

AULA DE GIT MÓDULO 10

Instalar o Git



Abrir o terminal e verificar a instalação

Git --version - Verifica instalação e versão do git

Na primeira vez que for utilizar o git configure nome e e-mail:

Git config --global user.name "Seu Nome"

Git config --global user.email seu@email.com

Para verificar se está tudo ok basta digitar:

Git config user.name - para nome

Git config user.email – para o e-mail

Ls – para verificar que diretório está

Cd .. – para voltar diretórios

Vá para o diretório raiz c:

Ls – pra confirmar que está no diretório raiz

Mkdir repositório – cria uma pasta de nome repositório

Cd repositório – para entrar no diretório repositório

Mkdir aula-git – para criar um diretório dentro da pasta repositório

Cd aula-git – para entrar na pasta

Até aqui o git ainda não entende que a pasta criada é para o repositório, é preciso informar, digite no terminal:

Git init – para iniciar o git dentro da pasta aula-git

Vai ser criado um arquivo .git diretório oculto dentro da pasta aula-git que é onde tudo acontece.

Agora clicar com o botão direito dentro da pasta aula-git e escolher a opção “Abri com o Code” para abrir o Visual Studio Code já com a pasta aula-git aberta.

Agora criar alguns arquivos:

Readme.md – arquivo markdown padrão do github

Pode digitar uma descrição dentro do arquivo

Agora crie:

Index.html – arquivo html, inserir os comandos básicos com a ! ou digitando doc e enter

Crie o arquivo:

Styles.css – arquivo de folha de estilo, adicionar o comando abaixo:

```
h1{  
  color:blue  
}
```

Agora voltar para o terminal bash e digite:

Git status – irá mostrar o status do git

Git add <nome do arquivo> - para adicionar arquivos no git

Git status – para verificar se o arquivo foi adicionado

Outra forma de adicionar vários arquivos de uma vez:

Git add . – para adicionar todos os arquivos

Agora o próximo comando é pra commitar os arquivos:

Git commit -m “primeiro commit”

Se der um git status vai ver que não tem mais arquivos para commitar

Abra seu github e crie um novo repositório de nome aula-git, deixe publico.

O github vai apresentar uma lista de comandos

Digitar no terminal:

git remote add origin <https://github.com/SimoneSS15/aula-git.git>

Depois o commando abaixo:

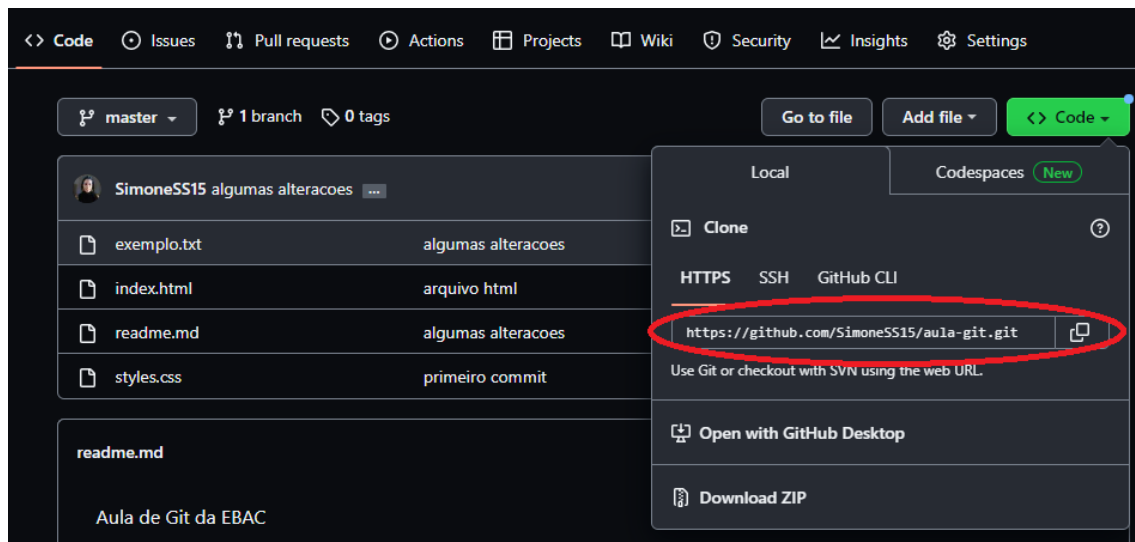
git push -u origin master (master é o nome da branch)

No meu caso não deu certo esses dois últimos comandos no terminal, então vou usar o github desktop para enviar os arquivos para o repositório do github.

Clonando um repositório

Excluir a pasta criada de nome aula-git no notebook

Acessar o github e clicar no repositório aula-git, conforma imagem abaixo clicar em “Code” e copiar o link apresentado.



Abrir o terminal bash e digitar:

Git clone <https://github.com/SimoneSS15/aula-git.git>

Observe a a pasta aula-git será criada com todos os arquivos dentro

Lembre-se de estar no diretório correto para digitar o comando, nesse caso tem que ser do diretório repositório.

Entrar na pasta aula-git e abrir os arquivos no VS code, modificar o arquivo readme.md salvar e voltar para o bash

Entrar na pasta aula-git pelo bash e digitar:

Git status

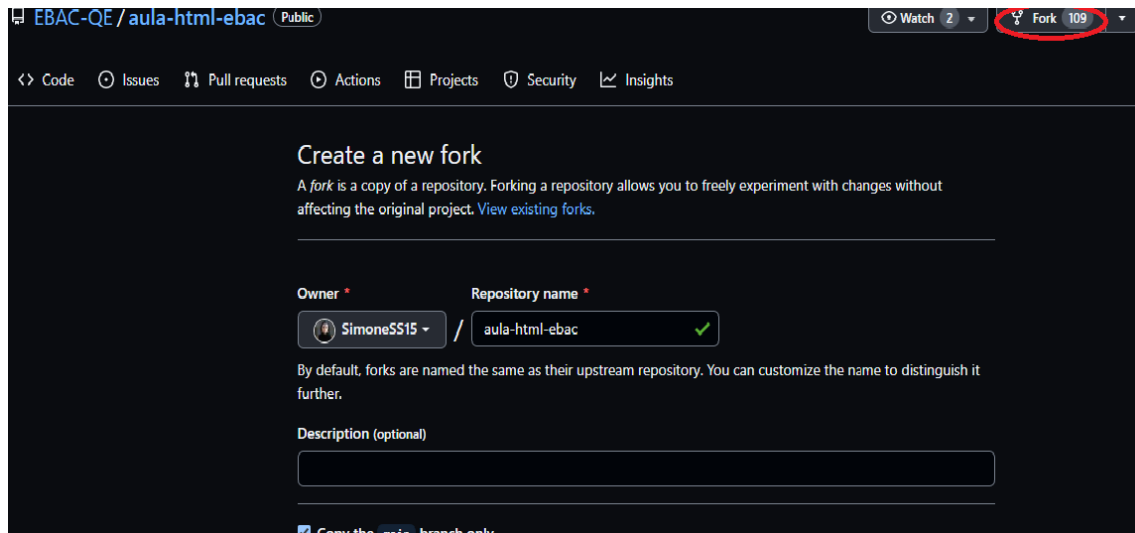
Observe que o arquivo readme.md aparece como modificado, fazer o commit e push usando o github desktop.

Outra forma de clonar um repositório, desta vez de outra pessoa.

Acesse <https://github.com/ebac-qe> e clica no repositório aula-html-ebac.

Poderia fazer o processo anterior de clonagem, no entanto não seria possível fazer as alterações e enviar novamente para o github.

Então clique na opção fork conforme imagem abaixo:



Vai ser criado um repositório diretamente na sua conta do github, fork é literalmente dar uma garfada no repositório de outra pessoa ou conta. Agora acesse o bash e clonar novamente o repositório que foi feito fork

Git clone <https://github.com/SimoneSS15/aula-html-ebac.git>

Dar um ls pra ver se foi criado realmente dentro da pasta repositório.

Puxando e ignorando modificações:

Acessar o meu github no repositório aula-html-ebac e criar um arquivo txt Exemplo.txt e commitar pelo próprio github.

Perceba que o arquivo foi criado no github, no entanto não está na sua máquina (no notebbok), verifique isso abrindo a pasta aula-html-ebac com o VS Code.

Entre na pasta acima pelo bash e digite o comando abaixo:

Git pull

Ele vai trazer o arquivo exemplo.txt para a pasta aula-html-ebac, visualize no bash e no VS Code.

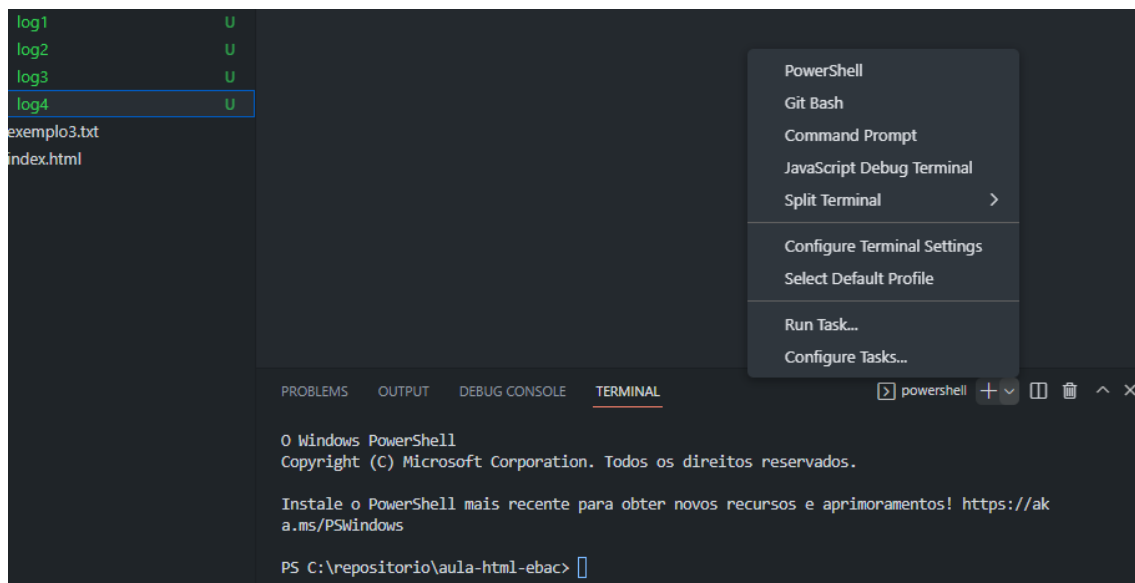
Então sempre antes de começar a trabalhar é bom dar o comando git pull para trazer para a máquina local qualquer alteração feita no github.

Para reforçar, acesse novamente o arquivo exemplo3.txt e faça uma alteração via github mesmo.

Volta pro bash e faz um git pull novamente, observe a alteração feita no VS Code.

Agora no VS Code crie uma pasta de nome Logs e dentro dela vários arquivos de nome log, log1, log2, log3, log4.

Abra o terminal do próprio VS Code, Terminal – new terminal, ao invés de usar o git bash, mas esse terminal não é colorido como o bash, então basta escolher o git bash no próprio terminal do VS Code conforme a imagem abaixo.



Agora dando um git status é possível ver os arquivos de log criados com a pasta, vamos adiciona-los dando o comando:

Git add .

Novamente o git status para verificação.

Acontece que de repente possa ser arquivos que eu não gostaria de adicionar, então eu posso usar uma comando:

Git reset – ele volta um procedimento anterior, desfaz o comando anterior.

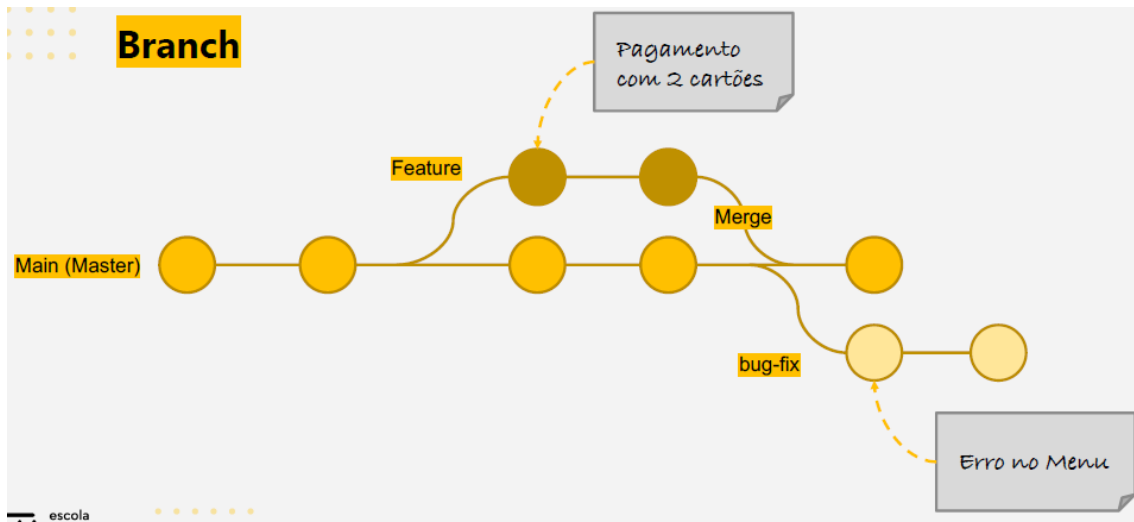
Para ignorar arquivos que não quero adicionar, crie um arquivo .gitignore e dentro dele adicionar a pasta Logs dessa forma:

Logs/

Salva as alterações e no terminal digite git status, perceba que não aparece mais os arquivos de logs, pois estão sendo ignorados. Qualquer arquivo criado e adicionado no arquivo .gitignore será ignorado, exemplo:

Crie um arquivo txt de nome arquivo_ignorado, e adicione no arquivo .gitignore, salva e veja o resultado.

Ramificações (Branches):



Abra o VS Code com a pasta repositório, abra o terminal escolha git bash e crie uma pasta:

Mkdir aula-cinco-git

Entre na pasta:

Cd aula-cinco-git

Vá para o github e crie um novo repositório com o nome de aula-cinco-git

Copiar os comando listados (um de cada vez) no terminal no VS Code

echo "# aula-cinco-git" >> README.md – vai criar o arquivo readme.md

git init – para avisar pro git que está pasta vai ser monitorada por ele.

git add README.md – adiciona o arquivo readme.md no stage do git

git commit -m "primeiro commit" - primeiro commit

git branch -M main - trocar de branch master para main

`git remote add origin https://github.com/SimoneSS15/aula-cinco-git.git` - faz a ligação do repositório local com o repositório remoto que é o github.

`git push -u origin main` – esse commando empurra tudo para o github, mas não funciona no meu notebook, então vou dar um push com o github desktop.

Visualize as alterações no github.

Agora crie uma nova branch

`Git granch funcionalidade1` - cria uma nova brach de nome funcionalidade1

`Git branch` – mostra quantas branch existem e em qual estou trabalhando

`git checkout funcionalidade1` - muda para a branch funcionalidade1

No VS Code crie um arquivo index.html na pasta aula-cinco-git e depois digite o comando `git status`, verifique que tem um arquivo para ser adicionado.

`Git add .` – adicionar o arquivo index.html

`Git commit -m "Funcionalidade 1"` – commita na branch funcionalidade1

`Git push origin funcionalidade1` - empurra para a branch funcionalidade1

Volta para branch main e faz um merge

`Git checkout main`

Git merge funcionalidade1 – junta as alterações da branch funcionalidade1 com a branch main

Git branch -d <nome da branch> para deletar uma branch

git checkout -b funcionalidade2 – já cria uma branch e entra nela

Note que ao criar essa nova branch de nome funcionalidade2 já existe os arquivos criados na branch anterior, os arquivos index.html e readme.md

Faça uma alteração no arquivo index.html depois commita e faz um push (não esquecer de fazer isso)

Git checkout main – para voltar pra branch main

Agora faz um merge:

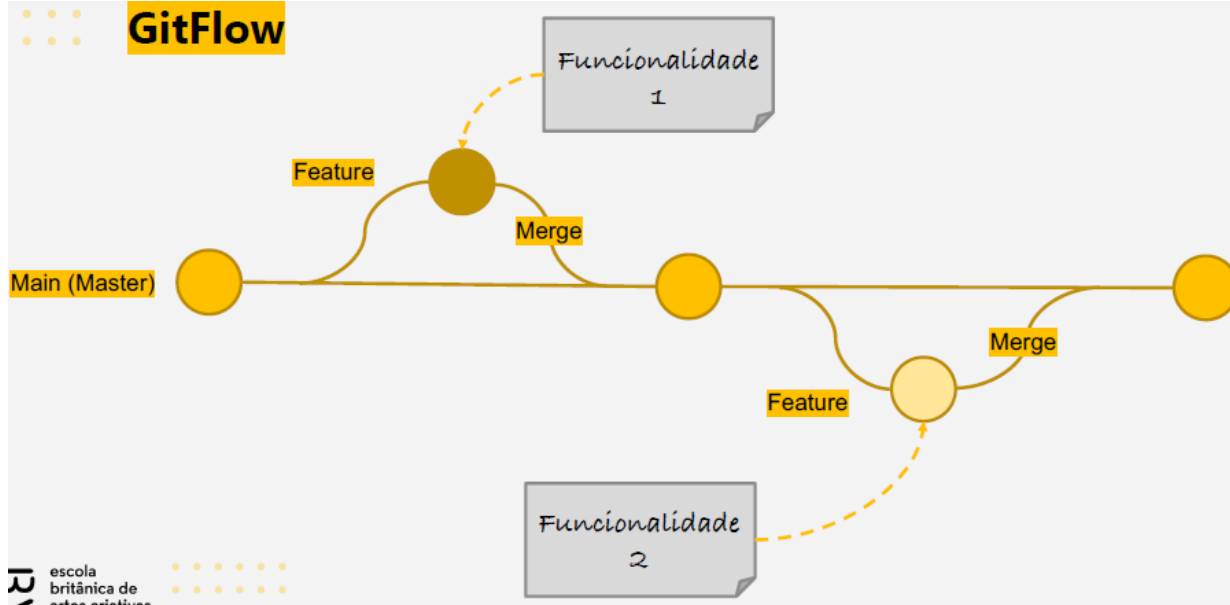
Git merge funcionalidade2 – puxa as alterações feitas na branch funcionalidade2 para a branch main

Git push origin main – para atualizar tudo na branch main

Abaixo uma representação de tudo que foi feito:



GitFlow



escola
britânica de
artes criativas



