



Como Usar GitHub com Chaves SSH

Tutorial Completo: Configuração de Chaves Pública e Privada

1 Gerar as Chaves SSH

No terminal (Linux/Mac) ou Git Bash (Windows), execute:

```
ssh-keygen -t ed25519 -C "seu-email@example.com"
```

Ou, para compatibilidade com sistemas mais antigos:

```
ssh-keygen -t rsa -b 4096 -C "seu-email@example.com"
```

- Pressione **Enter** para usar o local padrão (`~/.ssh/id_ed25519`)
- Defina uma senha (opcional, mas recomendado)

2 Adicionar a Chave ao SSH Agent

```
eval "$(ssh-agent -s)"  
ssh-add ~/.ssh/id_ed25519
```

3 Copiar a Chave Pública

```
cat ~/.ssh/id_ed25519.pub
```

Copie **todo o conteúdo** que aparecer na tela.

4 Adicionar a Chave no GitHub

1. Acesse **GitHub.com** → **Settings** → **SSH and GPG keys**
2. Clique em **New SSH key**
3. Dê um título descritivo (ex: "Meu Laptop")
4. Cole a chave pública no campo "Key"
5. Clique em **Add SSH key**

5 Testar a Conexão

```
ssh -T git@github.com
```

Deve aparecer uma mensagem como:

✓ **Sucesso:** "Hi username! You've successfully authenticated, but GitHub does not provide shell access."

6 Usar com Repositórios

Para clonar um repositório:

```
git clone git@github.com:usuario/repositorio.git
```

Para mudar um repositório existente de HTTPS para SSH:

```
git remote set-url origin git@github.com:usuario/repositorio.git
```



Dicas de Segurança

⚠ **Importante:**

- **Chave privada** (`id_ed25519`): NUNCA compartilhe, fica só no seu computador
- **Chave pública** (`id_ed25519.pub`): Esta você adiciona no GitHub
- Use uma senha forte para proteger a chave privada
- Você pode ter múltiplas chaves SSH para diferentes serviços

Vantagens das Chaves SSH

✓ Benefícios:

- Não precisa digitar senha a cada push/pull
- Mais seguro que HTTPS com senha
- Funciona com 2FA ativado
- Melhor para automação e scripts

Comandos de Verificação

```
# Listar chaves SSH disponíveis
ls -la ~/.ssh/

# Ver chave pública
cat ~/.ssh/id_ed25519.pub

# Testar conexão
ssh -T git@github.com

# Verificar remote do repositório
git remote -v
```

Solução de Problemas

Se der erro "Permission denied":

1. Verifique se a chave foi adicionada ao SSH agent
2. Confirme se a chave pública foi adicionada no GitHub
3. Teste a conexão com `ssh -T git@github.com`

Para Windows (Git Bash):

```
# Iniciar SSH agent automaticamente  
eval `ssh-agent -s`  
ssh-add ~/.ssh/id_ed25519
```



Tutorial criado para facilitar o uso do GitHub com autenticação SSH



Para mais informações, consulte a documentação oficial do GitHub