Git/Github

Comandos Git

Bora aprender mais a versionar os nossos códigos ?

Mas ai fica a dúvida, para que serve essa ferramenta.

Vamos colocar uma situação hipotética aqui para ficar mais claro

Situação-problema

Você está trabalhando com 4 desenvolvedores em um projeto completo. Cada um é responsável por uma parte do projeto e precisa desenvolver as funcionalidades ao mesmo tempo. Ao decorrer do projeto, teremos que juntar essas partes desenvolvidas em ambientes diferentes e fazer com que o projeto funcione completamente. Como você juntaria essas partes ?

Mandaria mensagem no whatsapp e iria pedir o arquivo zip ? NÃO

Copiaria o código e enviaria em alguma rede social para que o outro lado colasse no editor de código e estivesse pronto ? NÃO

O jeito mais simples de todos é fazer o versionamento com o git.

Ele armazena os códigos em uma "rede social" de projetos como Github, Gitlab, Biickbucket e entre outros.

Para isso, precisamos de alguns comandos para facilitar a nossa vida.

Terminal

Com o git devidamente baixado e instalado, vamos a sua configuração.

Os comandos serão os mesmos para o git, mas a abertura do terminal se diferem de acordo com o Sistema Operacional.

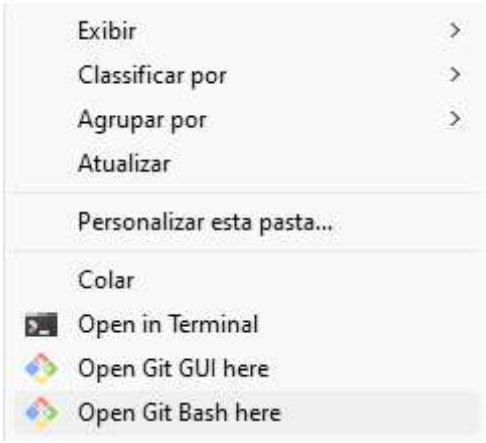
Linux/ MacOS

Para usuários de alguma distribuição linux ou macOS, normalmente o git já vem presente nesses sistemas, portanto, apenas abrir o terminal já é suficiente para utilizer os comandos git

Windows

Para o Windows, precisamos abrir o terminal gitbash. siga esses seguintes passos:

- Clique com o botão direito na área de trabalho ou no diretório desejado
- Procure a opção **OPEN GITBASH HERE.** Caso não encontre de primeira, clique **Mostrar mais opções**



Com isso, teremos o terminal:

E voilá, vamos para os comandos

Configurando

Para hospedar os códigos do seu computador (local) para o Github (remoto), temos que primeiro, fazer a ligação entre os dois. Para isso, digite no terminal os seguintes comandos

```
git config --global user.name "seu_nome_de_usuario_aqui"
git config --global user.email "seu_email_do_github"
```

⚠️ É importante que o email digitado aqui seja o mesmo do email cadastrado no github.

Pronto, agora começamos os jogos

Iniciando o Git

Para inicializar o repositório, utilizamos do comando:

```
git init
```

Isso irá gerar uma pasta .git que fica oculta no seu diretório

Mudando nome da branch

É necessário mudar o nome da ramificação desse projeto para main. Até um tempo atrás, a ramificação se iniciava com o nome de master.

Para mudar a branch para main, utilize o código

```
git branch -M main
```

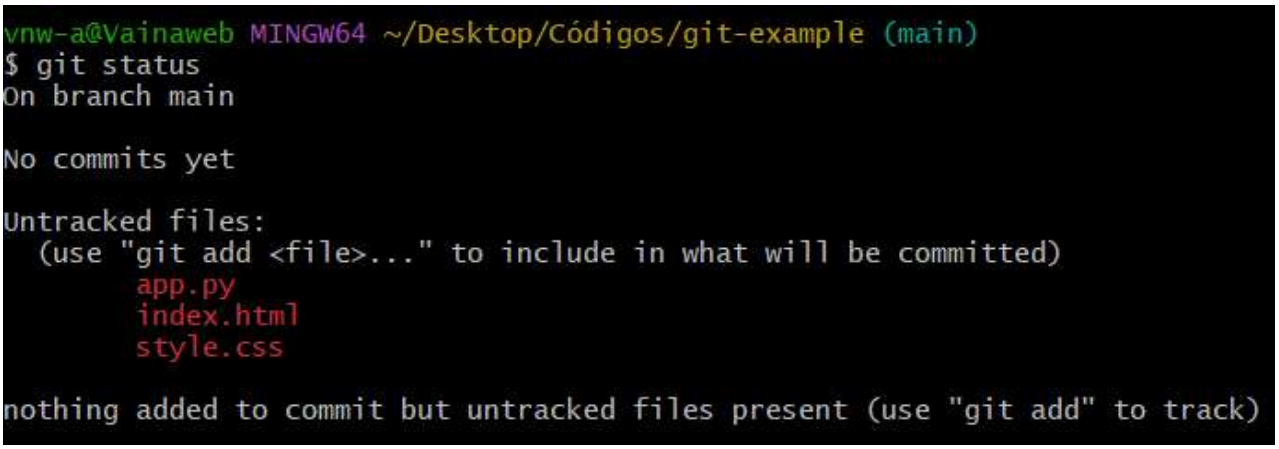
Verificando estados

Bom, sabemos que temos as versões do nosso projeto e para adicionarmos é necessário adicionar esse comando ao stash

Para fazer essa verificação, podemos nos basear pelas cores.

digite o comando:

```
git status
```



A cor VERMELHA indica que os arquivos não estão sendo monitorados ainda. Para monitorá-los, digitamos o comando

```
git add nome_do_arquivo
```

É possível fazer a adição de todos os arquivos ao mesmo tempo com o ponto (.)

```
git add .
```

Com isso, monitoramos os arquivos.

A cor VERDE indica que agora eles estão sendo monitorados e prontos para serem commitados.

Commitando

Agora vamos commitar os arquivos. Em palavras mais simples, adicionar mensagens aos envios feitos ao Github. Isso facilita a organização, rastreio e monitoramento de versões dos códigos.

Para commitar usamos o comando:

```
git commit -m "mensagem do commit"
```

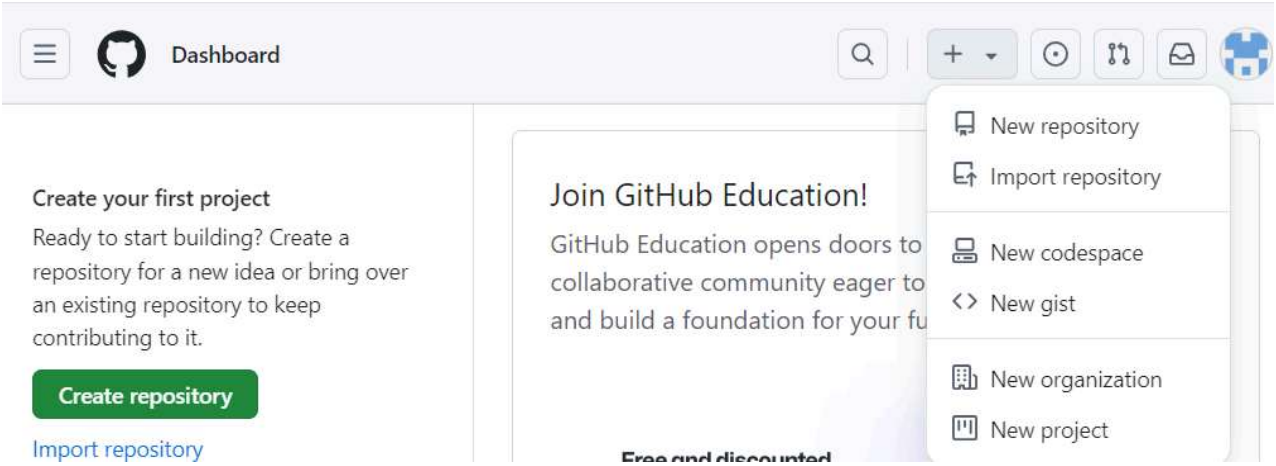
Quando o arquivo é commitado, ele é lançado para o stash, onde só precisamos enviar para o Github

Pronto, agora estamos prontos para enviar para o github.

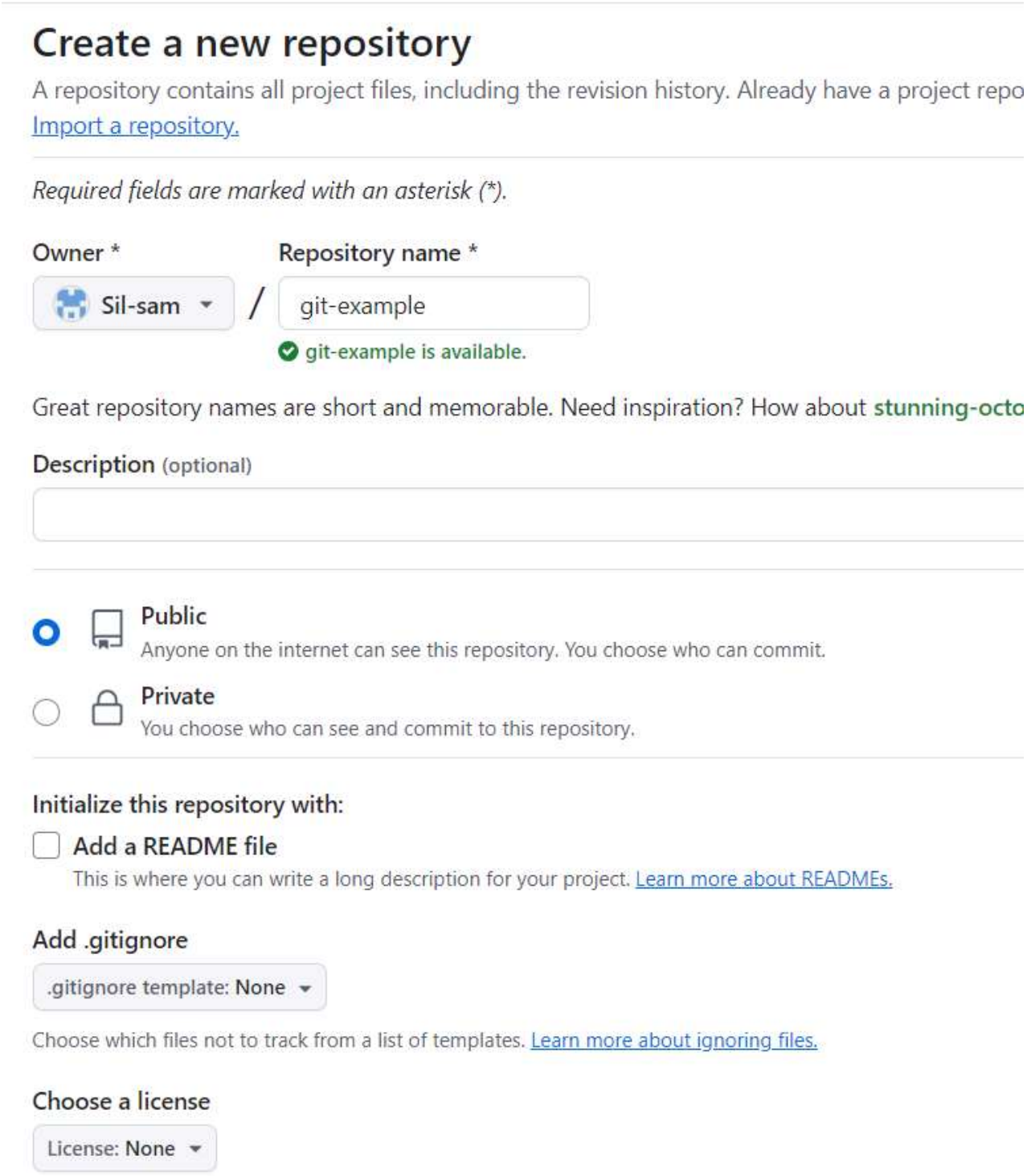
Link com o repositório

Antes de enviar para o github, precisamos linkar o nosso local com o remoto. Para isso, crie um repositório no serviço de hospedagem e pegue o link HTTPS.

Na tela inicial do github, procure o `+` e selecione a opção **new repository**



Seremos então redirecionados a uma nova tela e lá colocaremos o nome do repositório



> Nesse primeiro momento, vamos focar só no campo `Repository name *.` Conforme as aulas irão passando, entraremos mais afundo nessas outras opções

Vá até o fim da página e clique em **CREATE REPOSITORY**

Não se assustem com a interface que apareceu, como dito mais acima, iremos focar no básico a primeiro momento e conforme as aulas vão avançando, entraremos mais afundo com a estrutura.

Procure a opção `Quick setup - if you've done this kind of thing before` e pegue o link HTTPS

Vamos copiar esse link e voltar no terminal bash. Nele, digite o comando:

```
git remote add origin [link_do_repositorio_https]
```

substitua link_do_repositorio_https com o link gerado pelo github

para conferir, digite o comando:

```
git remote -v
```

E iremos nos deparar com o link do repositório remoto

Atualizando

Os seguintes comandos são responsáveis por alinhar os repositórios

```
git fetch origin
```

```
git pull origin main
```

Subindo

Agora, iremos enviar todos esses arquivos para o github através do comando

```
git push origin main
```

e assim, teremos algo semelhante a isso no terminal

Sucesso para todos, temos o nosso primeiro repositório criado, para conferir, atualize a página do github e visualize o seu projeto

🎉🎉🎉🎉🎉

Visão de comandos

Nesse arquivo, teremos uma visão geral dos comandos Git para versionamento. Abaixo, será encontrado o comando seguido de sua explicação

git init

É o comando que inicia um repositória (pasta) como um repositório Git

---

git clone

Copiamos todo o código de um repositório remoto (no Github) para o nosso repositório local (pasta no computador)

---

git status

Mostra os estados dos arquivos no repositório git. Nos é mostrado se um arquivo precisa ser adicionado ao stage, se foi adicionado ou se ja esta commitado

---

git add [arquivos]

Adiciona os arquivos para o próximo stage. Você só consegue commitar quando pelo menos 1 arquivo está adicionado

---

git commit -m "Sua mensagem aqui"

Faz um comentário nesse versionamento. Muito útil para a organização e o entendimento do que foi feito no seu trabalho

---

git branch

Vai listar suas ramificações existentes no seu repositório git

---

git branch [nome-da-branch]

Cria uma nova branch com um nome escolhido

---

git checkout

Faz a troca entre branchs.

---

git remote add [alias] [url]

Adiciona a URL do repositório remoto para a conexão com nosso repositório local

---

git fetch [alias]

Busca todas as branchs do git remote

---

git pull

Alinha as branchs, mantendo a atualização entre o remoto e o local

---

git push -U [alias] [branch]

Envia os códigos do nosso computador, para a branch especificada no Github