

http://linkedin.com/in/guijac

Curso DevOps

Aula 02 - Gerenciamento de Configurações em DevOps -
Exercícios

Prof. Esp. Guilherme Jorge Aragão da Cruz



guilherme.cruz@alumni.usp.br



linkedin.com/in/guijac

Exercícios

1. Criar uma conta Gitlab;
2. Criar uma conta Atlassian;
3. Instalar o Git;
4. Criar um grupo e projeto no Gitlab;
5. Clonar o repositório criado;
6. Configurar o Git em seu computador;
7. Realizar um *commit* do arquivo README.md.

Criar uma conta Gitlab

- https://gitlab.com/users/sign_in
- Se realizou login com conta Google ou similar, criar uma senha;
- Habilite o MFA (recomendado).

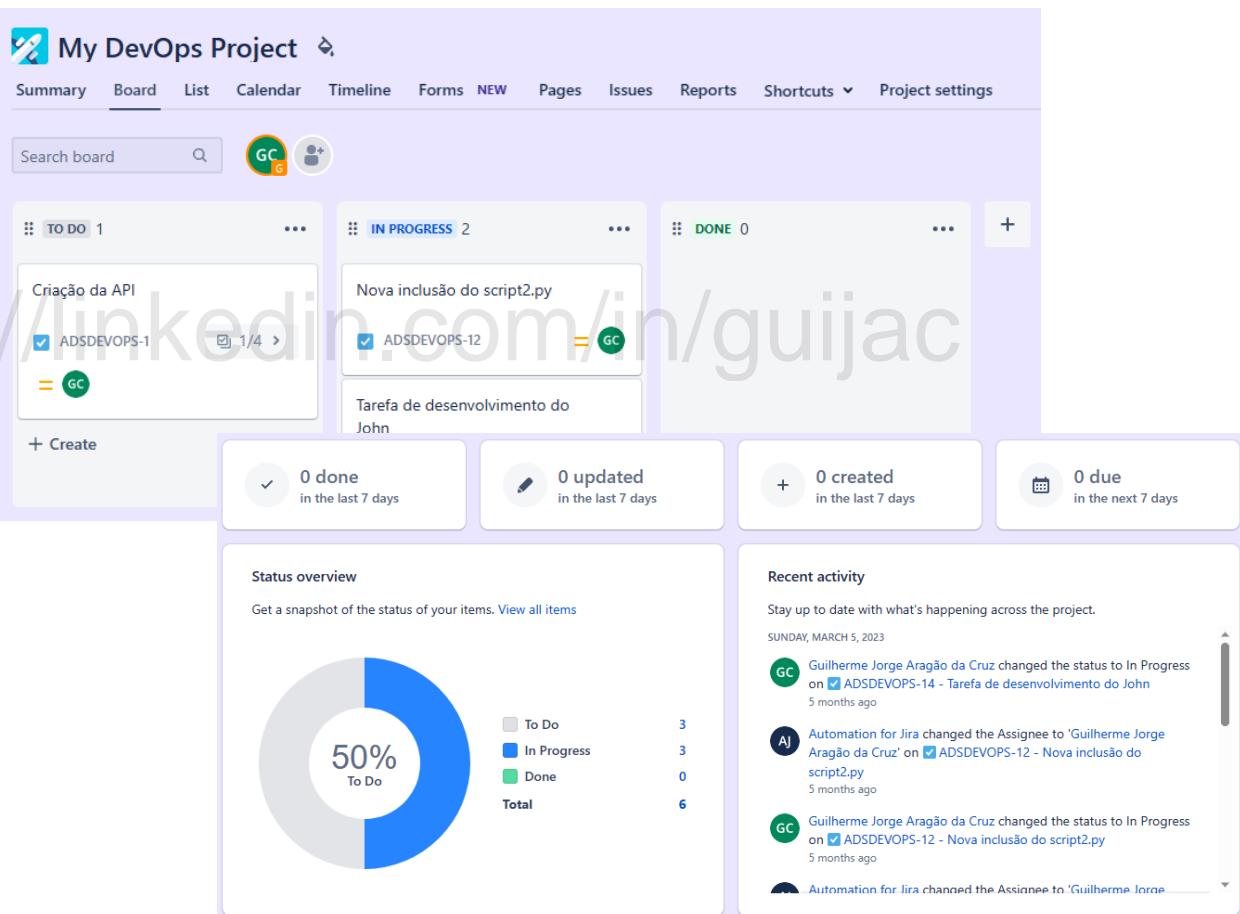


Fonte: Elaboração própria (2025)

The screenshot shows the "User Settings" page in the GitLab web interface. The left sidebar lists various settings: User Settings, Profile, Account (which is selected and highlighted in blue), Billing, Applications, Chat, Access Tokens, Emails, Password, Notifications, SSH Keys, GPG Keys, Preferences, Active Sessions, Authentication log, and Usage Quotas. The main content area has a yellow banner at the top stating "Pipeline failing? To keep GitLab spam and abuse free we ask that you verify your identity. Until then, shared runners will be unavailable. [Validate your account](#) or use your own runners." Below this, there are sections for "Two-factor authentication" (status: Enabled, with a "Manage two-factor authentication" button), "Service sign-in" (with options to connect to Google, Twitter, GitHub, Bitbucket, and Salesforce), "Change username" (current path: https://gitlab.com/guijac, with a "Update username" button), and "Delete account" (with a "Delete account" button and a note about the effects of account deletion).

Criar uma conta Atlassian

- <https://www.atlassian.com/br/software/jira/free>

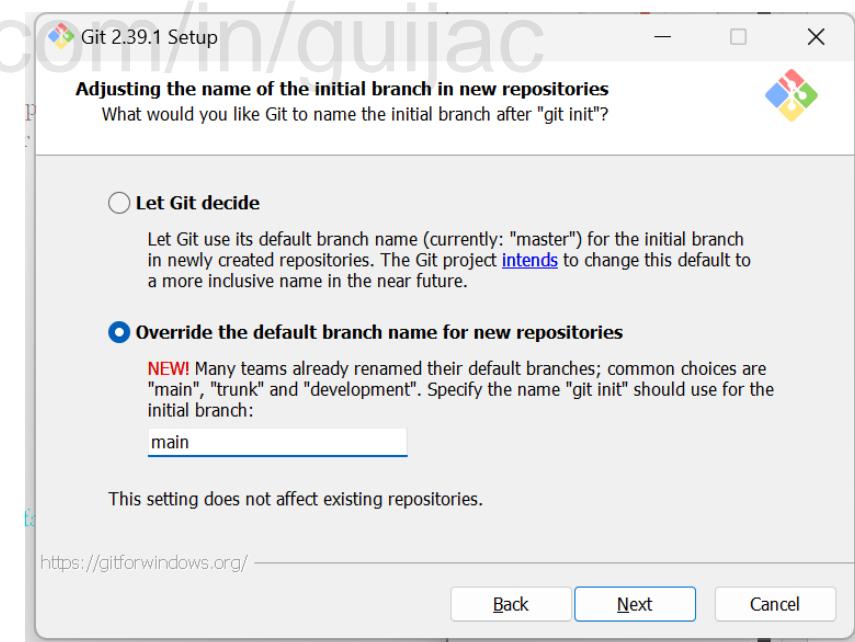
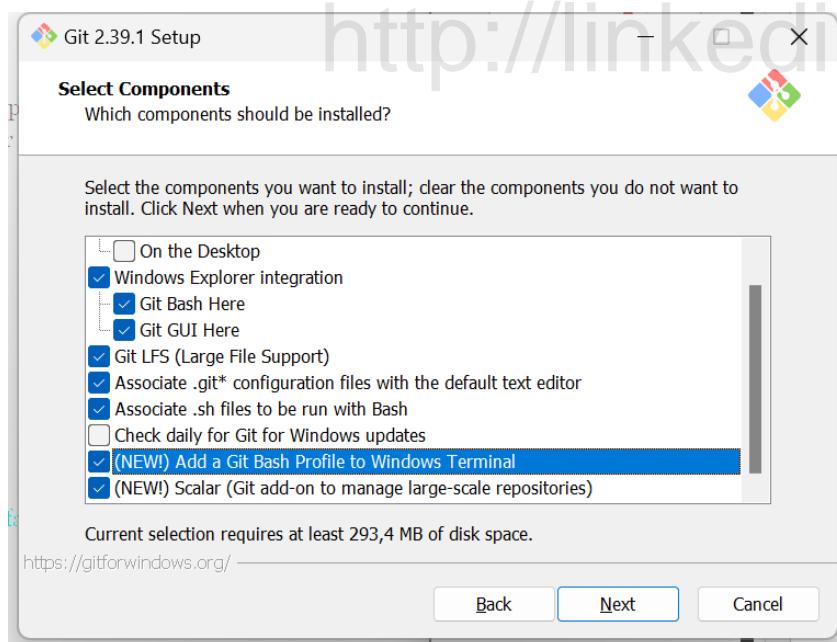


The image shows the Atlassian Jira software interface. On the left, there is a rounded rectangular box with the Jira logo and the text "Gestão de Projetos e Gestão de Mudanças". The main interface is titled "My DevOps Project". It features a "Board" tab selected, showing a Kanban board with three columns: "TO DO" (1 item), "IN PROGRESS" (2 items), and "DONE" (0 items). Below the board, there are four summary cards: "0 done in the last 7 days", "0 updated in the last 7 days", "0 created in the last 7 days", and "0 due in the next 7 days". To the right, there is a "Status overview" section with a pie chart showing 50% "To Do" and 50% "In Progress", and a "Recent activity" section listing recent changes made by users "GC" and "AJ".

Fonte: Elaboração própria (2025)

Instalar o Git

- <https://git-scm.com/downloads>;
- Selecione o nome default para branchs “main”;
- Mantenha as demais configurações e conclua a instalação;
- Reinicie o PC, para ler as alterações do PATH.



Criar um grupo e projeto no Gitlab

- <https://gitlab.com/groups/new>
- Por padrão, um grupo é sempre privado, caso queira **compartilhar** seus exercícios, **deixe o grupo público**.

Create group

Groups allow you to manage and collaborate across multiple projects. Members of a group have access to all of its projects.

Groups can also be nested by creating subgroups.

Group name Group name must start with letter, digit, emoji, or underscore. Can also contain periods, dashes, spaces, and parentheses. Your group name must not contain a period if you intend to use SCIM integration, as it can lead to errors.

Group URL

Visibility level
Who will be able to see this group? [View the documentation](#)

Private
The group and its projects can only be viewed by members.

Public
The group and any public projects can be viewed without any authentication.

Now, personalize your GitLab experience
We'll use this to help surface the right features and information to you.

Role

Who will be using this group?
 My company or team Just me

What will you use this group for?

Create group **Cancel**

Criar um grupo e projeto no Gitlab

- Após clicar em “Create group” siga para a criação do projeto do tipo “blank project”:

Group Git Command Tests was successfully created.

G Git Command Tests Free New subgroup New project ...

Subgroups and projects Shared projects Archived projects Search Name

1 **Create new subgroup**
Groups are the best way to manage multiple projects and members.

2 **Create new project**
Projects are where you can store your code, access issues, wiki, and other features of GitLab.

Create new project

Create blank project
Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Create from template
Create a project pre-populated with the necessary files to get you started quickly.

Import project
Migrate your data from an external source like GitHub, Bitbucket, or another instance of GitLab.

Run CI/CD for external repository
Connect your external repository to GitLab CI/CD.

Criar um grupo e projeto no Gitlab

- Certifique-se que o projeto está sendo criado dentro do grupo criado anteriormente (no exemplo, “git-command-tests”);
- Neste momento, nossos projetos podem ser privados;
- Inicialize o projeto com um arquivo “README.md”.

Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Project name Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL Project slug

Project deployment target (optional)

Visibility Level Private Public

Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.

The project can be accessed without any authentication.

Project Configuration Initialize repository with a README

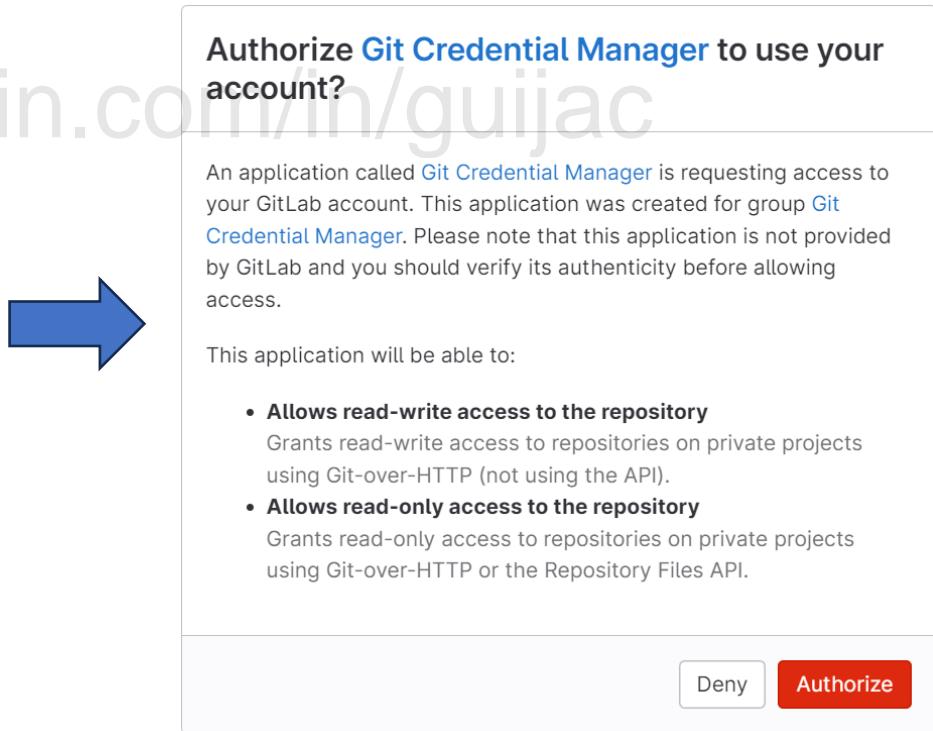
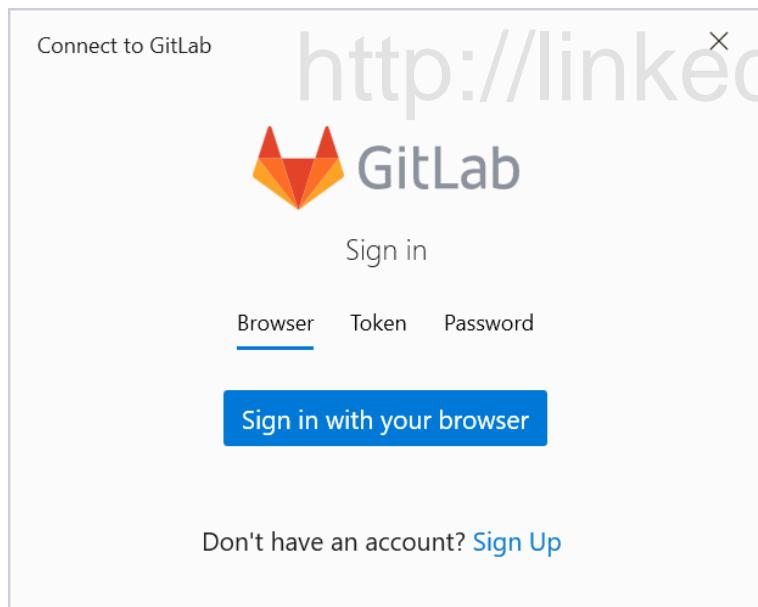
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Enable Static Application Security Testing (SAST)

Analyze your source code for known security vulnerabilities. [Learn more](#).

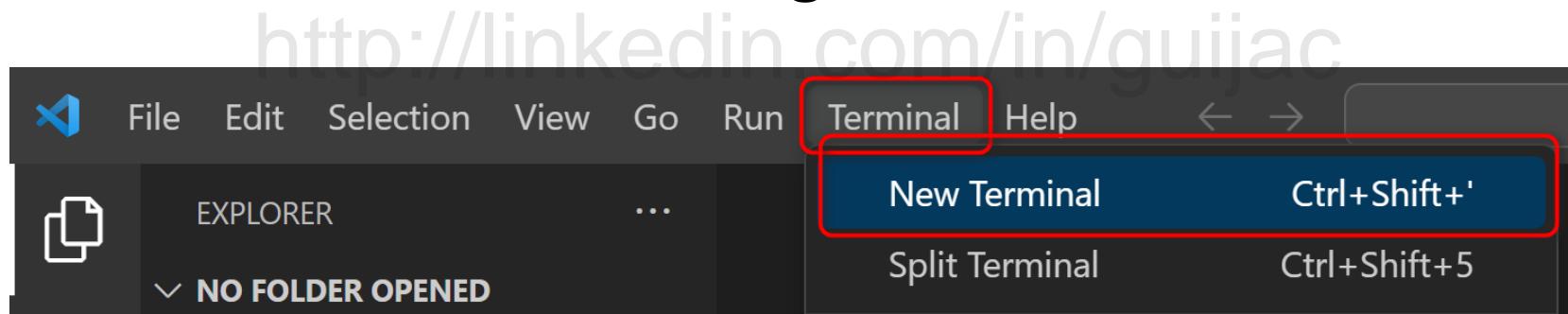
Clonar o repositório criado

- Ao abrir o VSCode através do clone realize a autorização de acesso do Git Credential Manager em sua conta Gitlab:



Configurar o Git em seu computador

- Já dentro do projeto clonado no VSCode, abra o terminal através do menu “Terminal” > “New Terminal”;
- Certifique-se que o clone ocorreu corretamente através do comando “**git status**”:



```
PS C:\Users\Guilherme\workspace\git-tests> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Configurar o Git em seu computador

- Configure seu usuário e e-mail dentro do VSCode, executando os comandos:

```
git config --global user.name "SEU_PRIMEIRO_NOME SEU_ULTIMO_NOME";
```

```
git config --global user.email "seu@email.com"
```

- Este comando só precisa ser realizado uma única vez.**

Realizar um *commit* do arquivo README.md

- Altere o arquivo README.md existente e certifique-se que o Git identificou a alteração dele, novamente através do comando “**git status**”:

```
PS C:\Users\Guilherme\workspace\git-tests> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Guilherme\workspace\git-tests>
```

Realizar um *commit* do arquivo README.md

- Adicione o arquivo alterado README.md em sua “staging área” através do comando “**git add .**” ou “**git add README.md**” ou “**git add –A**” (lembrem-se que neste caso, para um arquivo novo, todos estes parâmetros irão funcionar);
- Novamente, através do comando “**git status**”, certifique-se que o arquivo alterado está em fase de “*stage*”:

```
PS C:\Users\Guilherme\workspace\git-tests> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Realizar um *commit* do arquivo README.md

- Realize o *commit* do arquivo README.md, através do comando “**git commit -m "concluindo minha tarefa"**”;
- Mais uma vez, através do comando “**git status**”, certifique-se que o *commit* ocorreu com sucesso:

```
PS C:\Users\Guilherme\workspace\git-tests> git commit -m "concluindo minha tarefa"
[main 1d5cd80] concluindo minha tarefa
 1 file changed, 1 insertion(+), 91 deletions(-)
```

Realizar um *commit* do arquivo README.md

- Realize o “push” para enviar suas alterações ao servidor remoto através do comando “**git push**”, autorizando, se necessário, o acesso do Git Credential Manager em sua conta Gitlab.

```
PS C:\Users\Guilherme\workspace\git-tests> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (1/1), done.
Writing objects: 100% (3/3), 293 bytes | 293.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://gitlab.com/git-command-tests/git-tests.git
  5ec0585..1d5cd80  main -> main
```

Realizar um *commit* do arquivo README.md

- Certifique-se que o push ocorreu com sucesso, através do comando “**git status**”:

```
PS C:\Users\Guilherme\workspace\git-tests> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Realizar um *commit* do arquivo README.md

- Por fim, certifique-se também que a sua alteração foi refletida no Gitlab:

The screenshot shows a Gitlab repository interface. At the top, there is a header with a 'G' icon, the repository name 'Git Tests', and a lock icon. Below the header, there is a navigation bar with 'main' (a dropdown menu), a search bar with 'git-tests /', a '+' button, and buttons for 'History', 'Find file', 'Edit', and 'Code' (which is highlighted in blue).

Below the navigation bar, a commit message is displayed: 'concluindo minha tarefa' by 'Guilherme Cruz authored just now'. This message is highlighted with a red box.

The main content area shows a table of files with their last commit and update times:

Name	Last commit	Last update
.gitlab-ci.yml	Update .gitlab-ci.yml file	1 day ago
README.md	concluindo minha tarefa	just now

The 'README.md' row is also highlighted with a red box. Below the table, the 'README.md' file content is shown in a box:

```
Git Tests

Alterei meu README.md
```

Referências Bibliográficas

Atlassian. **Gerenciamento de configurações**. Disponível em <https://www.atlassian.com/br/microservices/microservices-architecture/configuration-management>. Acesso em 14 jan 2025;

BINS. **Guia dos principais comandos do GIT**. Disponível em <https://blog.dbins.com.br/guia-dos-principais-comandos-do-git>. Acesso em 14 jan 2025;

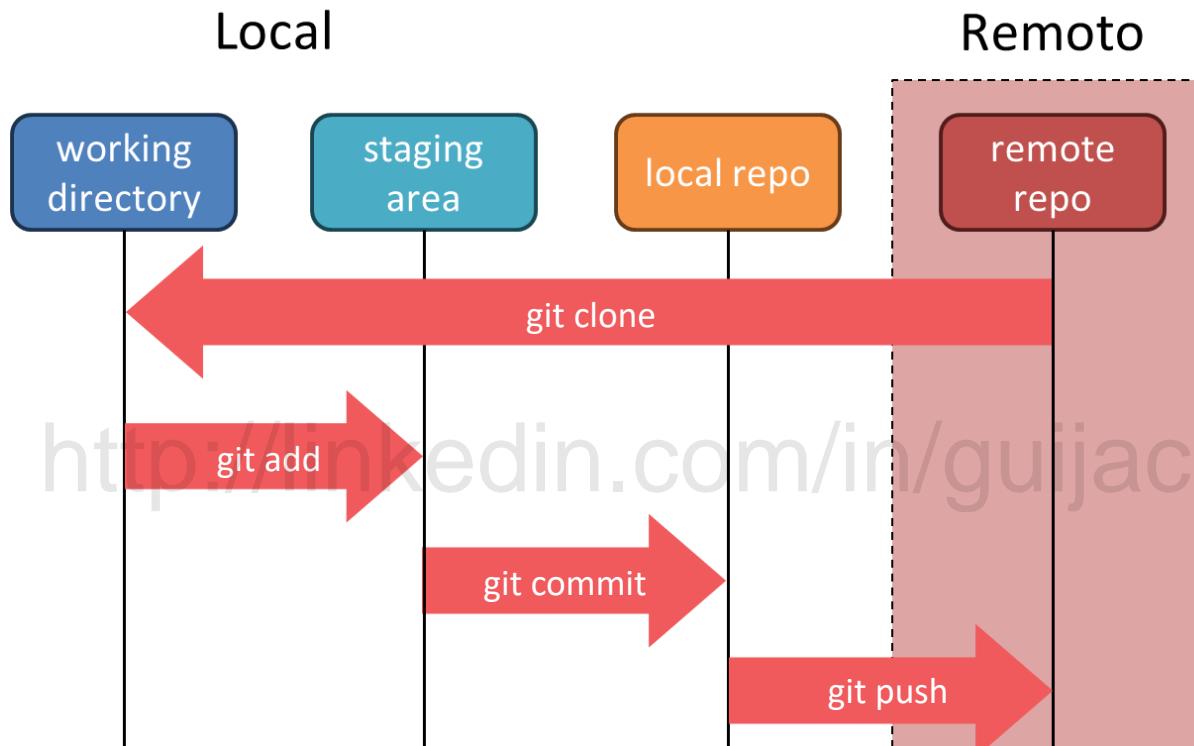
GIT. **Documentation**. Disponível em <https://git-scm.com/doc>. Acesso em 14 jan 2025;

GITHUB. **Git Guides**. Disponível em <https://github.com/git-guides>. Acesso em 14 jan 2025;

ROSA, Daniel. **10 comandos do Git que todo desenvolvedor deveria conhecer**. Disponível em <https://www.freecodecamp.org/portuguese/news/10-comandos-do-git-que-todo-desenvolvedor-deveria-conhecer/>. Acesso em 14 jan 2025;

SSL2BUY. **Infrastructure as Code (IaC) – A Developer's Perspective**. Disponível em <https://www.ssl2buy.com/cybersecurity/infrastructure-as-code-iac>. Acesso em 14 jan 2025;

Por hoje (agora sim!) é só!



Fonte: Elaboração Própria (2025)

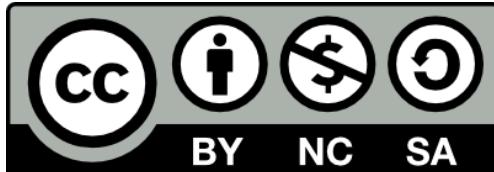
Prof. Esp. Guilherme Jorge Aragão da Cruz

 guilherme.cruz@alumni.usp.br

 linkedin.com/in/guijac

Licença

- Este conteúdo está licenciado sob a Licença Creative Commons Atribuição-NãoComercial-Compartilhagual 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.
- Para ver o texto completo da licença, acesse o <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