



Faculdades Integradas  
do Vale do Ivaí

# Estrutura de Dados

Dieimes Nunes de Souza

[prof\\_dieimessouza@ucpparana.edu.br](mailto:prof_dieimessouza@ucpparana.edu.br)

# Bem-vindos à aula!

# Agenda de hoje

- Apresentação
- Ementa
- Bibliografia
- Avaliações
- Grupo Whats
- Git e Github (Revisão)





# Apresentação



- Qual o seu nome?
- Mora em Ivaiporã?
- Quais são as suas motivações?
- Possui algum conhecimento em programação?
- Comente um pouco da sua história.

# Ementa

Definição e utilização de tipos abstratos de dados. Registros. Manipulação de arquivos. Variáveis do tipo ponteiro. Listas lineares e suas generalizações: listas ordenadas, listas encadeadas, pilhas e filas. Pesquisa sequencial. Pesquisa Binária. Árvore binária. Árvores balanceadas (AVL). Árvores de busca.



# Bibliografia básica

- GALANTE, Renata. Estruturas de dados. Porto Alegre: BOOKMAN. 2009.
- ZIVIANI, Nivio. Projeto de algoritmos. : com implementação em pascal e c. São Paulo: Cengage Learning, 2011.
- PIVA JUNIOR, Dilermando. Algoritmos e programação de computadores. Rio de Janeiro: Campus, 2012.

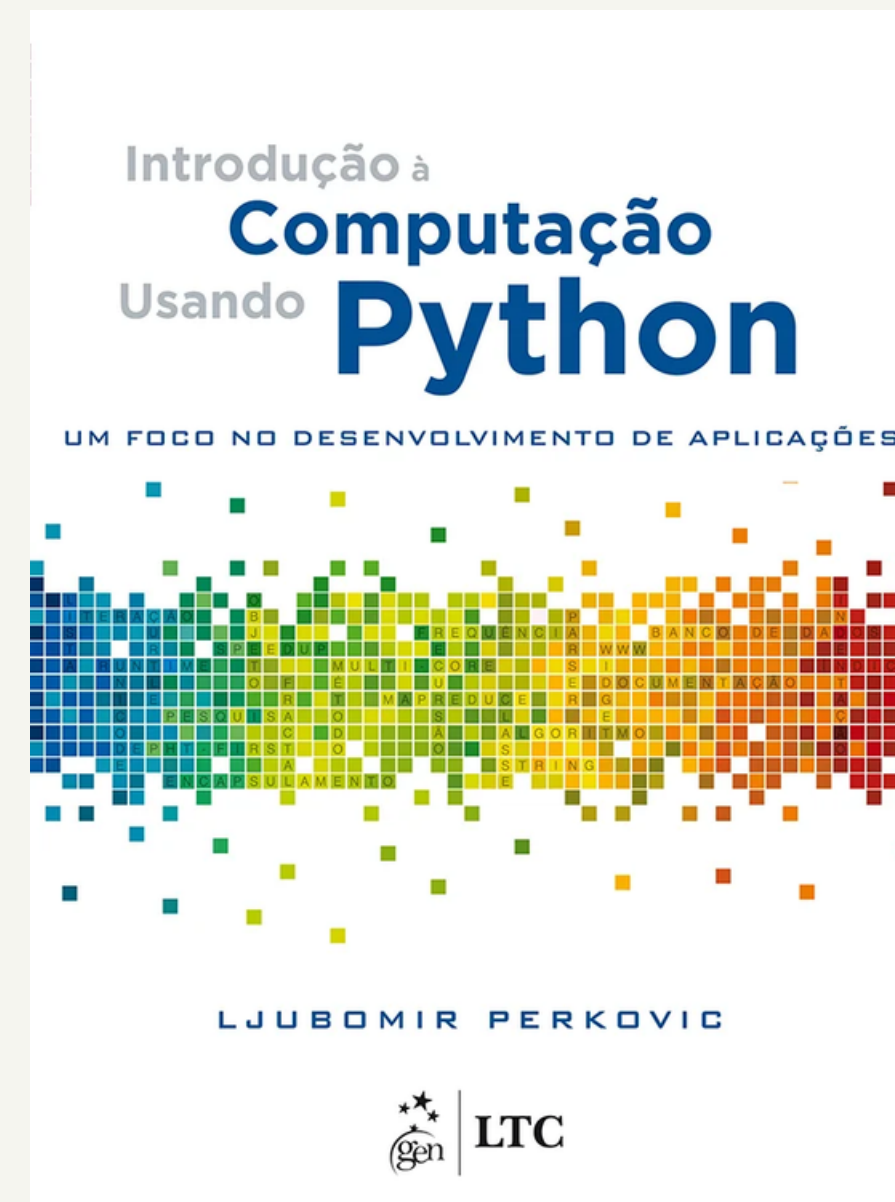
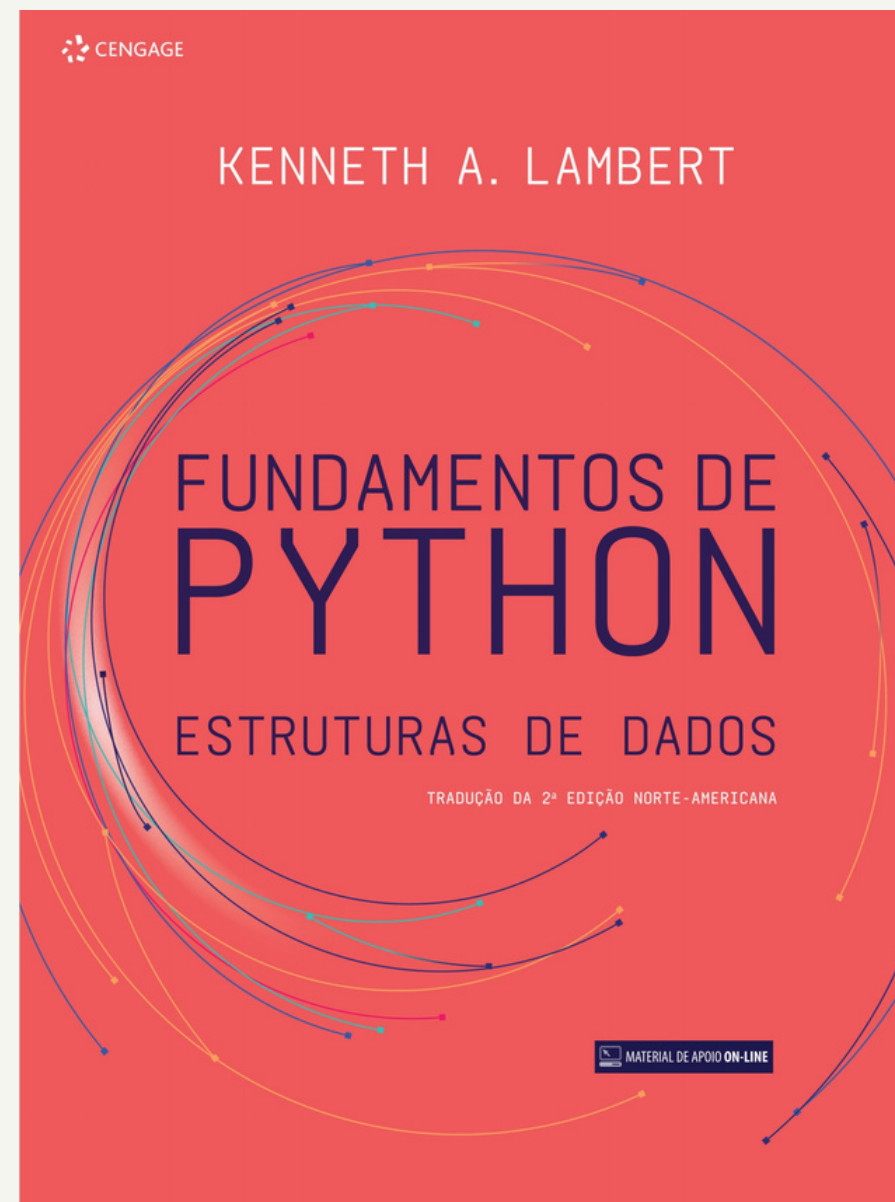


# Bibliografia Complementar

- AVILLANO, Israel de Campos.Algoritmos e pascal. Rio de Janeiro: Ciência Moderna, 2006.
- BOENTE, Alfredo.Construção de algoritmos:resumido.2 ed. Rio de Janeiro: Editora Rio, 2006.
- CORMEN, Thomas H.Algoritmos: teoria e prática.Rio de Janeiro: Campus, 2002.
- EDMONDS, Jeff.Como pensar sobre algoritmos. Rio de Janeiro: LTC, 2010.
- FORBELLONE, André Luiz Villar; Eberspacher, Henri Frederico.Lógica de programação: a construção de algoritmos e estruturas de dados.São Paulo: Makron Books, 1993.
- MEDINA, Marco; FERTIG, Cristina.Algoritmos e programação: teoria e prática. São Paulo: Novatec, 2006.
- MORAES, Celso Roberto.Estrutura de dados e algoritmos:uma abordagem didática. São Paulo: Berkeley Brasil, 2001.
- PREISS, Bruno.Estrutura de dados e algoritmos: padrões de projetos orientados a objetos com java.Rio de Janeiro: Campus, 2000.
- SZWARCFITER, Jayme Luiz.Estruturas de dados e seus algoritmos.3.Ed.São Paulo:LTC - LIVROS TÉCNICOS E CIENTÍFICOS.2014.



# Bibliografia - Indicação



# Avaliações

## Primeiro Bimestre

- Prova bimestral - 5,00
- Prova Êxito - 2,00
- Prova diagnóstica - 1,00
- Trabalhos e demais atividades 2,00



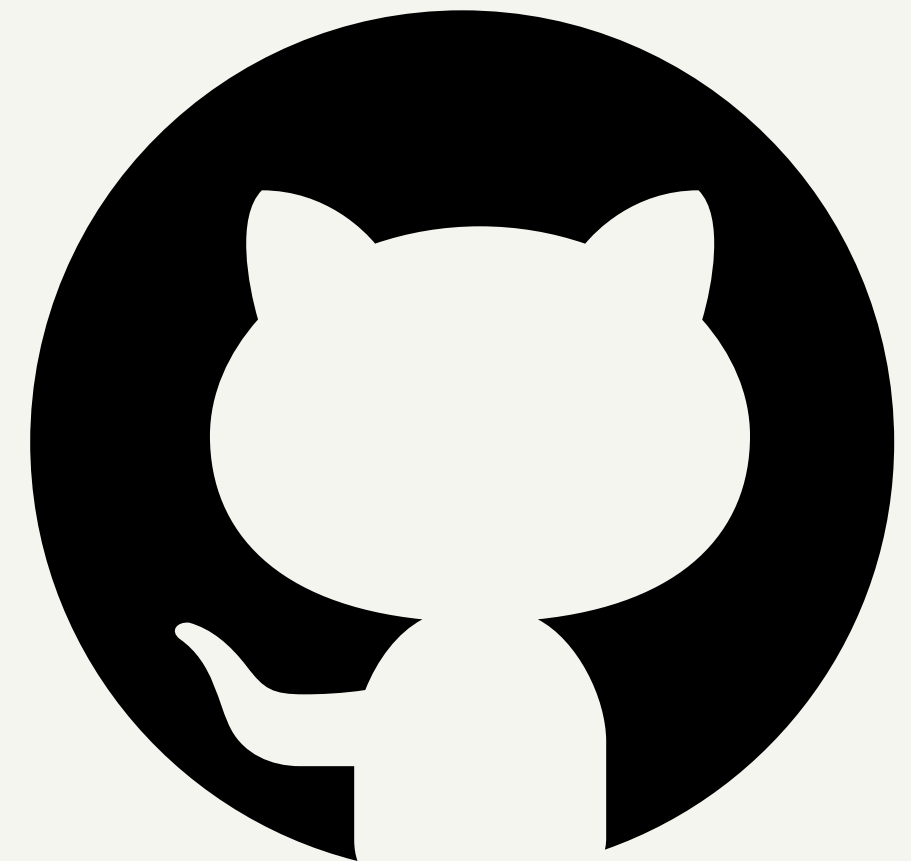


# Grupo no Whatsapp



# Instalação git Windows

- Acessar o site: <https://git-scm.com/downloads>
- Fazer o download do executável;
- E seguir as instruções;



# O que é git?

- Sistema de controle de versão mais utilizado do mundo atualmente;
- O Git é baseado em repositórios, que contêm todas as versões do código e também as cópias de cada desenvolvedor;
- Todos os objetos do Git são protegidos como criptografia para evitar alterações indevidas e maliciosas;
- O Git é um projeto de código aberto;



# O que é Github?

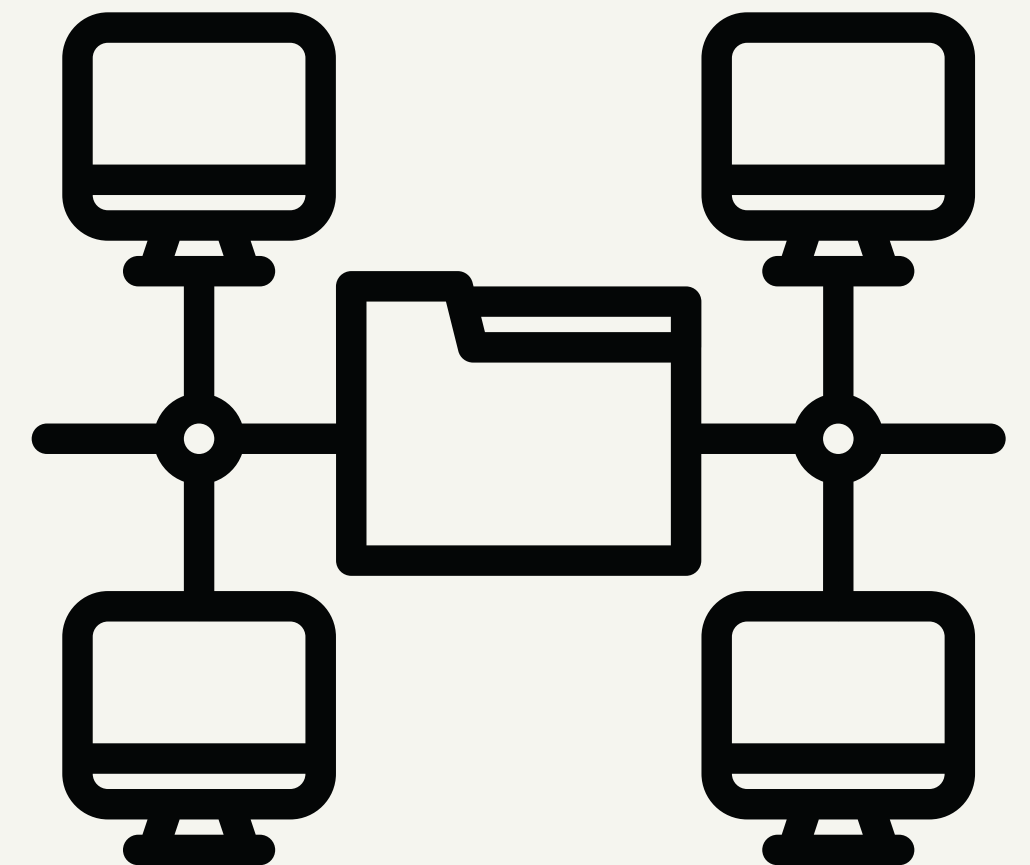
- É um serviço para gerenciar repositórios, gratuito e amplamente utilizado;
- Podemos enviar nossos projetos para o GitHub e disponibilizá-lo para outros devs;
- O GitHub é gratuito tanto para projetos públicos como privados;





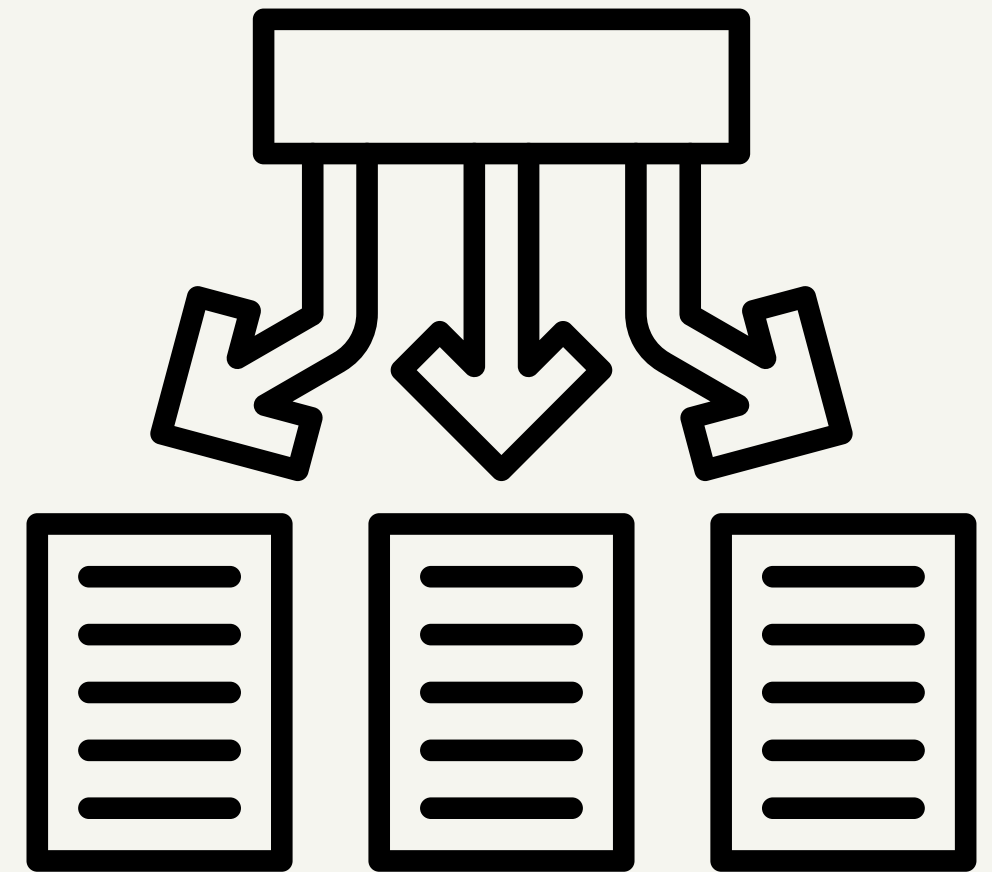
# O que é um repositório?

- É onde o código será armazenado;
- Na maioria das vezes cada projeto tem um repositório;
- Cada um dos desenvolvedores do time pode baixar o repositório e criar versões diferentes em sua máquina;



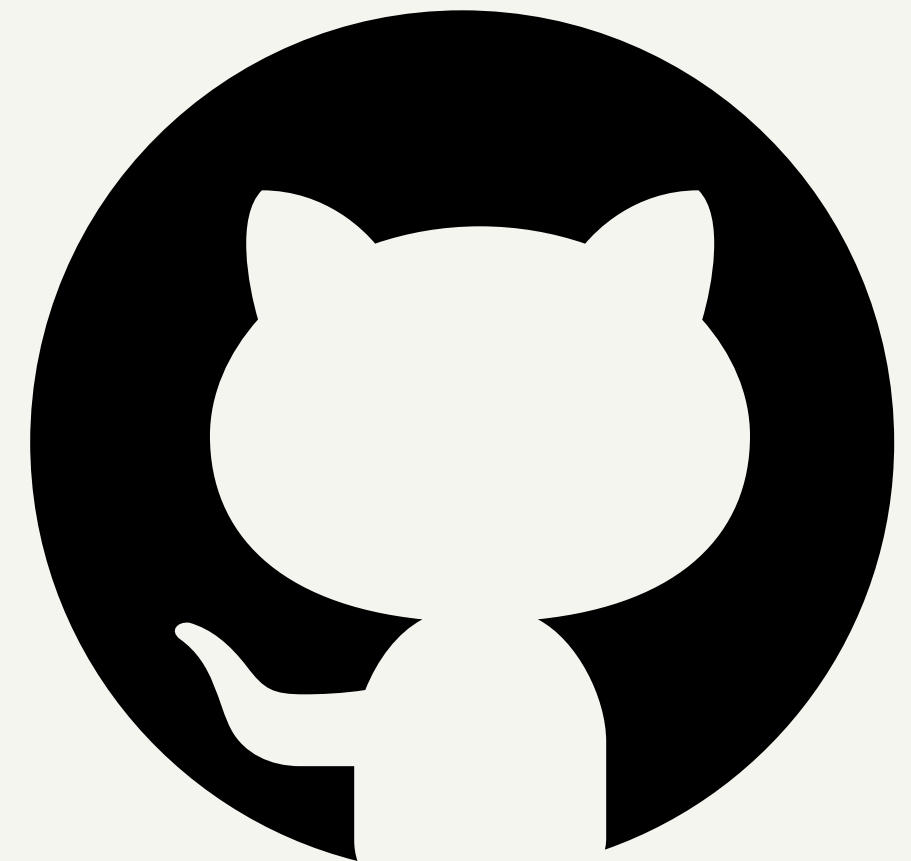
# Criando repositórios

- Para criar um repositório utilizamos o comando: **git init;**
- Após este comando o diretório atual será reconhecido pelo Git como um projeto e responderá aos seus demais comandos;



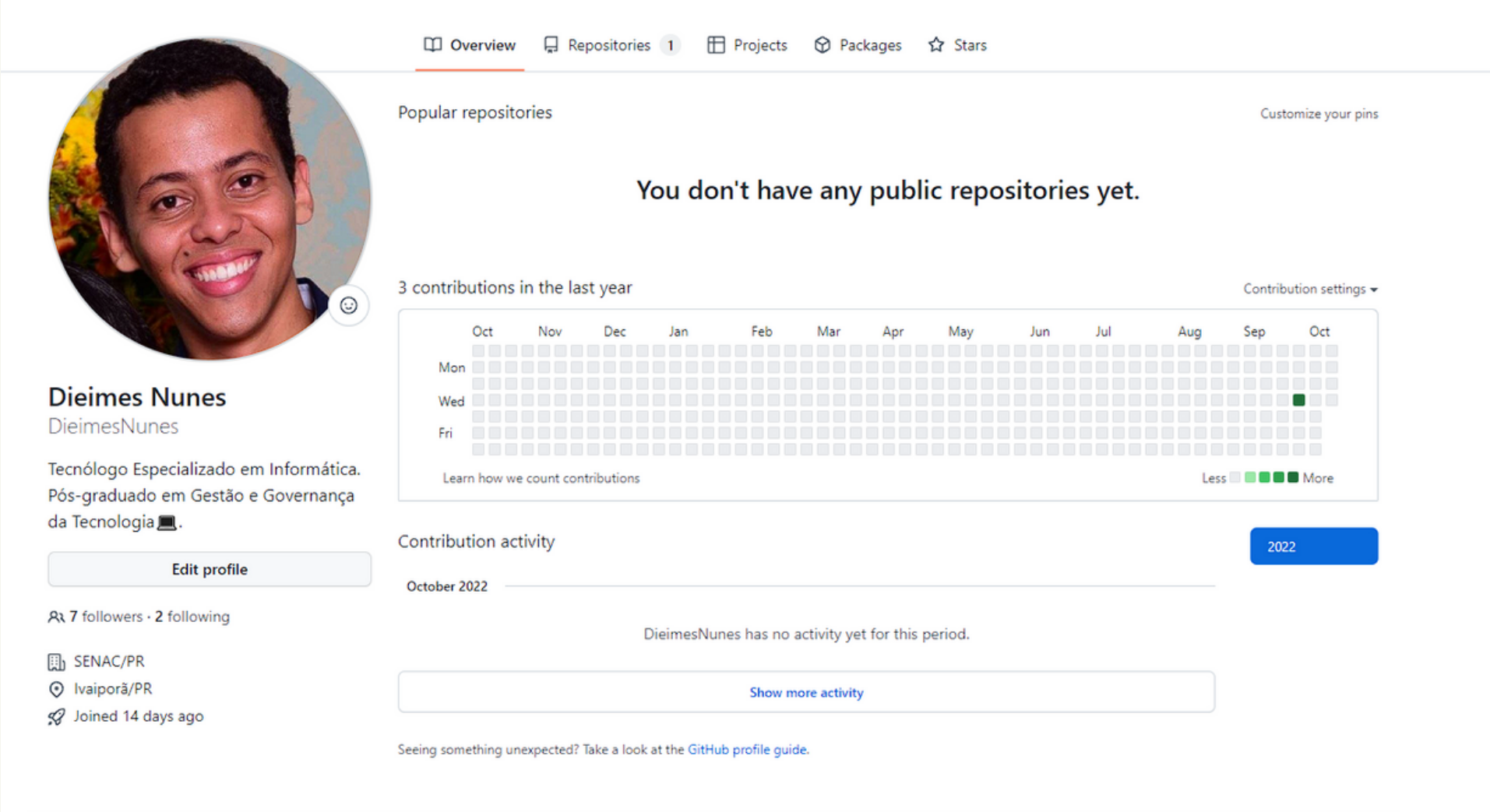
# Enviando repositórios para o Github

- Precisamos criar o projeto no Github, inicializar o mesmo no git em nosso computador, sincronizar com o Github e enviar;
- Só fazemos uma vez por projeto;



# Enviando repositórios para o Github

- Clicar na aba repositories



The screenshot shows a GitHub profile for 'Dieimes Nunes' (username: DieimesNunes). The profile includes a circular profile picture, a bio stating 'Tecnólogo Especializado em Informática. Pós-graduado em Gestão e Governança da Tecnologia', and statistics showing 7 followers and 2 following. The 'Repositories' tab is selected, showing a message: 'You don't have any public repositories yet.' Below this, a 'Contribution activity' section displays a calendar grid for October 2022, indicating 3 contributions in the last year. The grid shows activity on Monday, Wednesday, and Friday of October. A 'Show more activity' button is visible at the bottom of the calendar section.

Overview Repositories 1 Projects Packages Stars

Popular repositories Customize your pins

You don't have any public repositories yet.

3 contributions in the last year Contribution settings

	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
Mon													
Wed													
Fri													

Learn how we count contributions Less More

Contribution activity 2022

October 2022

DieimesNunes has no activity yet for this period.

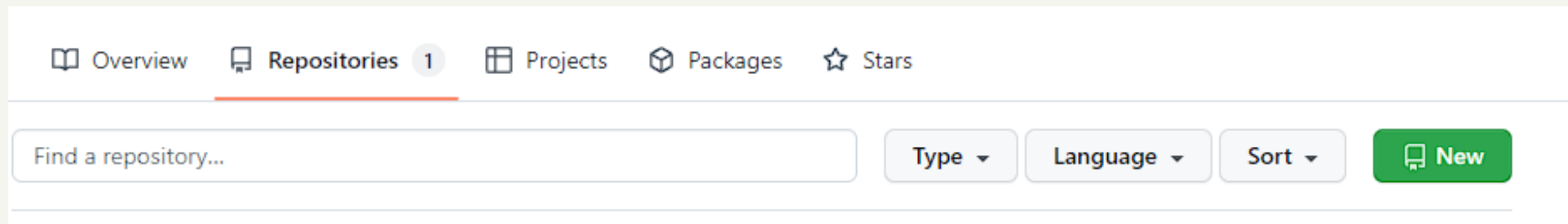
Show more activity

Seeing something unexpected? Take a look at the [GitHub profile guide](#).



# Enviando repositórios para o Github

Clicar no botão New



The image shows the top navigation bar of the GitHub interface. It includes tabs for 'Overview', 'Repositories' (which is active and has a '1' badge), 'Projects', 'Packages', and 'Stars'. Below the tabs is a search bar labeled 'Find a repository...'. To the right of the search bar are three dropdown menus: 'Type', 'Language', and 'Sort'. On the far right is a green button labeled 'New' with a repository icon.

Overview Repositories 1 Projects Packages Stars

Find a repository... Type Language Sort New

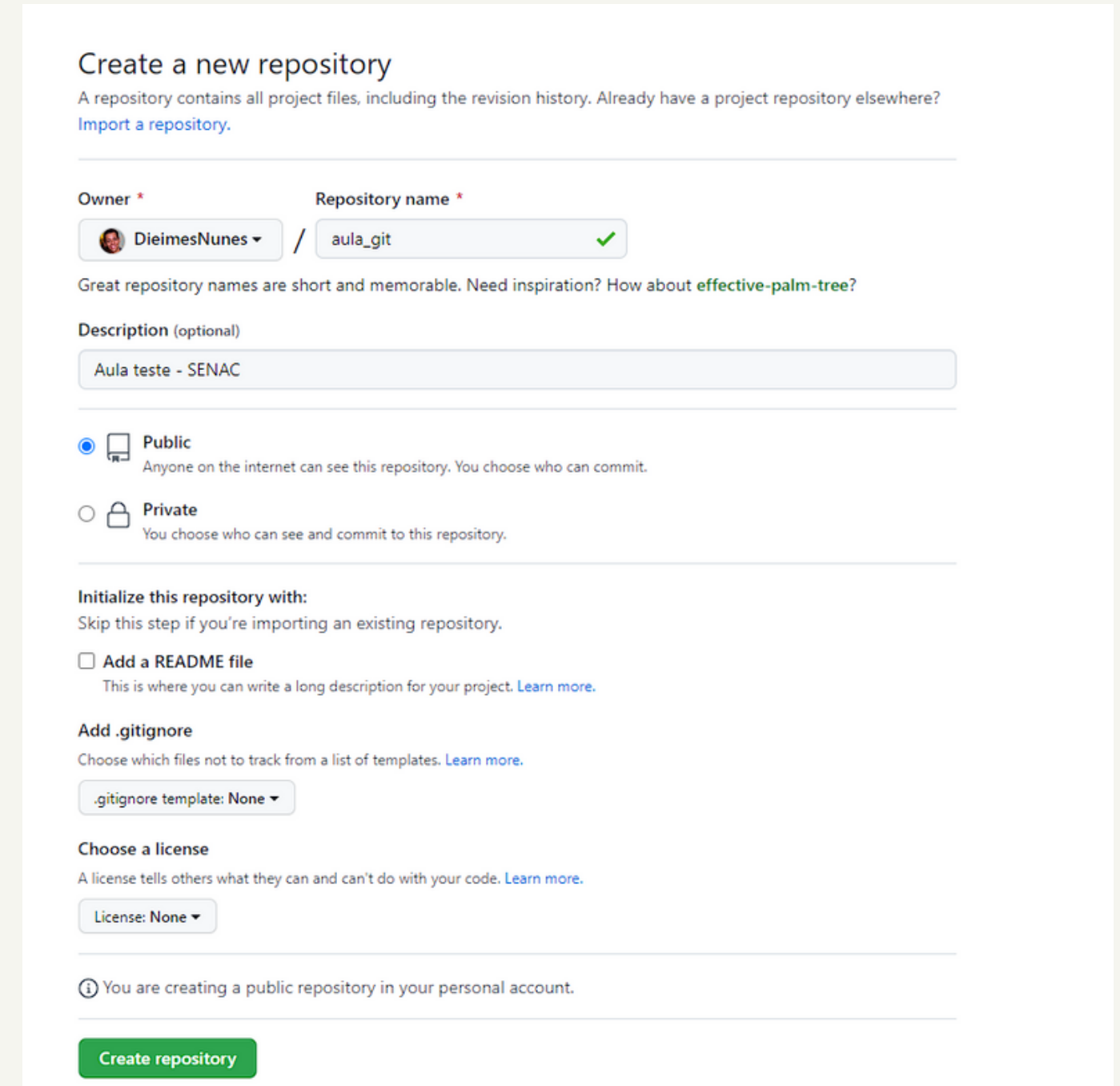
# Enviando repositórios para o Github

Inserir o nome do repositório;

Inserir uma descrição;

Selecionar public ou private;

Apertar o botão "Create repository"



The screenshot shows the GitHub 'Create a new repository' page. At the top, it says 'Create a new repository' and provides a brief explanation of what a repository is, along with a link to 'Import a repository'. Below this, there are two input fields: 'Owner' and 'Repository name'. The 'Owner' field is set to 'DieimesNunes' and the 'Repository name' field is set to 'aula\_git', which has a green checkmark next to it. Below these fields, there is a note about repository names being short and memorable, with a link to 'effective-palm-tree?'. The 'Description (optional)' field contains the text 'Aula teste - SENAC'. Underneath the description, there are two radio button options for repository visibility: 'Public' (selected) and 'Private'. The 'Public' option is accompanied by a lock icon and the text 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is accompanied by a lock icon and the text 'You choose who can see and commit to this repository.' Below the visibility options, there is a section titled 'Initialize this repository with:' which includes a link to 'Skip this step if you're importing an existing repository.' and a checkbox for 'Add a README file' with a link to 'Learn more.' Below this, there is a section titled 'Add .gitignore' with a link to 'Learn more.' and a dropdown menu for '.gitignore template' set to 'None'. Finally, there is a section titled 'Choose a license' with a link to 'Learn more.' and a dropdown menu for 'License' set to 'None'. At the bottom of the form, there is a green button labeled 'Create repository'.

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 \*

DieimesNunes / aula\_git ✓

Great repository names are short and memorable. Need inspiration? How about [effective-palm-tree?](#)

Description (optional)

Aula teste - SENAC

☒ 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

# Enviando repositórios para o Github

Nessa tela exibe o passo a passo dos comandos do git para enviar os arquivos para o Github

Quick setup — if you've done this kind of thing before


 Set up in Desktop

 or 

HTTPS

SSH


https://github.com/DieimesNunes/aula\_git.git



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


...or create a new repository on the command line

```
echo "# aula_git" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/DieimesNunes/aula_git.git
git push -u origin main
```



...or push an existing repository from the command line


```
git remote add origin https://github.com/DieimesNunes/aula_git.git
git branch -M main
git push -u origin main
```



...or import code from another repository

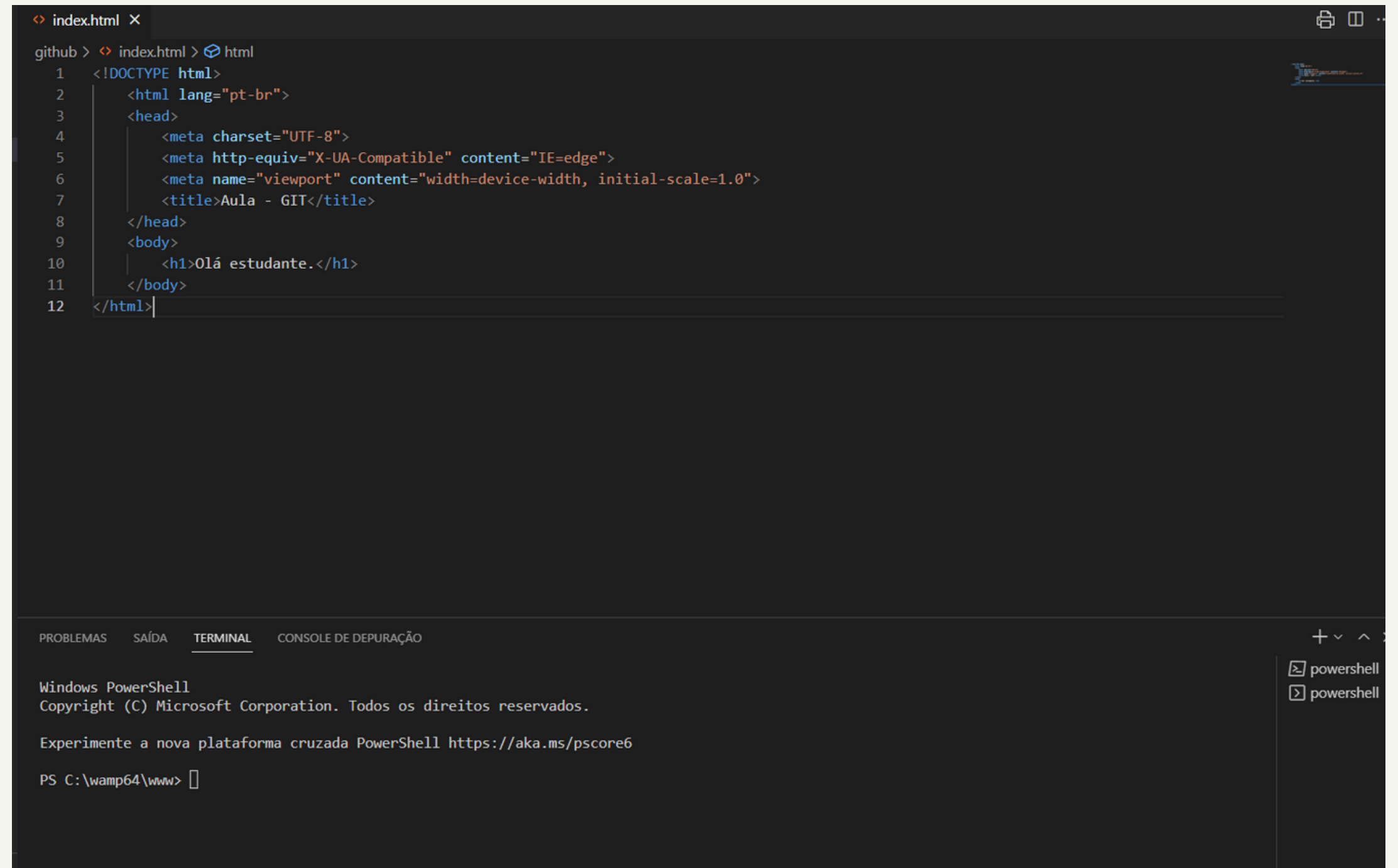
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

 **ProTip!** Use the URL for this page when adding GitHub as a remote.

# Enviando repositórios para o Github

Abra o terminal do Visual Studio Code e verifique se está na pasta correta



The image shows a screenshot of the Visual Studio Code editor interface. The main editor window displays a file named `index.html` with the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Aula - GIT</title>
8 </head>
9 <body>
10  <h1>Olá estudante.</h1>
11 </body>
12 </html>
```

Below the editor, the **TERMINAL** panel is open, showing the Windows PowerShell prompt. The terminal output includes the Windows PowerShell copyright notice and the current directory path:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\wamp64\www>
```

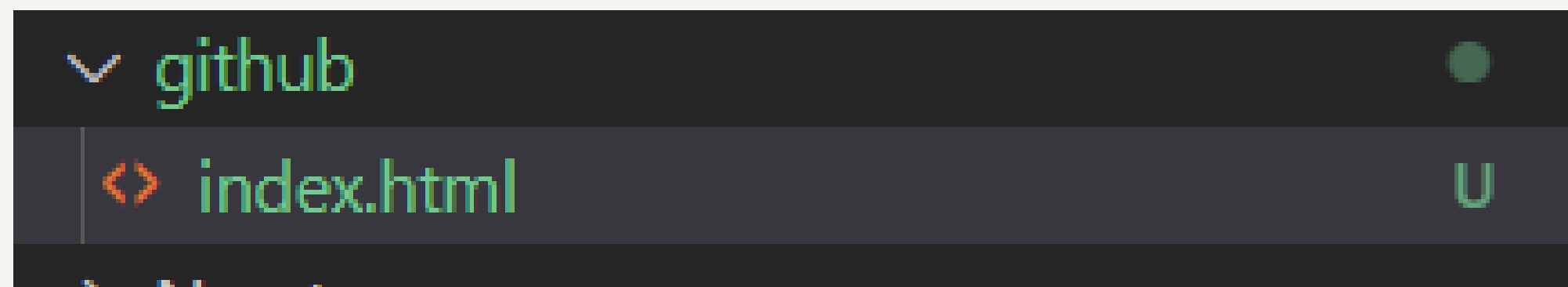


# Enviando repositórios para o Github

Inicializar um repositório com o comando git init

```
PS C:\wamp64\www\github> git init  
Initialized empty Git repository in C:/wamp64/www/github/.git/  
PS C:\wamp64\www\github> █
```

Observe que o Visual Studio Code muda de cor



# Enviando repositórios para o Github

Para adicionar um arquivo utilize o comando git add

```
PS C:\wamp64\www\github> git add index.html  
PS C:\wamp64\www\github> █
```

# Enviando repositórios para o Github

Utilize o comando `git commit -m` adicionar a versão e mensagem do seu arquivo.

```
PS C:\wamp64\www\github> git commit -m "Minha primeira atualização"
[master (root-commit) c85f363] Minha primeira atualização
 1 file changed, 12 insertions(+)
 create mode 100644 index.html
PS C:\wamp64\www\github> 
```

# Enviando repositórios para o Github

Utilize o comando `git branch -M main` para adicionar uma ramificação main em seu projeto.

```
PS C:\wamp64\www\github> git branch -m main  
PS C:\wamp64\www\github> |
```



# Enviando repositórios para o Github

Adicione o comando de origem, git remote add origin.

Atenção: copiar o comando no site do Github.

...or create a new repository on the command line

```
echo "# aula_git" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/DieimesNunes/aula_git.git
git push -u origin main
```



```
PS C:\wamp64\www\github> git remote add origin https://github.com/DieimesNunes/aula_git.git
PS C:\wamp64\www\github> 
```

# Enviando repositórios para o Github

Adicione o comando `git push -u origin main` para enviar o arquivo para main do seu repositório.

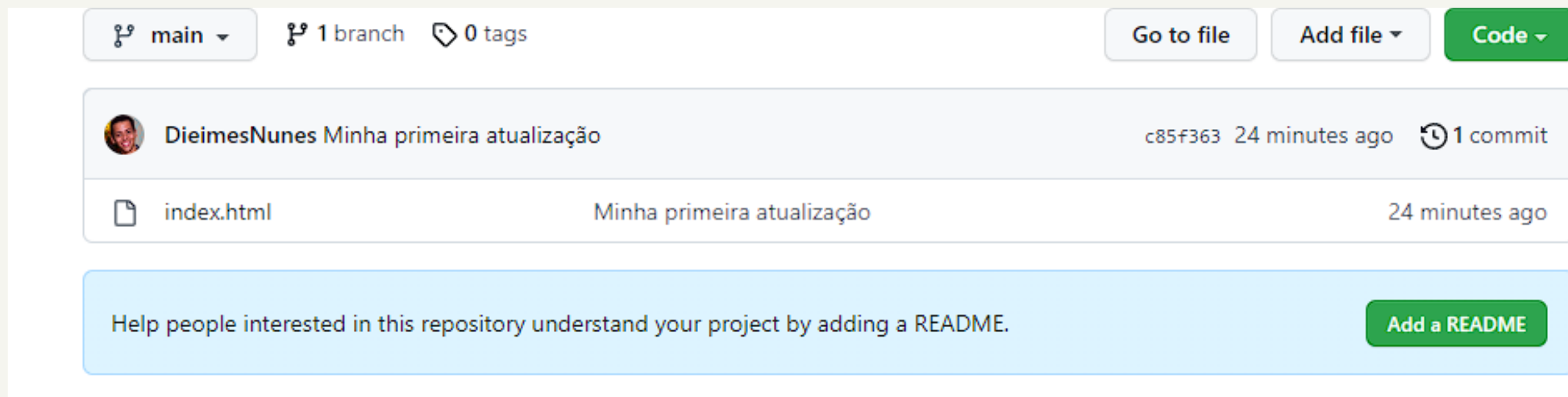
```
PS C:\wamp64\www\github> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 452 bytes | 452.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DieimesNunes/aula_git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\wamp64\www\github> 
```

# Enviando repositórios para o Github

- Pode acontecer a solicitação do login e a senha do Github, nesse caso deve inserir as informações solicitadas;
- Se o comando `git push -u origin` não funcionar deve então remover a origem e adicioná-lo novamente, como os comando `git remote -v`, `git remote rm origin`. Após a realizar utilizar os comandos `git remote add origin` para adicionar a origem do arquivo e depois `git push -u origin`.

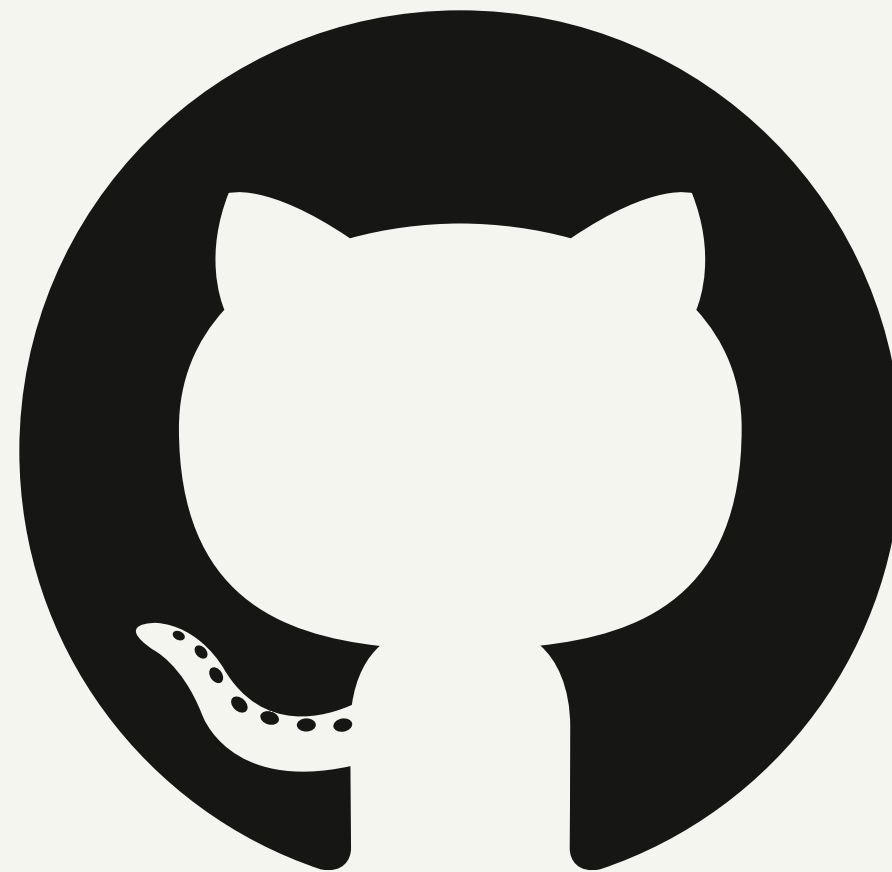
# Enviando repositórios para o Github

Pressione o F5 na página do Github e verifique se o arquivo está no seu repositório.



# Atividade

Crie um arquivo de texto e utilize os comandos do GIT visto na aula de hoje para enviar um arquivo no repositório do Github.



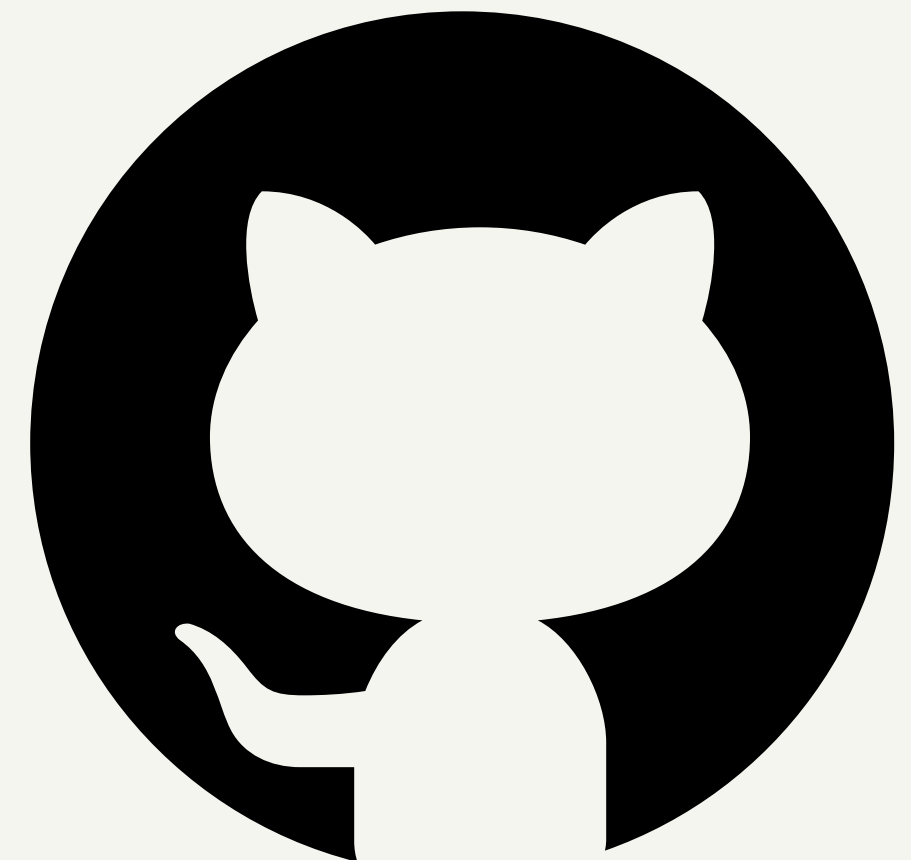


# Verificando mudanças do projeto

- As mudanças do projeto podem ser verificadas pelo comando git status;

```
PS C:\wamp64\www\github> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\wamp64\www\github> █
```



# Verificando mudanças do projeto

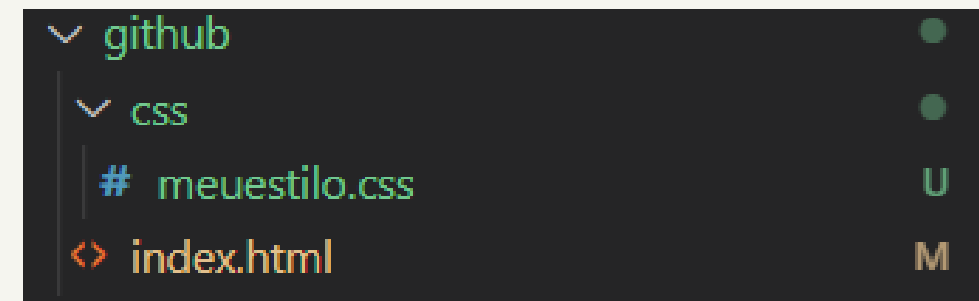
Na imagem abaixo, foi utilizado o comando `git status`, no qual foi informado que existe um arquivo `index.html` que foi modificado e uma pasta que ainda não foi incluído no repositório do Github.

```
PS C:\wamp64\www\github> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        css/

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\wamp64\www\github>
```



# Adicionando arquivos ao projeto

- Para adicionar arquivos novos no seu repositório utilize o comando `git add`;
- Podemos adicionar um arquivo específico como também diversos de uma vez só;
- Somente arquivos adicionado o git irá fazer o monitoramento;
- Atenção: Sempre adicione o seus arquivos para não perder nada por descuido;

# Adicionando um arquivo

Utilize o comando git add e nome do arquivo:

```
PS C:\wamp64\www\github> git add index.html
```

# Adicionando um arquivo

Verificando o status com o comando git status

```
PS C:\wamp64\www\github> git status
On branch main
Your branch is up to date with 'origin/main'.

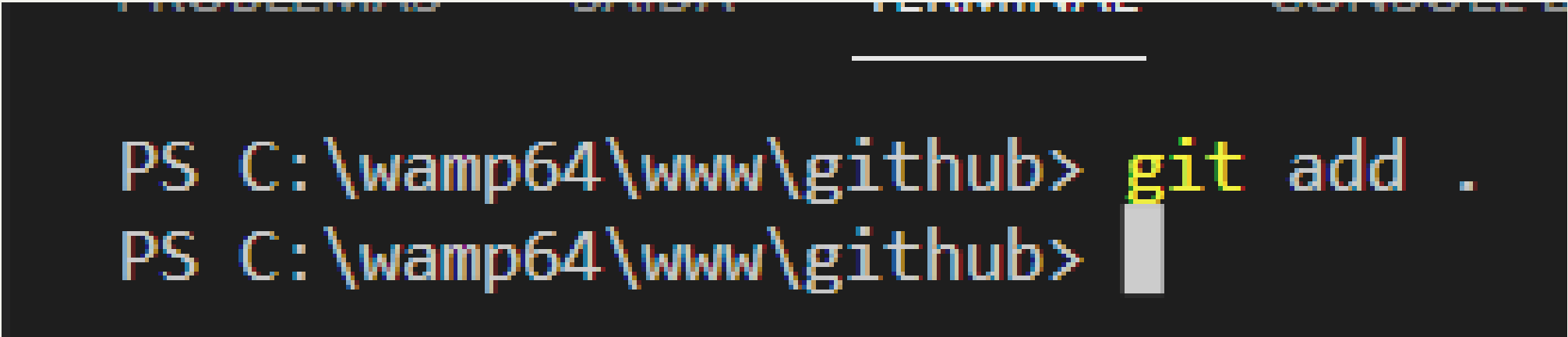
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        css/

PS C:\wamp64\www\github> █
```

# Adicionando vários arquivos

Utilize o comando git add .

A screenshot of a Windows command prompt window. The title bar at the top shows 'PROBLEMAS', 'ERRATA', 'TERMINAL', and 'CONSOLE'. The command prompt shows the directory 'C:\wamp64\www\github' and the command 'git add .' being entered. The word 'git' is highlighted in yellow. A grey cursor is visible at the end of the second line.

```
PS C:\wamp64\www\github> git add .  
PS C:\wamp64\www\github> 
```



# Salvando alterações

- As alterações salvas do seu projeto são realizadas pelo comando `git commit`;
- Podemos commitar arquivos específicos ou vários de uma vez com adição `-a`;
- É uma boa prática enviar uma mensagem a cada commit, com as alterações que foram feitas, adicione `-m` e em seguida a mensagem;

# Salvando alterações

Utilize o comando `git commit -a -m` para salvar todas alterações dos seus arquivos

```
PS C:\wamp64\www\github> git commit -a -m "Implementação do CSS"
[main f1784b6] Implementação do CSS
 2 files changed, 6 insertions(+)
 create mode 100644 css/meuestilo.css
PS C:\wamp64\www\github> █
```

# Enviando código para Github

- Para enviar os arquivos utilize o comando `git push`;
- Após esta ação o código que está no Github será atualizado baseando-se no código da sua máquina (local);

# Enviando código para Github

- Para enviar os arquivos utilize o comando git push;
- Após esta ação o código que está no Github será atualizado baseando-se no código da sua máquina (local);

```
PS C:\wamp64\www\github> git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 528 bytes | 264.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/DieimesNunes/aula_git.git
   c85f363..f1784b6  main -> main
PS C:\wamp64\www\github> █
```

# Enviando código para Github

main ▾


1 branch

0 tags



Go to file

Add file ▾

Code ▾

 DieimesNunes Implementação do CSS

f1784b6 7 minutes ago ⌚ 2 commits

 css	Implementação do CSS	7 minutes ago
 index.html	Implementação do CSS	7 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

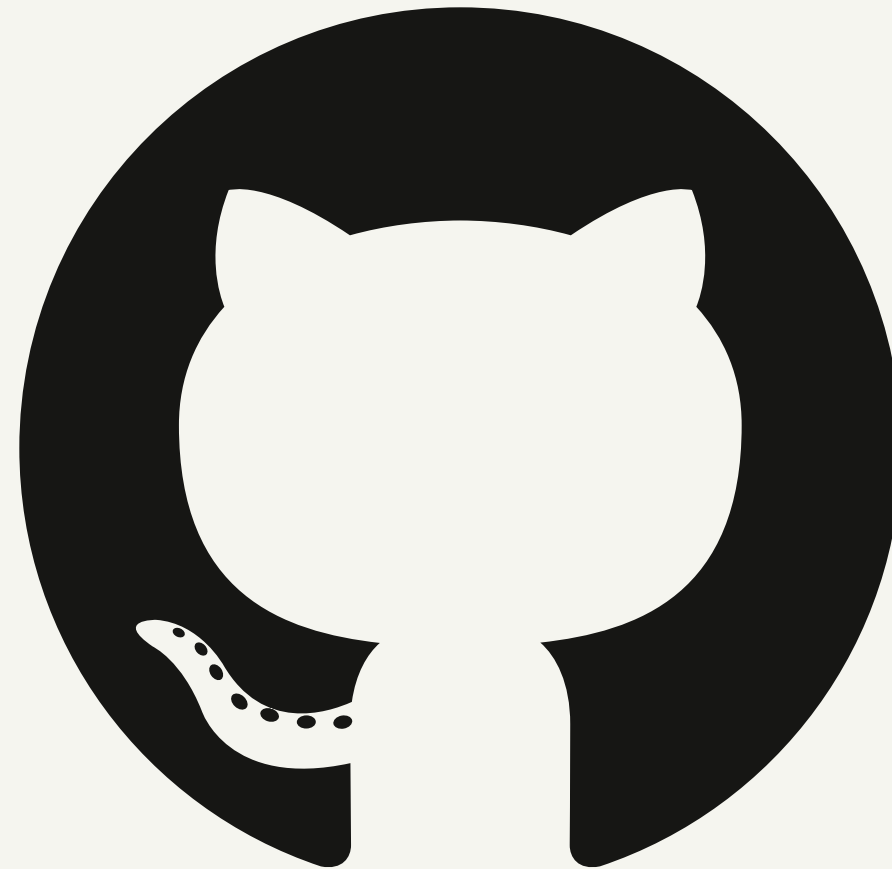
# Dúvidas e Comentários

Fique à vontade para fazer deste um debate aberto à perguntas e esclarecimentos antes de prosseguirmos.



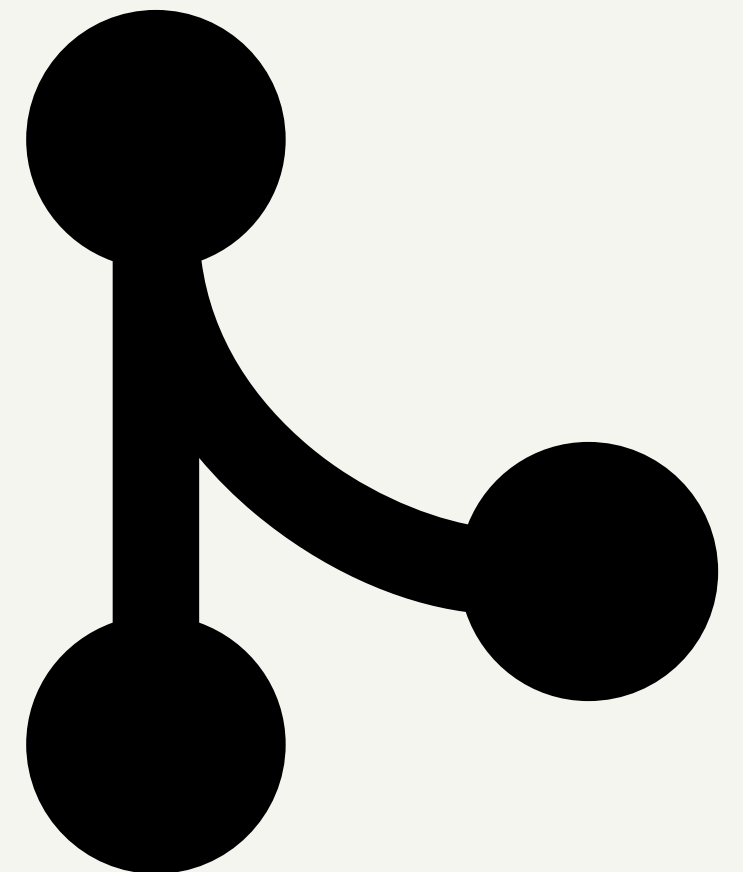
# Atividade

Crie cinco arquivos em texto ou HTML e adicione no seu repositório.



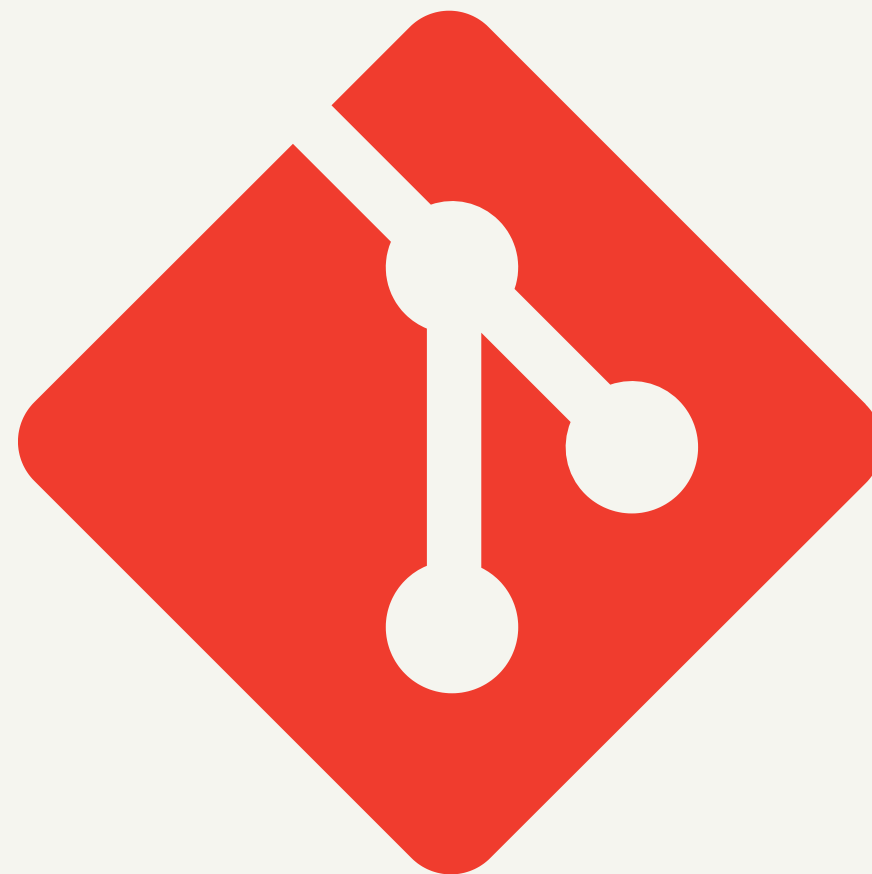
# O que é um branch

- Branch é a forma que o git separa as versões dos projetos;
- Quando um projeto é criado ele inicia na branch main;
- Após a finalização das alterações os branches são unidos para ter o código-fonte final;



# Criando e visualizando os branches

- Para visualizar os branches disponíveis utiliza o comando `git branch`;
- Para criar um branch você precisa utiliza o comando `git branch <nome>`;



# Visualizando os branches

Utilize o comando git branch

```
PS C:\wamp64\www\github> git branch
* main
PS C:\wamp64\www\github> 
```

# Visualizando os branches

Utilize o comando git branch

```
PS C:\wamp64\www\github> git branch
* main
PS C:\wamp64\www\github> 
```

Observe que o comando executado retorna o branch principal main.

# Criando o branch

Utilize o comando git branch <nome>

```
PS C:\wamp64\www\github> git branch funcionalidade_Star_Wars
PS C:\wamp64\www\github> 
```

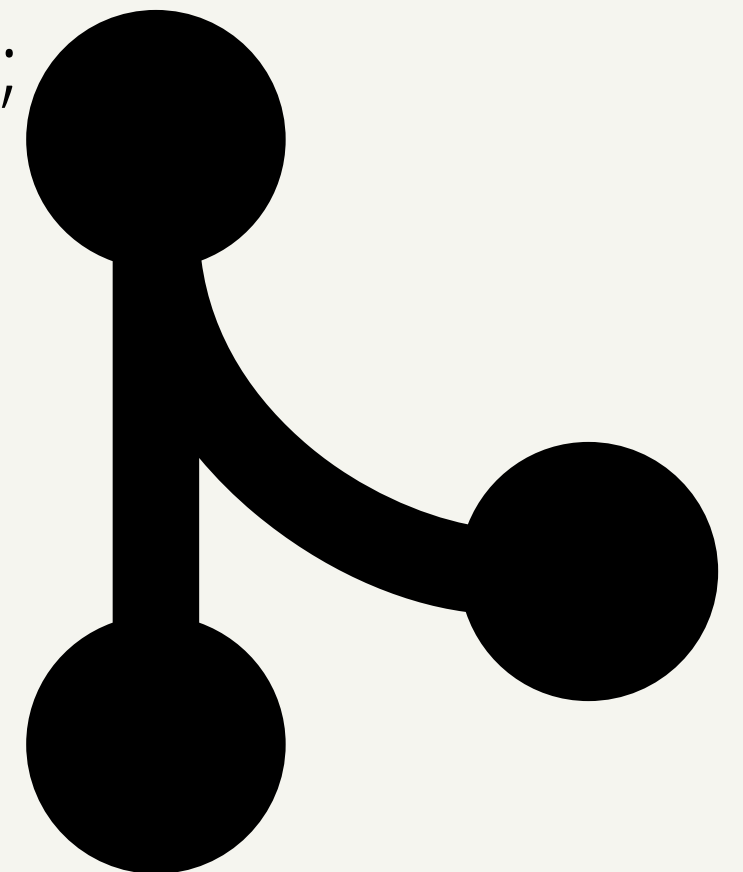
Verificando os branches:

```
PS C:\wamp64\www\github> git branch
funcionalidade_Star_Wars
* main
PS C:\wamp64\www\github> 
```



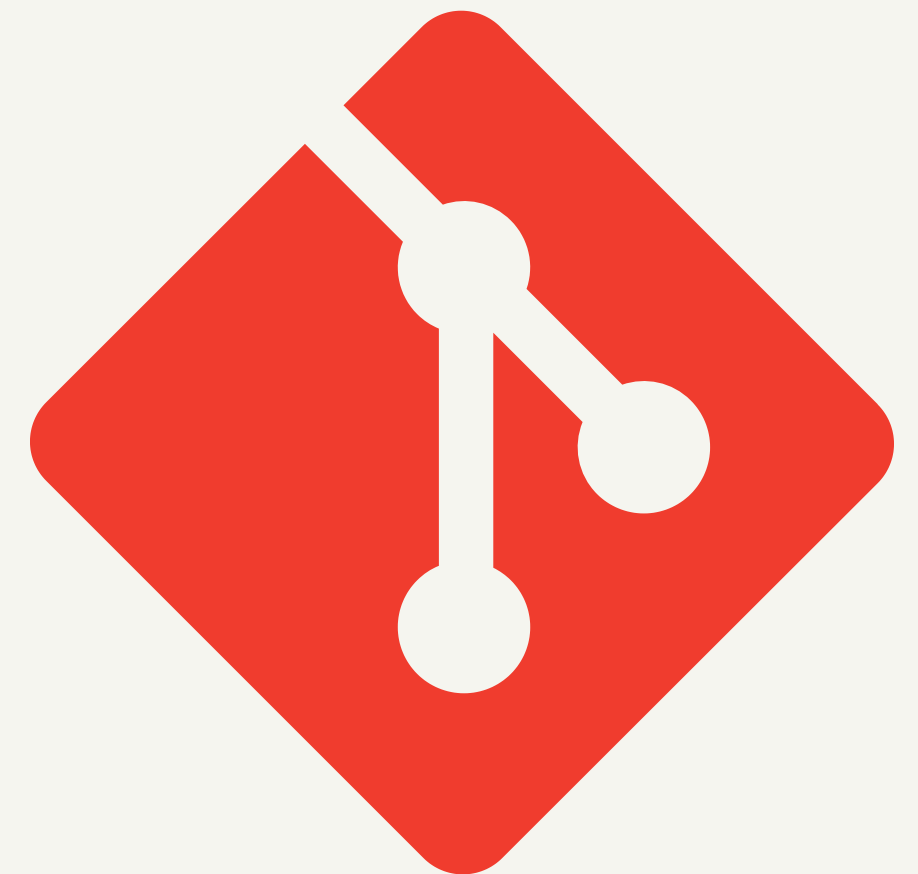
# Deletando branches

- Podemos deletar um branch com `-d` ou `--delete`
- Não é comum deletar um branch, normalmente guardamos o histórico do trabalho;
- Geralmente se usa o `delete` quando o branch foi criado errado;



# Mudando de branch

- Podemos mudar para outro branch utilizando o comando `git checkout -b <nome>;`



# Mudando de branch

Utilize o comando git checkout <nome>

```
PS C:\wamp64\www\github> git checkout funcionalidade_Star_Wars
Switched to branch 'funcionalidade_Star_Wars'
PS C:\wamp64\www\github> █
```

Utilize o comando git branch para visualizar qual branch está ativo

```
PS C:\wamp64\www\github> git branch
* funcionalidade_Star_Wars
  main
PS C:\wamp64\www\github> █
```

# Criando e mudando de branch

Há outra maneira de criar um branch com a utilização do comando `git checkout -b <nome>`. Esse comando irá criar um branch e mudar para ele.

```
PS C:\wamp64\www\github> git checkout -b funcionalidade_jedi
Switched to a new branch 'funcionalidade_jedi'
PS C:\wamp64\www\github> █
```

Utilize o comando `git branch` para visualizar qual branch está ativo

```
PS C:\wamp64\www\github> git checkout -b funcionalidade_jedi
Switched to a new branch 'funcionalidade_jedi'
PS C:\wamp64\www\github> git branch
  funcionalidade_Star_Wars
* funcionalidade_jedi
  main
PS C:\wamp64\www\github> █
```

# Entendo os branches

Foi inserido um paragrafo no HTML conforme a imagem abaixo, no branch funcionalidade\_jedi.

```
1  <!DOCTYPE html>
2    <html lang="pt-br">
3      <head>
4        <meta charset="UTF-8">
5        <meta http-equiv="X-UA-Compatible" content="IE=edge">
6        <meta name="viewport" content="width=device-width, initial-scale=1.0">
7        <link rel="stylesheet" href="meuestilo.css">
8        <title>Aula - GIT</title>
9      </head>
10     <body>
11       <h1>Olá estudante.</h1>
12       <p>Primeiro parágrafo.</p>
13       <p>Meste Yoda</p>
14     </body>
15  </html>
```

# Entendo os branches

O comando `git status` exibe que o arquivo `index.html` foi modificado e é necessário utilizar o comando `git commit -a -m <mensagem>` para atualizar o projeto.

```
PS C:\wamp64\www\github> git status
On branch funcionalidade_jedi
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\wamp64\www\github> █
```

# Entendo os branches

Se utilizamos o comando **checkout main** ou **checkout funcionalidade\_Star\_Wars** não haverá alteração no html.

```
PS C:\wamp64\www\github> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\wamp64\www\github> █
```

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="meuestilo.css">
  <title>Aula - GIT</title>
</head>
<body>
  <h1>Olá estudante.</h1>
  <p>Primeiro parágrafo.</p>
</body>
</html>
```



# Encontrando branches

- Branches novos são criados a todo tempo e o seu git pode não está mapeando eles;
- Com o comando **git fetch** você é atualizado de todos os branches;
- Este comando é útil para utilizar o branch de algum outro amigo (aluno) do seu projeto, por exemplo;

# Encontrando branches

No github foi adicionando um novo branch com o nome de teste1

Your branches				
teste1	Updated 5 hours ago by Dieimes Nunes	0   0	<a href="#">New pull request</a>	<a href="#">✎</a> <a href="#">🗑</a>
funcionalidade_jedi	Updated 3 hours ago by DieimesNunes	0   1	<a href="#">New pull request</a>	<a href="#">✎</a> <a href="#">🗑</a>

# Encontrando branches

Ao selecionar o comando **git branch** ele não consta na minha máquina o branch teste1. Desta forma será necessário utilizar o comando **git fetch**.

```
PS C:\wamp64\www\github> git branch
funcionalidade_Star_Wars
funcionalidade_jedi
* main
PS C:\wamp64\www\github> |
```

# Encontrando branches

Utilize o comando git fetch para encontrar novos branches e depois git checkout <nome> para ter acesso.

```
PS C:\wamp64\www\aula_git> git fetch -a
PS C:\wamp64\www\aula_git> git checkout funcionalidade_jedi
```

Resultado -->

```
PS C:\wamp64\www\aula_git> git branch
* funcionalidade_jedi
  main
PS C:\wamp64\www\aula_git>
```

# Enviando branches para o Github

Selecione o branch que desejar enviar com o comando **git checkout <nome>**.  
Adicione um commit com o comando **git commit -a -m <mensagem>** e em seguida utiliza o comando **git push** para enviar.

# Dúvidas e Comentários

Fique à vontade para fazer deste um debate aberto à perguntas e esclarecimentos antes de prosseguirmos.

# Unindo branches

- O código de dois branches distintos pode ser unido pelo comando `git merge <nome>`;
- Comando muito utilizado;
- É por meio dele que recebemos as contribuições e atualizações de outros colegas;

# Unindo branches

No branch funcionalidade\_jedi eu tenho um parágrafo a mais do que o main.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="meuestilo.css">
  <title>Página 2</title>
</head>
<body>
  <h1>Página 2.</h1>
  <p>Primeiro parágrafo.</p>
</body>
</html>
```

Branch main

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="meuestilo.css">
  <title>Página 2</title>
</head>
<body>
  <h1>Página 2.</h1>
  <p>Primeiro parágrafo.</p>
  <p>Segundo parágrafo.</p>
</body>
</html>
```

Branch funcionalidade\_jedi



# Unindo branches

No branch main utilize o comando `git merge <nome>` para unir as alterações.

```
PS C:\wamp64\www\aula_git> git merge funcionalidade_jedi
Updating 4347764..ef3e6f5
Fast-forward
 index.html      | 1 +
 pagina2.html    | 1 +
 2 files changed, 2 insertions(+)
PS C:\wamp64\www\aula_git> █
```

# Unindo branches

Ao utilizar o git status perceba que a mensagem que retorna é que branch está à frente do main, desta forma será necessário utilizar o comando git push para enviar no repositório.

```
PS C:\wamp64\www\aula_git> git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS C:\wamp64\www\aula_git> █
```

# Unindo branches

Ao utilizar o **git status** perceba que a mensagem que retorna é que branch está à frente do main, desta forma será necessário utilizar o comando **git push** para enviar no repositório.

```
PS C:\wamp64\www\aula_git> git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS C:\wamp64\www\aula_git> █
```

# Unindo branches

Utilize o **git push** para enviar no repositório.

```
PS C:\wamp64\www\aula_git> git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DieimesNunes/aula_git.git
    4347764..ef3e6f5  main -> main
PS C:\wamp64\www\aula_git> █
```