

# Preparação do Ambiente

---

Advanced Institute for Artificial Intelligence – AI2

<https://advancedinstitute.ai>

- ☐ Github References
- ☐ Atlassian Tutorials
- ☐ Git Book
- ☐ Na linha de comando: `git help comando`



# Sistemas de Controle de Versão

---



Figure: The time Machine Movie (1960)

## Motivação

- ❑ Gerência de alterações no código-fonte **com o passar do tempo**;
- ❑ Mantém registro de todas as modificações no código;
  - Utiliza alguma forma especial de Banco de dados
- ❑ Uma máquina do tempo:
  - Comparar versões anteriores
- ❑ Rastreamento de cada alteração individual feita por cada contribuinte, **evitando conflito com trabalhos simultâneos**;

## Importante

As alterações feitas em uma parte do software podem ser **incompatíveis** com aquelas feitas por **outro desenvolvedor trabalhando ao mesmo tempo**. Esse problema deve ser descoberto e resolvido **sem bloquear o trabalho do restante da equipe**.

## Motivação

- ☐ Compatível com o fluxo de trabalho dos desenvolvedores;
- ☐ Não fazer imposições sobre o modo de trabalho;
- ☐ Independência de plataforma;
- ☐ Parte essencial do dia a dia de equipe distribuídas de software;
- ☐ Rastreabilidade: capacidade de identificar quando um determinado bug foi inserido no código;
- ☐ Controle de Ramificações;



Git

---

- ☐ Conceitos
- ☐ Comandos
- ☐ Github
- ☐ Fluxos para Gerenciamento de Código



## O que é o Git?

- ☐ É um sistema de controle de versão de arquivos que rastreia o histórico de mudanças;
- ☐ Sistema de controle de versão moderno mais usado no mundo hoje;
- ☐ Desenvolvido em 2005 por Linus Torvalds;
- ☐ Arquitetura distribuída;
  - Em contraste com sistemas centralizados como *CVS* e *Subversion*
  - Toda cópia de trabalho é um repositório;
- ☐ Se concentra no conteúdo do arquivo em vez de nomes;
- ☐ Projetado com desempenho, segurança e flexibilidade em mente

## Características elementares do Git

- ☐ Provê um controle de versionamento completamente distribuído.
- ☐ Não precisa de uma conexão constante com um repositório central.
  - Desenvolvedores podem trabalhar em qualquer lugar e colaborar de **forma assíncrona**;
- ☐ Permite gerenciar as alterações localmente, antes de enviar para uma instância remota;

**ONE DOES NOT SIMPLY**

**UNDERSTAND GIT**

MemesHappen

## Setup Necessário - Chaves SSH

- ☐ Secure Shell Protocol
- ☐ Transferência remota de arquivos, gerenciamento de rede e acesso remoto ao sistema operacional.
- ☐ Usa um par de chaves para iniciar um handshake seguro entre partes remotas;
- ☐ Criptografia Assimétrica;

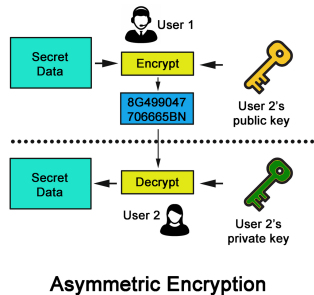


Figure: Criptografia Assimétrica

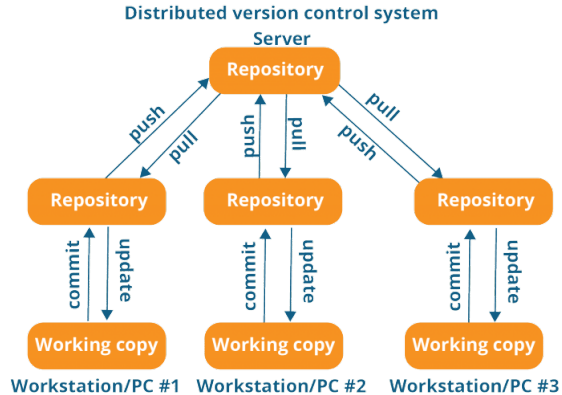


Figure: Repositórios Distribuídos

## O que é um repositório Git?

- ☐ Armazenamento virtual para projetos;
- ☐ Diretório especial `.git`;
  - contém todos os metadados Git necessários para o novo repositório;
  - subdiretórios para objetos, referências;
  - Arquivo **HEAD** que aponta para o commit em uso no momento;

## Inicializando um Repositório Git

- Definindo nome de usuário e email globalmente:

```
1 > git config --global user.name <name>  
2 > git config --global user.email <email>
```

## Inicializando um Repositório Git

### □ `git init`

- Converter um projeto existente e não versionado em um repositório do Git ou
- Inicializar um novo repositório vazio;
- cria um subdiretório `.git`

### □ `git clone`

- Maneira mais comum para os usuários obterem uma cópia de desenvolvimento;
- Em geral uma operação única;
- O repositório original pode estar localizado no sistema de arquivos local ou em protocolos com suporte a acesso por máquinas remotas;
  - Por exemplo, o protocolo SSH.

```
1 > git clone ssh://rmcobe@example.com:/repos/meurepo.git
2 Cloning into 'meurepo'...
```



## Salvando Alterações

- ☐ O conceito de "salvar" é um processo com mais nuances do que salvar em um processador de texto;
- ☐ "salvar" é sinônimo do termo Git "*commit*"
- ☐ Os comandos: `git add`, `git status` e `git commit` são usados em combinação para salvar uma fotografia do estado atual de um projeto Git

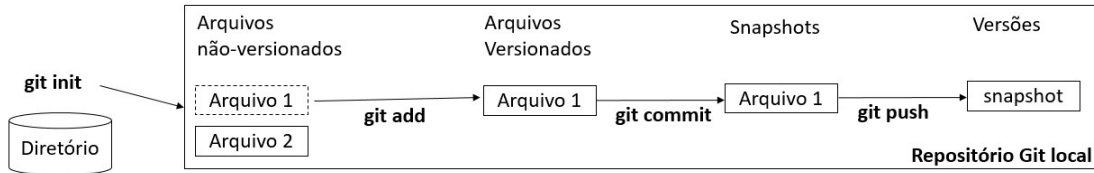
## Salvando Alterações

- ❑ O Comando `git add`:
  - Adiciona uma mudança no diretório de trabalho à área de teste
  - Informa ao Git que você deseja incluir atualizações para um arquivo específico no próximo *commit*;
  - Não afeta o repositório até que seja realizado o `git commit`
- ❑ Utilizado em conjunto com o comando `git status`;
- ❑ Um terceiro comando `git push` é essencial para um fluxo de trabalho Git colaborativo
  - Utilizado para enviar as alterações confirmadas para repositórios remotos para colaboração;
- ❑ `git pull` atualiza o repositório local com atualizações do repositório remoto;

## `git status`

Retorna:

- Lista de arquivos não-versionados
- Alterações realizados nos arquivos versionados



**Replica um repositório remoto**



**Atualiza repositório local a partir de um repositório remoto**



## Visualizando Diferenças

- ❑ Utilizando o comando `git diff`:
  - Compara as mudanças específicas no diretório de trabalho com o índice de objetos (HEAD)
  - Pode ser especificado apenas um arquivo;
  - Mostrando as mudanças que ainda não foram *staged*;
  - Opção `--cached`, o *diff* irá comparar as mudanças *staged* com o repositório local;
- ❑ Combinado com o comando `git log` para mostrar a diferença entre dois *commits* específicos.

```
1 > git log --oneline
2 611dc29 (HEAD -> master, origin/master, origin/HEAD) Added Section 2
3 85cadb5 Added Section 1
4 42b045b Added README.md
5 > git diff 42b0 611c
```

## Desfazendo Mudanças

- ❑ Para desfazer mudanças, podemos utilizar o comando `git checkout` como uma forma de navegar na história de *commits*

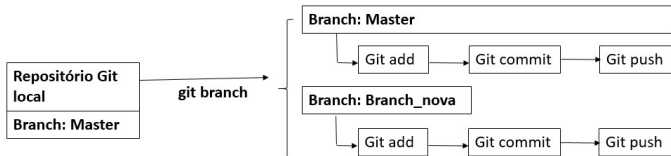
```
1 > git log --oneline
2 611dc29 (HEAD -> master, origin/master, origin/HEAD) Added Section 2
3 85cadb5 Added Section 1
4 42b045b Added README.md
5 > git checkout 42b045b README.md
6 > git commit "Revert to previous README.md"
```

- ❑ Usar o comando `git revert` quando mudanças já foram publicadas

## Branches

- ❑ `git branch` mostra a *branch* local atual.
- ❑ `git branch (nome)` cria uma nova *branch*.
- ❑ `git checkout (nome-branch)` troca para outra *branch*;
- ❑ `git merge` é usado para combinar alterações feitas em duas *branches* distintas;

## Criando uma ramificação do repositório



## Trocando contexto entre ramificações



## Combinando duas ramificações em uma existente





## Cuidando de conflitos

- ❑ Conflitos são situações em que o Git não consegue atualizar um arquivo com uma nova versão da *branch* automaticamente
- ❑ Nesses casos o Git avisa essa situação e pede para que o desenvolvedor resolva os conflitos

```
1  CONFLICT (content): Merge conflict in README.md
2  Automatic merge failed; fix conflicts and then commit the result.
3  > cat README.md
4  # README.md
5
6  <<<<<< HEAD
7  ## Section 1
8  =====
9  ## Section 3
10 >>>>>> conflict
```

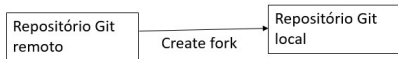


# Github

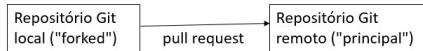
---

<https://github.com>

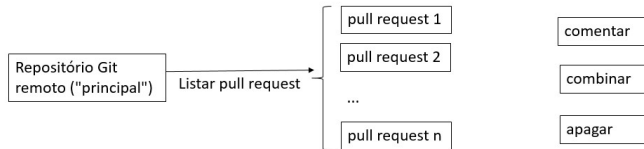
**Fork:** Criar um novo repositório a partir de um existente




**Pull request:** Enviar um pedido de alteração de um repositório replicado para o repositório original



**Aceitar, comentar ou apagar os pedidos de alteração enviados**



 [silviostanzani / teste](#)

Unwatch ▾ 1

★ Star 0

🍴 Fork 1

<> Code

🔔 Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

🛡 Security

📊 Insights

⚙ Settings

**Label issues and pull requests for new contributors**

Dismiss

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [help wanted](#) or [good first issue](#)

Filters ▾

🔍 is:issue is:open

🏷 Labels 9

🎯 Milestones 0


New issue

!

**Welcome to Issues!**

Issues are used to track todos, bugs, feature requests, and more. As issues are created, they'll appear here in a searchable and filterable list. To get started, you should [create an issue](#).

# Github - Issue



monstrando duas mensagens


Related Issues **Beta** [Try it.](#)


Write

Preview


AA B i “ < > ☰ ☷ ☹ @ 📎 ↶

O script atual mostra uma mensagem, é necessário alterar para mostrar duas mensagens

Attach files by dragging & dropping, selecting or pasting them. 

 Styling with Markdown is supported

Submit new issue

 [silviostanzani / teste](#)

Unwatch 1

Star 0

Fork 1

< > Code

! Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

## monstrando duas mensagens #1

[Edit](#)[New issue](#)

**silviostanzani** opened this issue 2 minutes ago · 0 comments



**silviostanzani** commented 2 minutes ago

Owner



O script atual mostra uma mensagem, é necessário alterar para mostrar duas mensagens



Write

Preview



Leave a comment

Attach files by dragging & dropping, selecting or pasting them.



Close issue

Comment

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet

Milestone



No milestone

Notifications

Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

1 participant

# Github - Fork

silviostanzani / teste

Unwatch 1

Star 0

Fork 1

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

No description, website, or topics provided.

Edit

Manage topics

2 commits

1 branch

0 releases

1 contributor

Branch: master


New pull request

Create new file

Upload files


Find File

Clone or download

 silviostanzani


Create pythonScript.py

Latest commit eb45ca5 2 minutes ago

 README.md


Initial commit

3 minutes ago

 pythonScript.py


Create pythonScript.py




2 minutes ago








 README.md


teste

# Github - Fork





 **TheLone-Gunmen / teste**  
forked from silviostanzani/teste






 Watch 0  Star 0  Fork 1



 Code  Pull requests 0  Projects 0  Wiki  Security  Insights  Settings




*No description, website, or topics provided.*  Edit



[Manage topics](#)

 2 commits  1 branch  0 releases  1 contributor

Branch: **master**     

This branch is even with silviostanzani:master.  Pull request  Compare

 <b>silviostanzani</b> Create pythonScript.py	Latest commit eb45ca5 3 minutes ago
 <a href="#">README.md</a>	Initial commit 4 minutes ago
 <a href="#">pythonScript.py</a>	Create pythonScript.py 3 minutes ago

 **README.md** 

# teste



## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).



base repository: **silviostanzani/teste** ▾

base: **master** ▾



head repository: **TheLone-Gunmen/teste** ▾

compare: **master** ▾

✓ **Able to merge.** These branches can be automatically merged.



**Create pull request**

Discuss and review the changes in this comparison with others.



🔗 **1** commit

📄 **1** file changed

💬 **0** commit comments

👤 **1** contributor



Commits on Oct 02, 2019



**silviostanzani**

Update pythonScript.py

Verified

d4943e3



Showing **1** changed file with **1** addition and **0** deletions.

Unified

Split

▼ **1** pythonScript.py 📄

...    ...    @@ -1 +1,2 @@

1    1    print("teste")

# Github - Pull Request

## Update pythonScript.py #2

[Edit](#)

 **Open** silviostanzani wants to merge 1 commit into `silviostanzani:master` from `TheLone-Gunmen:master` 

 Conversation **0**

 Commits **1**

 Checks **0**

 Files changed **1**



+1 -0 



**silviostanzani** commented 1 minute ago

Owner +  ...

segue a alteração pedida, incluir um novo comando print

  Update pythonScript.py

Verified d4943e3

Add more commits by pushing to the **master** branch on **TheLone-Gunmen/teste**.



**✓ This branch has no conflicts with the base branch**

Merging can be performed automatically.

**Merge pull request** or view [command line instructions](#).



Write

Preview

AA B i “ <>     @  

Reviewers



No reviews

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet

Milestone



No milestone

Notifications

[Customize](#)

## Dúvidas