

# **EXTRA 01 - CRIANDO REPOSITÓRIO REMOTO**

Disciplina de Backend - Professor Ramon Venson - SATC 2024

## tl;dr

- Criar conta no [Gitlab](#);
- Criar a página do repositório remoto no gitlab (sem inicializar);
- Organizar a pasta do seu projeto e inicializar o repositório local git ( `git init` );
- Realizar o primeiro commit ( `git add .` e `git commit -m "mensagem"` );
- Adicionar o endereço do repositório remoto ( `git remote add origin [URL]` );
- Fazer o envio das alterações locais ( `git push origin master` ).

Não esqueça de observar o nome do branch inicial (geralmente master ou main)

## Repositórios de Código

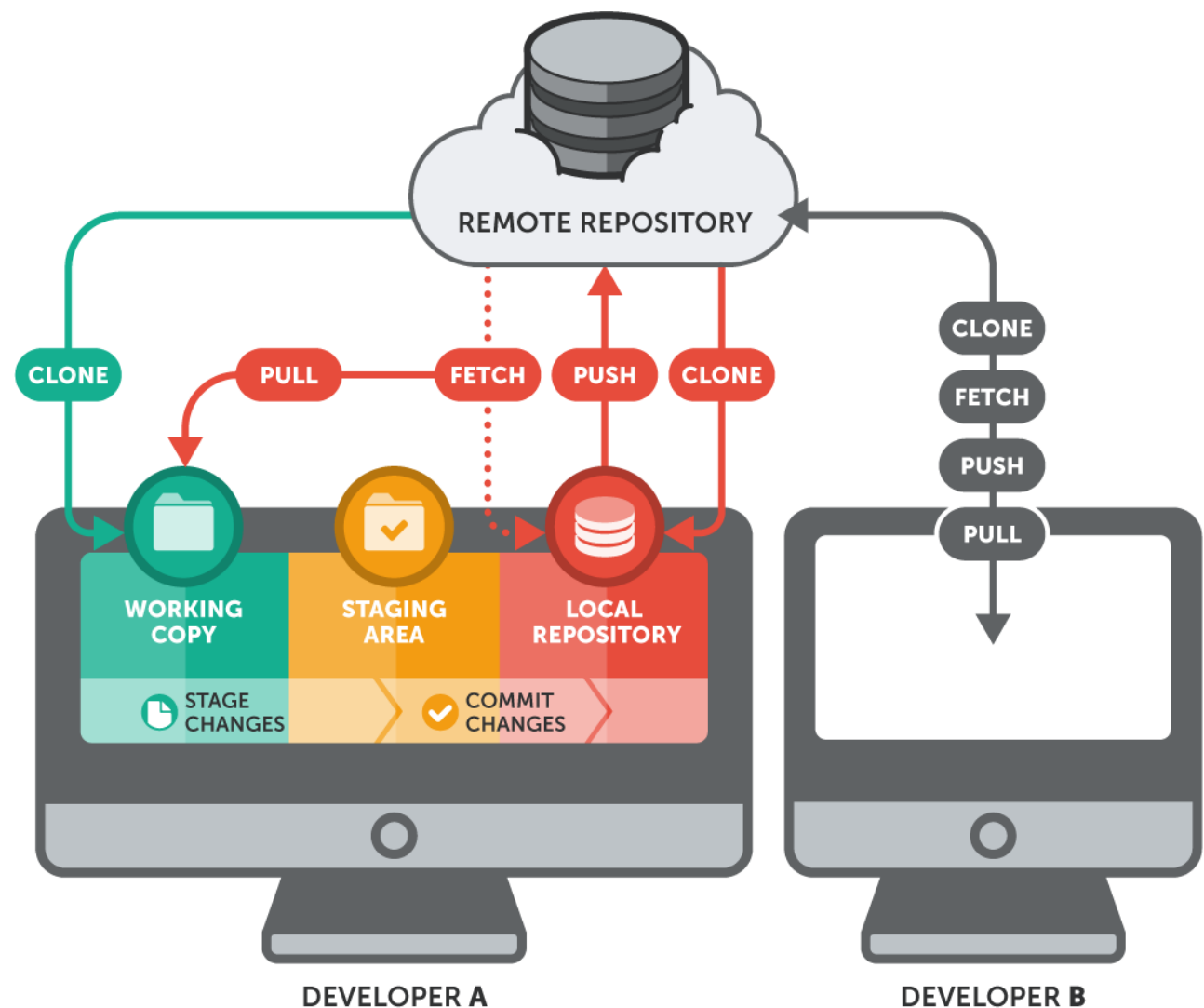
Da necessidade de compartilhar código-fonte e garantir a disponibilidade do código na nuvem surgiram vários gerenciadores de repositório de código-fonte, como é o caso do [Gitlab](#).

Este é o passo-a-passo para a criação de um repositório online e a sincronização com o repositório da máquina-local

- Um repositório local é definido como um repositório particular de cada programador
- Um repositório remoto é o repositório usado para sincronizar repositórios locais

Neste caso usaremos o [Gitlab](#) como repositório remoto.

É muito importante entender a diferença entre local e remoto!



## Criando uma conta



- Vá até o site [Gitlab](https://gitlab.com) e crie uma conta
- Lembre-se de utilizar o mesmo email e nome de usuários configurados no seu git local

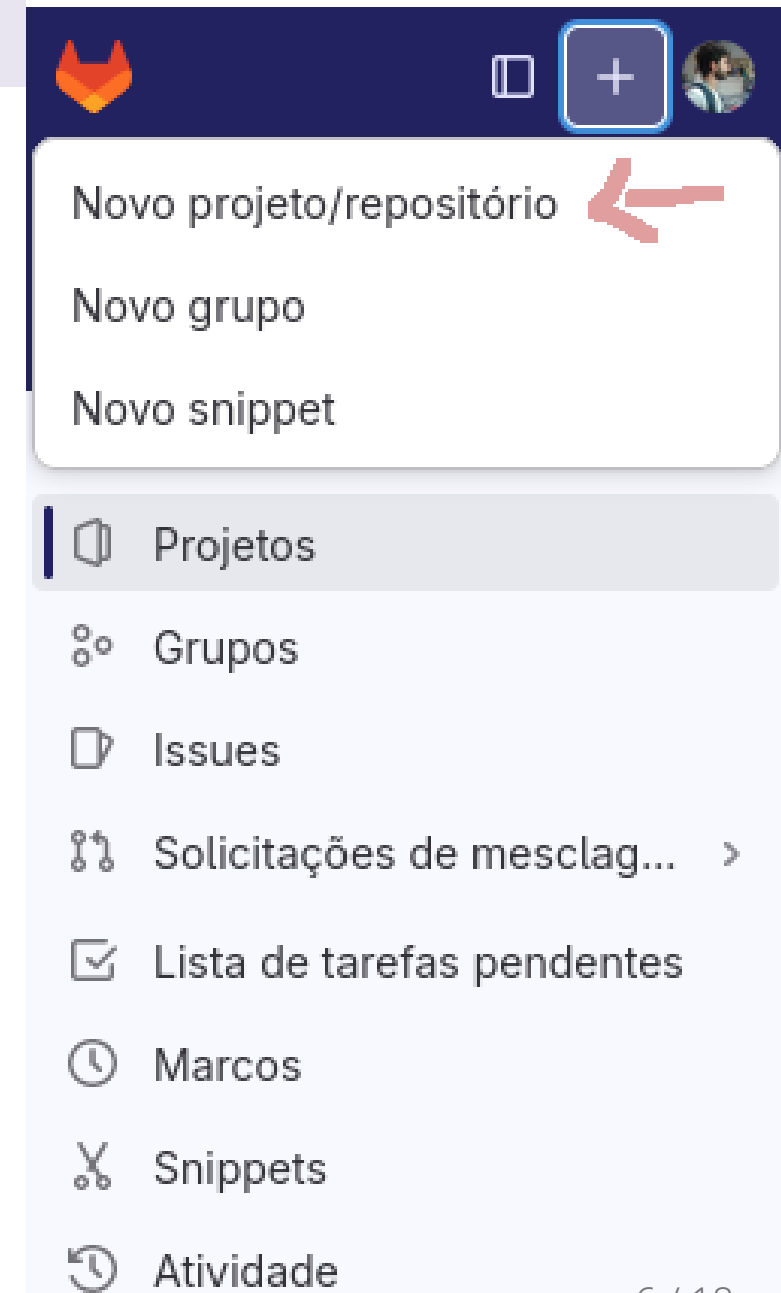
Ao final, faça o login na sua conta.

Para configurar usuário e email no git local use o comando `git config --global user.name "Seu Nome de Usuário"` e `git config --global user.email "seu.email@dominio.com"`

## Criando o repositório remoto (gitlab)

Para armazenar um novo projeto no Gitlab, precisamos criar seu repositório no site:

- Procure pelo **+** no menu e selecione a opção **Novo projeto/repositório**;
- Selecione a opção **criar projeto em branco**



Blank project	Create from template	Import project	CI/CD for external repo
<p><b>Project name</b></p> <input type="text" value="Test1"/>			
<p><b>Project URL</b></p> <input type="text" value="https://gitlab.com/firefox008899/"/>		<p><b>Project slug</b></p> <input type="text" value="test1"/>	
<p>Want to house several dependent projects under the same namespace? <a href="#">Create a group.</a></p>			
<p><b>Project description (optional)</b></p> <div>Description format</div>			
<p><b>Visibility Level</b> ⓘ</p> <p><input checked="" type="radio"/> <b>Private</b> Project access must be granted explicitly to each user.</p> <p><input type="radio"/> <b>Internal</b> The project can be accessed by any logged in user.</p> <p><input type="radio"/> <b>Public</b> The project can be accessed without any authentication.</p> <p><input type="checkbox"/> <b>Initialize repository with a README</b> Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.</p>			
<input type="button" value="Create project"/>		<input type="button" value="Cancel"/>	

- Preencha as informações necessárias
- Selecione a visibilidade do projeto como **Pública**
- **IMPORTANTE!:** desmarque a opção **Inicializar repositório com um README**

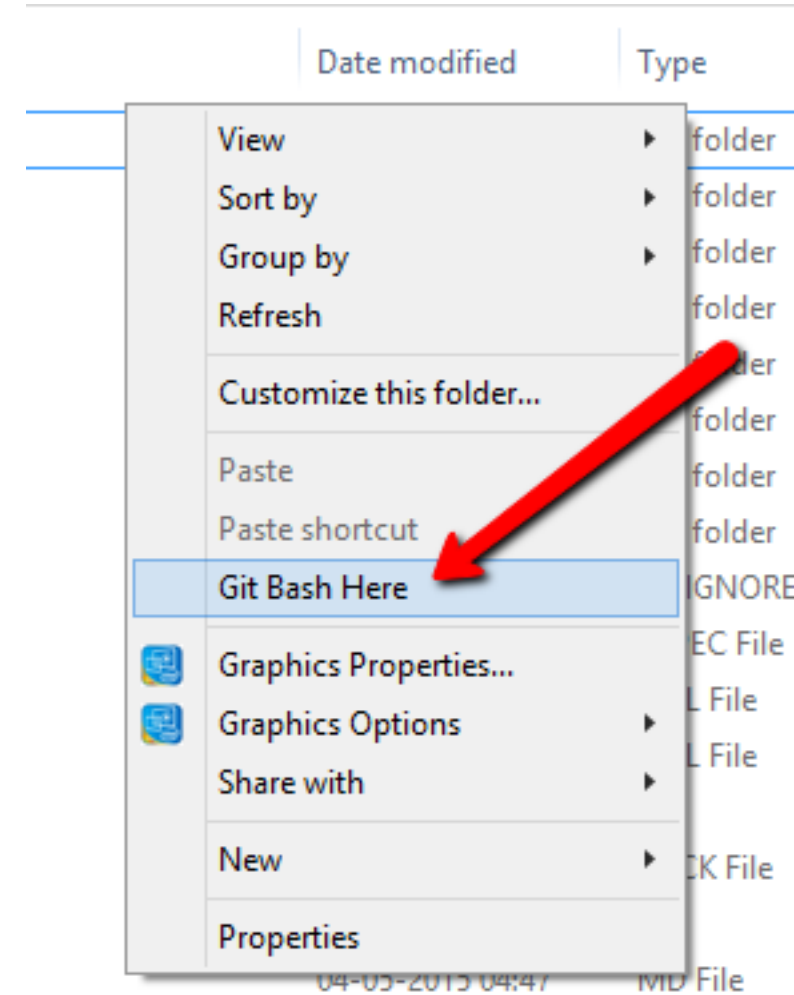
Caso a opção de `Inicializar repositório com um README` não seja desmarcada, o Gitlab inicializará um novo repositório por conta própria, o que deve causar conflitos com o repositório local.

É possível utilizar essa opção desde que você execute a clonagem do repositório remoto e atualize os arquivos a partir dele. *(Não recomendado para iniciantes)*



## Criando o repositório local (git)

- Antes de tudo, organize uma pasta que será a raiz do seu projeto.
  - Copie todos os arquivos para dentro dessa pasta e organize em outras pastas se necessário
- Utilize a opção do gerenciador de arquivos `Git Bash Here` para abrir o terminal na pasta raiz
- Execute o comando `git init` para inicializar o seu repositório local dentro da pasta.



**ATENÇÃO!** Sempre certifique-se de estar na **pasta raiz** no momento da criação do repositório.

A pasta raiz é a pasta que armazena todas as outras e será a primeira referência do seu repositório.

Utilize os comandos `pwd`, `git status` e `ls` para saber onde seu repositório foi criado.

A etapa de criação do repositório só é, de maneira geral, necessária **UMA ÚNICA VEZ** para cada projeto.

## Commit (local)

- Ainda dentro do repositório recém-criado, certifique-se de estar no local correto (pasta raiz do seu projeto)
- Se você não fez ainda, copie/organize todos os arquivos que você quer versionar (salvar)
- Use o comando `git add .` para adicionar todos os arquivos

Você pode substituir o `.` no comando pelo nome individual dos arquivos e pastas que você pretende adicionar. Isso é útil quando você deseja persistir na sua versão apenas alguns dos arquivos que você alterou.

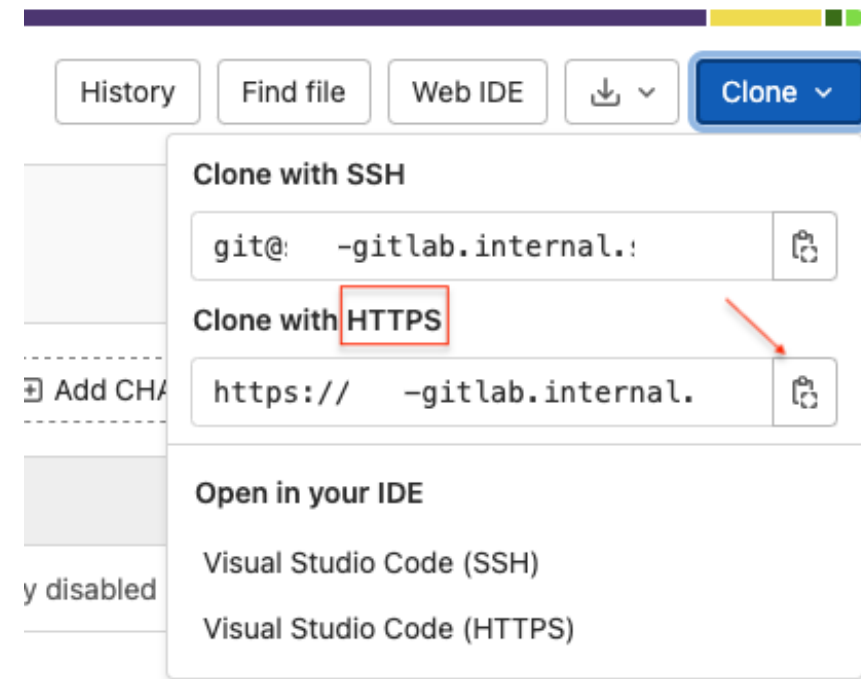
- Por fim, ainda na raiz, use o comando `git commit -m "Mensagem"`
  - Substitua `"mensagem"` por uma mensagem curta descrevendo as alterações. Por boas práticas, não utilize mensagens maiores do que 50 caracteres.
- Se tudo correr corretamente, você deve ser capaz de visualizar seu novo commit usando o comando `git log`

## Vinculando repositório remoto

Antes de enviar as alterações para o repositório **remoto** (gitlab) é necessário incluir a URL no seu novo repositório **local**.

- Para isso, use o comando `git remote add origin [URL]`

Substitua `[URL]` pelo endereço **HTTPS** do seu repositório, que pode ser encontrado na página criada, botão azul `Código`. Ex.: `git remote add origin https://gitlab.com/usuario/projeto`



Você pode utilizar o endereço SSH se preferir (geralmente mais seguro e prático). Porém será necessário criar uma chave assimétrica. Mais detalhes [aqui](#)

Também é possível mudar o `origin` por qualquer outro alias (apelido), que será necessário lembrar depois. Para ver todos os seus alias cadastrados, use o comando `git remote -v`

## Enviando alterações locais

Por fim, com a página do seu repositório remoto criada (e em branco) e seu repositório local com pelo menos um commit, utilize o comando `git push origin master` para enviar os commits.

Pronto! Se tudo correr bem, seu repositório remoto estará atualizado!

Lembre-se de substituir `origin` pelo alias do seu repositório remoto e `master` pelo nome do seu branch caso tenha utilizado qualquer outro nome nos dois casos.

Nada além dos **commits** será enviado ao repositório remoto, por isso sempre lembre de realizar o commit antes de executar esse comando.

Alterações do **Working Directory** (o estado atual do sistema de arquivos) ou enviadas ao **Stage Index** (quando usamos apenas o `git add`), nunca são sincronizadas.

Nenhum commit feito na máquina local será compartilhado sem o comando `git push`.



## Próximos passos

Após todos esses passos, seu repositório local e remoto estarão sincronizados. Sempre que realizar quaisquer alterações no código, lembre-se que é necessário executar os abaixo novamente para sincronizar:

- Adicionar os arquivos no index usando o comando `git add`
- Realizar o commit usando o `git commit`
- Enviar as alterações usando `git push`

Consulte os comandos completos nos passos anteriores

## Atualizando as alterações remotas

Para casos onde seu repositório é compartilhado (com outras pessoas ou outros computadores), lembre-se sempre antes de realizar alterações usar o comando `git pull origin master` para atualizar qualquer alterações vindas do repositório remoto.