

Guia iniciante para montar seu:

# Site ~ Portfólio



Lucas Santos Chitolina  
/lucas-chitolina  
@acechitolina  
[github.com/Chitolina](https://github.com/Chitolina)



Quer aprender a criar seu primeiro Site-Portfólio, ao mesmo tempo em que desenvolve algumas habilidades e noção sobre como funciona o esqueleto de uma página? Te prometo que ele não será apenas sobre criar blocos e arrastar, e também não exigirá de você conhecimento em programação, apenas um pouco de esforço e curiosidade!

Então vamos começar!



## 1ª INSTALAÇÃO CHOCOLATEY (WINDOWS)

Chocolatey é um gerenciador de pacotes do Windows. Ele facilita a instalação e atualização automática de aplicativos em sua máquina.

1. Clique em Iniciar e digite "powershell".
2. Clique com o botão direito do mouse em Windows Powershell e escolha "Executar como Administrador".
3. Cole o seguinte comando no Powershell e pressione Enter:

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object  
System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'  
))
```

OBS: Se preferir checar o código antes, não conseguir copiar, ou der erro, acesse (<https://docs.chocolatey.org/en-us/choco/setup>)

4. Responda 'Sim' quando solicitado.
5. Feche e reabra uma janela do PowerShell para começar a usar o choco.

## 2º INSTALE O HUGO

HUGO é um gerador de sites estáticos escrito com Golang, com rapidez e versatilidade para gerar/compilar conteúdos (você não precisa saber linguagem GO/HTML pra criar seu primeiro site; apenas para criar um post você vai precisar consultar um pouco de markdown, o que é bem tranquilo).

Abra o PowerShell chamando o gerenciador Chocolatey:  
(sem o >)

```
> choco install hugo -confirm
```

Para testar se ele está realmente instalado corretamente, execute o seguinte:

```
> hugo version
```

## 3º INSTALE O GIT/GITHUB

GitHub é um serviço baseado em nuvem que hospeda um sistema de controle de versão (SCV) chamado Git. Ele permite que os desenvolvedores colaborem e façam mudanças em projetos compartilhados enquanto mantêm um registro detalhado do seu progresso.

Instale o Git pelo PowerShell chamando o gerenciador Chocolatey:

```
> choco install git.install
```

Cheque a instalação:

```
> git version
```



## 4º CRIANDO UM SITE COM HUGO

**1.** No PowerShell, vá ao diretório onde Hugo ficou instalado, normalmente dentro da pasta lib de Chocolatey:

```
> cd C:\ProgramData\chocolatey\lib\hugo\
```

**2.** Digite:

```
> hugo new site nome_do_site
```

Isto irá criar um novo site Hugo na pasta chamada nome\_do\_site.

**3.** Adicionar um tema para seu site, que por enquanto está em branco.

Visite: [themes.gohugo.io] (<https://themes.gohugo.io/>) para uma lista de temas a serem considerados.

**4.** Primeiro, baixe o tema do GitHub e adicione-o ao diretório 'themes' do seu site:

```
> cd nome_do_site  
> git init  
> git submodule add https://github.com/EmielH/tale-hugo.git  
themes/tale-hugo
```

(este último link está disponível na página do tema que escolher, é interessante que dê uma olhada no que o tema sugere na hora de criar o submódulo, assim como na hora de fazer o 'echo').

**5.** Em seguida, adicione o tema à configuração do site:

(Hugo usa o config.toml, config.yaml ou config.json como o arquivo de configuração do site padrão. Nele você encontra configurações da sua bio, ícones, suas redes sociais, capas... é aí que você irá começar sua edição, e com o código abaixo você estará copiando o config.toml do tema que você escolheu para o config.toml do seu site ).

```
> echo 'theme = "nome_do_tema"' >> config.toml
```



## 5. ADICIONAR CONTEÚDO

Você pode usar o comando 'new' para fazer algumas coisas para você (como adicionar título e data):

```
> hugo new posts/my-first-post.md
```

Isto irá criar um arquivo em markdown na pasta 'post' para você postar seu conteúdo: imagens, texto, códigos...

Ele começará com um cabeçalho que indica que é um post, similar a esse:

```
title: "My First Post"  
date: 2019-03-26T08:47:11+01:00  
draft: false
```

Você começa adicionando o material a partir daqui e aconselho visitar a página do tema para aprender a customizar de acordo com as indicações do desenvolvedor.

## 6. NASCE O SITE

Execute o comando:

```
> hugo server -D
```

Irá aparecer um conteúdo no terminal e nele um endereço similar a:

Web Server is available at <http://localhost:1313/> (bind address 127.0.0.1)

Copie e cole o endereço no navegador e verá a primeira versão de seu site. Fique à vontade para editar (aconselho usar um editor como VSCODE) ou adicionar novos conteúdos e atualizar no navegador para ver as mudanças rapidamente. (Pode ser necessário forçar a atualização do navegador, algo como Ctrl-S no editor geralmente funciona).

## E AGORA, TENHO UM SITE JÁ?

Quase. Você precisa depositar todo essa pasta do site Hugo no seu Github (isto é interessante para mantê-lo como projeto e também para fazer o deploy do seu conteúdo para o formato de site):

**1. Crie uma conta no GitHub.**

**2. Criando uma chave SSH:**

No Windows, use o GitBash (terminal do git). O primeiro comando que iremos executar é o ssh-keygen, como vemos abaixo:

```
$ ssh-keygen -t rsa -b 4096 -C "email@bol.com.br"
```

É importante que coloque o email cadastrado no seu GitHub. Por padrão, esse arquivo ficará na pasta do usuário do Sistema, dentro da pasta .ssh, e o nome do arquivo será id\_rsa.

O segundo passo será uma senha para a nossa chave. Depois da chave iremos criar um agente SSH. Ele é um programa muito útil, que será o responsável por realizar a nossa autenticação com o servidor remoto, no nosso caso a comunicação Git local com o Git remoto.

**3. Para criá-lo, execute este comando:**

```
$ eval $(ssh-agent -s)
```

**4. Por fim, vamos executar o comando abaixo:**

```
$ ssh-add ~/.ssh/id_rsa
```

Nós informamos o caminho até o arquivo id\_rsa, que foi o nome que definimos para nossa chave (você pode alterar esse nome, mas como este é um guia iniciante, aconselho deixar assim).

5. Crie um repositório no Github remoto, que será usado para hospedar o conteúdo do site, pode colocar o mesmo nome do site.

6. No GitBash:

```
$ cd C:\ProgramData\chocolatey\lib\hugo\novo_site
```

Para a pasta do site que você criou.

7. Siga esta pequena série de comandos para fazer o correto upload dos arquivos locais para o repositório remoto.

Transferindo git base pro git remoto:

```
$ git init
```

```
$ git remote add origin linkdorepositóriogithub
```

(este link aparece na hora que você cria seu repositório no github, é o endereço remoto ao qual precisamos subir os arquivos)

```
$ git status
```

(mostra o estado atual do seu diretório de trabalho Git )

```
$ git add *
```

(adiciona todos os arquivos modificados na área de stage do Git)

```
$ git commit - m "primeiro commit"
```

(faz o commit dos arquivos modificados , ou novos)

```
$ git push origin main (ou master)
```

(usado para enviar conteúdo local ao GitHub)

--> vai pedir login e senha do github

E agora, tenho um site já?  
(quase, só falta uma última etapa)



## 7. DEPLOY DO REPOSITÓRIO NO NETLIFY

O Netlify é um serviço para hospedar sistemas que tem sua stack baseada em tecnologias front-end, que basicamente não precisam de banco de dados ou linguagens back-end para funcionar.

(você pode usar o [GitHub Pages](#) como opção alternativa)

1- Crie sua conta no Netlify vinculada ao Github.

2a- Precisamos gerar um arquivo `netlify.toml` na raiz do site (crie pelo VSCODE, na pasta com o nome do site que você criou) com o seguinte conteúdo:

```
[context.production.environment]
```

```
HUGO_VERSION = "x.xx.x"
```

(descubra a versão do seu hugo com o `> hugo version`)

2b- Ou crie um arquivo com estes parâmetros da imagem. Você também pode consultar o que significa cada parâmetro na página do netlify e lá eles também te indicam um `netlify.toml` de referência:

(<https://docs.netlify.com/configure-builds/file-based-configuration/>)

```
[build]
  command = "hugo"
  publish = "public"

[build.environment]
  HUGO_VERSION = "0.89.4"

[context.production.environment]
  HUGO_VERSION = "0.89.4"
  HUGO_ENV = "production"
  HUGO_ENABLEGITINFO = "true"

[context.split1]
  command = "hugo --gc --minify --enableGitInfo"

[context.split1.environment]
  HUGO_VERSION = "0.89.4"
  HUGO_ENV = "production"

[context.deploy-preview]
  command = "hugo --gc --minify --buildFuture -b $DEPLOY_PRIME_URL"

[context.deploy-preview.environment]
  HUGO_VERSION = "0.89.4"

[context.branch-deploy]
  command = "hugo --gc --minify -b $DEPLOY_PRIME_URL"

[context.branch-deploy.environment]
  HUGO_VERSION = "0.89.4"

[context.next.environment]
  HUGO_ENABLEGITINFO = "true"
```



**3-** Depois de logado na conta do Netlify, clique no botão “New site from GIT”.

**4-** Você será redirecionado para integrar o repositório do seu site no Netlify. Você encontra e clica no seu projeto na lista.

**5-** Aparecerá alguns campos, caso eles "Build command" e "Publish directory" não estejam completos, preencha respectivamente com hugo e public.

**6-** Clique "Deploy site", que vai buscar os arquivos no repositório indicado, rodar o comando hugo que gera os arquivos estáticos do site dentro de uma pasta public dentro do servidor da Netlify.

**7-** Edite o subdomínio para que fique igual ao que você configurou na baseURL do seu config.toml .

(que no caso foi copiado do config.toml do tema que você escolheu, lembra do echo? Você pode modificar para o seu gosto no config.toml)

> Domain settings > Domain management > Options > Edit site name

Se o nome que você escolheu já estiver em uso, você terá que voltar no config.toml do seu site e modificar o nome da baseURL (que você quer pro seu site) e fazer novo commit e push .



Depois de clicar no botão de DEPLOY, o Netlify vai fazer o restante.

Agora, se não houver erro, você TEM o seu site!

Algumas observações:

- Edite seu config.toml de acordo com o que planeja construir para sua página;
- Através do comando `> hugo new posts/nome_do_post.md`, você criará um post, que geralmente fica salvo na pasta content/posts;
- Imagens que queira usar normalmente ficarão na pasta static, veja exemplo de como chamá-las no config.toml;
- Para atualizar novos posts do Hugo, que às vezes não se atualizam sozinhos na hora do deploy:  
`> hugo` (no diretório raíz)
- Sempre que modificar seus arquivos locais, para atualizar no Netlify, você precisa commitar novamente;
- É um site que te permite escrever projetos, postar fotos, currículo, scripts de alguma linguagem, linkar endereços eletrônicos. Todavia não possibilita cadastro de informações como login e compra de produtos. Como está indicado, é uma boa opção gratuita e iniciante para leigos curiosos buscando aprender diferentes habilidades além da sua área de atuação.

Espero que o material tenha sido útil. Caso eu tenha escrito alguma informação equivocada, ou você tenha algum problema/dúvida em alguma etapa, ficarei feliz com o contato. Abaixo deixo uma imagem do meu site-portfólio e alguns endereços de redes. Se gostou, curta e/ou compartilhe.



Netlify: <https://chitolina.netlify.app>

Instagram: @acechitolina

Github: [github.com/Chitolina](https://github.com/Chitolina)

Repositório do site: <https://github.com/Chitolina/2ndSite>