

Criando Rotas Dinâmicas



JS {mdx.frontmatter_slug}.js

src > pages > blog > JS {mdx.frontmatter_slug}.js > ...

```
1 import * as React from 'react'
2 import { graphql } from 'gatsby'
3 import Layout from '../components/layout'
4
5 const BlogPost = ({ data, children }) => {
6   return (
7     <Layout pageTitle={data.mdx.frontmatter.title}>
8       <p>{data.mdx.frontmatter.date}</p>
9       {children}
10     </Layout>
11   )
12 }
```

```
14 export const query = graphql`
15   query ($id: String) {
16     mdx(id: {eq: $id}) {
17       frontmatter {
18         title
19         date(formatString: "MMMM D, YYYY")
20       }
21     }
22   }
23 `
```

```
25 export const Head = ({ data }) => <title>{data.mdx.frontmatter.title}</title>
26
27 export default BlogPost
```

JS index.js x

src > pages > blog > JS index.js > [🔗] BlogPage

```
1  import * as React from 'react'
2  import { Link, useStaticQuery, graphql } from 'gatsby'
3  import Layout from '../components/layout'
4
5  const BlogPage = () => {
6
7    const data = useStaticQuery(graphql`
8      query {
9        allMdx(sort: { frontmatter: { date: DESC } }) {
10          nodes {
11            frontmatter {
12              date(formatString: "MMMM D, YYYY")
13              title
14              slug
15            }
16            id
17            excerpt
18          }
19        }
20      }
21    `)
```

2

Página com os links
dinâmicos

3

JS index.js X

src > pages > blog > JS index.js > [🔗] BlogPage

```
23   return (  
24     <Layout>  
25       <p>Esses são os últimos posts:</p>  
26       {  
27         data.allMdx.nodes.map(node => (  
28           <article key={node.id}>  
29             <h2>  
30               <Link to={`/blog/${node.frontmatter.slug}`}>  
31                 {node.frontmatter.title}  
32               </Link>  
33             </h2>  
34             <p>Posted: {node.frontmatter.date}</p>  
35           </article>  
36         ))  
37       }  
38     </Layout>  
39   )  
40 }  
41  
42 export const Head = () => <title>Posts</title>  
43  
44 export default BlogPage
```

Página com os links
dinâmicos

