

CRIADO PELA FERNANDA MAKI HIROSE

o que é git? o que é versionamento?

git = software de controle de versão

github = plataforma de rede social para programadores

comandos para terminal windows e linux

tab: auto completa o texto

seta para cima: navega entre os últimos comandos executados

comandos para terminal no windows

listas arquivos/pastas: dir

voltar de pasta: cd ..

criar uma pasta: mkdir pasta

deletar pasta: del pasta

comandos para terminal no linux

listas arquivos/pastas: ls

voltar de pasta: cd ..

criar uma pasta: mkdir pasta

deletar pasta: rm -rf pasta

quem foi o criador do git?

o mesmo criador do Linux, o Linus Torvalds

primeiros passos

- baixe o git: <https://git-scm.com/>

- baixe o github para desktop: <https://desktop.github.com/>

após baixar, clique no git > apenas mude uma configuração para: [use visual studio code as git's default editor](#) depois clique em [next](#)

no github para desktop > logue na sua conta > da parte de email coloque o seu email > continue

criando um repositório

create a new repository on your hard drive >

dê o nome do repositório, coloque uma descrição, coloque um local path, adicione um readme, adicione gitignore se precisar, na licença escolha 'mit license > create repository

clique em publish repository > keep this code private'

vá no github da internet > copie o link de clone do repositório, clone ele dentro do visual studio no terminal (git clone + link), adicione os códigos dentro desse local clonado

faça um commit e um push

issues

no seu repositório tem a parte de issues

1- quando um issue já foi resolvido, feche o issue. se você não quer que tenha mais comentários clique em 'lock issue'

2- para escrever algo em negrito: ****use 2 asteriscos****

3- para enviar uma imagem arraste a imagem, vai aparecer como código, mas no preview vai aparecer como imagem normal

4- [se você fizer isso] (e aqui passar o link, ao clicar na mensagem o link irá ser aberto)

5- para colocar o link de uma issue coloque # e o número dela

6- para fazer uma lista coloque um *

saved replies

vá em settings > saved replies

aqui você pode criar replies salvos, então quando você for responder alguém pode selecionar algum saved reply

para aparecer o saved reply ao responder alguém basta clicar na setinha onde se localiza a área para enviar mensagens

markdown

****negrito****

itálico

~~texto excluído~~

título 1

título 2

título 3

--- faz uma linha

__*em negrito e em itálico*__

listas com números:

1. lista

2. lista

1. lista

2. lista

caixinhas:

☐ criar a página principal

caixinhas marcadas:

☒ finalizar a reunião com o cliente

criar tabelas:

nome | idade | sexo

Fernanda | 17 | feminino

colocar código:

```
`print(ler)`
```

```
'''
```

(ou com 3 crases)

```
'''
```

para responder alguém:

clique nos 3 pontinhos > quote reply

tag HEAD

a tag HEAD é o ramo atual

stash

todos os arquivos que não foram comitados ficam dentro do stash, então é possível se mover para outra branch sem dar errado, toda vez que você criar uma branch crie um stash

- **criando um stash:** `git stash save` "Título da ação do stash"

- **para listar o stash:** `git stash list`

(no git stash list vai listar o índice do stash criado)

- para excluir um stash: `git stash pop 1`
(o 1 é apenas um exemplo, é para passar o número do índice)
- para limpar um stash: `git stash clear`

branches e merge

o ramo master é o ramo principal (quando você dá um commit você está no ramo master)
você pode colocar o código do ramo master para o origin (quando você dá um push vai para o github)
é possível criar branch (ramificações) e juntar elas como o merge

para criar uma branch vá no aplicativo do github para desktop > branch > new branch
para fazer um merge > vá no aplicativo do github para desktop > vá no merge > clique em 'branch' > merge into current branch > clique no branch que você quer juntar com o master

se der um erro falando que vai ter um conflito, pode aceitar do mesmo jeito, depois vá no seu arquivo e na área que tiver o erro clique em 'aceitar a mudança atual', depois dê um commit e um push

para apagar uma branch > vá no aplicativo do github para desktop > branch > delete

alerta: não faça o rebase, porque ele reescreve todo o código

comando de branch

- para saber qual é a branch atual: `git branch`
- renomeando a branch: `git branch -m nome-antigo nome-novo`
- mudando de branch: `git checkout nome-da-branch`
- deletar uma branch: `git branch -d nome-da-branch`

hospedar um site no github

ele só hospeda sites simples, como html, css e javascript

coloque o seu projeto em um repositório, vá no seu repositório > settings > github pages > source = master branch

adicionar arquivos com mais de 100mb

1ª etapa

primeiro, por se tratar de uma extensão, é necessário que seja instalado.

`git lfs install`

é necessário executar o comando apenas uma vez por conta de usuário

2ª etapa

iremos informar quais extensões de arquivos o Git LFS irá gerenciar.
podemos adicionar manualmente no `.gitattributes`.

`*.h5 filter=lfs diff=lfs merge=lfs -text`

ou via terminal:

`git lfs track "*.h5"`

nesse exemplo estamos utilizando a extensão ".h5", não esqueça de modificar o comando para o seu cenário.

3ª etapa

certifique-se que o .gitattributes seja rastreado.

`git add .gitattributes`

4ª etapa

basta seguir com o commit do seu arquivo.

`git add model.h5`

`git commit -m "add file with trained neural network"`

`git push origin master`

ver o histórico dos commits

digite: `git log`

(para visualizar os commits de uma forma mais rápida clique em 'page up' e 'page down', para sair deste modo clique a tela 't')

ver o histórico dos commits em uma pasta específica:

digite: `git log nome-da-pasta`

para sair do histórico digite: `:q`

trazer o histórico de commits de uma forma resumida: `git log --oneline`

trazer o histórico de forma mais gráfica no terminal: `git log --graph`

trazer o histórico de forma mais gráfica fora do terminal: `gitk`

para reverter commits: `git revert 11a5b`

(no seu github vai aparecer o id dos commits), adicione o id que é o commit que você quer voltar, apenas os 5 primeiros caracteres (o reset também volta atrás, mas faça com o revert para não ter que decorar mais códigos)

dica para criar commit: sempre com a primeira letra maiúscula, frase no imperativo, exemplo: Remove a funcionalidade.

como adicionar imagem no readme

crie um issue e coloque uma foto, vá carregar a url da foto, copie esse código, depois exclua a issue, adicione a url copiada e cole no seu readme