	<p align="center">Centro Federal de Educação Tecnológica de Minas Gerais Campus VIII – Varginha Curso Técnico em Informática</p>	
<p><i>Disciplina</i> Lab. Aplicações Web 1</p>	<p align="center">Git e Github</p>	<p><i>Professor</i> Lázaro Eduardo da Silva</p>

Nesta aula, vamos estudar como subir um projeto no GitHub. Primeiro é importante estar claro que existe um software que chama git e ele permite criar um client e um server que faz o controle de versão do seu código. Os laboratórios do CEFET possuem a versão cliente do git instaladas. Para instalar na sua casa acesse o endereço abaixo:

<https://git-scm.com/downloads>

Para verificar se o seu computador possui esse software instalado, acesse o terminal e digite o comando abaixo
git --version

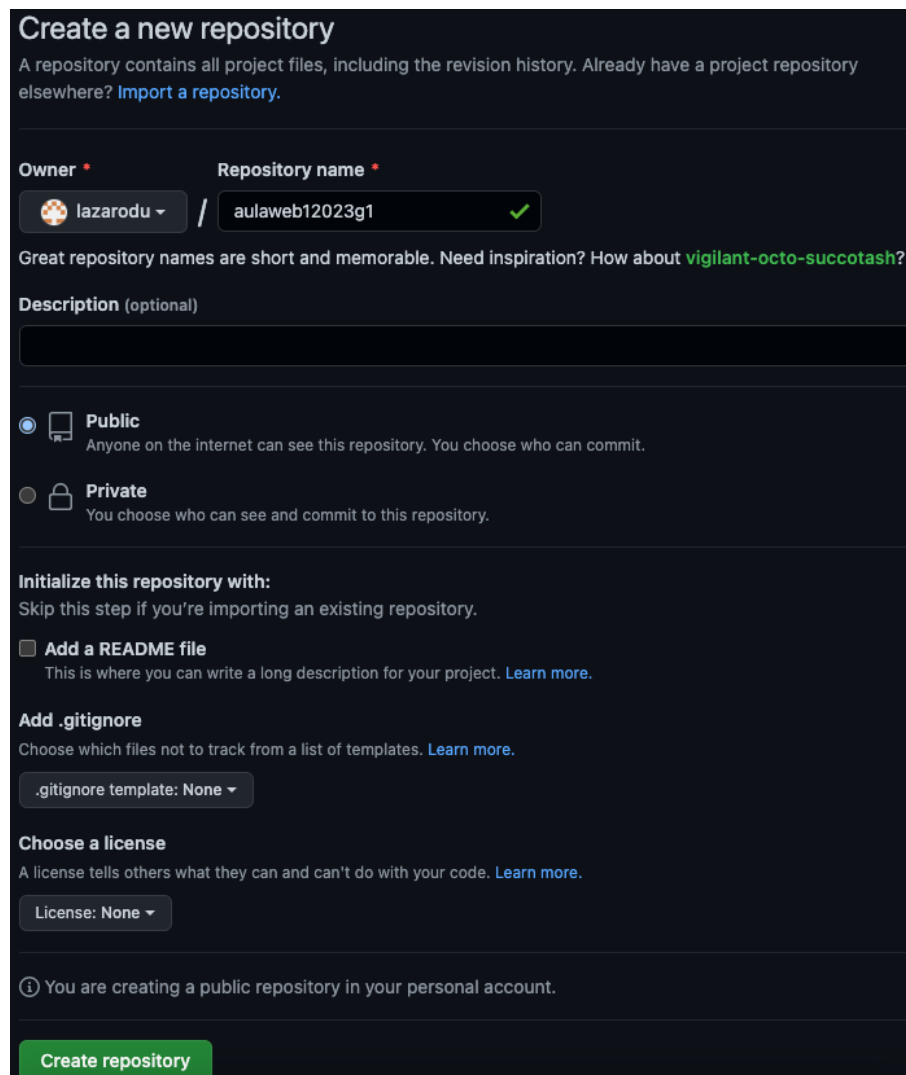
Caso apareça a versão do software você está apto a executar os comandos.

O github é um software na nuvem que executa o git server e permite que se crie repositórios online para receber os arquivos de um repositório no computador que você estiver utilizando.

A primeira parte a realizar é criar uma conta no github. O nome que você escolher para o seu usuário é muito importante, pois será através dele que você conseguirá vincular a pasta no seu computador com o repositório criado no github. O meu usuário é lazarodu, consequentemente o endereço do meu github é <https://github.com/lazarodu>.

Passos para iniciar um novo projeto


Criada a conta no github, você deve criar um repositório para vincular ao seu projeto.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * **Repository name ***

 lazarodu / aulaweb12023g1 ✓

Great repository names are short and memorable. Need inspiration? How about [vigilant-octo-succotash?](#)

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾


Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

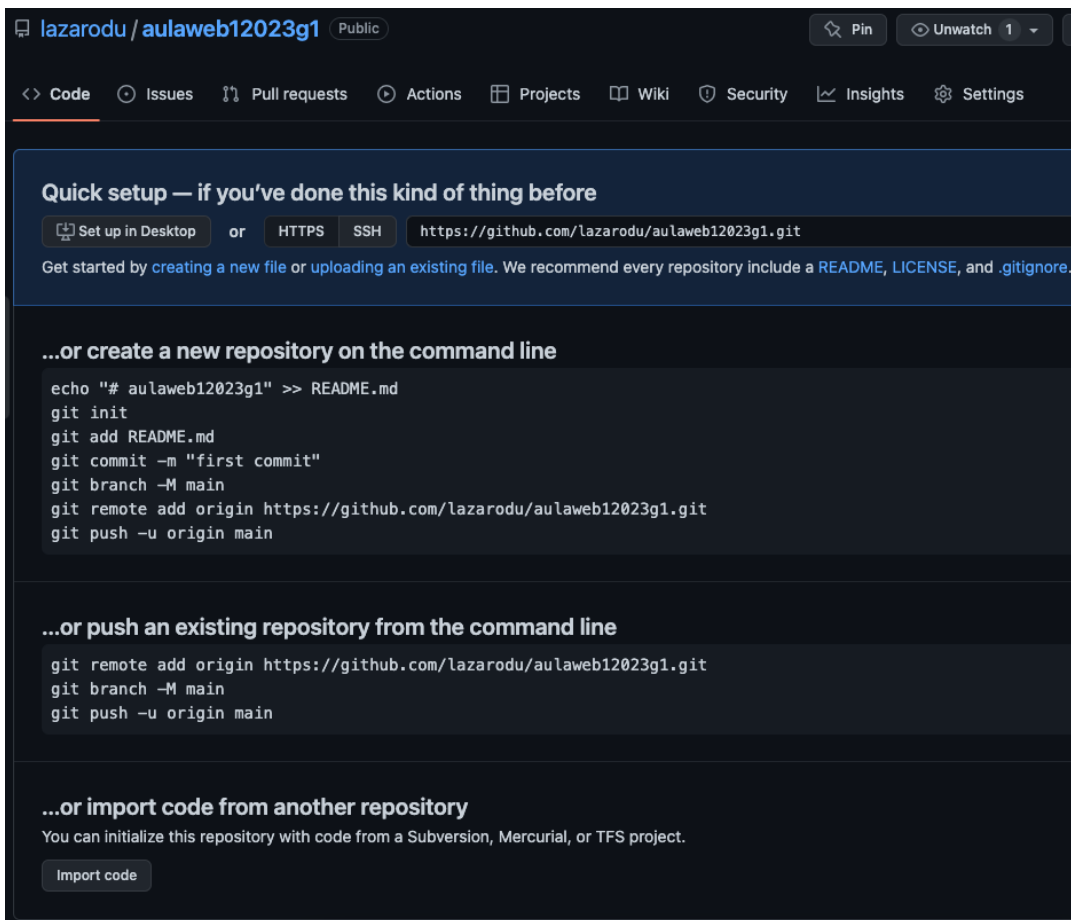
ⓘ You are creating a public repository in your personal account.

Create repository

Todos os repositórios que criarmos em aula devem ser públicos para que vocês criem um portfólio de atividades no seu github.

	<p align="center">Centro Federal de Educação Tecnológica de Minas Gerais Campus VIII – Varginha Curso Técnico em Informática</p>	
<p><i>Disciplina</i> Lab. Aplicações Web 1</p>	<p align="center">Git e Github</p>	<p><i>Professor</i> Lázaro Eduardo da Silva</p>

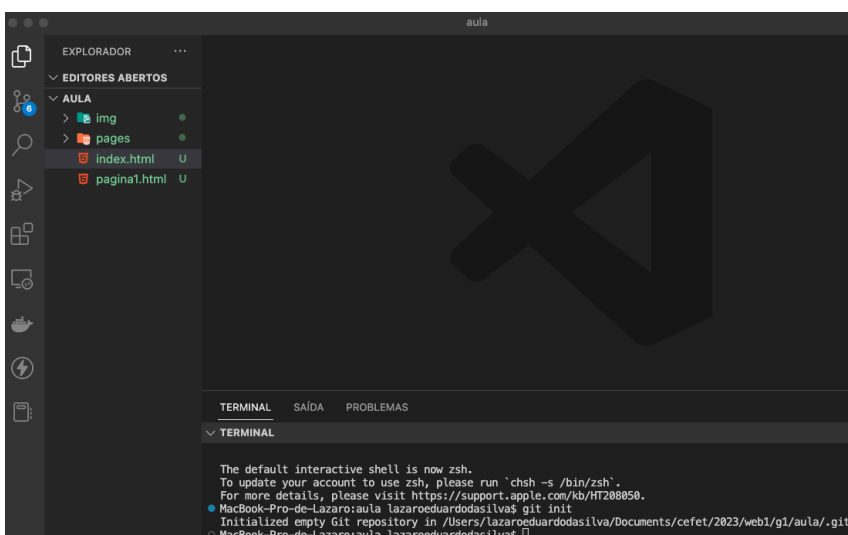
Ao criar um repositório o github irá colocar informações de como vincular um novo repositório, ou vincular um existente, ambos em linha de comando.



Para o nosso caso, o projeto já estará criado no nosso computador e precisaremos vinculá-lo ao github. Para isso, o primeiro passo é informar ao github qual conta está localizada o repositório, para isso, no terminal execute os comandos:

```
git config --global user.name "SEU_NOME_DE_USUARIO"
```

```
git config --global user.email "SEU_E-MAIL"
```




Feito isso, podemos executar o comando para inicializar o repositório atual no git. O comando é:

```
git init
```

Observe que os arquivos e diretórios ficaram destacados e, no caso da imagem, o número 6 apareceu no segundo ícone do visual studio.

Este número, indica que temos 6 arquivos para enviar para o servidor git.

	<p align="center">Centro Federal de Educação Tecnológica de Minas Gerais Campus VIII – Varginha Curso Técnico em Informática</p>	
<p><i>Disciplina</i> Lab. Aplicações Web 1</p>	<p align="center">Git e Github</p>	<p><i>Professor</i> Lázaro Eduardo da Silva</p>

O próximo passo é configurar o repositório remoto que esta pasta local estará vinculada. Para a pasta que criei o comando é:

```
git remote add origin https://github.com/lazarodu/aulaweb12023g1.git
```

O git remote add é o comando, origin é o nome dado ao diretório remoto e o endereço é do repositório do github.

Feito isso, vamos criar uma branch (ramificação). As branches são importantes para dividir o que está em produção e o que está sendo testado. Geralmente, damos o nome de mais para a branch principal. Para isso, vamos executar o comando abaixo no terminal:

```
git branch -M main
```

Com a branch criada, vamos acrescentar os arquivos nesta versão de código que queremos subir para o github. Para isso, vamos executar o comando abaixo

```
git add .
```

O comando git add acrescenta os arquivos e o ponto indica que todos os arquivos a partir da pasta atual serão acrescentados nesta versão.

Após esse comando, vamos fazer um commit para comentar o que foi desenvolvido. Para isso, o comando abaixo deve ser executado:

```
git commit -m "comentário"
```

Este comando deve listar os arquivos que serão inseridos nesta versão de código.

```
MacBook-Pro-de-Lazaro:aula lazaroeduardodasilva$ git add .
MacBook-Pro-de-Lazaro:aula lazaroeduardodasilva$ git commit -m "first commit"
[main (root-commit) 081e088] first commit
5 files changed, 79 insertions(+)
create mode 100644 img/cefet.jpeg
create mode 100644 img/pessoas.jpeg
create mode 100644 index.html
create mode 100644 pages/pagina2.html
create mode 100644 pagina1.html
MacBook-Pro-de-Lazaro:aula lazaroeduardodasilva$
```

Por fim, vamos enviar este código para o servidor github executando o comando abaixo:

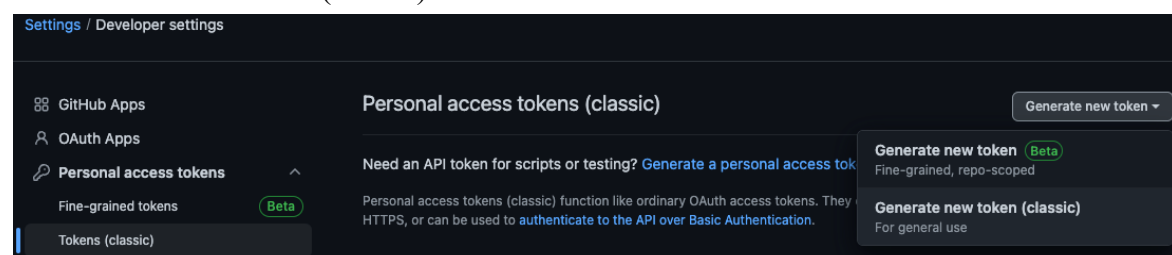
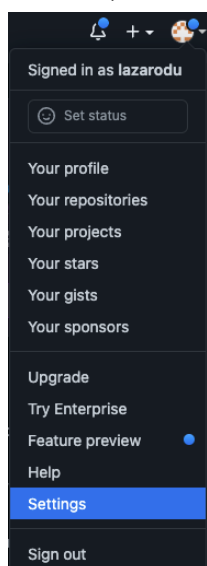
```
git push origin main
```


Ao executar este comando o Visual Studio Code deve pedir o login e senha para envio dos arquivos. Caso ele apareça uma janela, clique no cancelar que irá aparecer um campo na parte superior para você colocar o usuário, caso não, ele deve pedir o usuário do seu github no terminal.

Feito isso, ele pede a senha, porém, essa senha não é a que você utiliza para fazer login no github e sim uma senha específica de enviar código. Para cria-la você deve clicar no ícone do seu usuário e no menu que aparecer, clicar na opção Settings (Configurações)

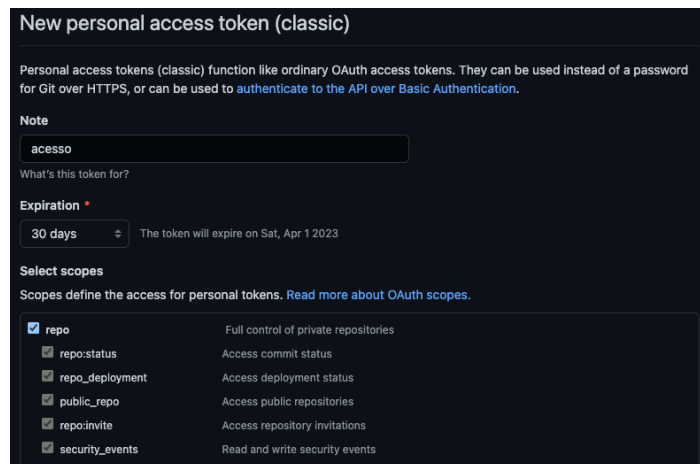
Na página que carregar, escolha o último menu do lado esquerdo chamado Developer settings

Posteriormente, clique no menu Tokens (classic), depois em Generate new token e por fim, em Generate new token (classic)

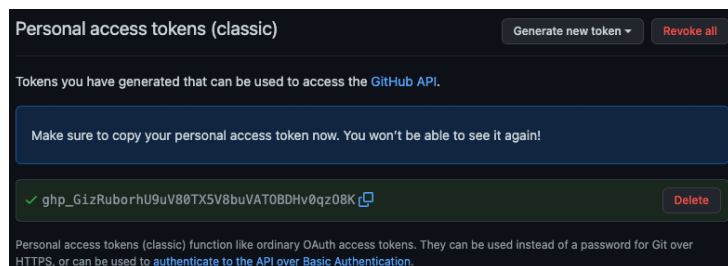


	<p align="center">Centro Federal de Educação Tecnológica de Minas Gerais Campus VIII – Varginha Curso Técnico em Informática</p>	
<p>Disciplina Lab. Aplicações Web 1</p>	<p align="center">Git e Github</p>	<p>Professor Lázaro Eduardo da Silva</p>

Na janela que abrir, coloque um nome no campo Note, marque todos as caixas de seleção e clique em Generate token no final da tela.

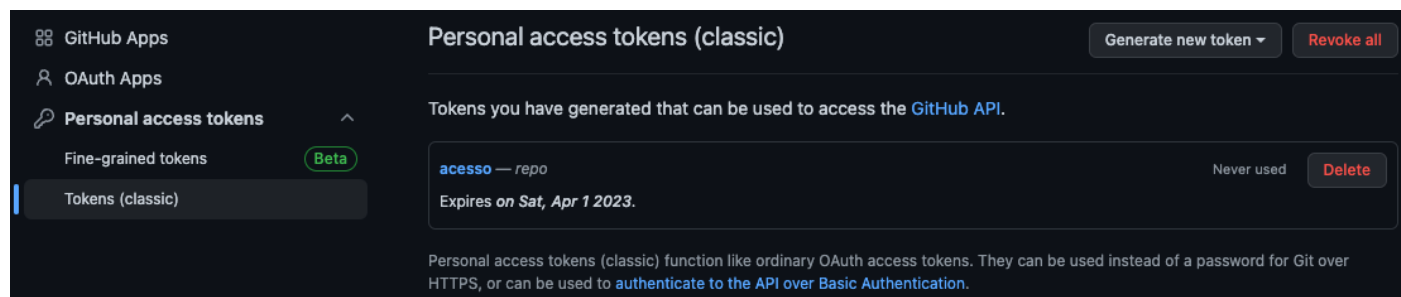


Estes caracteres que aparecem, são a senha que você deve utilizar para autenticar o envio dos arquivos do repositório

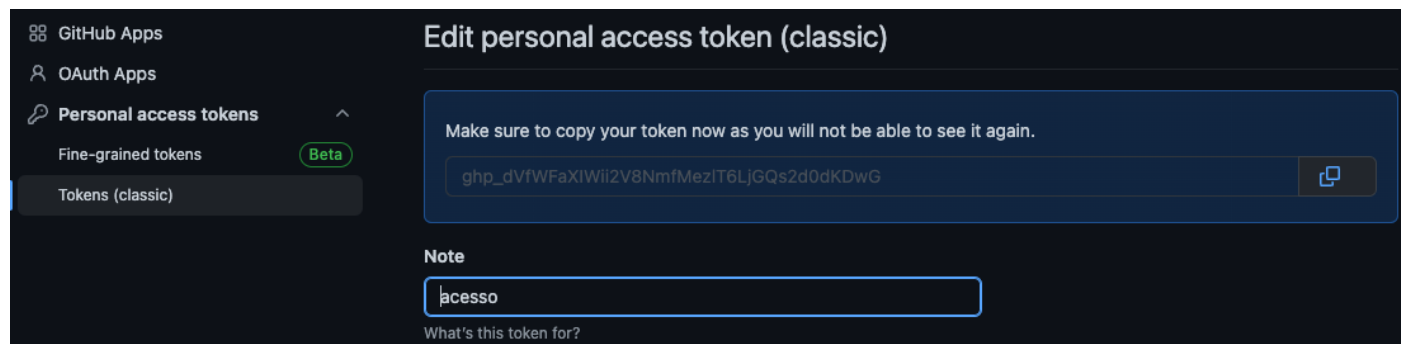


Copie ela e cole no local depois de colocar o usuário. Caso você esteja fazendo este processo pelo terminal, ele não mostrará nada, mas se você clicar com o botão direito do mouse e ir em colar a senha será colada e ao pressionar enter o envio ocorrerá.

Esta senha aparecerá somente quando você a criar. Caso você não a guarde, você tem a opção de renová-la. Ao acessar novamente, o local de gerar o token, ele deve aparecer conforme imagem abaixo:




Clique no nome dado para o token, no meu caso acesso. Na próxima janela, clique em Regenerate token, e na próxima clique em Regenerate token novamente. Deve aparecer a tela abaixo:



Observe que o token é diferente do anterior, copie-o e utilize-o para fazer novos pushes para o github.

Caso sua autenticação ocorra da forma correta, o seu terminal deve mostrar algumas linhas parecidas com as da imagem abaixo:

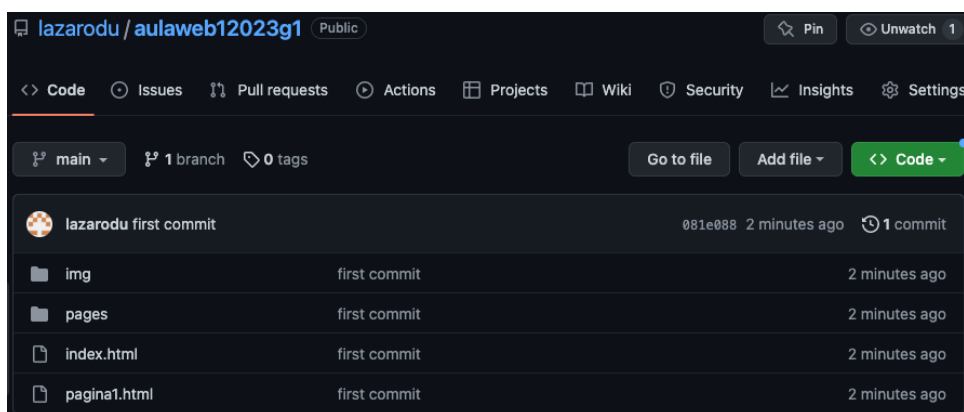
	<p align="center">Centro Federal de Educação Tecnológica de Minas Gerais Campus VIII – Varginha Curso Técnico em Informática</p>	
<p><i>Disciplina</i> Lab. Aplicações Web 1</p>	<p align="center">Git e Github</p>	<p><i>Professor</i> Lázaro Eduardo da Silva</p>

```

MacBook-Pro-de-Lazaro:aula lazaroeduardodasilva$ git push origin main
git: 'credential-erase' is not a git command. See 'git --help'.
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 134.41 KiB | 26.88 MiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/lazarodu/aulaweb12023g1.git
 * [new branch]      main -> main
MacBook-Pro-de-Lazaro:aula lazaroeduardodasilva$

```

Feito isso, os arquivos devem aparecer na pasta online, acesse-a para conferir



Deveremos seguir até aqui, toda vez que um novo projeto for iniciado e um novo repositório do github for criado.

Passos para continuar um projeto em um novo computador

Ainda podemos ter a situação de querer continuar um projeto que foi trabalhado em uma outra aula e você não esta na mesma máquina que fez o deploy no github.

Para isso, você deve fazer um clone da pasta online para a sua pasta local executando o comando abaixo:

```
git clone https://github.com/lazarodu/aulaweb12023g1.git .
```

Este comando irá clonar na pasta local (.) o conteúdo da pasta remota que você colocou o endereço.

Após fazer as alterações, você deve informar ao github qual o usuário que você utilizará para atualizar a sua pasta executando os comandos:

```
git config --global user.name "SEU_NOME_DE_USUARIO"
git config --global user.email "SEU_E-MAIL"
```

Depois acrescentar as alterações realizadas nesta versão executando o comando:


```
git add .
```

Comentar o que foi desenvolvido executando o comando:

```
git commit -m "comentário"
```

E enviar as atualizações para o seu repositório executando o comando:

```
git push origin main
```

	<p align="center">Centro Federal de Educação Tecnológica de Minas Gerais Campus VIII – Varginha Curso Técnico em Informática</p>	
<p><i>Disciplina</i> Lab. Aplicações Web 1</p>	<p align="center">Git e Github</p>	<p><i>Professor</i> Lázaro Eduardo da Silva</p>

Passos para continuar um projeto no mesmo computador

Uma terceira situação que também pode ocorrer é o caso de você se sentar no mesmo computador que havia trabalhado na aula anterior, mas não saber se os arquivos que estão neste computador estão atualizados com os que estão na nuvem. Isso pode gerar um erro após realizar as alterações.

Portanto, se você estiver nessa situação, antes de realizar qualquer alteração execute o comando abaixo:

```
git pull origin main
```

Este comando irá sincronizar seu repositório local com o repositório remoto.

Depois de realizar as alterações, faça como no procedimento acima, informe ao github qual o usuário que você utilizará para atualizar a sua pasta executando os comandos:

```
git config --global user.name "SEU_NOME_DE_USUARIO"
```

```
git config --global user.email "SEU_E-MAIL"
```

Depois acrescente as alterações realizadas nesta versão executando o comando:

```
git add .
```

Comente o que foi desenvolvido executando o comando:

```
git commit -m "comentário"
```

E envie as atualizações para o seu repositório executando o comando:

```
git push origin main
```