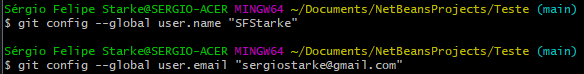
Git & GitHub





|  |  |  |
| --- | --- | --- |
|  |  |  |

Ou 

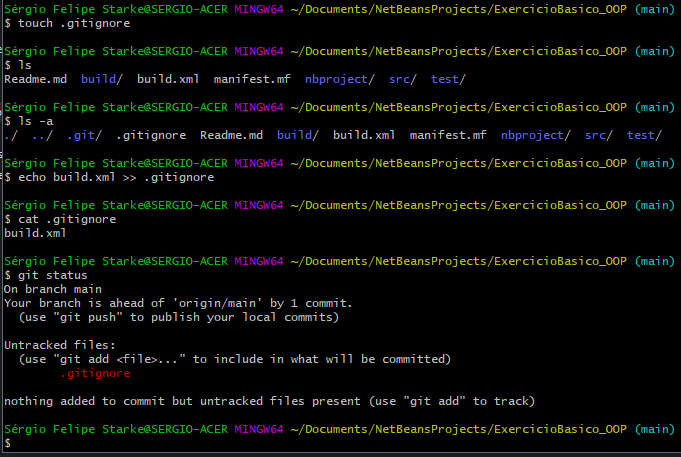
|  |  |  |  |
| --- | --- | --- | --- |
| Sabre quantas branch existem  Alternando entre as branchs  Mudar nome da branch  Criar nova branch  Unir a branch com a principal |  | | |
| Puxar alterações do Remoto para LocalHost | |  |





git pull origin main –allow-unrelated-histories

**Criar GitIgnore**



\*bfm = Botão Função do Mouse.

**“git init” Incluir** Progeto ao **Repositório** **Git** no **NetBeans**:

No Progeto / \*bfm / Controle de Verção / Inicializar Repositório Git... / OK /

Mostrar Alteração:

No Progeto / \*bfm / Git / Mostrar Alterações / IMPORTANTE: Efetuar “Ignore”/

**“git add .”** Adicionar **ADD**:

No Progeto / \*bfm / Git / Adicionar /

**git commit -m**  ”titulo commit”  **Commit** Progeto no **NetBeans**:

1. No Progeto / \*bfm / Git / Fazer Commit... / titulo commit / Fazer Commit /

|  |  |
| --- | --- |
| 2. No Progeto / \*bfm / Git / Mostrar Alterações / IMPORTANTE: Efetuar “Ignore” Selecionando arqivos com \*bfm / Selecione Arquivos, \*bfm, Fazer Commit... / “Adicionar mensagem”, Fazer Commit... |  |

**“git push –u origin** main**” Enviar** para **GitHub**:

No Progeto / \*bfm / Git / Remoto / Enviar... / Próximo / Finalizar /

**“git pull” Puxar** do **GitHub** para **NetBeans**:

No Progeto / \*bfm / Git / Remoto / Puxar... / Próximo / Finalizar /(Quando necessário: Mesclar)

**Comandos:** (GitBash. “terminal”)

**git –version** / **git status** / **clear** /

Para navegar entre as branch:

**git checkout** \*nome da ramificação / ou “**cd** \*nome da branch”

Início:

(Cria o Repositório “Vasio”)

1 **git init**

(Prepara arquivo para o Commit ”Stady”)

2 **git add** nome do arquivo (Ex: Readme.md)/ arquivos.md = MarkDown



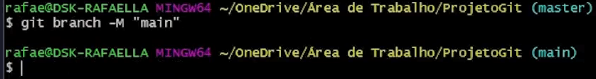
Fazer Commit:

1 **git commit -m**  ”titulo commit” (m de message|Cada Commit, Recomenda-se Mensagem Expecífica)



Renomear nomenclatura principal:

**git branch –M** ”Nome da Branch Principal. Ex; de master para main ”



Vincular Repositório **LocalHost** em Repositório **Remoto**:

Antes é necessario criar um Repositório no GitHub, por conta da url “endereço”.

Vincular ao GitHub: ”origin” É o “apelido” do repositório do GitHub...

1 **git remote add** origin url...



”Empurra” do Git para o GitHub:

2 **git push –u origin** main(git, Comando padrão|push –u, Envia|origin, Nome do Repositório do GitHub| main, Nome da Branch Principal)

****

Vercionamento ”**Alterações**, nova verção”:

1 **git add .** (O ponto, seleciona todos os arquivos existentes no projeto)

Ou especifico; Ex: **git add** Readme.md arquivos.md = MarkDown

2 **git commit –m**  ”titulo commit” (“Salvar” Commit Alterado LocalHost)



(git remote add, só é necessário na primeira etapa ”Vinculação” no GitHub)

(“Empurra” o Rapositório do Git, para o GitHub)

3 **git push** originmain(git, Comando padrão|push, Envia|origin, Nome do Repositório do GitHub| main, Nome da Branch Principal)



Branch ”Ramificação”:

Primeiro se **cria** a **branch** **alternativa**, para só então fazer as edições: (**main** passa a ser **novo-botao**)

1 **git checkout –b “**\*novo-botao**”**



Quando concluido o desenvolvimento:

2 **git add .**

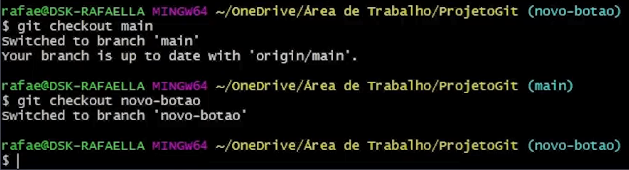
3 **git commit –m** “novo botao” (Commit, da Branch, com nova mensagem)



4 **git push** origin\*novo-botao

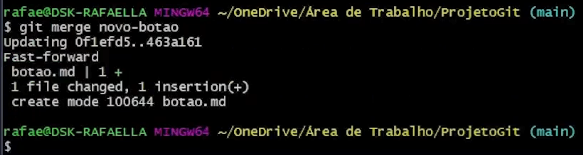


Para alterar à outra branch: git checkout \*nome da ramificação Ex; main

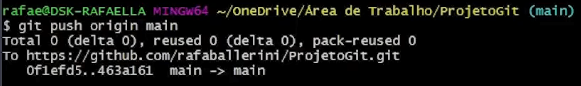


Merge ”Unir”:

1 **git merge** \*novo-botao



2 **git push** origin **main**



**Pull** Alterações do Repositório, Remoto para LocalHost:

No gitBash, dentro da pasta para a atualização / **git pull**

Cópiar Repositório;

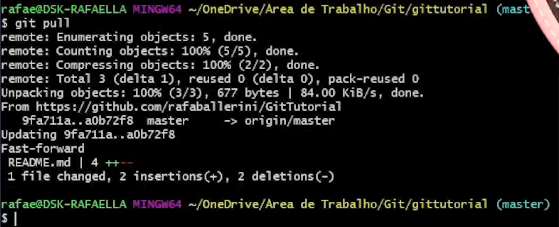
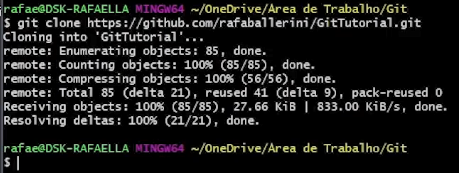
Fork “De Outro GitHub, para meu GitHub”:

No GitHub de outro desenvolvedor, selecione o repositório / Fork /selecione meu perfil.

Clonar “Para o PC”: \*Botão Função Mouse

No GitHub, selecione o repositório desejado / Code / botão copiar.

No PC, criar pasta para clonar / \*bfm gitBash / **git clone** \*link.git.

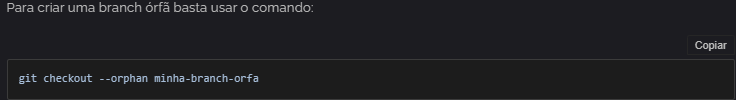
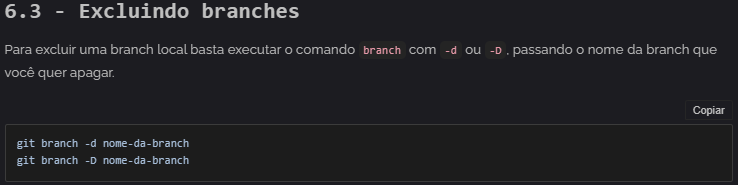
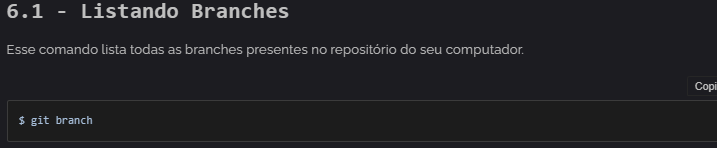
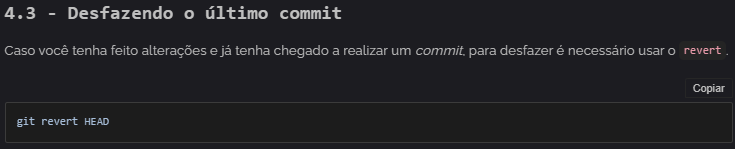


**Pull Request** (“Pedido, Requisição” para Alteração do Progeto de outro GitHub):

No meu GitHub, selecione o repositório de outro desenvolvedor / depois da alteração / Comente no texto de Commit changes / Botão **Commit changes** / Contribute / **Open pull request** / **Create pull request** / Comente sua alteração / **Create pull request**

**Aceitar Pull Request**

No GitHub, selecione seu Repositório / Pull Requestes / Selecione o request / **Merge pull request**.

****

**MarkDown**

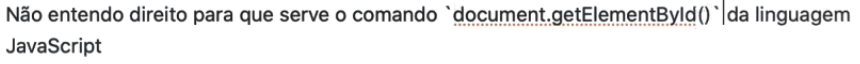
|  |  |  |
| --- | --- | --- |
| Função  Titulo Principal  Titulo Secundário  Titulo Terciário  Linha  Itálico  Negrito  Negrito & Itálico  Riscadas  E-Mogi | Marcas ’Símbolos’  # “Titulo”  ## “Titulo”  ### “Titulo”  --- ou \*\*\*  \*Texto\* ou \_  \*\*Texto\*\* ou \_\_  \_\*Texto\*\_  ~~Texto~~  : | Resultado  Titulo1  Titulo2  Titulo3  \_\_\_\_\_\_\_\_  *Texto*  **Texto**  ***Texto***  ~~Texto~~ |

Criar Link:



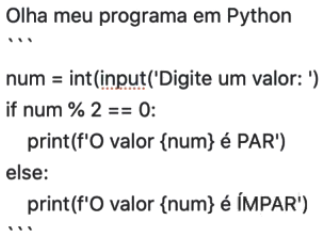
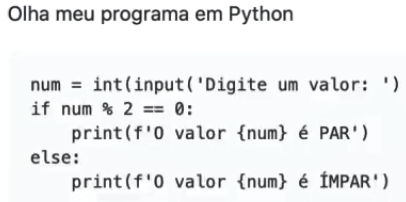
Preview: 

Crase para especificar Comando:



Preview: 

Três Crases, para formatar em código:

|  |  |
| --- | --- |
| \* ou – |  |

**Conceitos:**

|  |  |
| --- | --- |
| **FORK**  **BRANCH**  **MERGE**  **REMOTE**  **PUSH**  **PULL**  **FETCH**  **ISSUE**  **PULLREQUEST** | Importa e ou copia para o git e github, progeto de outro desenvolvedor.  Ramificação do progeto:    Unir “fundir” as branch das commit.  É o Link, entre o app da criação das commit, à plataforma.  “Empurra” o commit do localhost, para Remot.”plataforma”  “Puxa” o repositório do GitHub, para o pc.”localhost” (Faz Merge)  “Buscar, Trazer” o repositório do GitHub, para o pc. (Não faz Merge)  “Publicação, Questão. Consulta de duvida”  “É uma Fork do progeto alternativo. (Solução)” |

**Git**: Software de Controle de Verção. V.C.S. “Versionamento”

**GitHub**: Plataforma “Rede Social” p. Hospedar na web as\*commit. Ele trabalha com \*repositório. ”Local onde é guardado Arquivos”.

\*Repositório é o diretório”pasta” onde são armasenados os progetos.

\*Commit é o gerenciamento de dados e controle de verções. Torna permanente um conjunto de alterações, ou seja, de efetivar as alterações.”Salva”

Chronicle