



GatsbyJS

Material Complementar



# Marianne Salomão

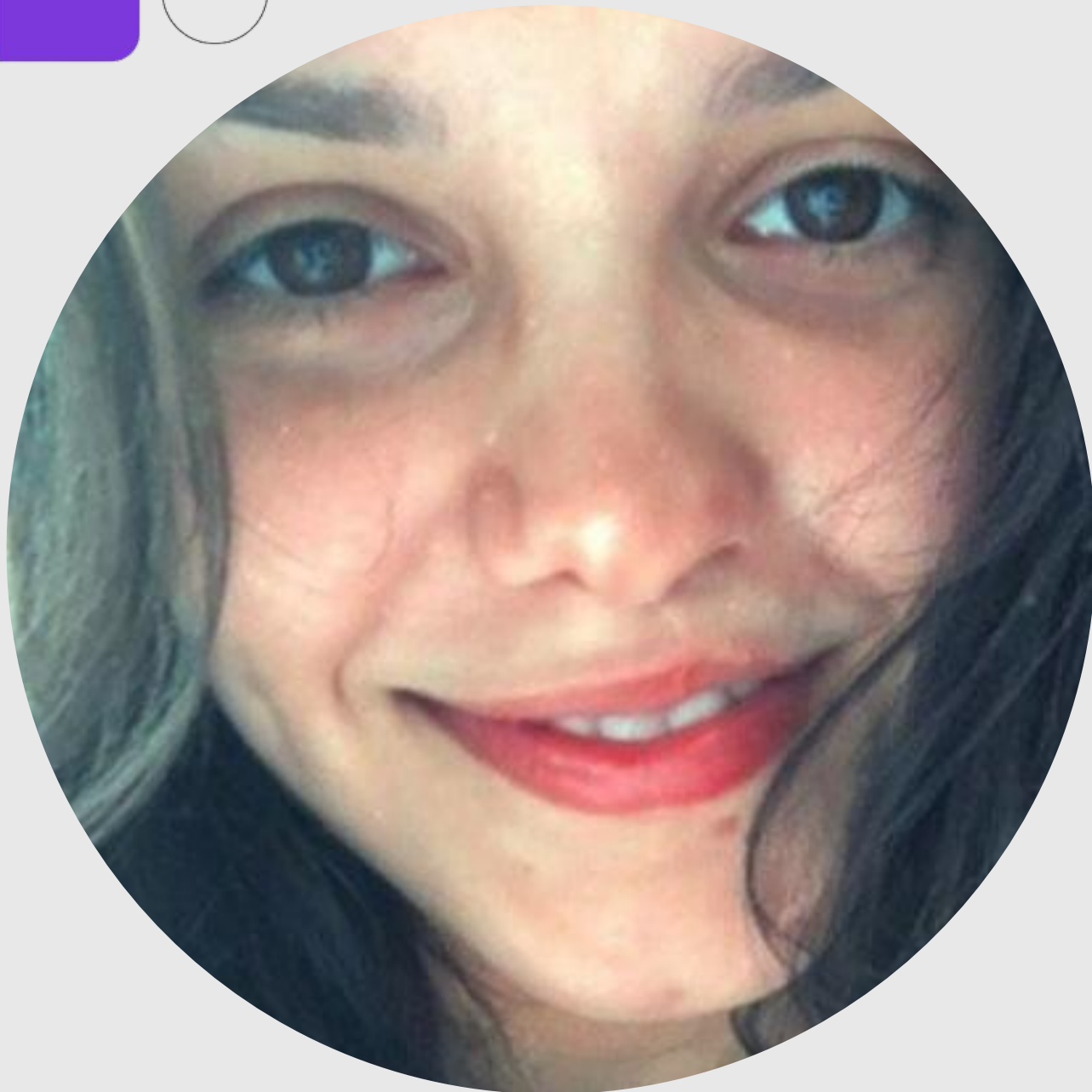
Cloud Engineer na IBM Brasil, desenvolvedora de software full - stack e instrutora de TI.



[linkedin.com/in/mariannesalomao](https://linkedin.com/in/mariannesalomao)



[github.com/mariannesalomao](https://github.com/mariannesalomao)



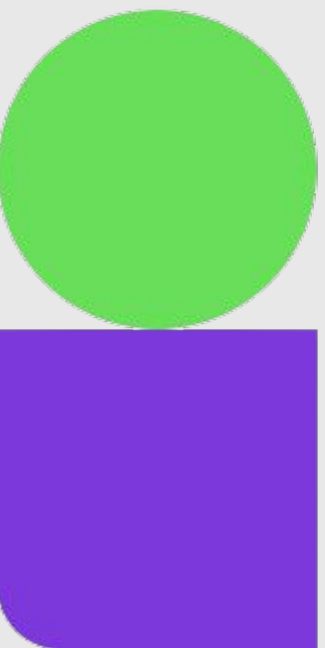
#PraCegoVer: Fotografia da autora Marianne Salomão



# 01. O que é o GatsbyJS?

Gatsby ou GatsbyJS é uma estrutura baseada na biblioteca React e GraphQL que facilita a criação de sites e aplicativos da web.

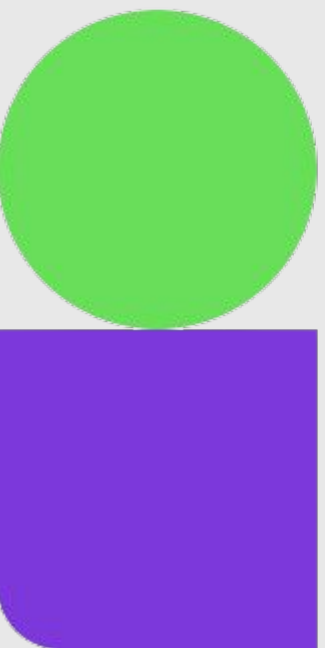
Ele é construído no JAMstack – ou seja, uma estratégia para construir sites/aplicativos baseados em JavaScript do lado do cliente (ou outros scripts), APIs reutilizáveis e Markup pré-construído. Este tipo de site possui alto desempenho, escalabilidade e segurança.



# 01. O que é o GatsbyJS?

No entanto, o Gatsby é considerado um gerador de site estático (SSG) como outras tecnologias Jamstack (Jekyll, Next, Hugo etc) com base no que compartilha com eles. Mas, na realidade, ele pode fazer muito mais do que você normalmente pode fazer com um gerador de site estático.

Você pode pensar no Gatsby como um framework React para construir sites e aplicativos web complexos. Com o Gatsbyjs, você não está limitado a sites estáticos. Você pode criar um site de blog, comércio eletrônico ou qualquer site complexo usando as ferramentas mais recentes, como React, GraphQL, CMSs headless etc.





## 02. Projeto

Você precisa instalar o Gatsby CLI (interface de linha de comando). Esta CLI permitirá que você gere um novo projeto Gatsby. E está disponível como um pacote npm. Então vá para o seu terminal e execute este comando npm:

```
npm install -g gatsby-cli
```



## 02. Projeto

Gerando um novo projeto:

```
gatsby new [PROJECT DIRECTORY] [STARTER URL]
```

No terminal, mude para o diretório que você gostaria de salvar seu projeto (por exemplo, cd Desktop) e execute:

```
gatsby new ibaslogic-gatsby-tutorial https://github.com/gatsbyjs/gatsby-starter-hello-world
```



## 02. Projeto

Isso criará uma nova pasta de projeto chamada `ibaslogic-gatsby-tutorial` (você pode nomeá-la como quiser) e incluirá todos os arquivos iniciais. Uma vez que seu novo site Gatsby tenha sido inicializado com sucesso, abra-o com seu editor de texto favorito.

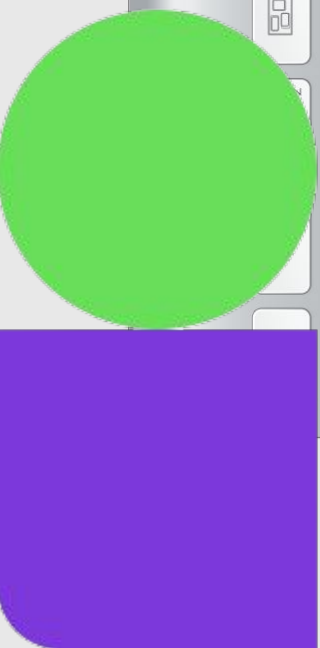


## 02. Projeto

Quando se trata de criação de páginas, esse diretório é onde Gatsby procura quando está descobrindo quais páginas estáticas seu site precisa. Portanto, todos os arquivos que você colocar dentro desse diretório representarão as páginas do seu site.

Então, vamos começar com o arquivo `index.js`. Enquanto isso, substitua o código pelo seguinte (próxima página):





```
import React from "react"
```

```
const Index = () => {
```

```
  return (
```

```
    <div>
```

```
      <h1>Home page</h1>
```

```
      <h2>I'm Ibas, a teacher and a Gatsby.js developer</h2>
```

```
    </div>
```

```
  )
```

```
}
```

```
export default Index
```



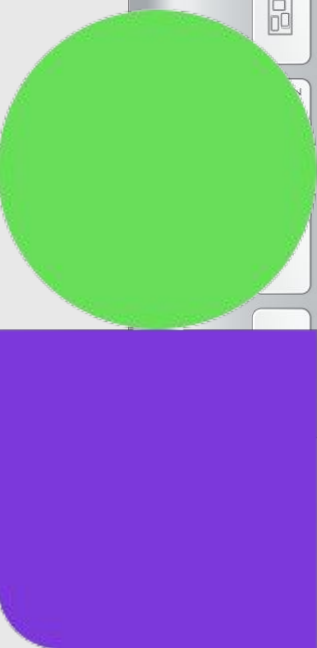
## 02. Projeto

Ele começa olhando para a pasta `src/pages` para descobrir quais páginas estáticas ele deve criar. No nosso caso, ele percebe que temos apenas um arquivo, `index.js`. Ou seja, nosso site terá apenas uma única página.

Agora o nome do arquivo também é importante. Aqui, o arquivo é chamado `index.js`. Isso é semelhante a como `index.html` será a página padrão para a página inicial de um site.

Com Gatsby, `index.js` será a página inicial padrão. Para criar uma nova página, tudo o que você precisa fazer é adicionar um novo arquivo ao diretório `src/pages`.

Neste tutorial, criaremos três páginas adicionais. As páginas Blog, Contato e Sobre. Então vamos criar esses arquivos – `blog.js`, `contact.js` e `about.js` na pasta `src/pages`.



No arquivo `blog.js`, adicione o seguinte código:

```
import React from "react"
```

```
const Blog = () => {
```

```
  return (
```

```
    <div>
```

```
      <h1>Blog Page</h1>
```

```
      <p>Blog posts will be displayed here!</p>
```

```
    </div>
```

```
  )
```

```
}
```

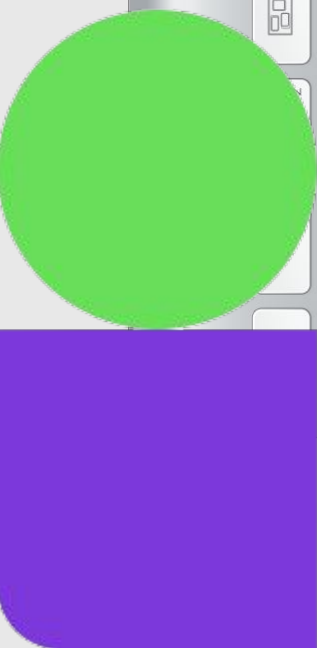
```
export default Blog
```

## 02. Projeto

Salve o arquivo e navegue até `localhost:8000/blog/` na barra de endereços do seu navegador para ver a página. Aqui, o "blog" na URL é o nome do arquivo da página.







Da mesma forma, entre no arquivo `contact.js` e adicione este código:

```
import React from "react"
```

```
const Contact = () => {
```

```
  return (
```

```
    <div>
```

```
      <h1>Contact Page</h1>
```

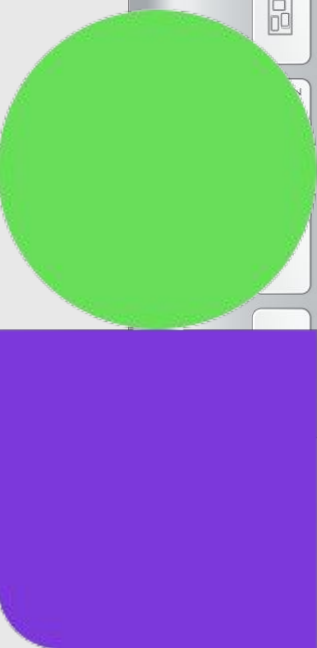
```
      <p>You can find Ibas on Twitter via @ibaslogic</p>
```

```
    </div>
```

```
  )
```

```
}
```

```
export default Contact
```



Para o arquivo `about.js`, adicione isto:

```
import React from "react"
```

```
const About = () => {
```

```
  return (
```

```
    <div>
```

```
      <h1>About Page</h1>
```

```
      <p>Ibas is a web developer, teacher and a creative writer.</p>
```

```
    </div>
```

```
  )
```

```
}
```

```
export default About
```

# 03. Layouts

Começando com o componente Header. No diretório src, crie uma nova pasta chamada componentes. Dentro desta pasta, adicione um arquivo chamado header.js e adicione o seguinte código:

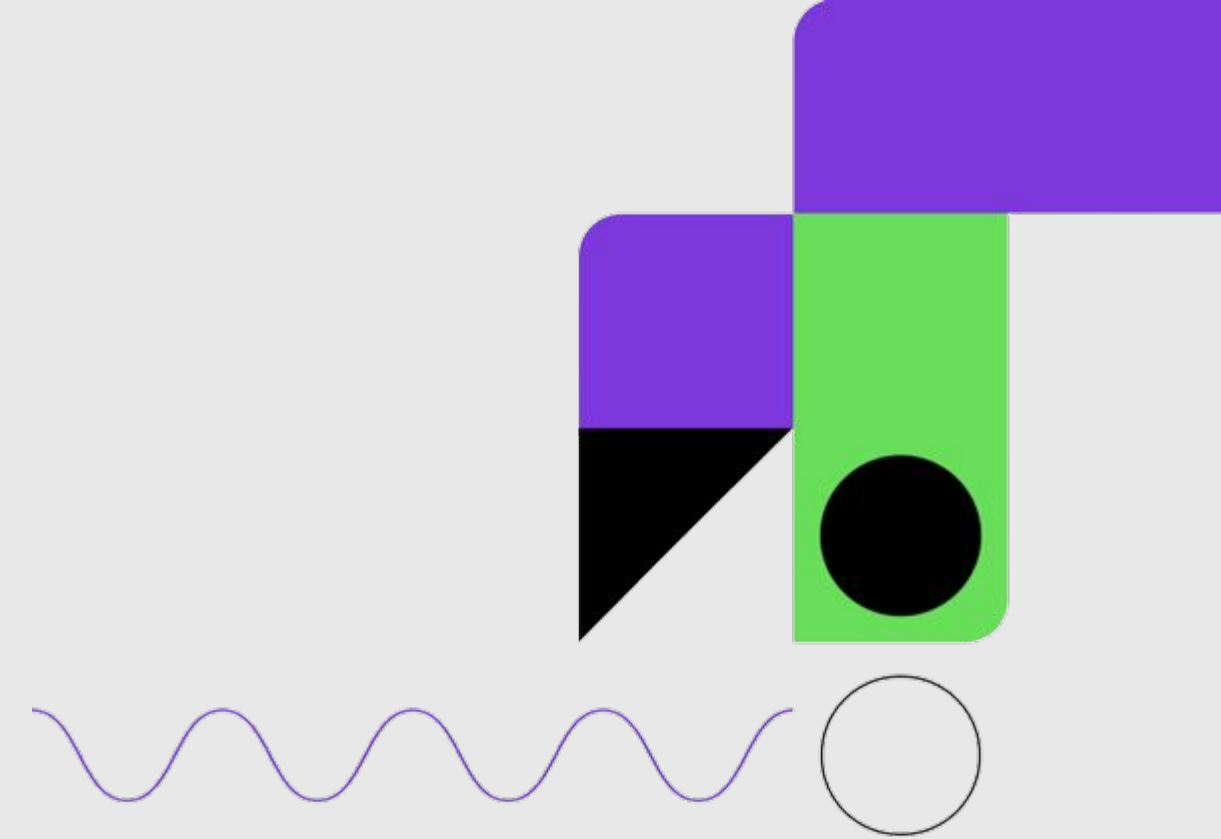
```
import React from "react"
```

```
const Header = () => {
```

```
  return <div>This is the header content</div>
```

```
}
```

```
export default Header
```

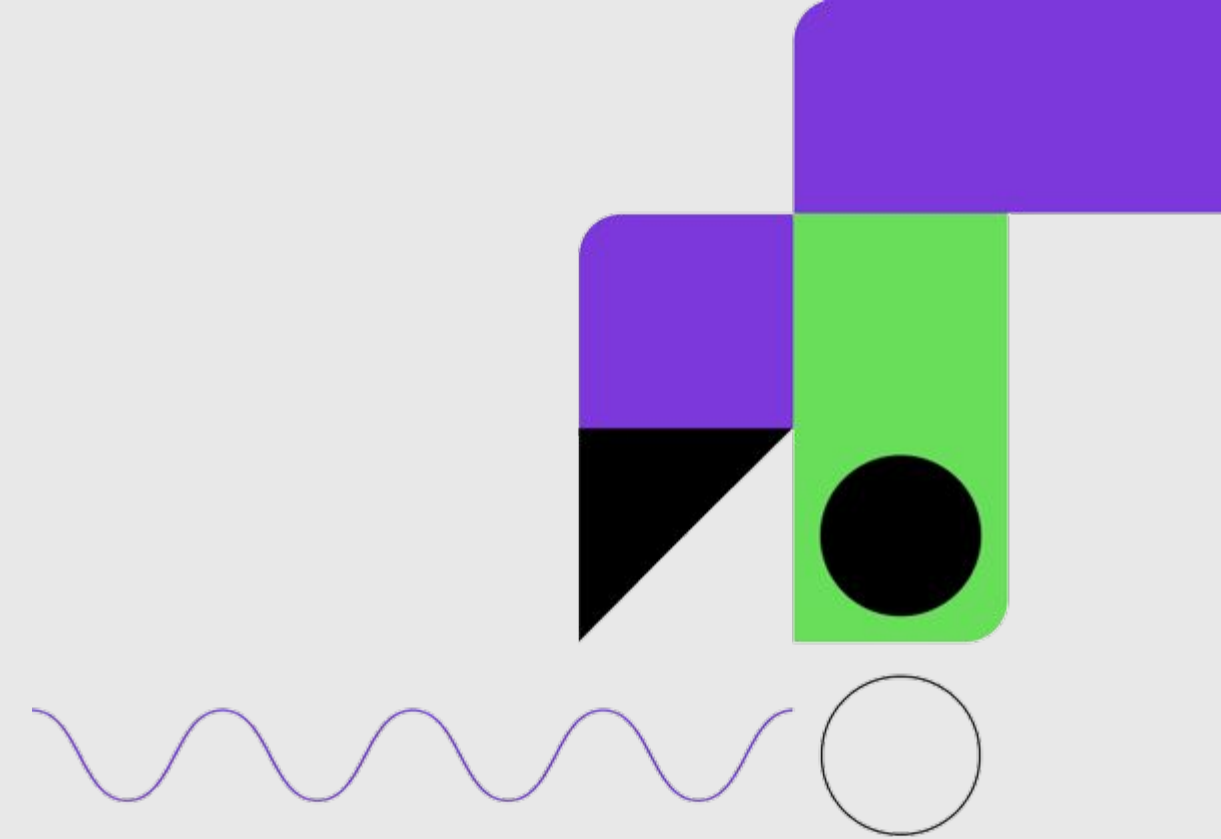


# 03. Layouts

Por enquanto, estamos renderizando um texto simples.

O arquivo não está fazendo nada ainda. Precisamos importá-lo em algum lugar. Não nos arquivos das páginas, mas no arquivo de layout. Então vamos criar o arquivo. Entre na pasta de componentes e crie o arquivo layout.js.

Então comece adicionando este código (próxima página):



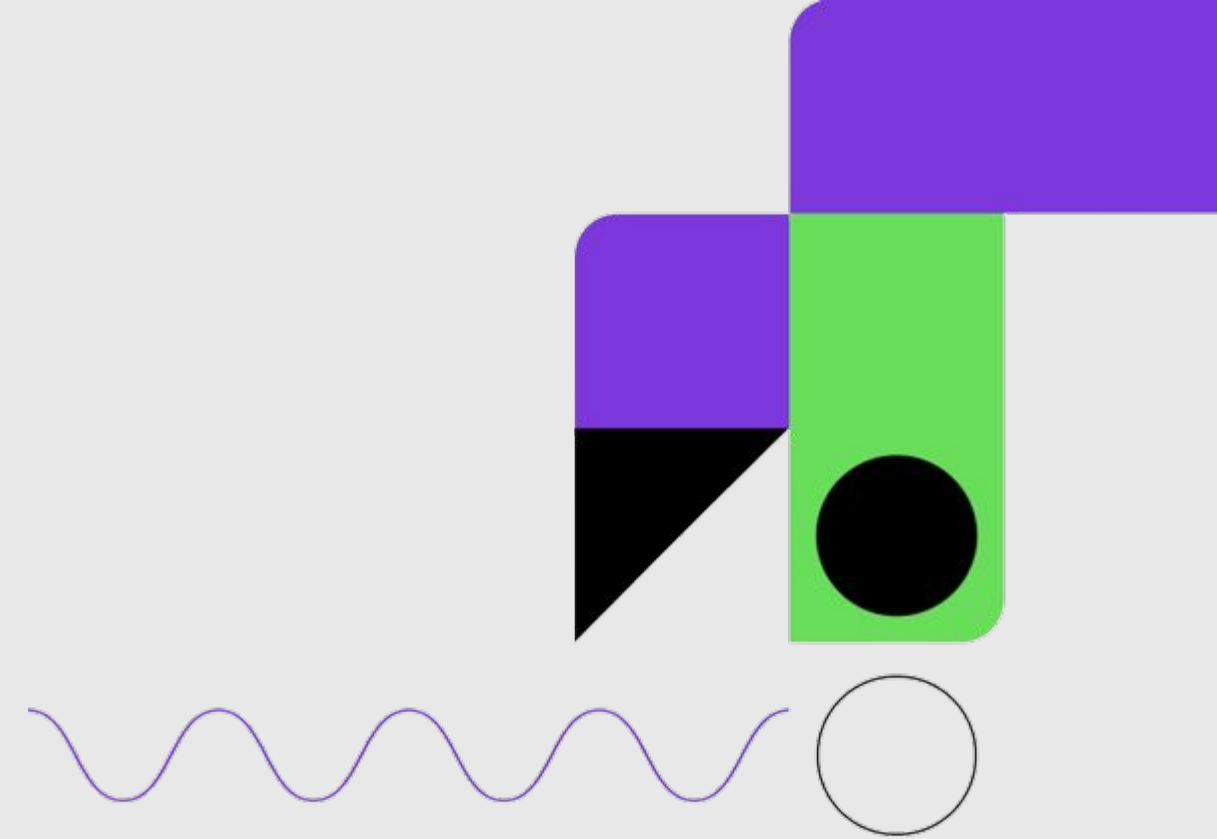


# 03. Layouts

```
import React from "react"
import Header from "../header"
```

```
const Layout = props => {
  return (
    <div>
      <Header />
      {props.children}
    </div>
  )
}
```

```
export default Layout
```



# 03. Layouts

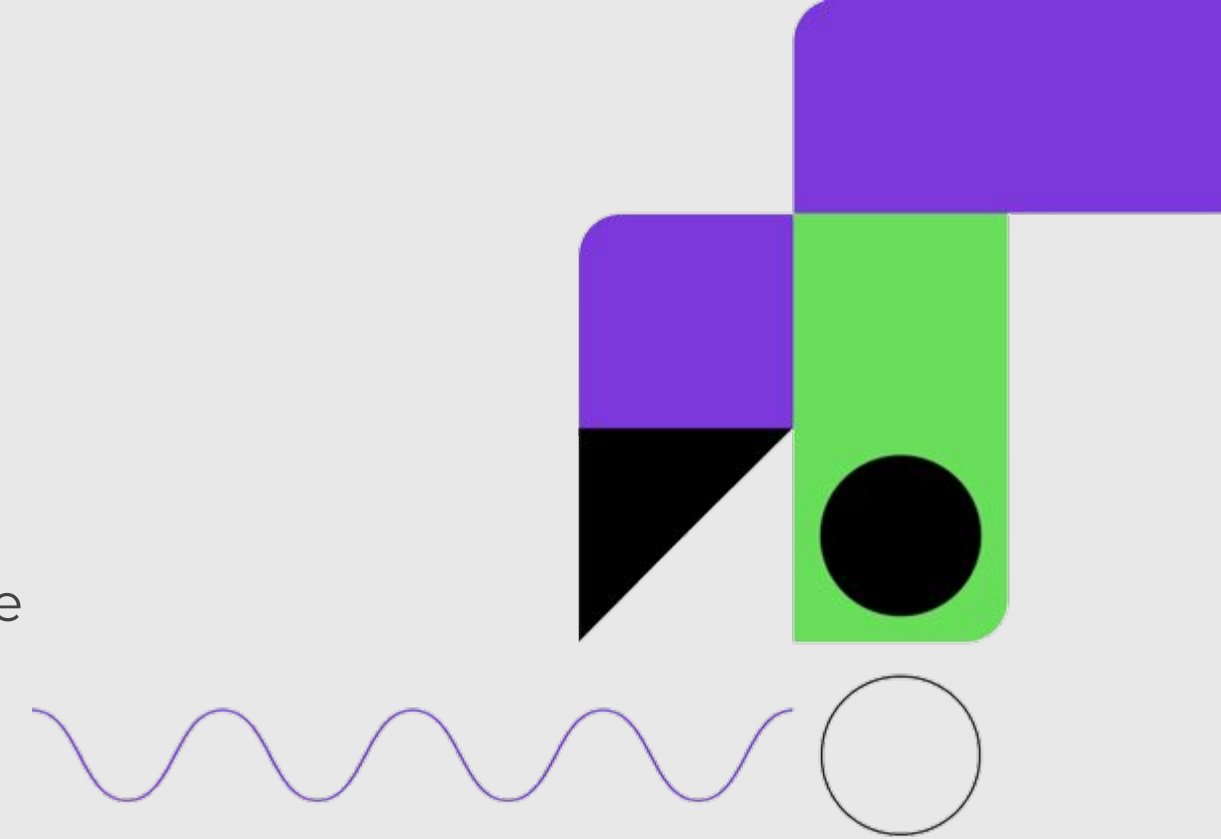
Em seguida, importe este novo componente de Layout para todos os arquivos de componentes de suas páginas.

Por exemplo, o arquivo index.js agora ficará assim:

```
import React from "react"
import Layout from "../components/layout"

const Index = () => {
  return (
    <Layout>
      <h1>Home page</h1>
      <h2>I'm Ibas, a teacher and a Gatsby.js developer</h2>
    </Layout>
  )
}

export default Index
```





# Fechamento

Vimos diversos assuntos até aqui, ainda há muito a ser abordado. Esse material pode servir como consulta e guia de estudos.

Até a próxima pessoal!

# Referência Bibliográfica

<https://ibaslogic.com/gatsby-tutorial-from-scratch-for-beginners/>

