Curso DevOps

Aula 02 - Gerenciamento de Configurações em DevOps - Exercícios

Prof. Esp. Guilherme Jorge Aragão da Cruz

guilherme.cruz@alumni.usp.br

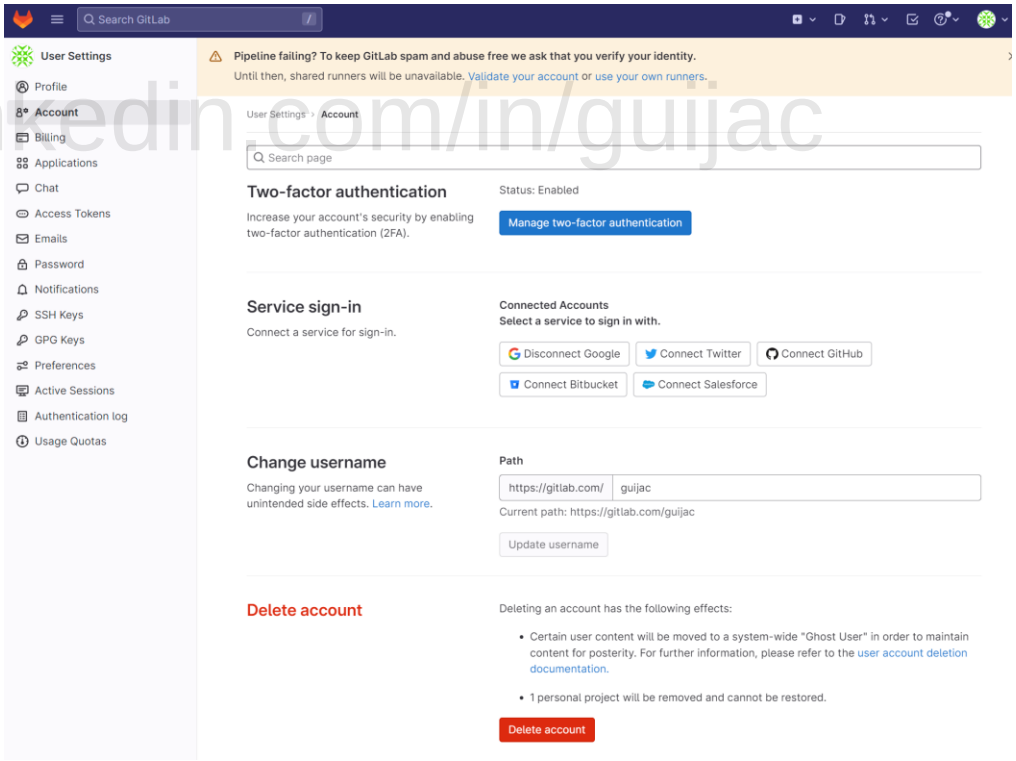
linkedin.com/in/guijac

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).



GitLab

Versionamento de Fontes, Integração e Entrega Contínua

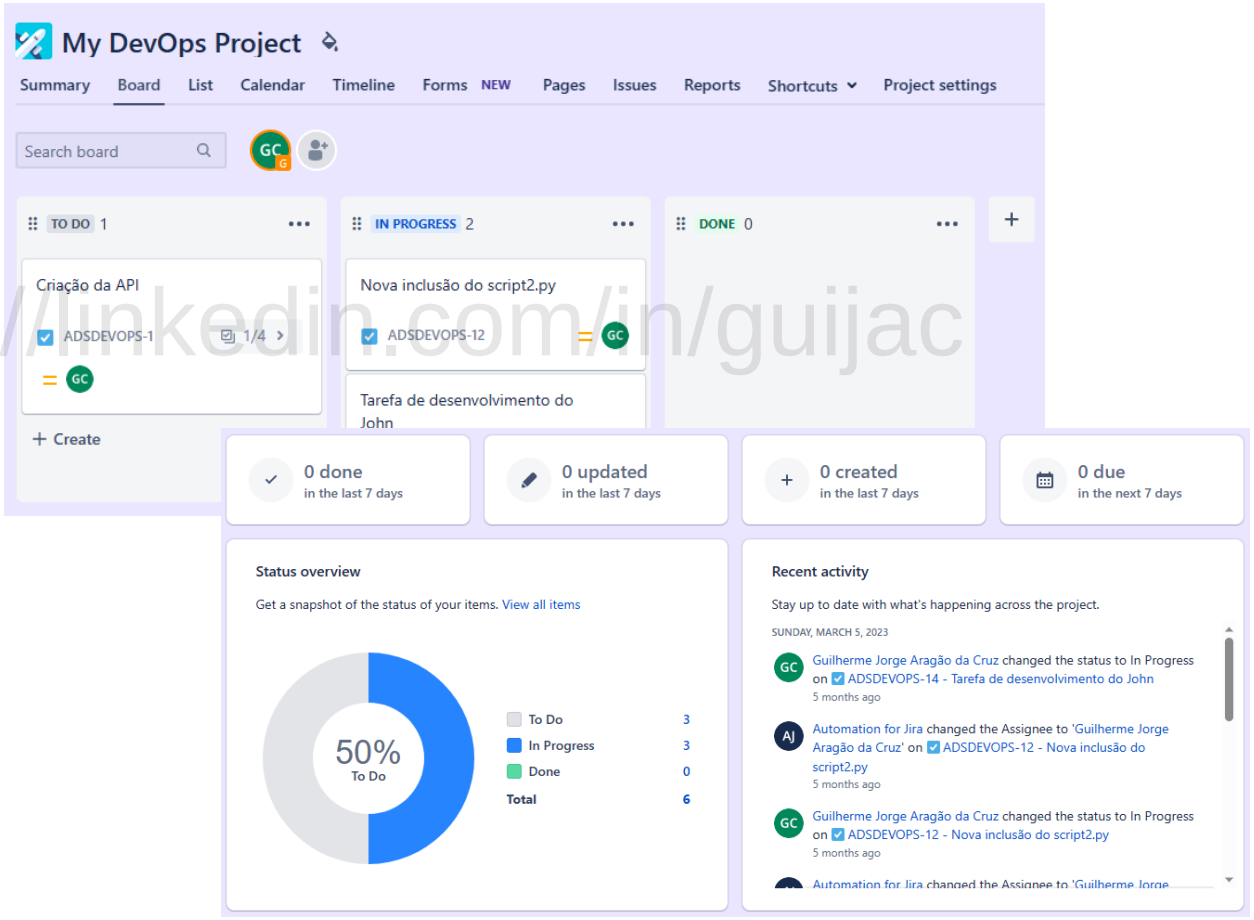
Fonte: Elaboração própria (2025)

Criar uma conta Atlassian

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

Jira

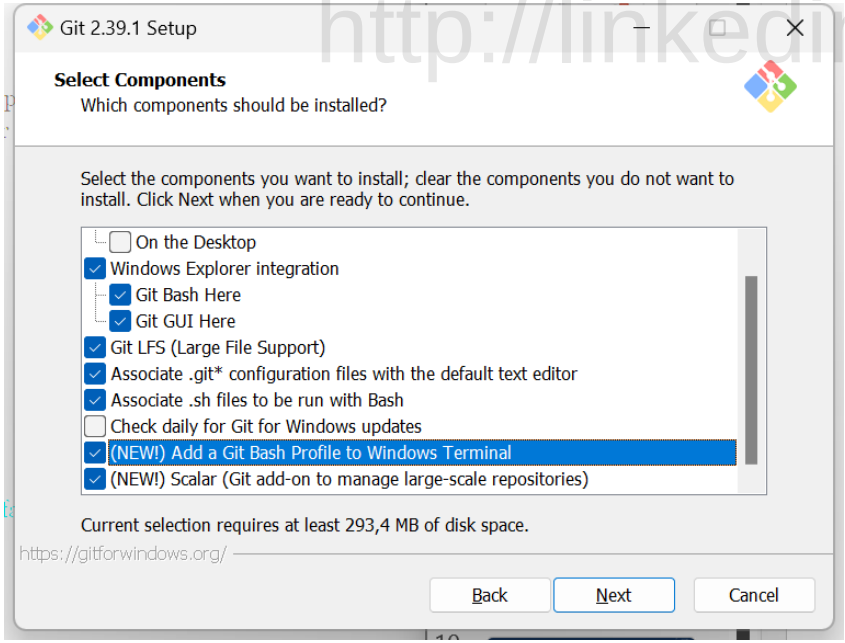
Gestão de Projetos e
Gestão de Mudanças

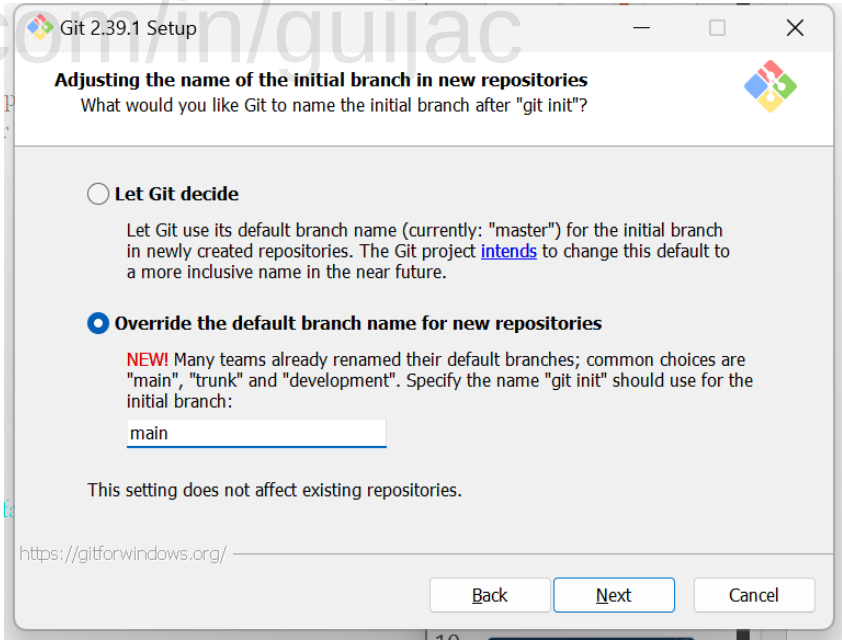


Fonte: Elaboração própria (2025)

Instalar o Git

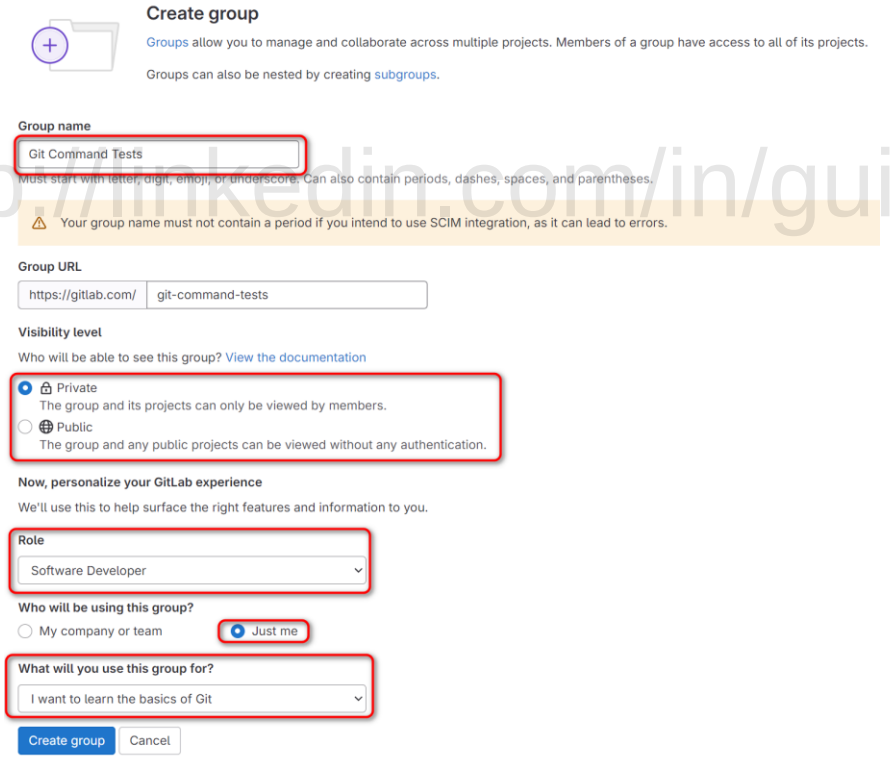
- https://git-scm.com/downloads;
- Selecione o nome default para branchs “main”;
- Mantenha as demais configurações e conclua a instalação;
- Reinicie o PC, para ler as alterações do PATH.





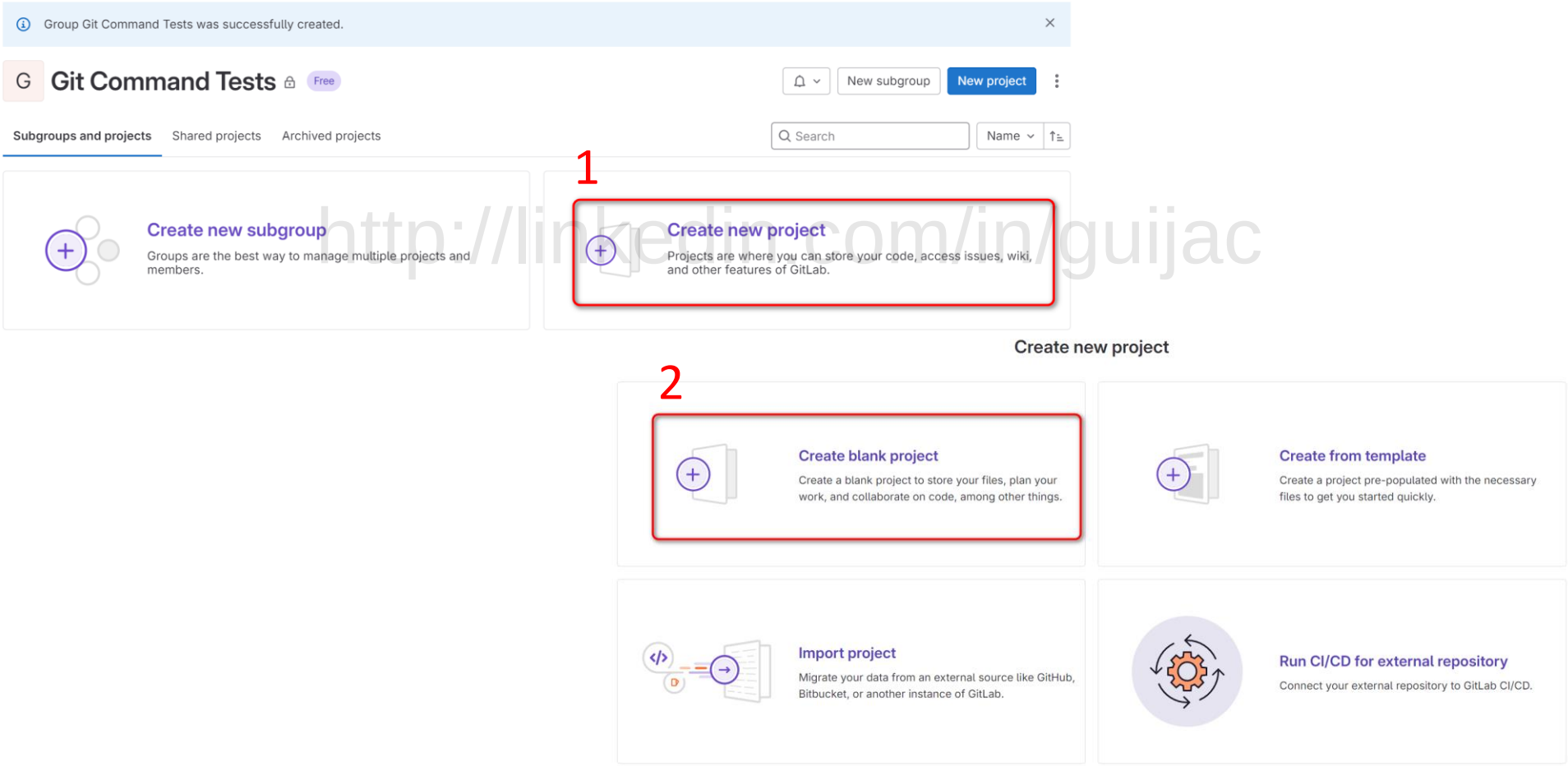
Criar um grupo e projeto no Gitlab

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



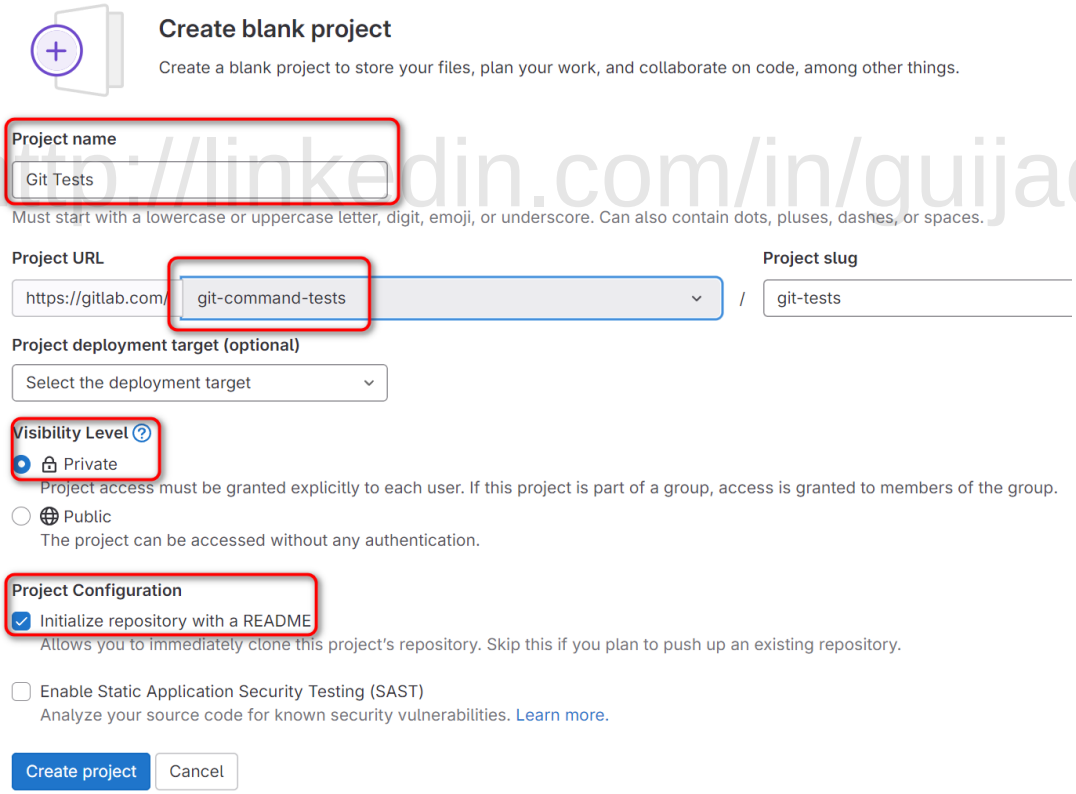
Criar um grupo e projeto no Gitlab

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



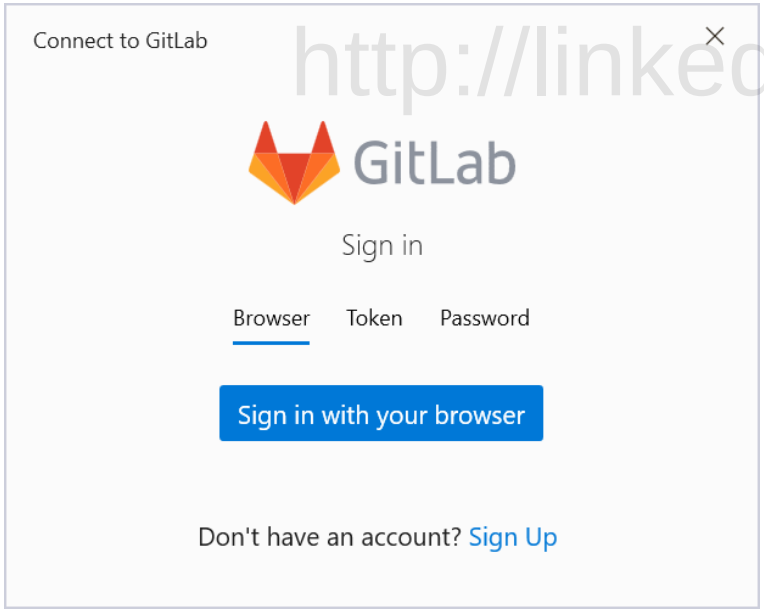
Criar um grupo e projeto no Gitlab

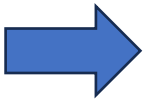
- Certifique-se que o projeto está sendo criado dentro do grupo criado anteriormente (no exemplo, “git-command-tests”);
- Neste momento, nossos projetos podem ser privados;
- Inicialize o projeto com um arquivo “README.md”.

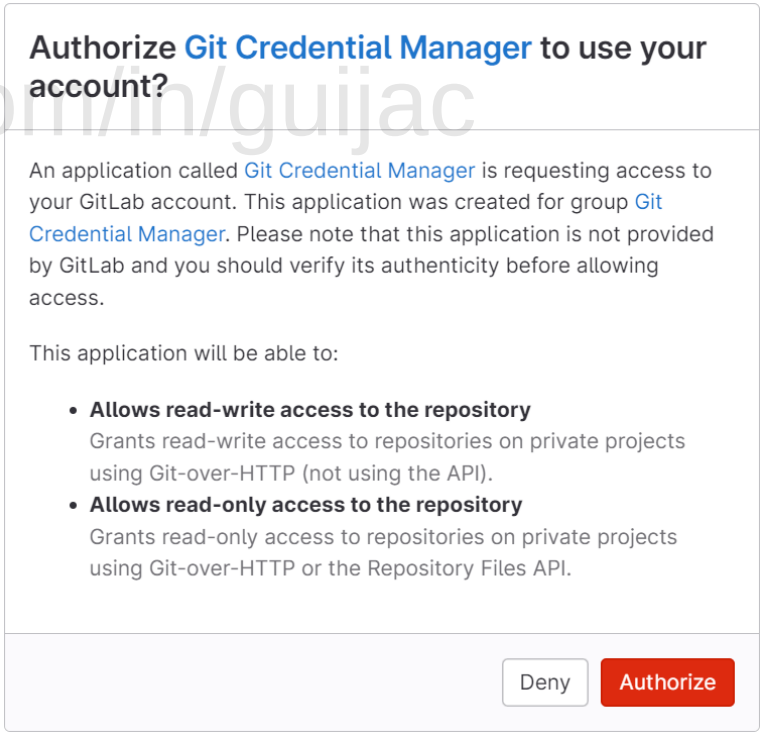


Clonar o repositório criado

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

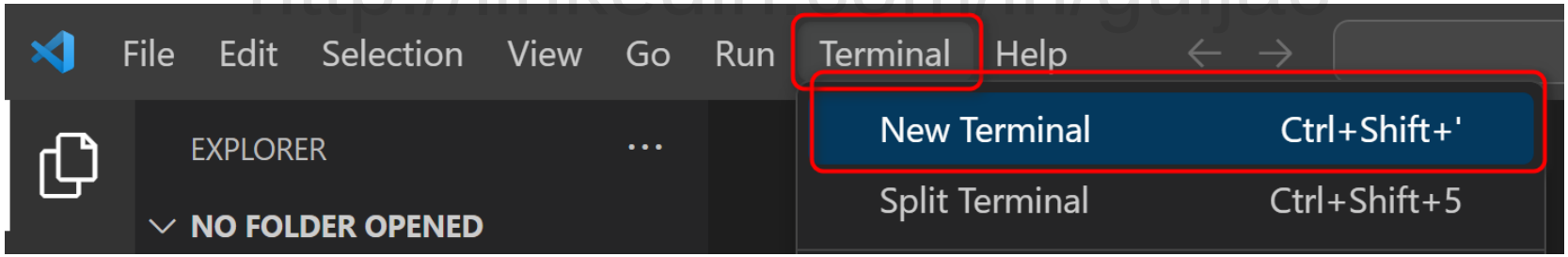






Configurar o Git em seu computador

- 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

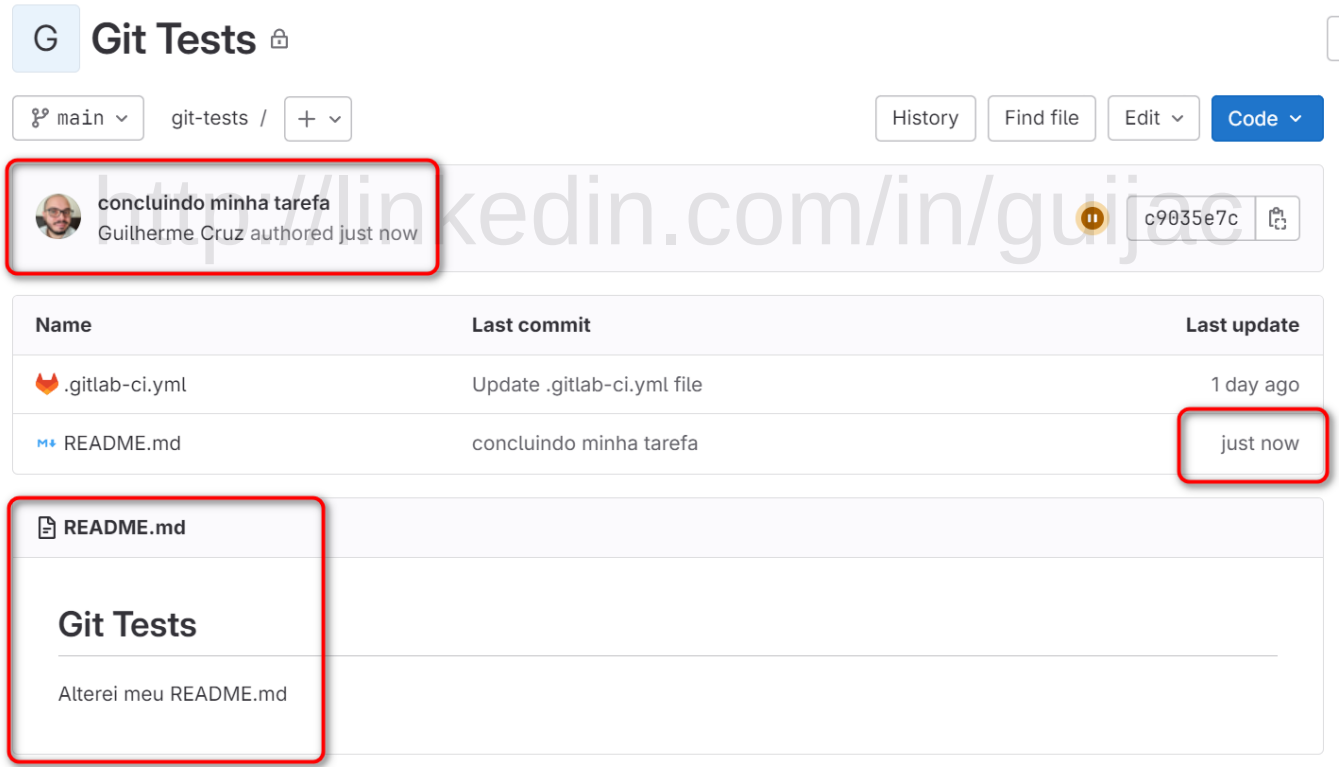
- Certifique-se que o push ocorreu com sucesso, através do comando “**git status**”:

```
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:



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;

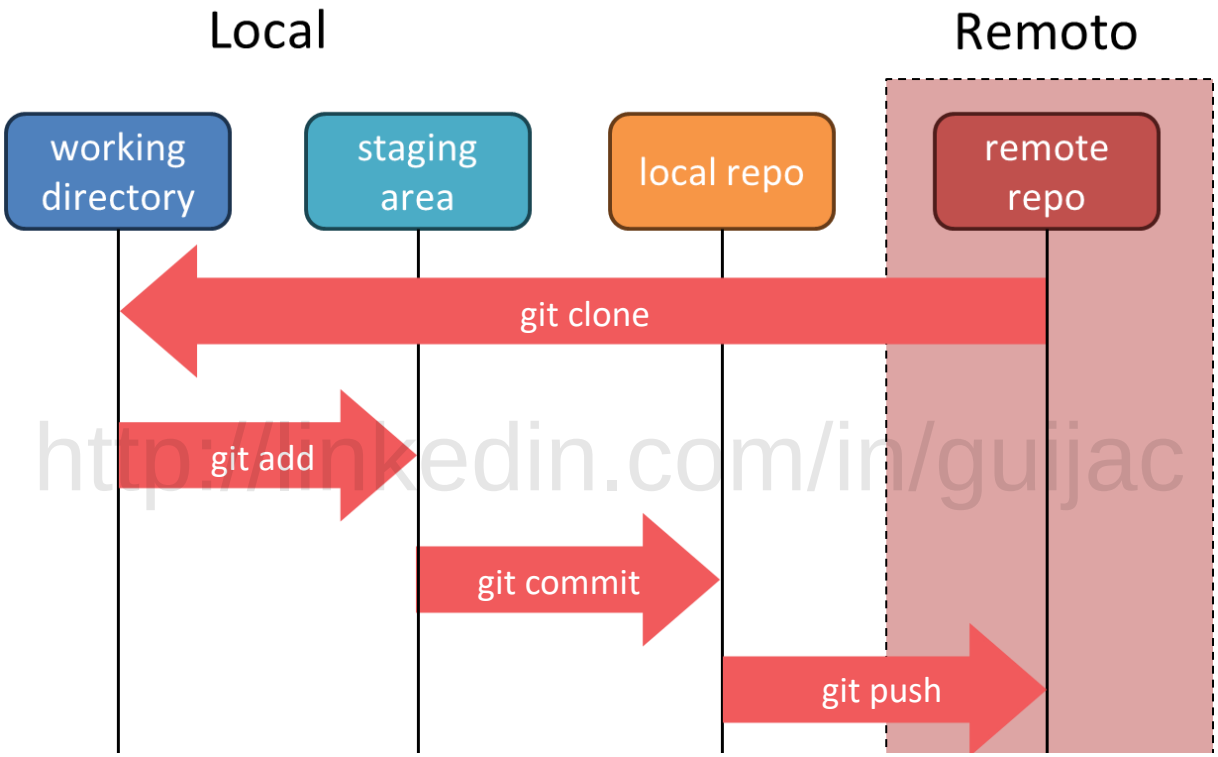
GIT. **Documentation**. Disponível em https://git-scm.com/doc. Acesso em 14 jan 2025;

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-CompartilhaIgual 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