**Anotações (:**

* **git init**: Inicializa um novo repositório Git local na pasta atual.
* **git clone** [URL]: Copia um repositório Git remoto para o seu computador.
* **git status:** Mostra o estado atual do repositório, incluindo os arquivos modificados, adicionados ou removidos.
* **git add [arquivos]:** Adiciona as alterações dos arquivos selecionados à área de preparação (staging).
* **git commit -m "mensagem":** Cria um novo ponto na história do repositório com as alterações da área de preparação e uma mensagem descritiva.
* **git pull:** Baixa e mescla as alterações do repositório remoto para o ramo atual.
* **git push:** Envia as alterações do ramo local para o repositório remoto. git branch: Lista todos os ramos (branches) no repositório local.
* **git checkout [branch]:** Altera para o ramo especificado, permitindo que você trabalhe em uma linha de desenvolvimento diferente.
* **git merge [branch]:** Incorpora as alterações de um ramo específico no ramo atual
* **git config –list:** É usado para exibir todas as configurações do Git que estão definidas no nível do sistema, do usuário e do repositório local. Ele lista todas as configurações com seus respectivos valores.
* **git config --global user.email:** É a chave de configuração que armazena o endereço de e-mail do usuário do Git. Esse endereço de e-mail é usado para identificar o autor dos commits nos repositórios.
* **git config --global user.name:** É a chave de configuração que armazena o nome do usuário do Git. Esse nome é também utilizado para identificar o autor dos commits nos repositórios.
* **git remote add origin <URL\_do\_repositorio\_remoto>:** É usado para conectar o seu repositório Git local a um repositório remoto.
* **git push origin <Nome\_Do\_Repositorio>:**  É usado para enviar o conteúdo do repositório local para um repositório remoto.

**Como configurar a chave SSH no seu computador:**

A **autenticação por SSH** é mais segura e conveniente do que usar nome de usuário e senha para acessar o repositório remoto.

O **SSH** — Secure Shell, ou algo próximo a “cápsula de proteção” em português — é um protocolo usado para entrar na hospedagem de um site e fazer os mais variados tipos de mudanças nas configurações.

1. **ssh-keygen -t ed25519 -C meuemail.com**

**ssh-keygen**: Esse é o comando em si, usado para gerar as chaves SSH.

**-t ed25519:** É uma opção que especifica o tipo do algoritmo de criptografia a ser usado para gerar o par de chaves. Nesse caso, está definido como ed25519, que é um algoritmo moderno e seguro recomendado para uso atualmente.

-**C meuemail.com**: É uma opção que permite adicionar um comentário à chave pública gerada. O comentário é geralmente um identificador ou informação que ajuda a identificar a chave em questão. Neste exemplo, o comentário é definido como meuemail.com, mas você pode substituí-lo por qualquer informação relevante.

1. **Cat “chave publica”**

O comando cat irá mostrar o conteúdo da chave pública no terminal. Você pode copiar esse conteúdo e adicioná-lo ao servidor Git ou qualquer outro serviço que requer a chave pública para autenticação SSH.

1. **eval $(ssh-agent)**

Inicia o agente SSH no seu terminal atual. O agente SSH gerencia chaves privadas e permite que você as utilize sem precisar digitar a senha a cada operação.

1. **ssh-add “chave privada”**

Adiciona a chave privada gerada anteriormente (id\_ed25519) ao agente SSH para que ele possa usá-la para autenticação sem senha. Isso significa que, a partir de agora, você poderá fazer operações Git (e outras operações SSH) sem precisar inserir a senha da chave privada todas as vezes.

**EXEMPLO**

ssh-keygen -t ed25519 -C [meuemail@gmail.com](mailto:meuemail@gmail.com)

cat id\_ed25519.pub

eval $(ssh-agent)

add id\_ed25519

OBS: Sempre se atentar onde foi o caminho que as chaves foram criadas.