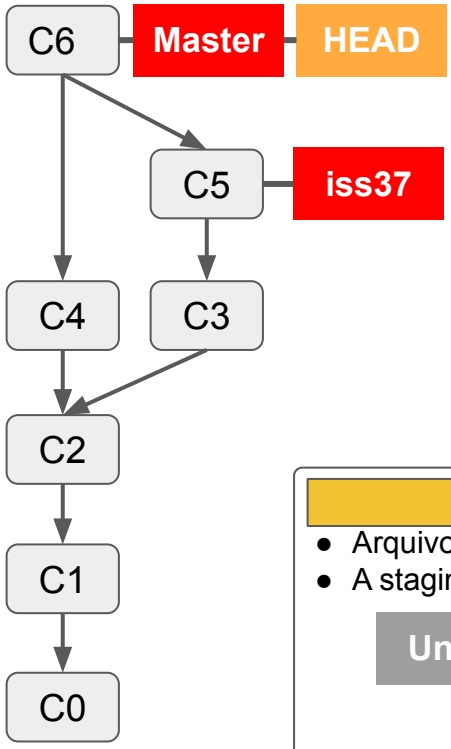
Git CheatSheet

Referências: https://git-scm.com/book/en/v2; https://www.atlassian.com/git/tutorials; https://training.github.com/downloads/github-git-cheat-sheet.pdf

Princípios básicos sobre Git

- Git é um **software** para controle de versão de código ("versionamento").
- Ele acompanha mudanças do código como uma **árvore de estados**, a qual você pode controlar.
- Ramificações se chamam **branches**.
- Estados são **commits**, e têm "**hashes**" como ID (códigos alfa-numéricos).
- Cada branch aponta para um commit, enquanto o ponteiro especial "**HEAD**" aponta para a branch em uso.
- Este controle está salvo em uma pasta "**.git**", dentro do repositório. Para ser repositório, essa pasta tem que existir.
- Em geral, nós sincronizamos o repositório local (no nosso PC), com um **repositório remoto**, que fica no **Github**, **Bitbucket**, **Gitlab**, etc...



Comandos do git (repositórios)

git init	transforma a pasta atual em um repositório git. Se passar argumento [dir] pra ele, transforma a pasta [dir] em repositório.
git clone [url]	clona e segue o repositório git que está no link [url]
git status	mostra como está o repositório, na branch atual.
git config	configura o git no atual repositório, a flag --global configura o git para todos os seus repositórios.
git log	mostra histórico de versão. git log --oneline mostra o histórico com comentários resumidos.

"Branching"

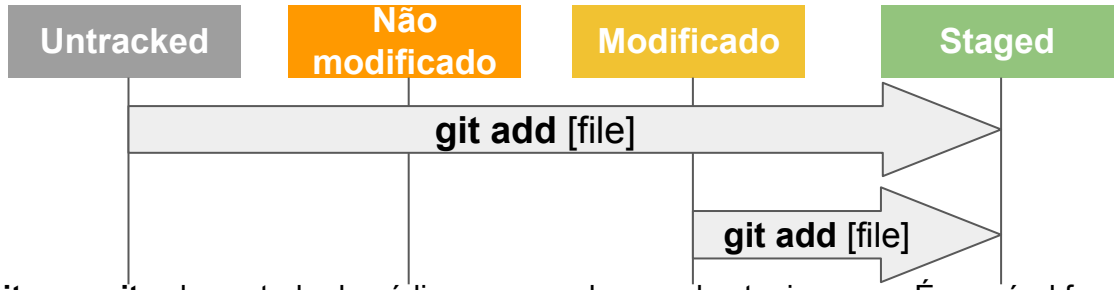
git branch	lista branches no repositório local.
git branch [branch]	cria branch [branch]. Se passar --track [new] [remote], ela segue [remote]
git checkout [branch]	muda HEAD para [branch]
git merge [branch]	junta [branch] à atual, gerando um commit do merge.

Comunicando com remoto

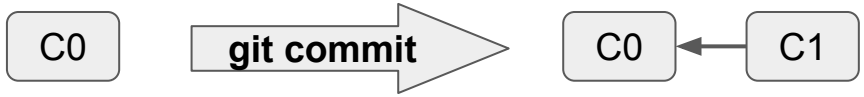
git push	leva mudanças locais para remoto
git pull	traz mudanças remotas para local
git push [remote] [branch]	envia mudanças locais de [branch] para o repo remoto na branch [remote].
git pull [branch]	traz mudanças apenas de [branch]
git fetch	baixa histórico do repo remoto.

Fluxo de arquivos no Git

- Arquivos estão tracked (git acompanha mudanças) ou untracked (git não acompanha).
- A staging area é uma área onde colocamos as mudanças a serem salvas em commit.



- **git commit** salva estado do código com mudanças da staging area. É possível fazer comentários descrevendo o commit. Se passar a flag -m, o texto a seguir será a descrição.



git diff [file]	mostra as modificações de [file] (que esteja em "modificado").
git checkout [file]	desfaz mudanças feitas em [file] (que esteja em "modificado").
git show [commit]	mostra as mudanças em [commit]
git reset	remove tudo da staging area, mas mantém em "modificado" (caso tracked)
git reset [commit]	desfaz todos os commits após [commit]. Pode ser usado com [file].
git log --follow [file]	mostra histórico de mudanças de [file] (caso tracked).

Obs: o arquivo .gitignore define arquivos que devem ser totalmente ignorados pelo git, sempre.