

GIT (Versão Prévia)

LOGIN:

- `git config --global user.name "SEU USUÁRIO"`
- `git config --global user.email "SEU EMAIL"`

LOGOUT:

- `git config --global --unset user.name`
- `git config --global --unset user.email`
- `git config --global --unset credential.helper`
- `cmdkey /delete:git:https://github.com` (WINDOWS)
- `git credential reject` (LINUX)

CLONE (Para projetos existentes):

- `git clone <URL> diretório*`
- * Definir o diretório é opcional, caso defina ele criará armazenará o repositório em um diretório com o nome que você definiu, caso não defina, o nome do diretório será o mesmo nome do repositório

INIT (PARA PROJETOS NOVOS):

- `git init`
- `git branch -M main` (Para informar que você usará a branch main)
- `git remote add origin https://github.com/SEUUSUARIO/SEUREPOSITORIO.git`

ENVIAR ARQUIVOS:

- **Preparar:**
 - 1 arquivo por vez:
 - `git add example1.php`
 - 2 ou mais arquivos:
 - `git add example1.php example2.php`
 - Todos os arquivos pendentes
 - `git add .`
- **Commit:**
 - `git commit -m "Descrição das suas alterações"`
- **Enviar para o github ou outro:**
 - `git push -u origin main`
 - OBSERVAÇÃO: CASO NÃO CONSIGA FAZER O PUSH, VERIFIQUE OS SEGUINTE CASOS:
 - Você está logado com sua conta? Caso não tenha certeza se está logado com sua conta, faça o logout e repita os passos do login
 - Caso tenha certeza de que está logado com sua conta, verifique se você tem acesso ao repositório remoto, esse problema geralmente acontece quando você está tentando utilizar um repositório de outro usuário e não tem acesso a ele.

SINCRONIZAR REPOSITÓRIO:

`git pull`

OUTRAS DICAS

LEMBRE SEMPRE DE ENTRAR NO SEU REPOSITÓRIO PARA DAR OS COMANDOS DE ENVIO OU DE SINCRONIZAÇÃO

Lembre de fazer o logout ao final da aula para que outros alunos não tenham acesso a sua conta.

COMANDOS ÚTEIS PARA O USAR O PROMPT DE COMANDO

cd: altera o diretório

ls (Linux) ou dir (Windows): Lista os arquivos do diretório atual

mkdir: cria diretórios

nano nomeDoArquivo.extensão (Linux com nano instalado): Pode criar um arquivo novo

echo SEU TEXTO > [File-Name].txt: Pode criar um arquivo novo

DETALHANDO MELHOR CADA COMANDO

git config: Utilizado para configurar as opções de personalização do Git no seu computador. Você pode definir configurações de usuário, como nome e e-mail, que serão usadas nos seus commits.

git config --global user.name "SEU USUÁRIO": Este comando define o nome de usuário para todos os repositórios Git no seu computador. O parâmetro --global aplica a configuração globalmente, o que significa que, a menos que seja sobrescrito em um nível mais específico (como o nível do repositório), este será o nome de usuário usado em todos os commits que você fizer em qualquer repositório no seu sistema.

git config user.name "SEU USUÁRIO": Este comando define o nome de usuário apenas para o repositório Git atual em que você está trabalhando. Ele não tem o parâmetro --global, então a configuração é aplicada somente ao repositório local. Isso é útil quando você quer ter um nome de usuário diferente para um projeto específico, separado da sua configuração global.

Em resumo:

Global (--global): Define o nome de usuário para todos os repositórios no seu computador.

Local (sem --global): Define o nome de usuário apenas para o repositório atual.

git clone [url]: Cria uma cópia local de um repositório remoto. O [url] é o endereço do repositório que você deseja clonar.

git init: Inicializa um novo repositório Git local. Esse comando cria um novo subdiretório .git, que contém todos os arquivos necessários do repositório.

git branch -M [nome]: Renomeia a branch atual para [nome]. O parâmetro -M indica que, se a branch de destino já existir, ela deve ser substituída sem aviso. É possível usar também somente git branch [nome].

git remote add [nome] [url]: Adiciona um novo repositório remoto com o apelido [nome] para a URL especificada [url]. Isso permite que você referencie o repositório remoto de forma mais fácil.

git add [arquivo]: Adiciona um ou mais arquivos à área de staging (preparação) para que eles sejam incluídos no próximo commit. Se você passar um diretório ou . (ponto), adicionará todos os arquivos modificados.

git commit -m "[mensagem]": Cria um novo commit contendo as alterações atualmente na área de staging. A [mensagem] é uma descrição curta do que foi alterado.

git push -u [remote] [branch]: Envia os commits da branch local para a branch remota. O parâmetro -u define a branch remota como upstream, o que significa que futuros comandos git pull irão automaticamente trazer as mudanças dessa branch. [remote] é o nome do repositório remoto (geralmente "origin") e [branch] é a branch para a qual você está empurrando as mudanças (geralmente "main").

git pull: Atualiza o repositório local para o estado mais recente do repositório remoto, trazendo e integrando as mudanças.