



Setup GitHub

- ☐ Criar sua conta no GitHub
- ☐ Instale o Git desktop
- ☐ Extensões GitGraph Git History no VS CODE
- ☐ Faça login em sua conta GitHub no navegador

Setup do ambiente – *git init*



1º configurar as credenciais do usuário que origina a publicação

- 1.1 - `git config --global --unset user.name`
- 1.2 - `git config --global --unset user.email`
- 1.3 - `git config --global user.name "fulano"`
- 1.4 - `git config --global user.email fulano@blabla.com`
- 1.5 - `git config --global --list`

2º `git add .` (pra validar o novo branche)

- 2.1 `git add .`

3º `git commit` (pra validar o novo branche)

- 3.1 `git commit -m "1º commit"`

4º Criar a branche padrão (xxxx-cedupflix)

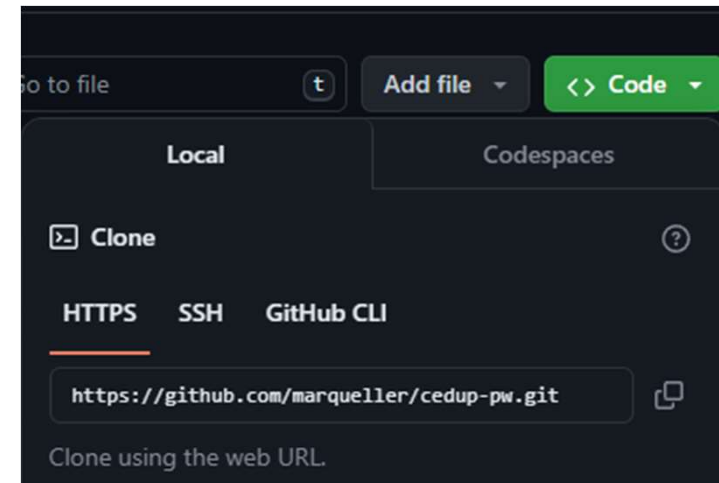
- 4.1 - `git branch -M xxxx-cedupflix`

5º Faça o add remoto (repositório do professor cadastrando seus usuários/alunos)

- 5.1 - `git remote add origin https://github.com/marqueller/cedup-pw.git`

6º `git push` (pra enviar o projeto “commitado” para o repositório na nuvem)

- 6.1 `git push -u origin xxxx-cedupflix`



Setup do ambiente – *Baixando arquivo da nuvem – git clone*



1º Após renomeada a pasta `c:/web` para `c:/web_bkp`, Recrie a pasta “zerada” no diretório `c:\web`

2º no VsCode execute o git nessa pasta zerada

2.1 `git init .`

3º execute o comando de clone

3.1 `git clone -b 8007-cedupflix --single-branch <<repositório>>`

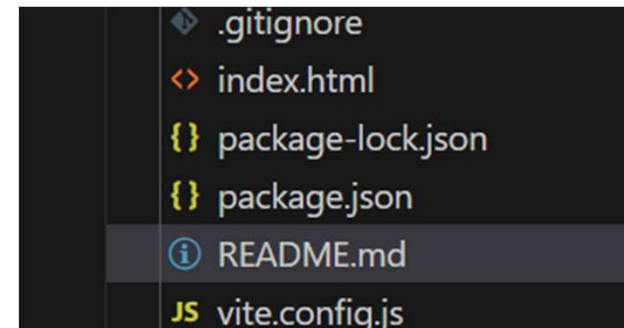
4º Criar a branche padrão (xxxx-cedupflix)

4.1 `git branch -M xxxx-cedupflix`

5º Para entrar no diretório padrão

5.1 `CD xxxx-cedupflix`

6º Renomear pasta `c:/cedup-pw` para `c:/xxxx-cedupflix`



7 º Faça as alterações conforme aula;

7.1 Use o arquivo README.md para “colar os RF’s (Requisitos Funcionais) e RNF’s (Requisitos não Funcionais) do projeto

8º Faça o add remoto após alterações

5.1 - `git remote add origin https://github.com/marqueller/cedup-pw.git`

9º git push (pra enviar o projeto “commitado” para o repositório na nuvem)

6.1 `git push -u origin xxxx-cedupflix`

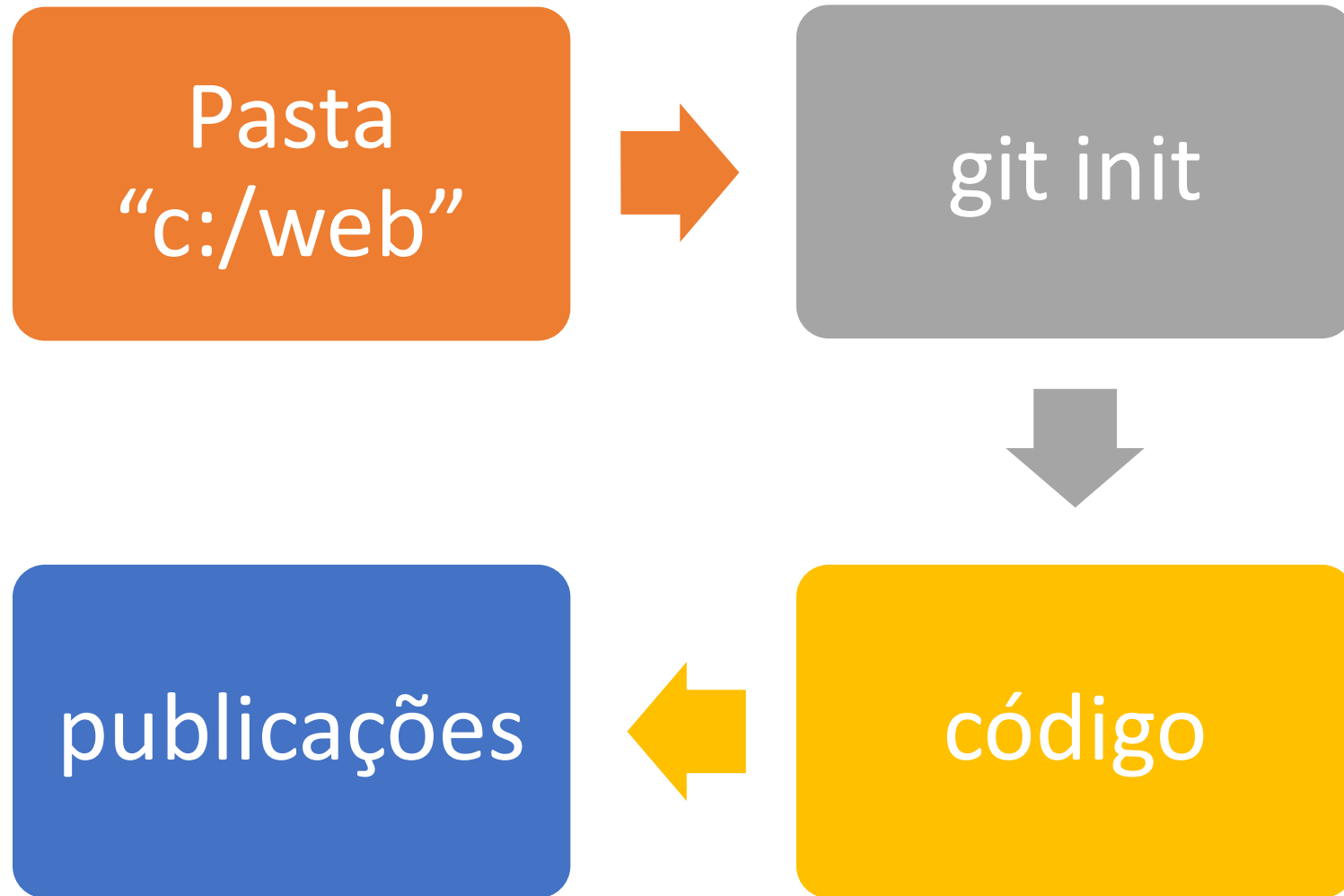
Padronização do sistema de arquivos



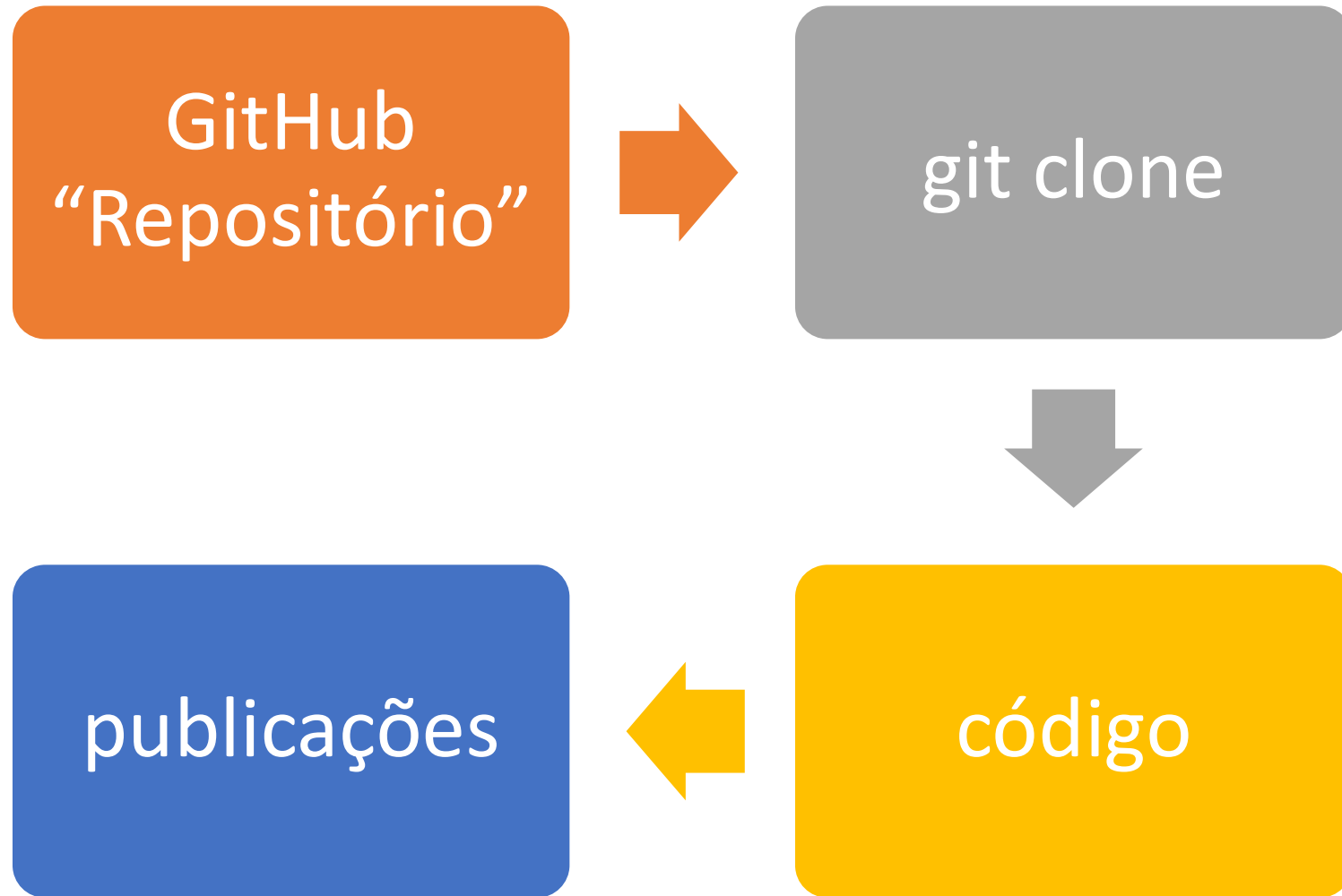
xxxx-cedupflix



Fluxo do comando “*git init*”



Fluxo do comando “*git clone*”



Fluxo do comando “*git commit*”



git commit

- ☐ git log – para ver todos os commits se há algum
- ☐ git diff – para ver as alterações
- ☐ git status - para ver alterações que faltam ser commitadas
- ☐ git add – para adicionais as alterações no commit
- ☐ git status - para rever

Fluxo do comando “*git commit*”



Branche (galho)???

- * Cria um padrão de versões para testes
- * Preserva a versão mãe (testada e funcional)



Fluxo do comando “*git commit*”



Branche (galho)???

- *git branch -M main* – comando para renomear a branch de acordo com a necessidade
- *git remote add origin <https://github.com/marqueller/001-cedup-be>* - comando para “informar” ao git local que estará linkado ao git remote.
- *git remote get-url origin* – comando para informar qual o endereço do git remote (GitHub) está linkado.

Fluxo do comando “*git commit*”



mensagens

Título (Objetivo)

Corpo (explicação
detalhada)

Diferença entre os commits

Razão da mudança

Informações extras.. cuidados, etc

`git commit -m “mensagem”`

Detalhe da configuração das credenciais



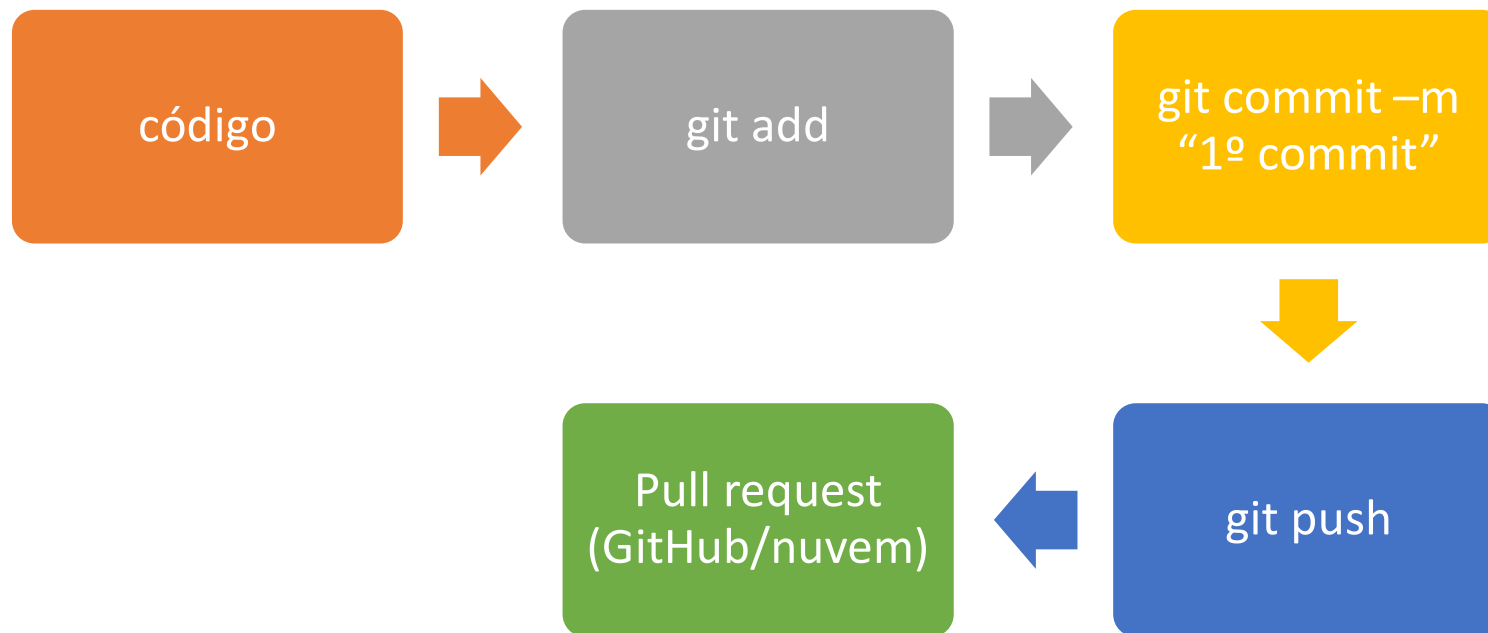
```
git config --global user.name "Fulano"
```

```
git config --global user.email "fulano@blabla.com"
```

```
git config --list
```

```
git remote get-url origin
```

Fluxo do comando “*git commit*”



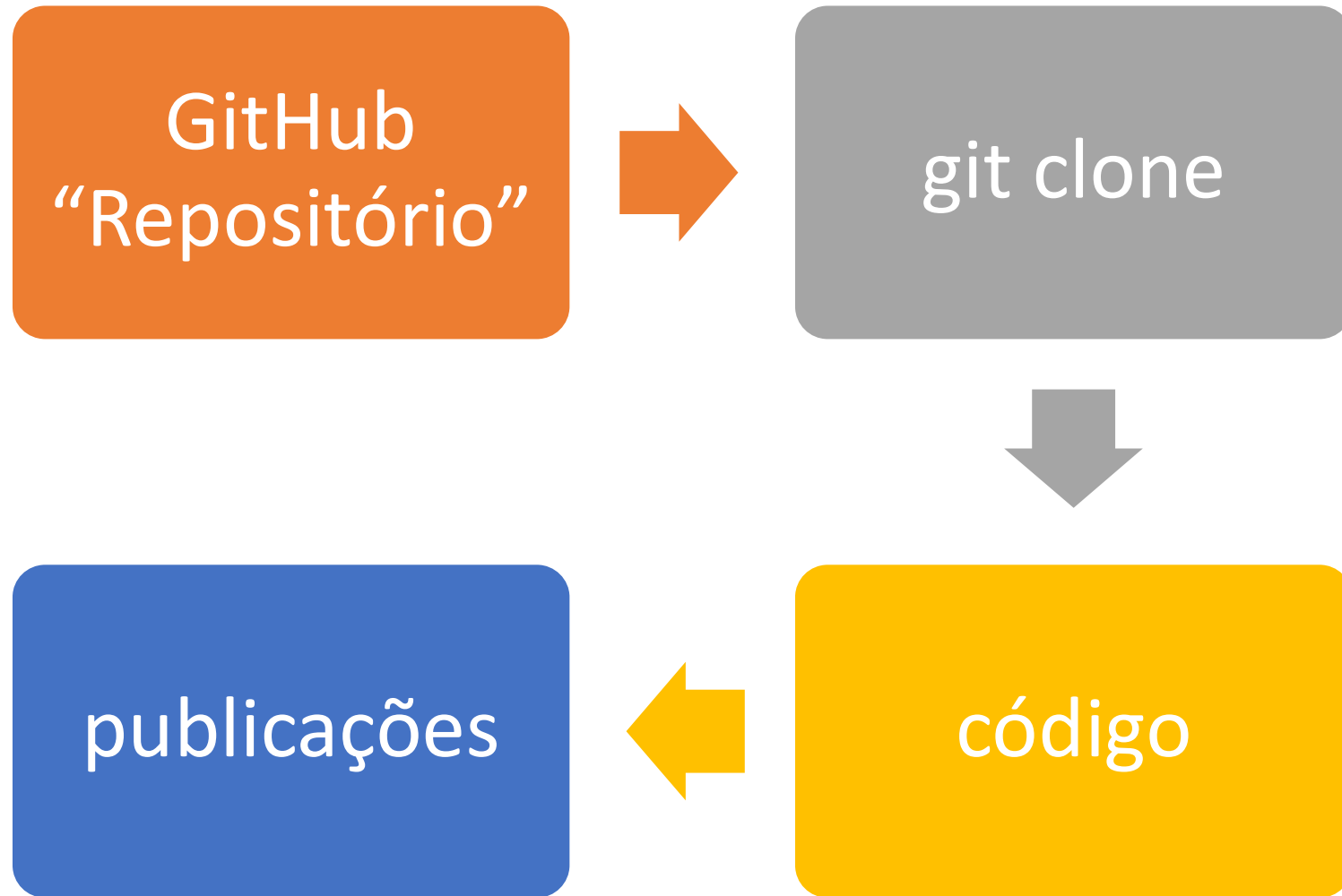


git push (publicação/envio)???

- *git push -u origin main* – comando executado na PRIMEIRA VEZ que é publicado/enviado projeto ou arquivo ao repositório remoto.
- *Sendo a primeira vez também se faz necessário fazer login**.*
- *git push* – para outros “pushs”.

**login talvez se faça necessário alterar as configurações de contas no Windows.

Fluxo do comando “*git clone*”



Fluxo do comando “*git checkout*”

