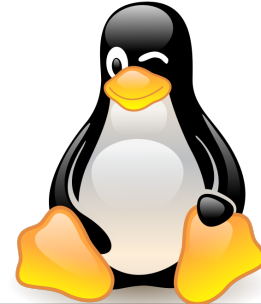


# Usando chaves SSH com o Git (/pt/2017/04/09/usando-chaves-ssh-com-o-git/)

09 de Abril de 2017 (/2017/04/09/)

19

Sobre



(<https://kamarada.github.io/assets/img/TuxKamarada.png>)

O Projeto Linux Kamarada é um projeto novo que visa divulgar e promover o Linux como um sistema operacional robusto, seguro, versátil e fácil de usar, adequado para o uso diário seja em casa, no trabalho ou no servidor. Os focos do projeto são principalmente distribuição e documentação.

(<http://www.facebook.com/LinuxKamarada>)

([twitter.com/LinuxKamarada](https://twitter.com/LinuxKamarada))

([github.com/kamarada](https://github.com/kamarada))

(<https://kamarada.github.io/pt/2017/04/09/usando-chaves-ssh-com-o-git/>)

Esse *post* é para os desenvolvedores. Se você utiliza o sistema de controle de versão Git (<https://git-scm.com/>) em conjunto com algum servidor como o GitHub (<https://github.com/>) ou o Bitbucket (<https://bitbucket.org/>) para hospedar e gerenciar seus projetos, deve saber que por padrão a conexão com esses servidores é feita pelo protocolo HTTPS ([https://pt.wikipedia.org/wiki/Hyper\\_Text\\_Transfer\\_Protocol\\_Secure](https://pt.wikipedia.org/wiki/Hyper_Text_Transfer_Protocol_Secure)). Isso obriga você a digitar usuário e senha toda vez que vai executar um comando como `git pull` ou `git push`.

Usando o protocolo SSH ([https://pt.wikipedia.org/wiki/Secure\\_Shell](https://pt.wikipedia.org/wiki/Secure_Shell)), você pode se conectar a servidores remotos e se autenticar para utilizar seus serviços. Tanto o GitHub como o Bitbucket permitem que o Git se conecte a seus servidores via SSH em vez de HTTPS. A conexão feita com criptografia de chaves dispensa o fornecimento de usuário e senha para cada comando.

Veremos nesse *post* como utilizar o GitHub e o Bitbucket com chaves SSH.



(/files/2017/04/git-

Copyright © 2018 Antônio Vinícius. Todos os direitos reservados. Baseado no Material Site template for Jekyll PLUS (<https://github.com/vinyanalista/material-jekyll-plus>).

Autor

## Verifique se há um cliente SSH instalado

Para utilizar o protocolo SSH, você precisa ter um cliente SSH instalado no seu computador. No openSUSE (<https://www.opensuse.org/>), ele já vem instalado por padrão.

Por desencargo de consciência, apenas para verificar, abra o terminal e execute:

```
$ ssh -V
```

Esse comando deve retornar informações sobre a versão do cliente SSH instalado:

```
OpenSSH_7.2p2, OpenSSL 1.0.2j-fips 26 Sep 2016
```

Caso o sistema informe que o comando **ssh** não foi encontrado, você pode instalar o cliente do OpenSSH (<https://www.openssh.com/>) executando:

```
# zypper in openssh
```

## Verifique se você já possui um par de chaves SSH

Para utilizar o protocolo SSH, você precisa gerar um par de chaves SSH (uma pública e uma privada). Se você nunca utilizou o SSH, pode pular esse tópico e seguir para o próximo. Se você já o utilizou alguma vez (como em um *post* anterior, em que fizemos acesso remoto a um servidor com o openSUSE (</pt/2017/02/18/levante-um-servidor-com-o-linux-opensuse-leap-parte-2/>)), é possível que já tenha um par de chaves SSH. Nesse caso, pode não ser necessário gerar um novo par de chaves.

Como não custa verificar, abra o terminal e execute:

```
$ ls -lah ~/.ssh
```

Esse comando deve listar os arquivos da pasta `~/.ssh`, que é onde o SSH guarda seus arquivos de configuração:

```
total 40K
drwx----- 2 viny users 4,0K Abr  9 11:42 .
drwxr-xr-x 54 viny users 12K Abr  9 12:00 ..
-rw----- 1 viny users 3,2K Abr  9 10:48 id_rsa
-rw-r--r-- 1 viny users 748 Abr  9 10:48 id_rsa.pub
-rw----- 1 viny users 5,1K Abr  9 11:59 known_hosts
```

Se ele informar que a pasta `~/.ssh` não existe, não se preocupe: significa que você ainda não tem um par de chaves SSH. Nesse caso, prossiga para o próximo tópico.

Por padrão, as chaves públicas SSH são nomeadas como:

- `id_dsa.pub`;



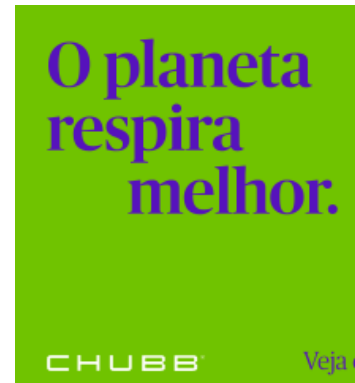
([https://secure.gravatar.com](https://secure.gravatar.com/avatar/5415f4267b44aa9f51b6f89ac4d1656b)  
/avatar

/5415f4267b44aa9f51b6f89ac4d1656b)

Antônio Vinícius



<http://www.vinyanalista.com.br>  
(<http://www.vinyanalista.com.br>)



- `id_ecdsa.pub`;
- `id_ed25519.pub`; ou
- `id_rsa.pub`.

No exemplo, a pasta `~/.ssh` possui um par de chaves SSH (a privada chamada `id_rsa` e a pública, `id_rsa.pub`) criadas hoje (Abr 9 10:48).

Por questões de segurança, é recomendado que você gere um novo par de chaves SSH ao menos uma vez por ano. Se você já possui um par de chaves SSH que foi criado há mais de ano, é recomendado que você prossiga para o próximo tópico.

Se você já possui um par de chaves SSH e quer reutilizá-lo, pule o próximo tópico.

## Gere um novo par de chaves SSH

Para gerar um novo par de chaves SSH, abra o terminal e execute o seguinte comando (substitua `seuemail@exemplo.com.br` pelo seu endereço de *e-mail*):

```
$ ssh-keygen -t rsa -b 4096 -C "seuemail@exemplo.com"

Generating public/private rsa key pair.
Enter file in which to save the key (/home/seunomedei
```

O programa pergunta onde salvar a chave privada (`id_rsa`).

Pressione **Enter** para aceitar a localização padrão.

Caso você já tenha uma chave privada, ele pergunta se deve sobrescrevê-la:

```
/home/seunomedeusuario/.ssh/id_rsa already exists.
Overwrite (y/n)?
```

Nesse caso, digite `y` e tecla **Enter**.

Em seguida, você deve digitar e confirmar uma frase-senha (<https://pt.wikipedia.org/wiki/Frase-passe>) (do inglês *passphrase*, entenda como "uma senha longa"):

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

Digite a frase-senha e tecla **Enter**. Depois, faça isso de novo para confirmá-la.

O programa cria o par de chaves SSH em `~/.ssh`.

Toda a interação deve gerar algo parecido com:

```

seunomedeusuario@seucomputador:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/seunomedeusuario/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/seunomedeusuario/.ssh/id_rsa
Your public key has been saved in /home/seunomedeusuario/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:aUAiURCvurcsQx98Ber80zMLqwYlWup1aFD5JAU6X9U seunomedeusuario@seucomputador
The key's randomart image is:
+---[RSA 4096]-----+
|  =* * . . . .      |
|  . * =. . E        |
|  o . * . o         |
| ..=0.. o .         |
| .=*.. . S          |
|+o O o .           |
|= +.=              |
|o=..o=             |
|o=+o o*            |
+----[SHA256]-----+org/product
seunomedeusuario@seucomputador:~$

```

## Adicione a chave privada ao agente SSH

Se você não quiser digitar sua frase-senha toda vez em que for se conectar via SSH, você pode adicionar sua chave privada ao agente SSH (<https://pt.wikipedia.org/wiki/Ssh-agent>), responsável por gerenciar as chaves SSH e armazenar suas frases-senha de forma segura.

Para isso, inicie o agente SSH em *background*:

```
$ eval "$(ssh-agent -s)"
```

A saída desse comando é o identificador do processo ([https://pt.wikipedia.org/wiki/Identificador\\_de\\_processo](https://pt.wikipedia.org/wiki/Identificador_de_processo)) (*process identifier* ou **PID**) do agente SSH:

```
Agent pid 21201
```

Finalmente, adicione sua chave privada SSH ao agente:

```
$ ssh-add ~/.ssh/id_rsa
```

Digite a frase-senha e tecle **Enter**:

```
Enter passphrase for /home/seunomedeusuario/.ssh/id_rsa:
```

O comando confirma que a chave foi adicionada ao agente:

```
Identity added: /home/seunomedeusuario/.ssh/id_rsa (.ssh/id_rsa)
```

Feito isso, você já pode configurar a conexão via SSH com o GitHub ou o Bitbucket.

# Adicione sua chave SSH à sua conta do GitHub

Para se conectar ao GitHub utilizando SSH, você deve associar sua chave pública à sua conta do GitHub.

Sua chave pública está no arquivo `~/.ssh/id_rsa.pub`.

Para copiá-la, utilize o comando **xclip**:

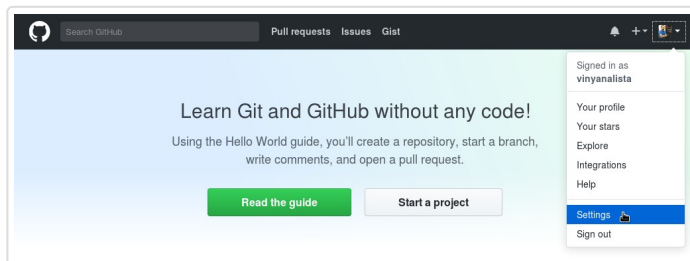
```
$ xclip -sel clip < ~/.ssh/id_rsa.pub
```

O **xclip** (<https://github.com/astrand/xclip>) é um utilitário que permite, via interface textual, acessar a área de transferência (*clipboard*) ([https://pt.wikipedia.org/wiki/%C3%81rea\\_de\\_transfer%C3%AAncia](https://pt.wikipedia.org/wiki/%C3%81rea_de_transfer%C3%AAncia)) da interface gráfica.

Caso ele não esteja instalado no seu sistema, você pode instalá-lo executando:

```
# zypper install xclip
```

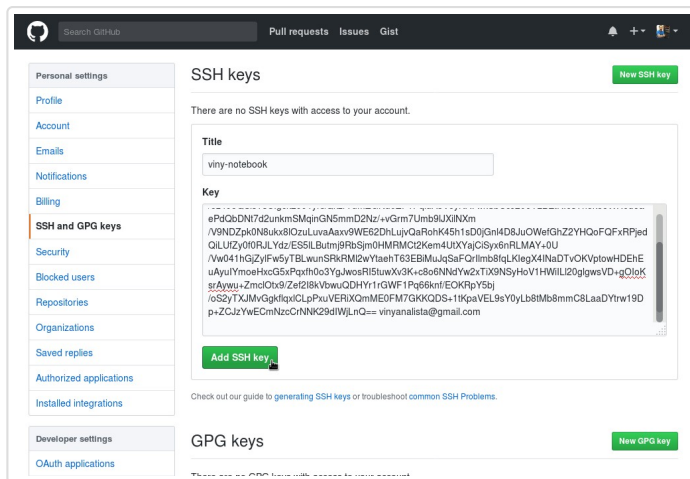
Usando o navegador, acesse a página inicial do GitHub em [github.com](https://github.com/) (<https://github.com/>) e entre na sua conta clicando em **Sign in**. No canto superior direito da página, clique na sua foto de perfil e depois em **Settings** (configurações):



(/files/2017/04/git-ssh-01.jpg)

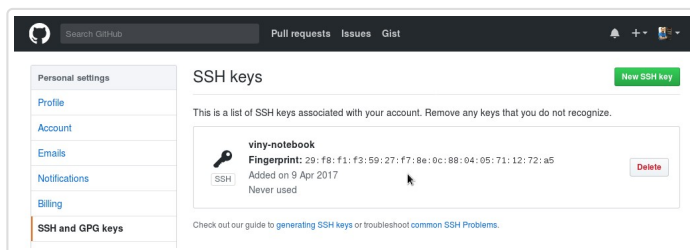
Na barra lateral, clique em **SSH and GPG keys** (chaves SSH e GPG ([https://pt.wikipedia.org/wiki/GNU\\_Privacy\\_Guard](https://pt.wikipedia.org/wiki/GNU_Privacy_Guard))). Depois, clique em **New SSH key** (nova chave SSH).

Preencha o campo **Title** (título) com um nome descritivo para a nova chave (pode ser, por exemplo, o nome do seu computador) e cole sua chave pública no campo **Key** (chave). Finalmente, clique em **Add SSH key** (adicionar chave SSH):



(/files/2017/04/git-ssh-02.jpg)

Agora a chave aparece na lista de chaves SSH associadas à conta:



(/files/2017/04/git-ssh-03.jpg)

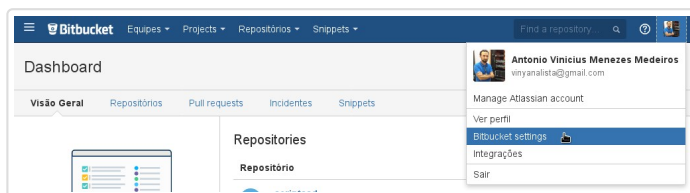
## Adicione sua chave SSH à sua conta do Bitbucket

O processo para o Bitbucket é bastante semelhante.

Copie sua chave pública utilizando o comando **xclip**:

```
$ xclip -sel clip < ~/.ssh/id_rsa.pub
```

Usando o navegador, acesse a página inicial do Bitbucket em [bitbucket.org](https://bitbucket.org/) (<https://bitbucket.org/>) e entre na sua conta clicando em **Log in**. No canto superior direito da página, clique na sua foto de perfil e depois em **Bitbucket settings** (configurações do Bitbucket):

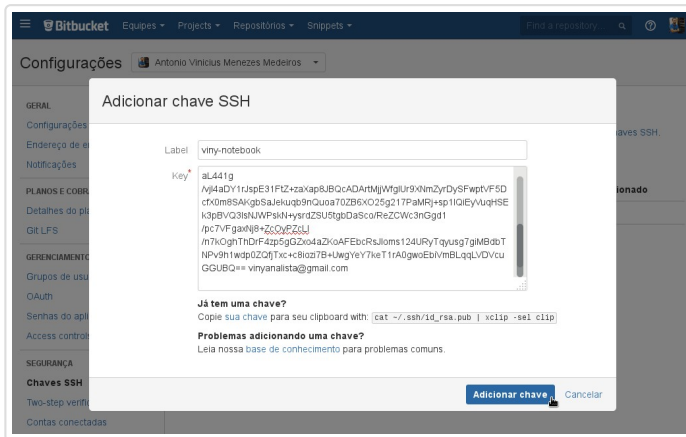


(/files/2017/04/git-ssh-04-pt.jpg)

Note que a interface do Bitbucket é traduzida, mas apenas em parte.

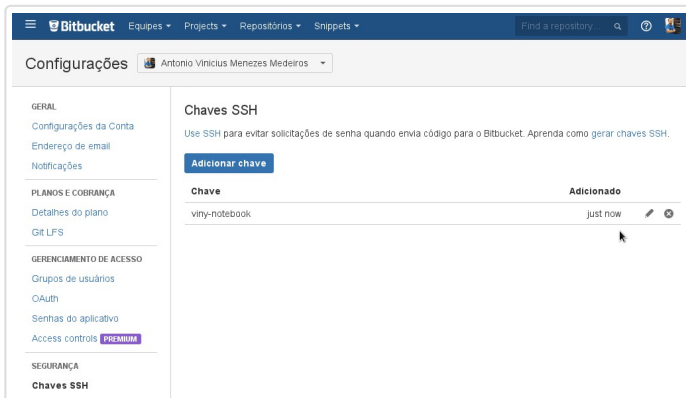
Na barra lateral, na seção **Segurança**, clique em **Chaves SSH**. Depois, clique em **Adicionar chave**.

Preencha o campo **Label** (rótulo) com um nome descritivo para a nova chave (pode ser, por exemplo, o nome do seu computador) e cole sua chave pública no campo **Key** (chave). Finalmente, clique em **Adicionar chave**:



(/files/2017/04/git-ssh-05-pt.jpg)

Agora a chave aparece na lista de chaves SSH associadas à conta:



(/files/2017/04/git-ssh-06-pt.jpg)

## Teste a conexão via SSH

Tanto no GitHub quanto no Bitbucket, é possível testar a conexão antes de usá-la propriamente com o Git.

Se você adicionou sua chave SSH à sua conta do GitHub, abra o terminal e execute:

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

Com esse comando, você vai tentar acessar remotamente o servidor do GitHub.

Se você ainda não se conectou ao GitHub via SSH, o cliente SSH pergunta se pode confiar na chave fornecida pelo servidor do GitHub:

```
The authenticity of host 'github.com (192.30.253.112)'
RSA key fingerprint is SHA256:nThbg6kXUpJWGL7E1IG0CsI
Are you sure you want to continue connecting (yes/no)
```

Nesse caso, digite `yes` e tecla **Enter**. Essa pergunta só é feita na primeira vez. O cliente SSH adiciona o servidor do GitHub à lista de computadores conhecidos:

```
Warning: Permanently added 'github.com,192.30.253.11':
```

Como essa conexão é apenas para teste (o GitHub fornece controle de versão com Git, não acesso remoto via SSH), o servidor informa que você conseguiu se autenticar, mas encerra o acesso remoto via SSH:

```
Hi seunomedeusuario! You've successfully authenticat
```

Teste concluído com sucesso, você já pode utilizar o SSH com o GitHub.

Se você adicionou sua chave SSH à sua conta do Bitbucket, o teste é semelhante:

```
$ ssh -T git@bitbucket.org
The authenticity of host 'bitbucket.org (104.192.143
RSA key fingerprint is SHA256:zzXQ0XSRBEiUtuE8AikJYK
Are you sure you want to continue connecting (yes/no
Warning: Permanently added 'bitbucket.org,104.192.14
logged in as seunomedeusuario.

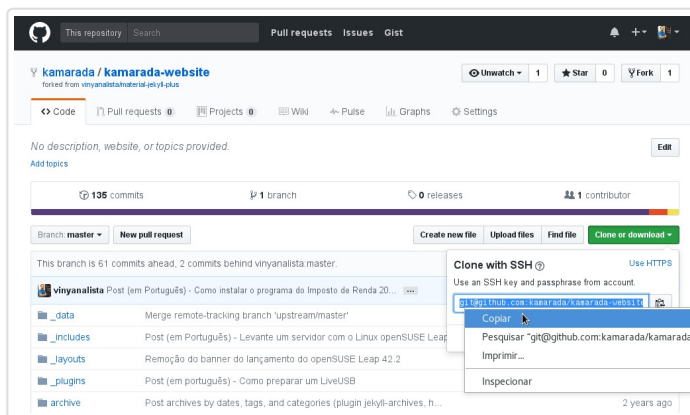
You can use git or hg to connect to Bitbucket. Shell
```

Teste efetuado com sucesso, você já pode utilizar o SSH com o Bitbucket.

## Clone um repositório via SSH

Agora que sabemos que podemos conectar ao GitHub ou ao Bitbucket via SSH, vejamos como clonar um repositório utilizando o SSH em vez do HTTPS.

No GitHub, acesse o repositório de um projeto, clique em **Clone or download** e copie o URL (<https://pt.wikipedia.org/wiki/URL>) para clonar o repositório usando SSH:



(/files/2017/04/git-ssh-07-pt.jpg)

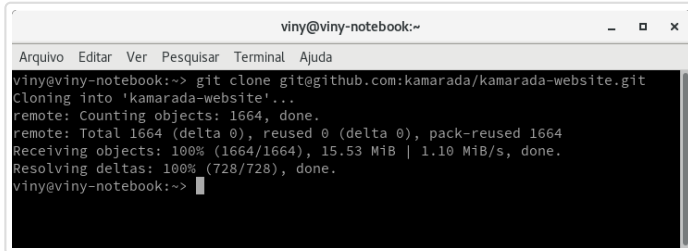
O URL de um repositório do GitHub se parece com:



```
git@github.com:seunomedeusuario/nomedoseuprojeto.git
```

Abra o terminal e execute o comando `git clone` passando o URL copiado (dica: para colar no terminal, utilize a combinação de teclas **Ctrl + Shift + V**).

Observe que o Git clona o repositório via SSH sem pedir senha:

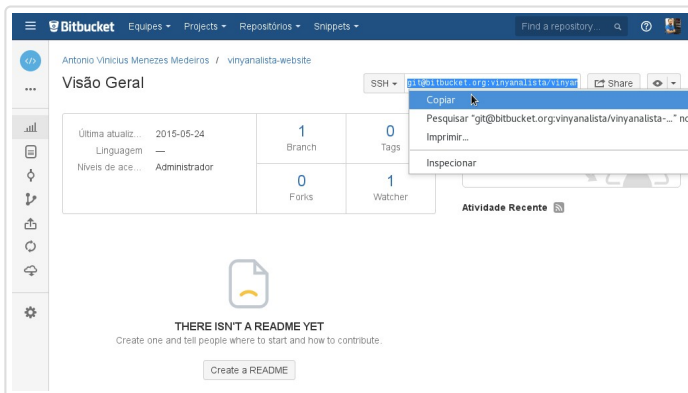


```
viny@viny-notebook:~$ git clone git@github.com:kamarada/kamarada-website.git
Cloning into 'kamarada-website'...
remote: Counting objects: 1664, done.
remote: Total 1664 (delta 0), reused 0 (delta 0), pack-reused 1664
Receiving objects: 100% (1664/1664), 15.53 MiB | 1.10 MiB/s, done.
Resolving deltas: 100% (728/728), done.
viny@viny-notebook:~$
```

(/files/2017/04/git-ssh-08-pt.png)

O processo é semelhante se você utiliza o Bitbucket.

No Bitbucket, acesse o repositório de um projeto e copie o URL para clonar o repositório usando SSH:

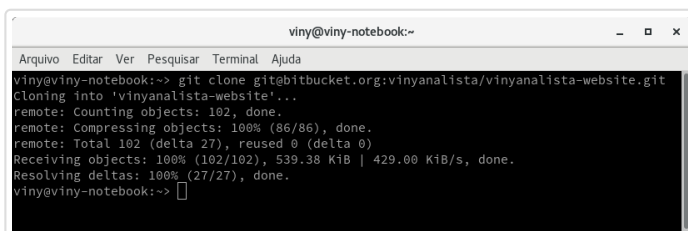


(/files/2017/04/git-ssh-09-pt.jpg)

O URL de um repositório do Bitbucket se parece com:

```
git@bitbucket.org:seunomedeusuario/nomedoseuprojeto.git
```

Abra o terminal e execute o comando `git clone` passando o URL copiado:



```
viny@viny-notebook:~$ git clone git@bitbucket.org:vinyanalista:vinyanalista-website.git
Cloning into 'vinyanalista-website'...
remote: Counting objects: 102, done.
remote: Compressing objects: 100% (86/86), done.
remote: Total 102 (delta 27), reused 0 (delta 0)
Receiving objects: 100% (102/102), 539.38 KiB | 429.00 KiB/s, done.
Resolving deltas: 100% (27/27), done.
viny@viny-notebook:~$
```

(/files/2017/04/git-ssh-10-pt.png)

Com o Bitbucket, o Git também clona o repositório via SSH sem pedir senha.

## Mude um repositório para SSH

Os repositórios que acabamos de clonar via SSH seguirão utilizando o protocolo SSH para futuros comandos do Git como `git pull` e `git push`. Mas repositórios clonados anteriormente com HTTPS seguirão utilizando o protocolo HTTPS.

Também podemos configurar esses repositórios para utilizar SSH.

Para fazer isso, abra o terminal e mude para a pasta do seu repositório local.

Liste os repositórios remotos existentes e seus URLs com o comando:

```
$ git remote -v
```

Sua saída deve ser parecida com:

```
origin  https://github.com/seunomedeusuario/nomedosei
origin  https://github.com/seunomedeusuario/nomedosei
```

Mude o URL do seu repositório remoto com o comando:

```
git remote set-url origin git@github.com:seunomedeusu
```

Execute o comando `git remote -v` mais uma vez para verificar que o URL do repositório remoto foi alterado:

```
origin  git@github.com:seunomedeusuario/nomedoseupro:
origin  git@github.com:seunomedeusuario/nomedoseupro:
```

Pronto. Feito isso, o Git passará a utilizar o SSH em vez do HTTPS para sincronizar esse repositório local com o seu equivalente remoto.

Espero que essas dicas possam ser úteis para você como têm sido para mim desde que comecei a usar o Git. Se ficou com dúvida ou algo deu errado, não deixe de comentar! Até a próxima!

## Referências

- Connecting to GitHub with SSH - User Documentation - GitHub (<https://help.github.com/articles/connecting-to-github-with-ssh/>)
- Set up SSH for Git - Atlassian Documentation (<https://confluence.atlassian.com/bitbucket/set-up-ssh-for-git-728138079.html>)
- Changing a remote's URL - User Documentation - GitHub (<https://help.github.com/articles/changing-a-remote-s-url/>)

## Gostou? Que tal compartilhar?

19

## Comentários

**Comentários** **Comunidade** **1 Iniciar sessão** ▼

 **Recomendar**  **Partilhar**

**Mostrar primeiro os mais votados** ▼

**INICIE SESSÃO COM O** **OU REGISTE-SE NO DISQUS** 

Seja o primeiro a comentar!

TAMBÉM NO LINUX KAMARADA

### openSUSE Leap 42.2: versão ótima para

1 COMENTÁRIO • há um ano



**Marcos** — openSUSE  
42.2 .. Ótimo e estável  
..Excelente sua

### openSUSE Leap 42.1 se torna a primeira

1 COMENTÁRIO • há 2 anos



**Qwide** — O Leap está  
incrível! Me fez abandonar  
até o dual boot.

### Mantenha seu sistema sempre atualizado

1 COMENTÁRIO • há 2 anos



**Rafael Flores** — Muito  
bom, bem simples e

### Levante um servidor com o Linux openSUSE Leap

1 COMENTÁRIO • há um ano



**J4** — Parabéns!!! Os  
usuários do opensuse