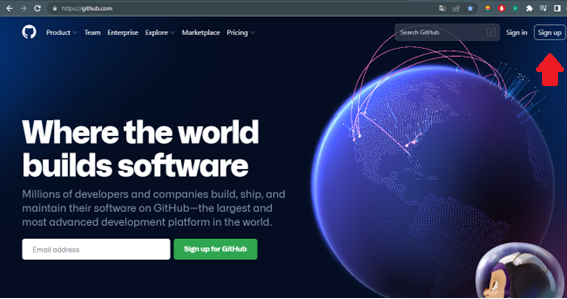
**GIT E GITHUB**

Documentação para o passo a passo da instalação do git e seus principais comandos. Também será mostrado a criação da conta no github.

* **GIT:** O Git é um sistema de controle de versionamento.
* **GITHUB:** O GitHub é um serviço baseado em nuvem que hospeda um sistema de controle (git)

**CRIAÇÃO DA CONTA NO GITHUB**

No site <https://github.com/> clicar em “**Sing up**” (Figura 1) e seguir as instruções do site passo a passo.

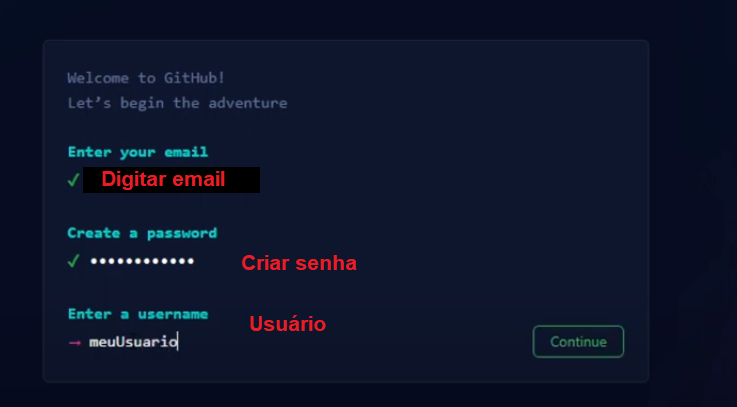


**Figura 1 – Site Github**

**1º PASSO**

* Digitar email válido
* Criar senha
* Definir nome de Usuário

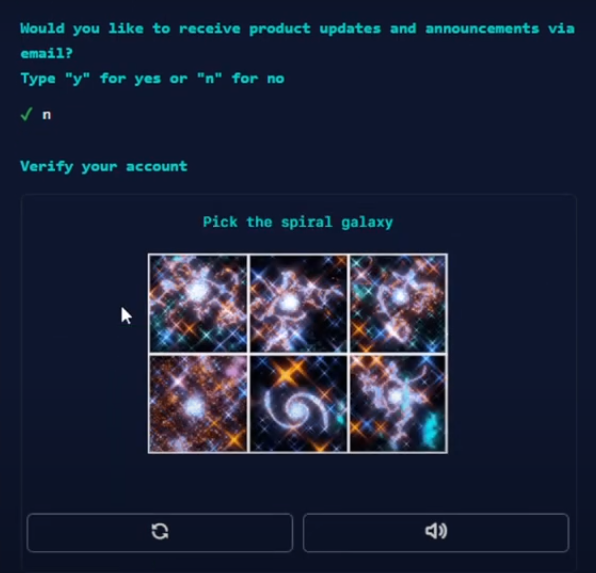
Conforme Figura 2



**Figura 2**

**2º PASSO**

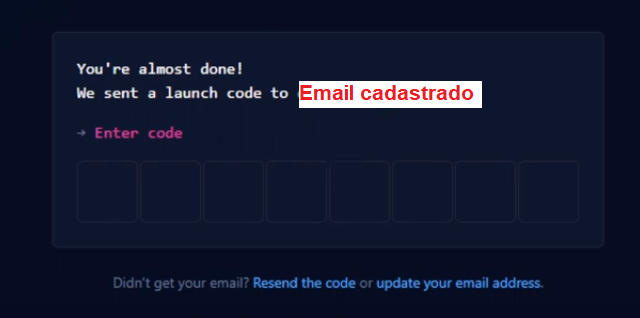
* Responder “y” (yes) ou “n” para aceitar receber atualizações e anúncios via email
* Verificação de conta clicando na imagem (conforme Figura 3).



**Figura 3**

**3º PASSO**

* Inserir Código recebido via email para confirmação da conta (Figura 4)

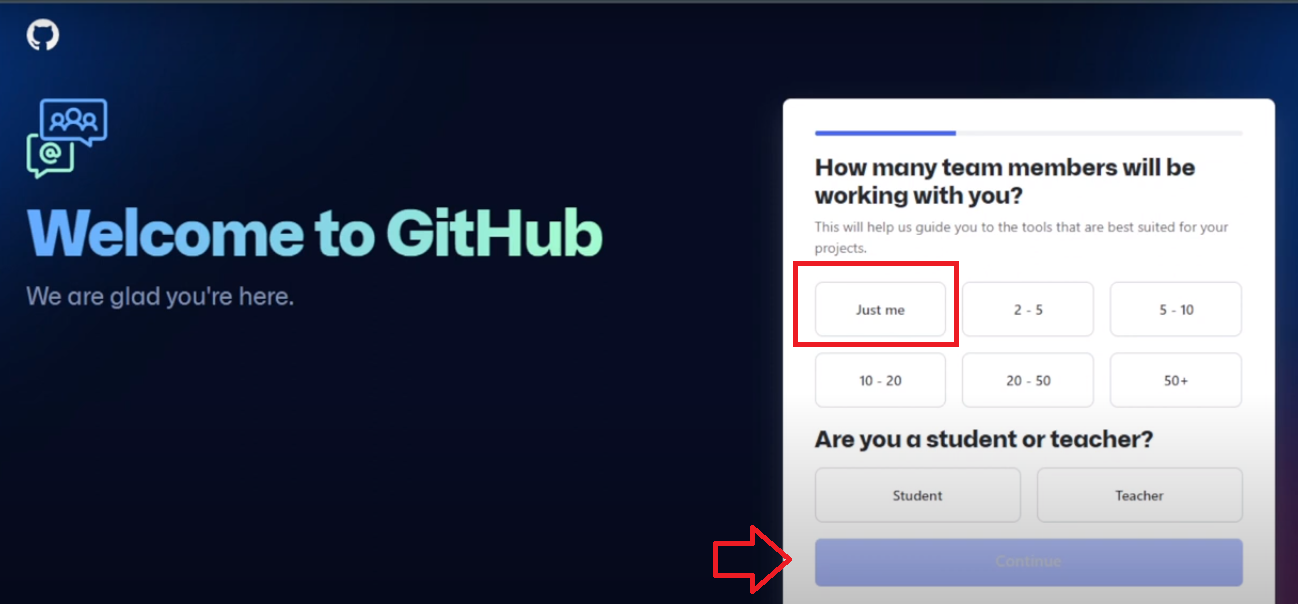


**Figura 4**

**4º PASSO**

* Clicar em “**Just me**”
* Escolher a opção “**Students**”
* Selecionar “**Continue**”

Conforme Figura 5

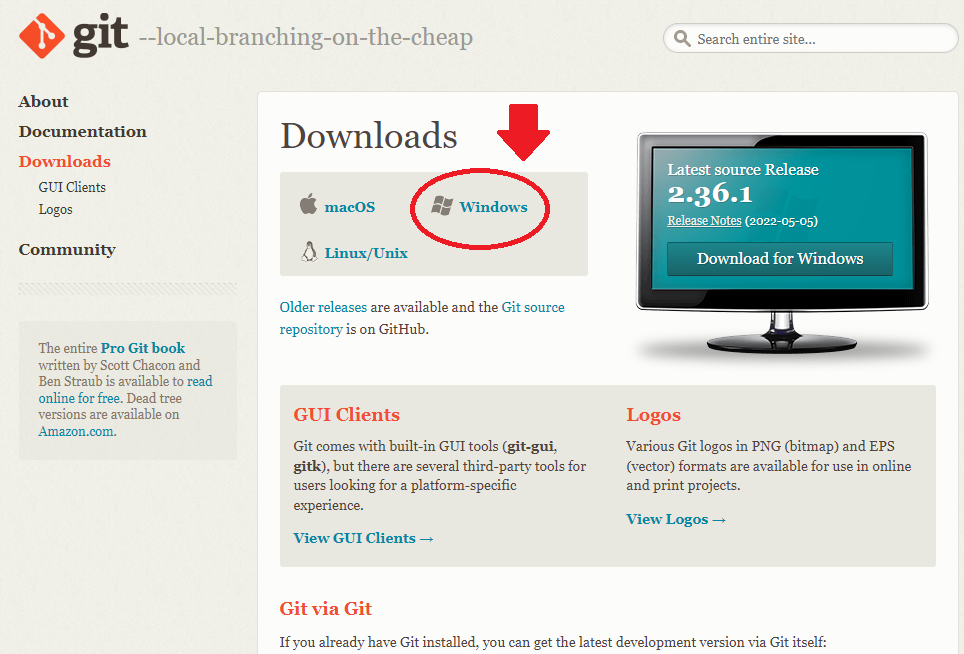
****

**Figura 5**

Pronto! Conta criada com sucesso.

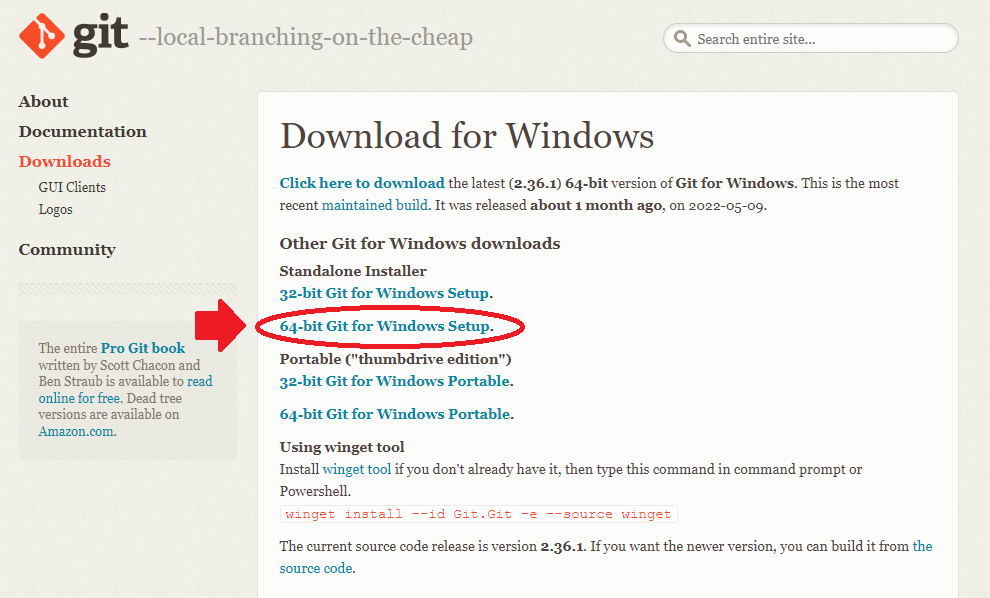
**INSTALAÇÃO DO GIT NO WINDOWS**

Acessar o site do git <https://git-scm.com/downloads> e selecionar a opção “**Windows**” na seção de “**Donwloads**” como mostra (Figura 6):



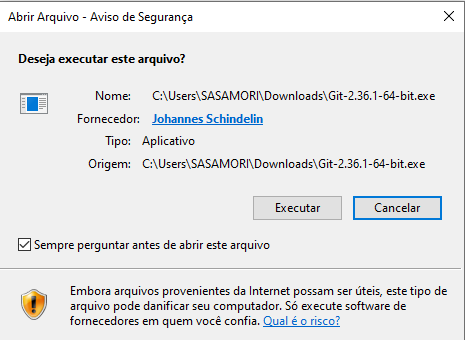
**Figura 6 - Downloads**

Clicar na opção “**64-bit Git for Windows Setup**” para baixar um instalador “Git-2.36.1-64-bit.exe” (Figura 7).

****

**Figura 7**

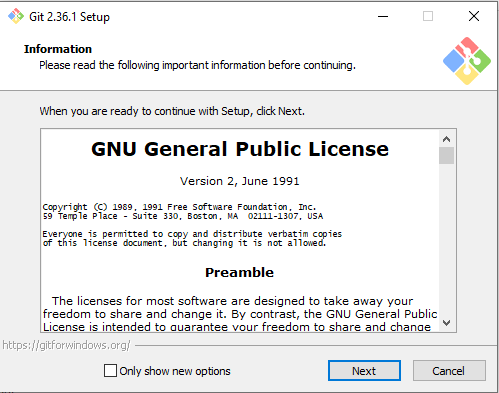
Clicar em “**Executar**” para realizar a instalação (Figura 8)



**Figura 8**

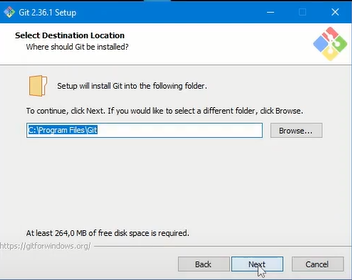
Para os procedimentos a seguir clicar em “**Next**” para realizar a instalação do git.

* Informação



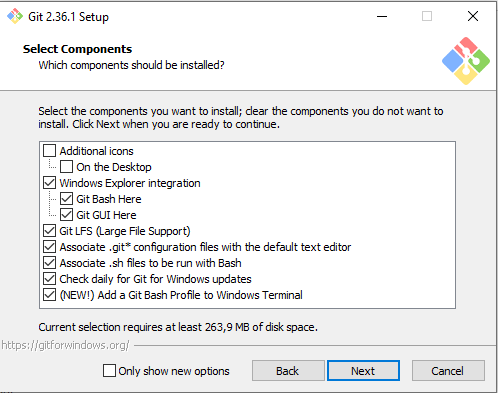
**Figura 9**

* Diretório de instalação



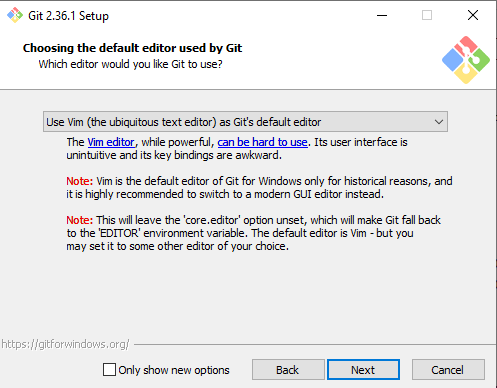
**Figura 10**

* Seleção de componentes



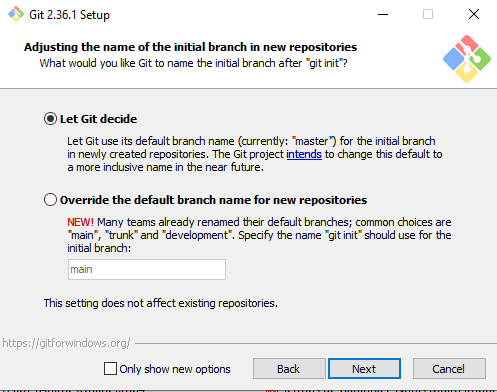
**Figura 11**

* Opção de Editor



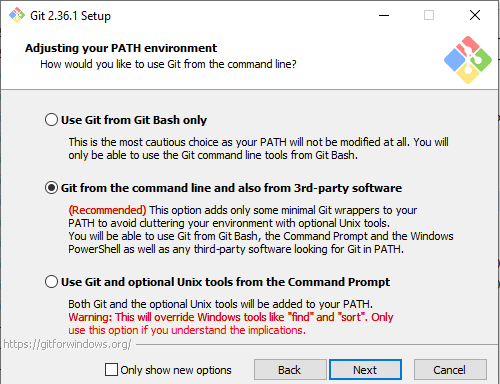
**Figura 12**

* Definição da branch inicial



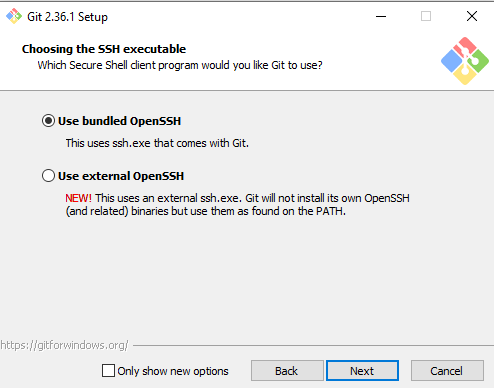
**Figura 13**

* Configuração ajustes



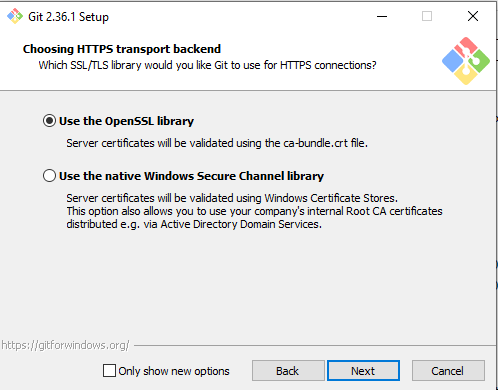
**Figura 14**

* Escolha do SSH executável



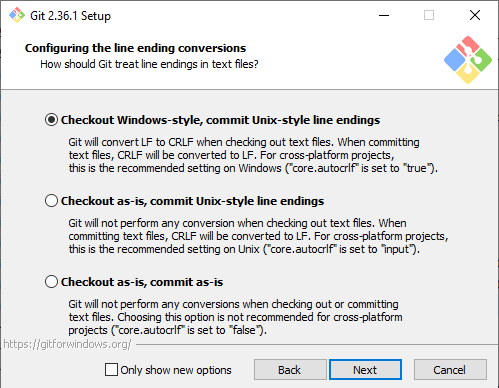
**Figura 15**

* Ajustes



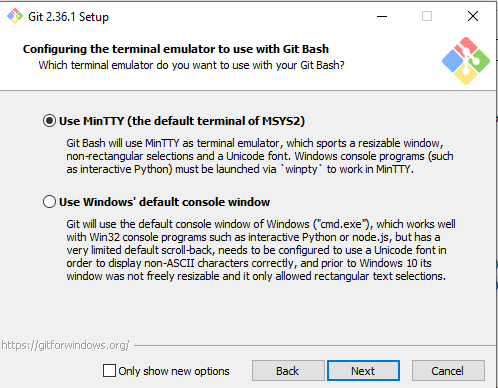
**Figura 16**

* Configuração



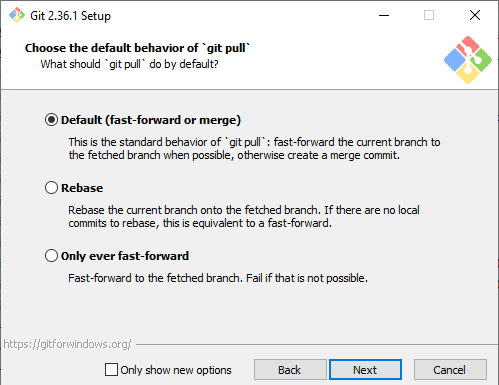
**Figura 17**

* Configuração



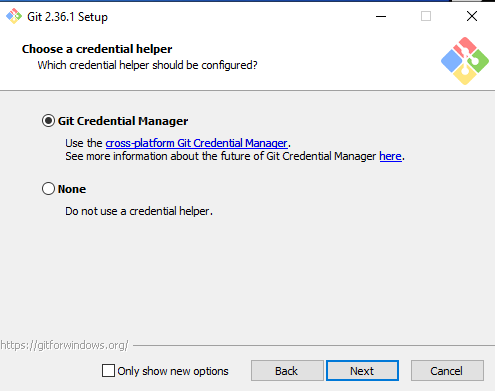
**Figura 18**

* Definição “git pull”



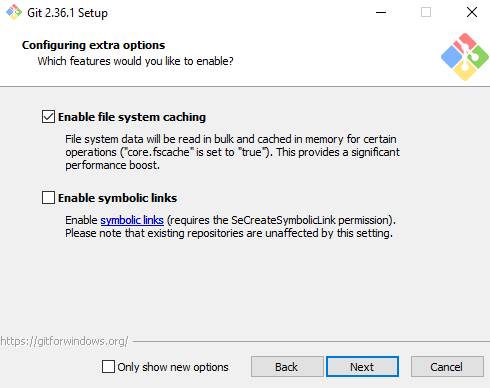
**Figura 19**

* Credenciais



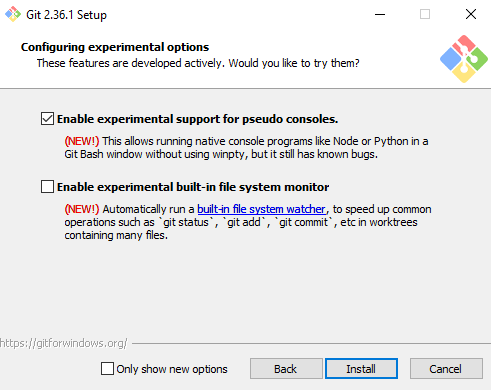
**Figura 20**

* Opções extras



**Figura 21**

* Instalação do git



**Figura 22**

**Concluído**

Após a instalação configurar seu nome de usuário e endereço de e-mail. Isto é importante porque cada commit usa esta informação.

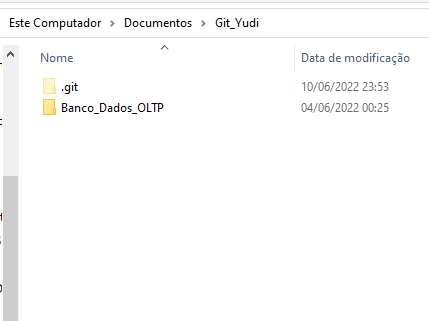
Os comandos são:

* git config --global [user.name](http://user.name/) “login”
* git config --global user.email “email”

**CRIANDO REPOSITÓRIO**

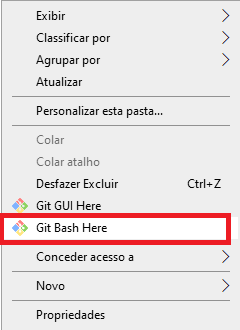
**PASSO 1**

* Criar uma pasta no computador local (Figura 23)



**Figura 23**

* Clicar com o botão direito dentro da pasta criada e selecionar em “ Git Bash Here”



**Figura 24**

**git init**

Esse comando transforma o diretório atual em um repositório do git. É adicionado um subdiretório criando uma pasta oculta (.git)

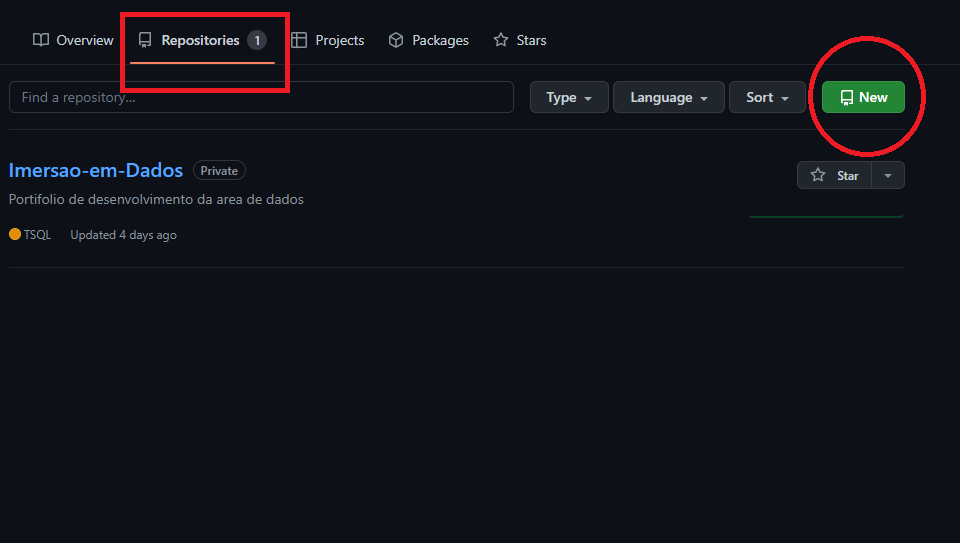


**Figura 25**

**PASSO 2**

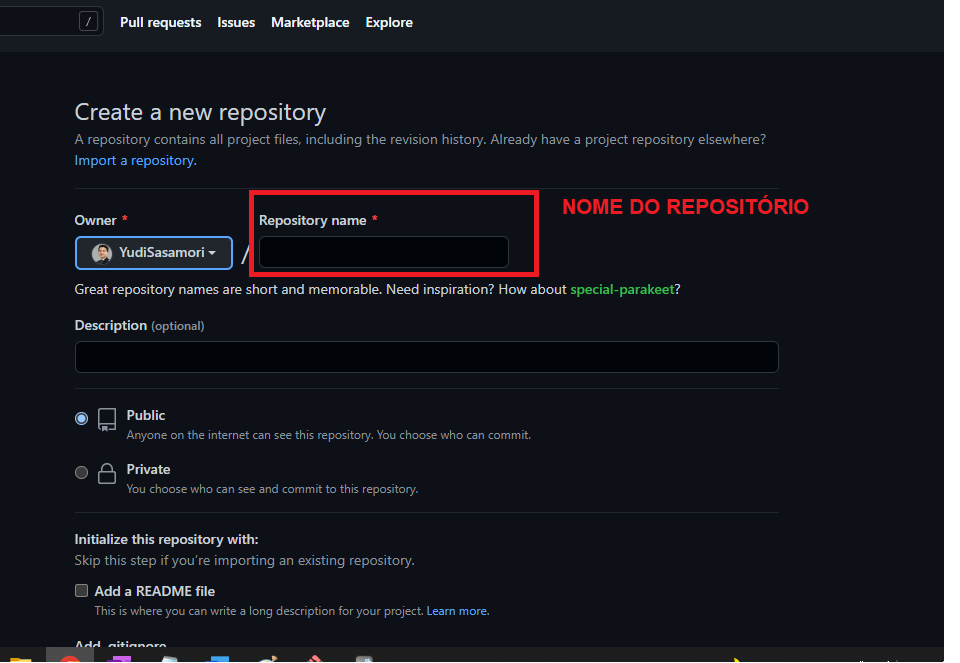
* Acessar a página do github e se logar
* Clicar em “Repositories”
* Clicar em “New”

Conforme Figura 26



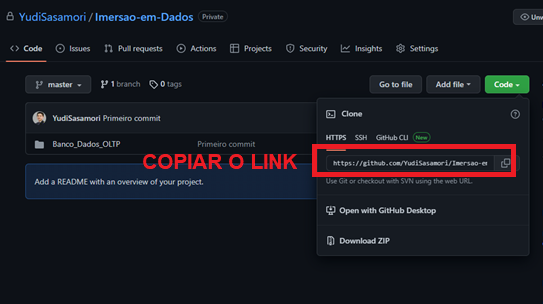
**Figura 26**

* Colocar o nome do repositório (Conforme a Figura 27)



**Figura 27**

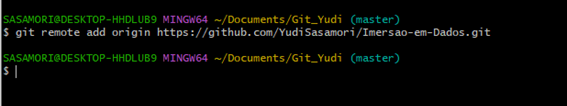
* Na opção “Code” clicar e copiar o link conforme Figura 28



**Figura 28**

**git remote add origin**

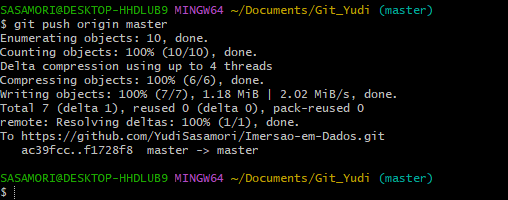
* Após digitar o comando **git remote add origin** colar o link copiado (Como mostra a Figura 29)
* O comando **git remote add origin** vai criar um novo registro de conexão para um repositório remoto.



**Figura 29**

**git status**

* Exibe as condições do diretório de trabalho com os arquivos armazenados no computador com o conteúdo que possui no github
* Os arquivos em vermelho mostram os arquivos que estão diferentes com os do repositório do github



**Figura 30**

**git add**

* Quando criamos, modificamos ou excluímos um arquivo, essas alterações ocorrerão em nosso ambiente local e não serão incluídas no próximo commit (a menos que alteremos as configurações).

**git commit**

* git commit -m "mensagem explicando a mudança no código"
* Este comando é como definir um ponto de verificação no processo de desenvolvimento, para o qual você pode voltar mais tarde, se necessário.

**git push origin “master”**

* Após confirmar as alterações, a próxima coisa que a fazer é enviar as alterações para o servidor remoto, é o que esse comando realiza.

**CONCEITO pull e branch**

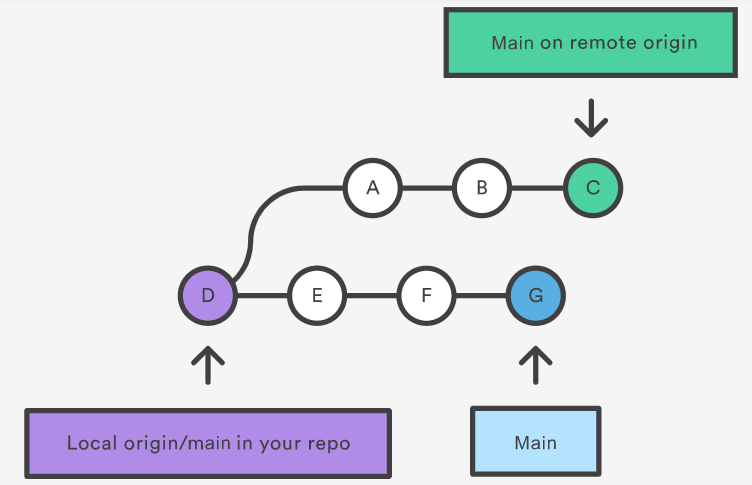
**PULL**

O comando **git pull** é usado para buscar e baixar conteúdo de repositórios remotos e fazer a atualização imediata ao repositório local para que os conteúdos sejam iguais. O comando **git pull** é a combinação de dois outros comandos, o **git fetch**, seguido do **git merge** (ATLASSIAN, 2022).

* **git fetch** baixa commits, arquivos e referências de um repositório remoto para seu repositório local.
* **git merge** permite que você pegue as linhas de desenvolvimento independentes criadas pelo **git branch** e as integre em uma ramificação única.

O **git pull** baixa todas as alterações a partir do ponto de divergência entre o local e a ramificação principal. Na figura 31, o ponto é E. O **git pull** busca os commits remotos divergentes, que são A-B-C. O processo de pull então cria o commit de merge local que tem o conteúdo dos novos commits remotos divergentes. (ATLASSIAN, 2022).

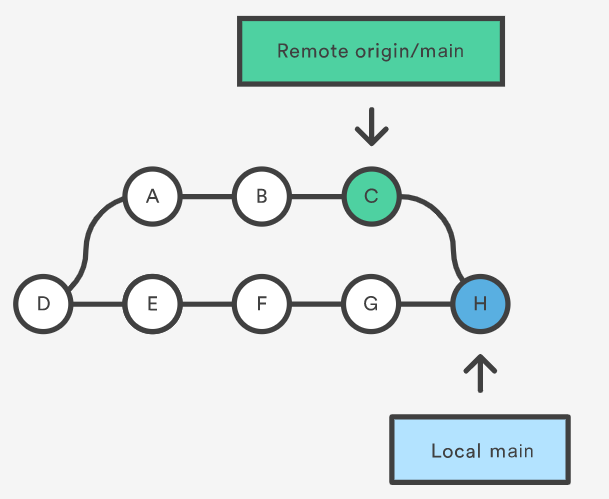
**Figura 31**



**Fonte:** www.atlassian.com

Na figura 32, é possível ver o novo commit H. Esse commit de merge é novo e armazena o conteúdo dos commits remotos A-B-C e uma mensagem de log combinada. Essa é uma de várias estratégias de merge de git pull (ATLASSIAN, 2022).

**Figura 32**

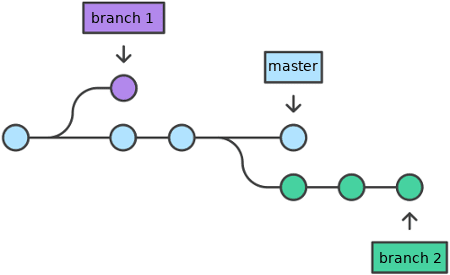


**Fonte:** www.atlassian.com

**BRANCH**

As branchs (ramificações) é como se fosse uma linha independente de desenvolvimento, todo projeto precisa conter pelo menos uma que é criada automaticamente após o primeiro commit, chamada de main/master.

**Figura 33**



**Fonte: www.dev.to/eduardoopv**

Na figura 33 mostra um repositório com duas linhas isoladas de desenvolvimento, uma para um recurso pequeno e outra para um recurso de duração mais longa. Ao desenvolver esses recursos nas ramificações, é possível não apenas trabalhar em ambos em paralelo, mas também manter a ramificação master livre de códigos questionáveis. (ATLASSIAN, 2022).

Para melhor entendimento podemos Imaginar a branch como uma "linha do tempo", possuímos a master onde o progresso de desenvolvimento ocorre naturalmente, ao criarmos uma nova branch criamos uma nova "linha do tempo" com o mesmo código da master, a diferença é que podemos realizar alterações tranquilamente, já que estamos em uma "linha do tempo" paralela (VELTRONI, 2022).

**REFERÊNCIA**

**ATLASSIAN.** git pull. Acesso em jun/2022. Disponível em: <https://www.atlassian.com/br/git/tutorials/syncing/git-pull>

**ATLASSIAN.** git pull. Acesso em jun/2022. Disponível em: https://www.atlassian.com/br/git/tutorials/using-branches

**VELTRONI,** Luiz Eduardo Prado. Conceito de Branch. Acesso em jun/2022.

Disponível em: <https://dev.to/eduardoopv/conceito-de-branch-git-e-github-2bfe>