

Fundamentos de DevOps

<http://linkedin.com/in/guijac>

Aula 03 - Git e Incorporação de Modificações

Prof. Esp. Guilherme Jorge Aragão da Cruz

 guilherme.cruz@alumni.usp.br

 linkedin.com/in/guijac

Roteiro

- Revisitando *Commits*;
- Outros Comandos Git:
 - git checkout;
 - git revert;
 - git reset;
 - git merge;
 - git pull;
 - Juntando as Peças.
- Referências Bibliográficas.

Revisitando *Commits*

- *Commits* são “**save states**” de um determinado ponto de modificação em seu projeto;
- São identificados através de um checksum SHA-1, que representa as mudanças efetivamente realizadas em um commit.

<http://linkedin.com/in/guijac>

a7327d4e3196eb874032
d187c253192376c887eb

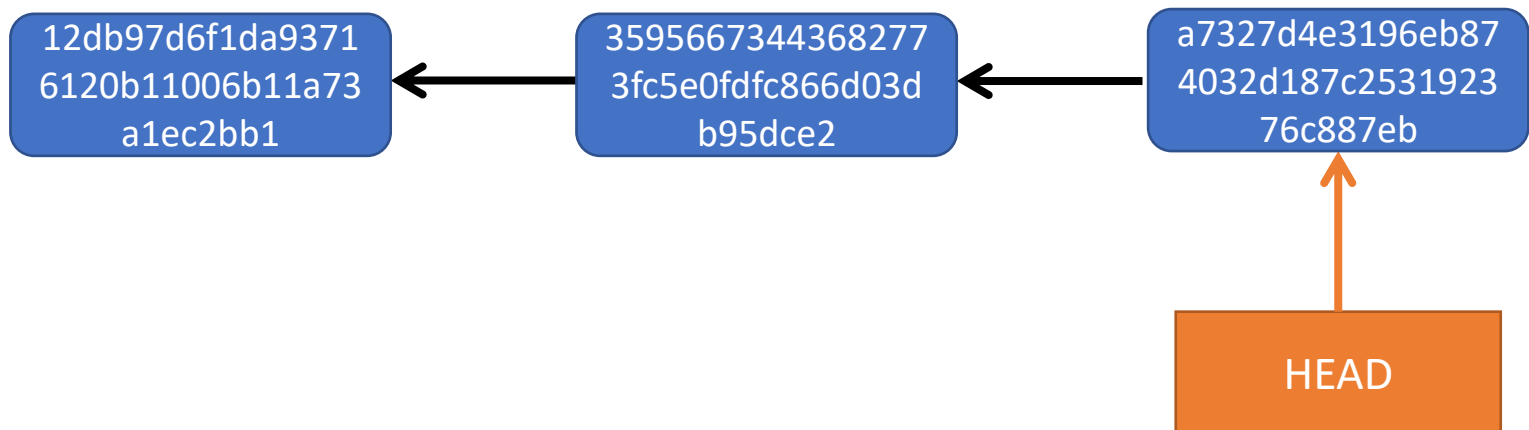


Fonte: [1597: Git - explain xkcd](#)

Revisitando *Commits*

- *Commits* são ligados da mesma forma que uma **estrutura de dados do tipo lista**, apontando sempre ao commit anterior pelo qual ele se originou;
- Por padrão, uma referência/ponteiro chamada **“HEAD”** aponta para o **último commit** realizado.

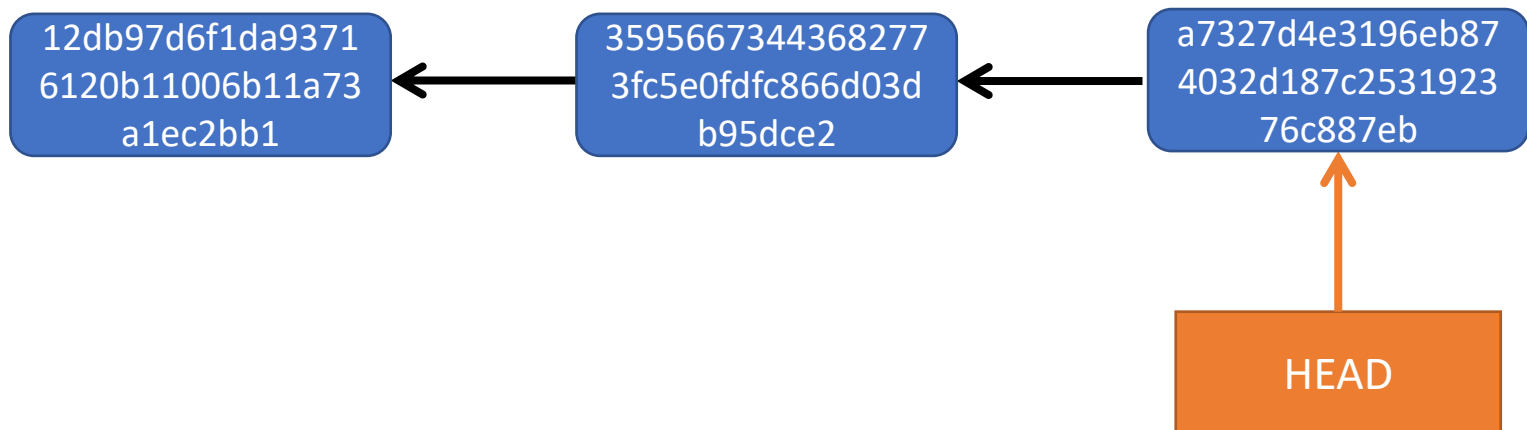
<http://linkedin.com/in/guijac>



Outros Comandos Git: checkout

- Através do comando “git checkout” conseguimos movimentar o ponteiro “HEAD” para outros commits.

<http://linkedin.com/in/guijac>

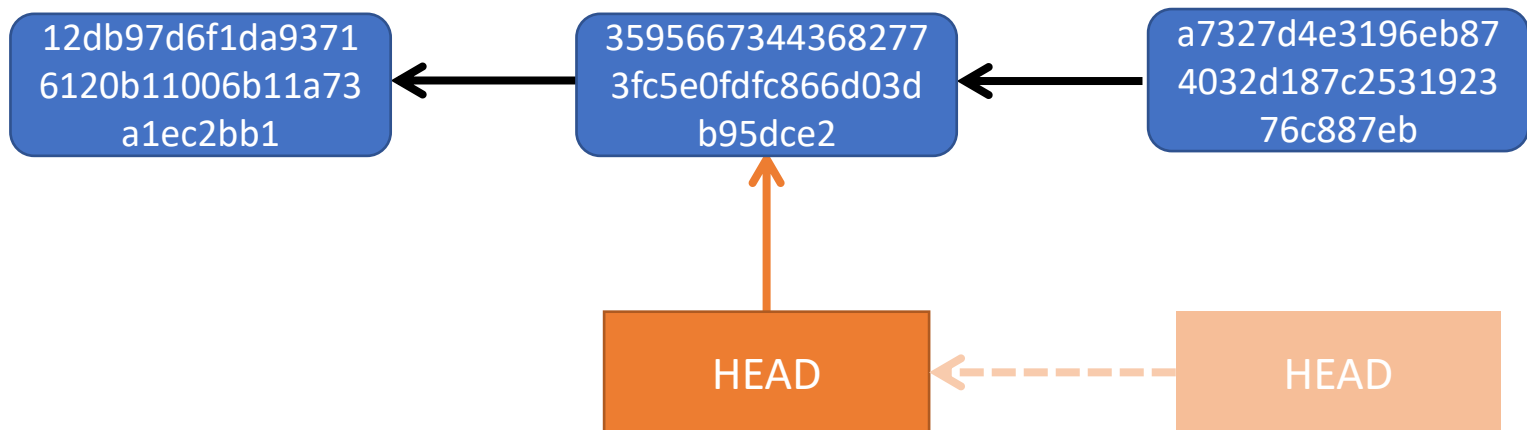


```
git checkout 35956673
```

Outros Comandos Git: checkout

- Através do comando “git checkout” conseguimos movimentar o ponteiro “HEAD” para outros commits.

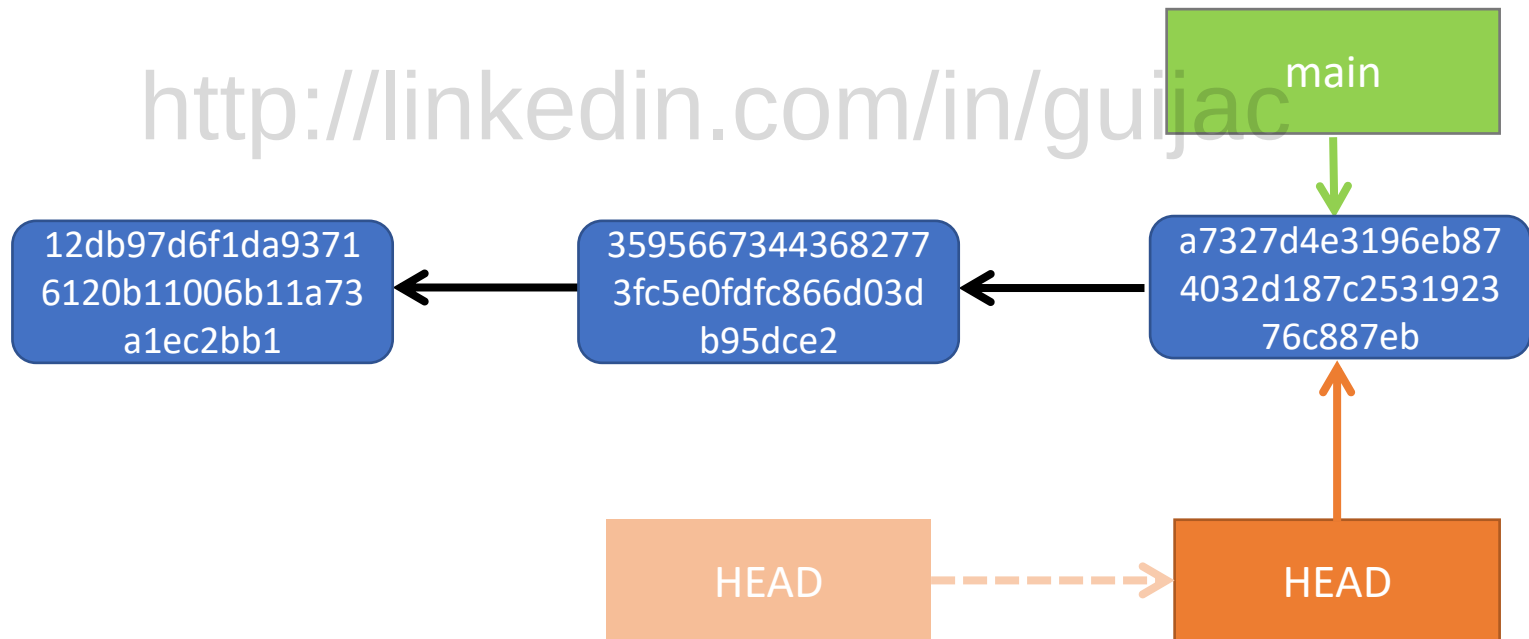
<http://linkedin.com/in/guijac>



```
git checkout 35956673
```

Outros Comandos Git: checkout

- O comando também possibilita voltar para a ponta da lista (*branch* principal “main”) ou qualquer outra *branch* pertencente ao projeto, além de **criar** poder uma nova ***branch*** (com o parâmetro -b).

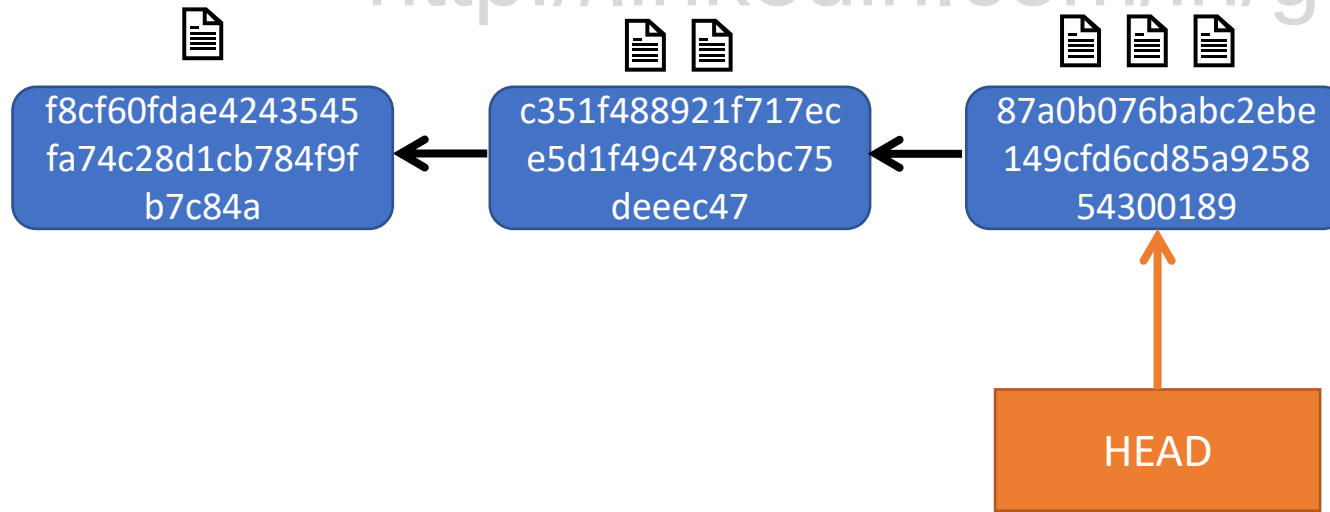


```
git checkout main
```

Outros Comandos Git: revert

- Comando do tipo “desfazer”; entretanto não há uma **exclusão** do estado especificado;
- Será gerado um novo *commit*, desfazendo as modificações posteriores à ele, mas preservando o histórico de *commits*.

<http://linkedin.com/in/guijac>

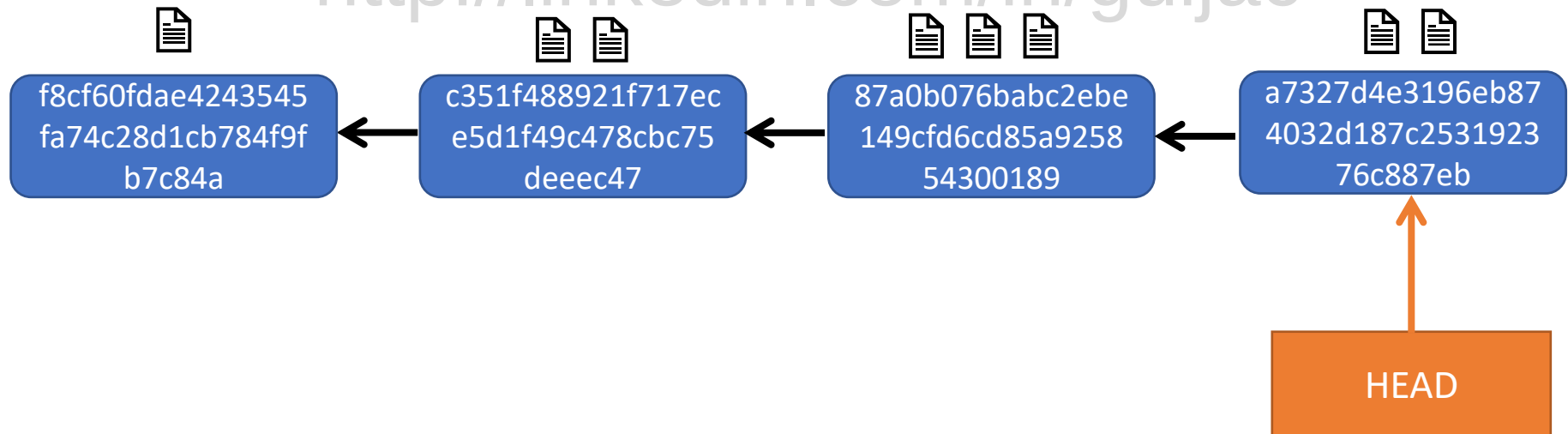


```
git revert 87a0b076
```


Outros Comandos Git: revert

- Comando do tipo “desfazer”; entretanto não há uma **exclusão** do estado especificado;
- Será gerado um novo *commit*, desfazendo as modificações posteriores à ele, mas preservando o histórico de *commits*.

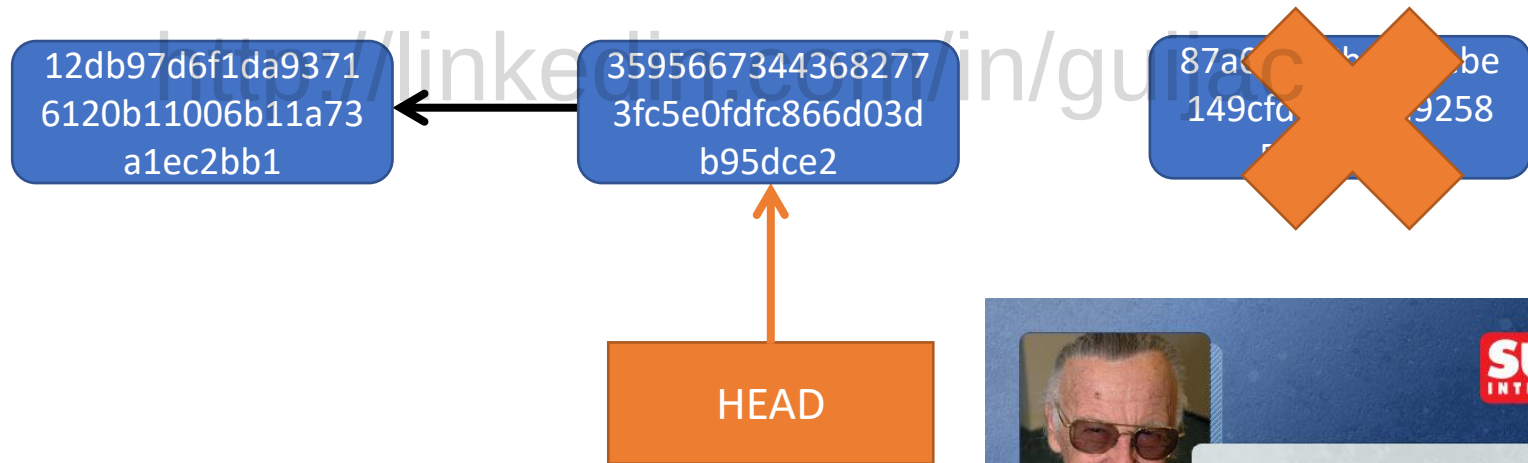
<http://linkedin.com/in/guijac>



```
git revert 87a0b076
```

Outros Comandos Git: reset

- Este comando modifica os ponteiros da branch atual e da HEAD **juntas**, tornando um commit inacessível;
- Uso não recomendado e não deve ser utilizado em repositórios compartilhados.

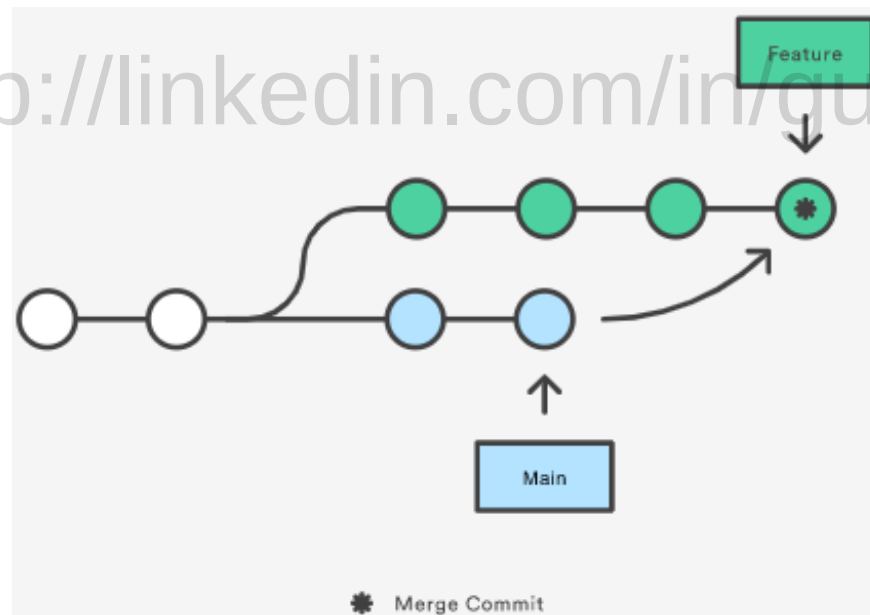


```
git reset 87a0b076
```



Outros Comandos Git: merge

- Cria um novo estado com base nos recursos existentes em outra *branch*, unindo os históricos;
- Modificações constantes na *main* são tratadas com o chamado “*merge back*”.

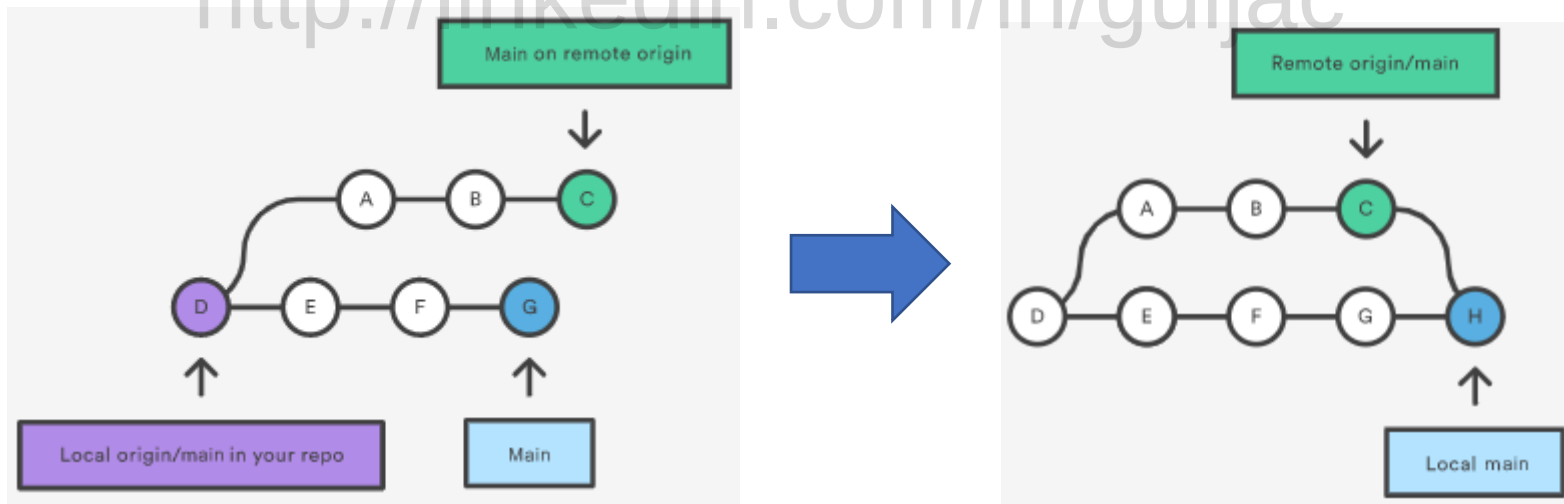


Fonte: Atlassian (2025)

```
git merge main
```

Outros Comandos Git: pull

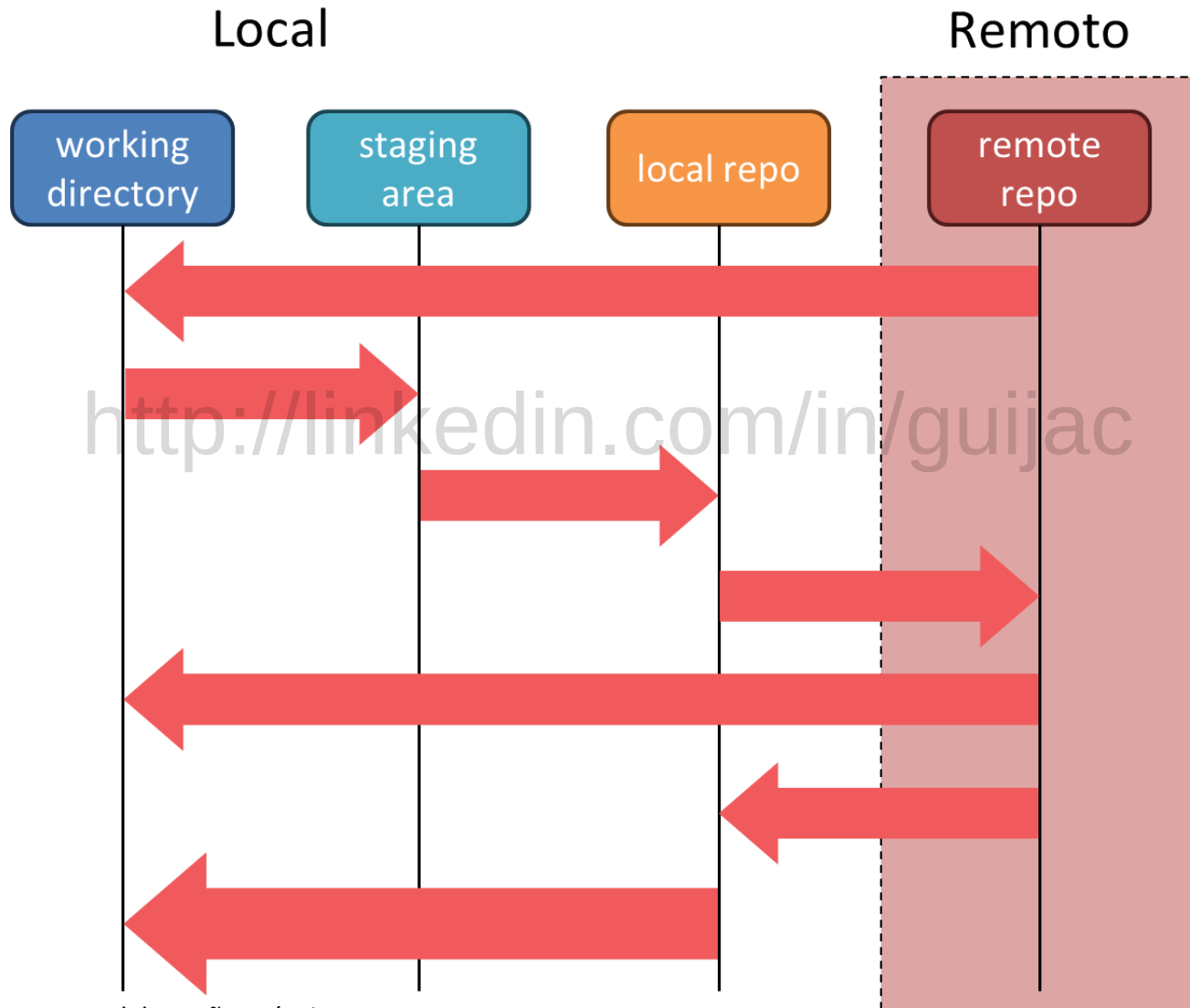
- Realiza um **sincronismo** do seu repositório **local** com o repositório **remoto**;
- É executado o comando **git fetch** (baixar o conteúdo do repositório remoto) em seguida do git merge.
- “H” contém “A, B e C”, sendo um novo *commit*.



Fonte: Atlassian (2025)

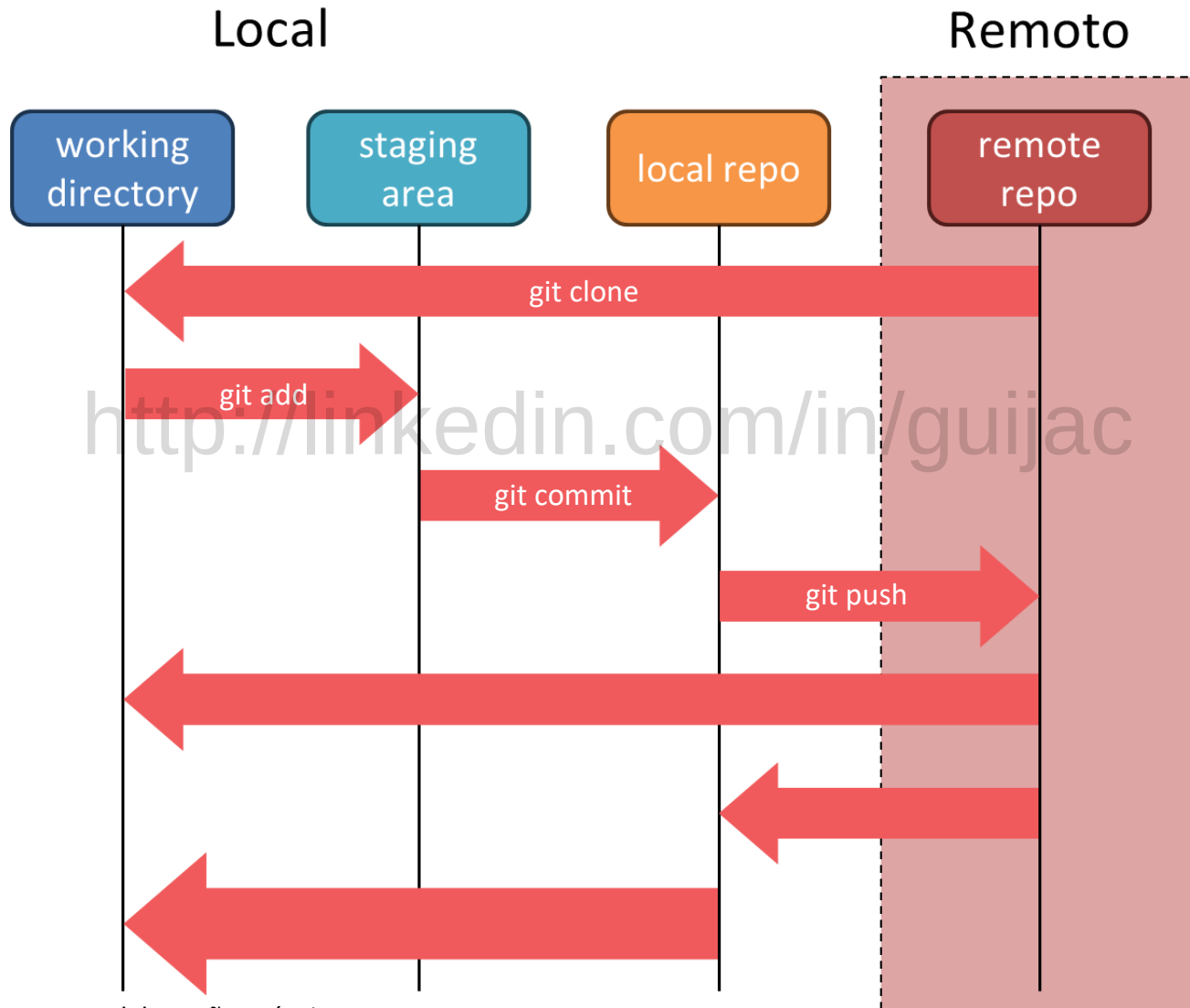
```
git pull
```

Juntando as Peças



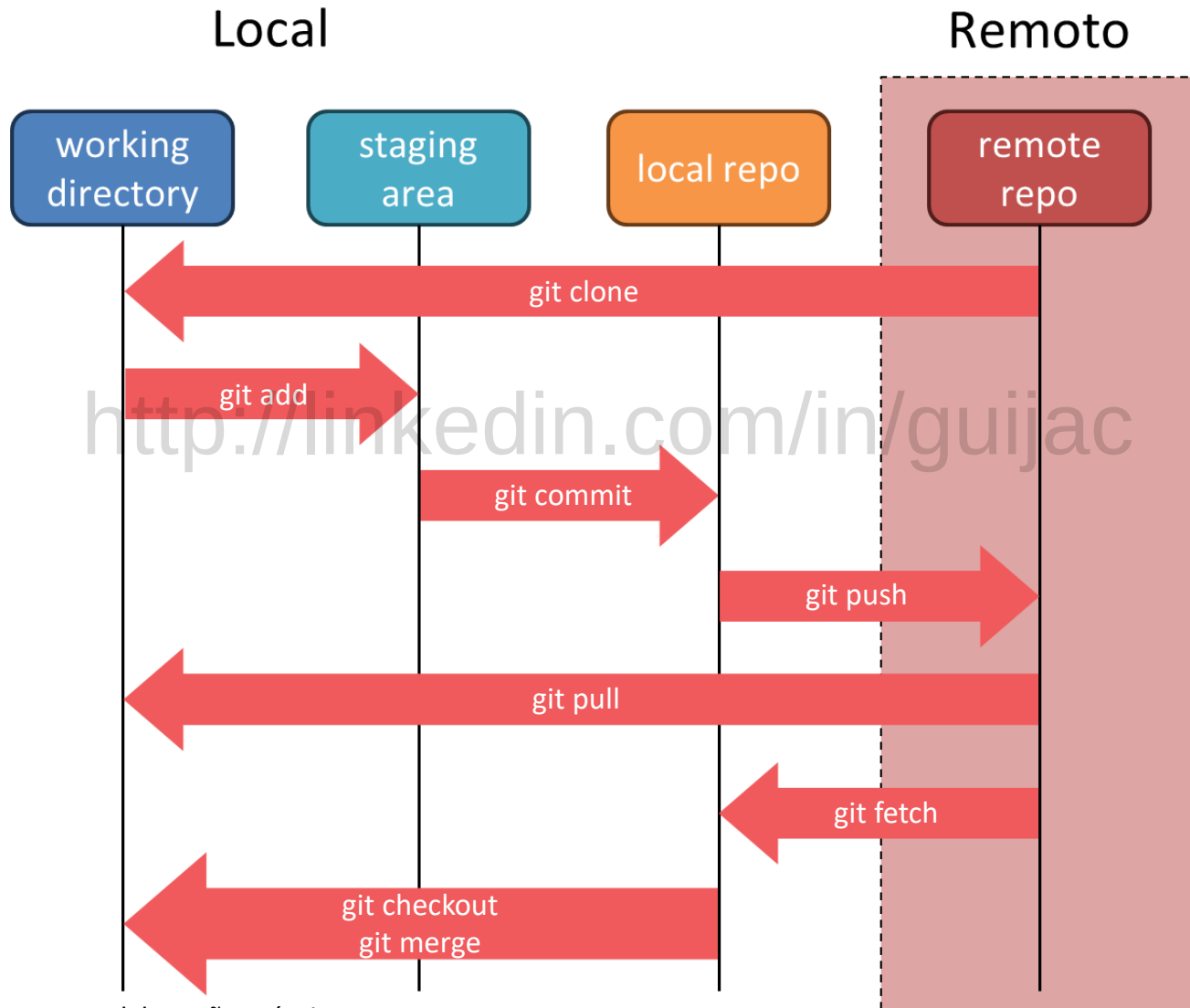
Fonte: Elaboração Própria

Juntando as Peças



Fonte: Elaboração Própria

Juntando as Peças



Fonte: Elaboração Própria

Referências Bibliográficas

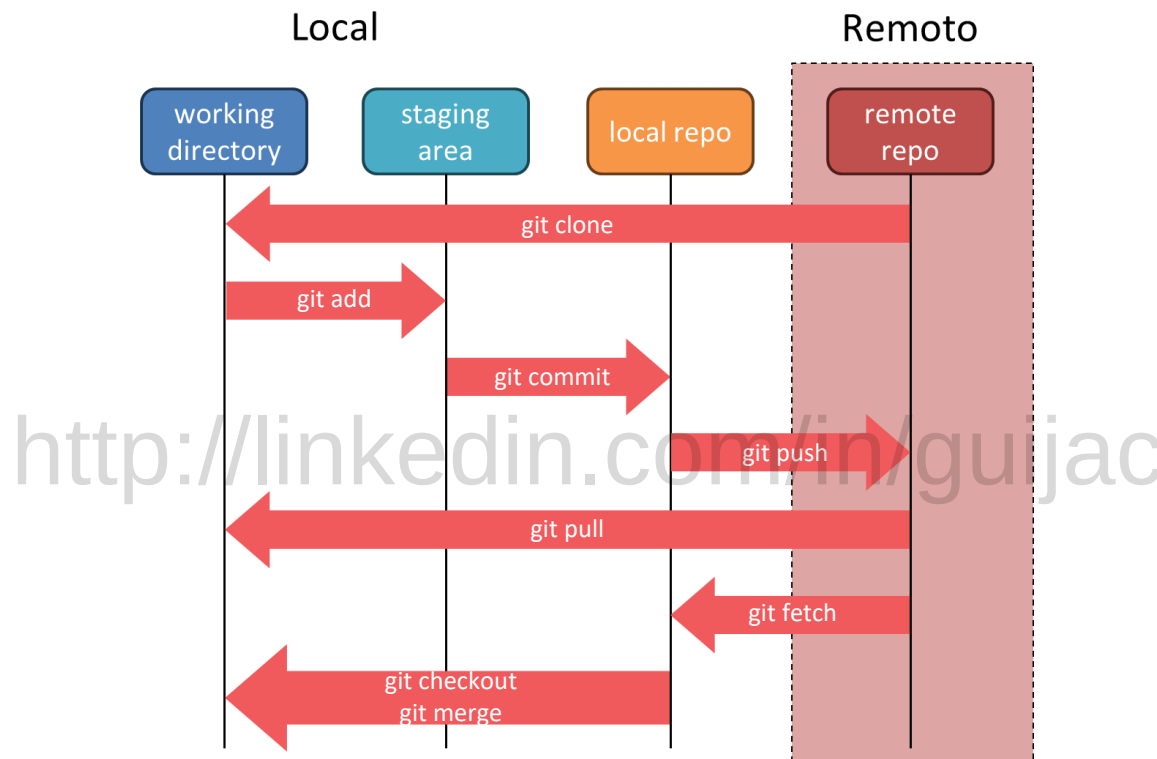
GIT. Documentation. Disponível em <https://git-scm.com/doc>. Acesso em 18 jan 2025;

GITHUB. Git Guides. Disponível em <https://github.com/git-guides>. Acesso em 18 jan 2025;

GOMES, Daniane. Fiz m*rda no GIT, e agora?. <https://danianepg.medium.com/fiz-m-rda-no-git-e-agora-f6575750e5f8>. Acesso em 18 jan 2025;

FERREIRA, Renato Cordeiro. MAC0475 2021 - Aula07.1 - Git Avançado. Disponível em <https://uclab.xyz/sistemas-complexos-2021-aula07-1>. Acesso em 18 jan 2025.

Por hoje (de teoria!) é só!



Fonte: Elaboração Própria (2025)

Prof. Esp. Guilherme Jorge Aragão da Cruz

✉ guilherme.cruz@alumni.usp.br

in linkedin.com/in/guijac

Licença

- Este conteúdo está licenciado sob a Licença Creative Commons Atribuição-NãoComercial-Compartilhalgual 4.0 Internacional (CC BY-NC-SA 4.0).
- Todos os direitos autorais sobre este conteúdo pertencem ao autor, e este material não pode ser usado comercialmente sem autorização expressa.
- Material elaborado no contexto da disciplina **Fundamentos de DevOps** do **Centro Universitário Senac**, para fins educacionais.
- As referências a marcas, produtos e tecnologias têm caráter exclusivamente educacional, não havendo vínculo institucional ou comercial com as organizações citadas.
- Para ver o texto completo da licença, acesse <https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>.

Prof. Esp. Guilherme Jorge Aragão da Cruz

 guilherme.cruz@alumni.usp.br

 linkedin.com/in/guijac