



Guia prático para iniciantes



# Igor Santos

- Bacharel em Sistemas de Informação pela ULBRA
- FullStack Web Developer – Navalha Digital
- PHP Developer há 4 anos
- Coordenador da Comunidade PHPRS
- Gamer nas horas vagas ;D



# Provavelmente você já se deparou com uma dessas situações no seu dia-a-dia de trabalho:

- Dificuldade em determinar a versão mais recente do seu código;
- Colegas de equipe com versões de código diferente;
- Sobescrita de códigos;
- Perca de tempo para unificar trabalho;
- Não é possível regredir o projeto para uma versão anterior que se encontrava ao encontrar um bug.

# O que o versionamento nos permite?

- Gerenciar histórico dos arquivos;
- Fácil colaboração durante o desenvolvimento;
- Versões estáveis do código;
- Criar novas funcionalidades sem influenciar o código em funcionamento.

# Mas porque usar o Git?

## **Ponto de vista profissional**

- Grande popularidade ao Github;
- Grande maioria dos projetos open source hoje em dia se encontram no GitHub;
- Foi criado por nada menos que Linus Torvalds o criador do Linux em 2005;
- Fácil aprendizagem;
- Hoje em dia dominar o Git é uma habilidade essencial para desenvolvedores.

# Mas porque usar o Git?

## **Ponto de vista técnico**

- Rapidez;
- Autonomia;
- Confiabilidade;
- Controle de histórico;
- Ramificação do projeto;
- Facilidade de mesclagem;
- Aumento de produtividade.

# Git for Windows

Bom, a maneira mais comum de usar o Git é pela linha de comando, Acessível através de um terminal.

É o jeito que a maior parte dos bons profissionais do mercado utiliza o Git e será nossa escolha aqui.

Particularmente recomendo o Cmder, caso se interessar acesse:

➤ **<https://link.cmder>**

λcmder

Para isso utilizaremos o Git para Windows.



# Instalando o Git no Windows

➤ <https://git-for-windows.github.io/>







git

Como Usar

# Gerando Chave SSH (ssh key)

Tem como objetivo fazer conexões automáticas no gitlab usando o protocolo SSH sem a necessidade de informar senha

Para gerar a chave ssh digite no terminal o comando abaixo, substituindo o email, pelo seu email registrado no perfil do gitlab

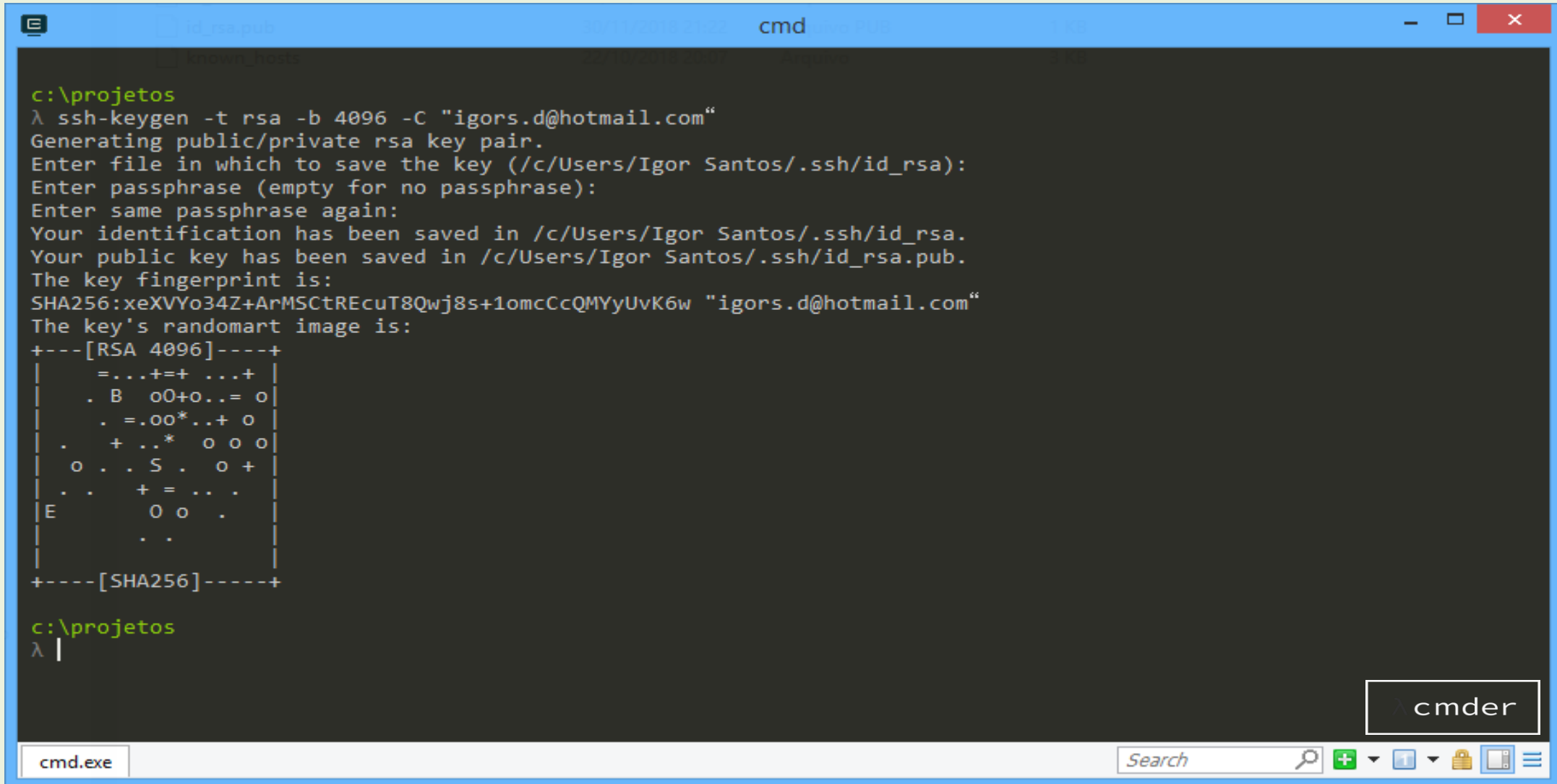
```
ssh-keygen -t rsa -b 4096 -C "meu-email@email.com.br"
```

Normalmente as chaves são geradas no diretório:

```
C:\Users\seu.usuario\.ssh\
```



# Gerando Chave SSH (ssh key)



```
c:\projetos
λ ssh-keygen -t rsa -b 4096 -C "igors.d@hotmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c:/Users/Igor Santos/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c:/Users/Igor Santos/.ssh/id_rsa.
Your public key has been saved in /c:/Users/Igor Santos/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:xeXVYo34Z+ArMSCtREcuT8Qwj8s+1omcCcQMYyUvK6w "igors.d@hotmail.com"
The key's randomart image is:
+---[RSA 4096]---+
|      =...+=+ ...+ |
|      . B  o0+o..= o |
|      . =.oo*..+ o  |
|      . + ..*  o o o |
|      o . . S .  o + |
|      . .   + = .. . |
| E      O o   .      |
|      . .              |
+-----[SHA256]-----+

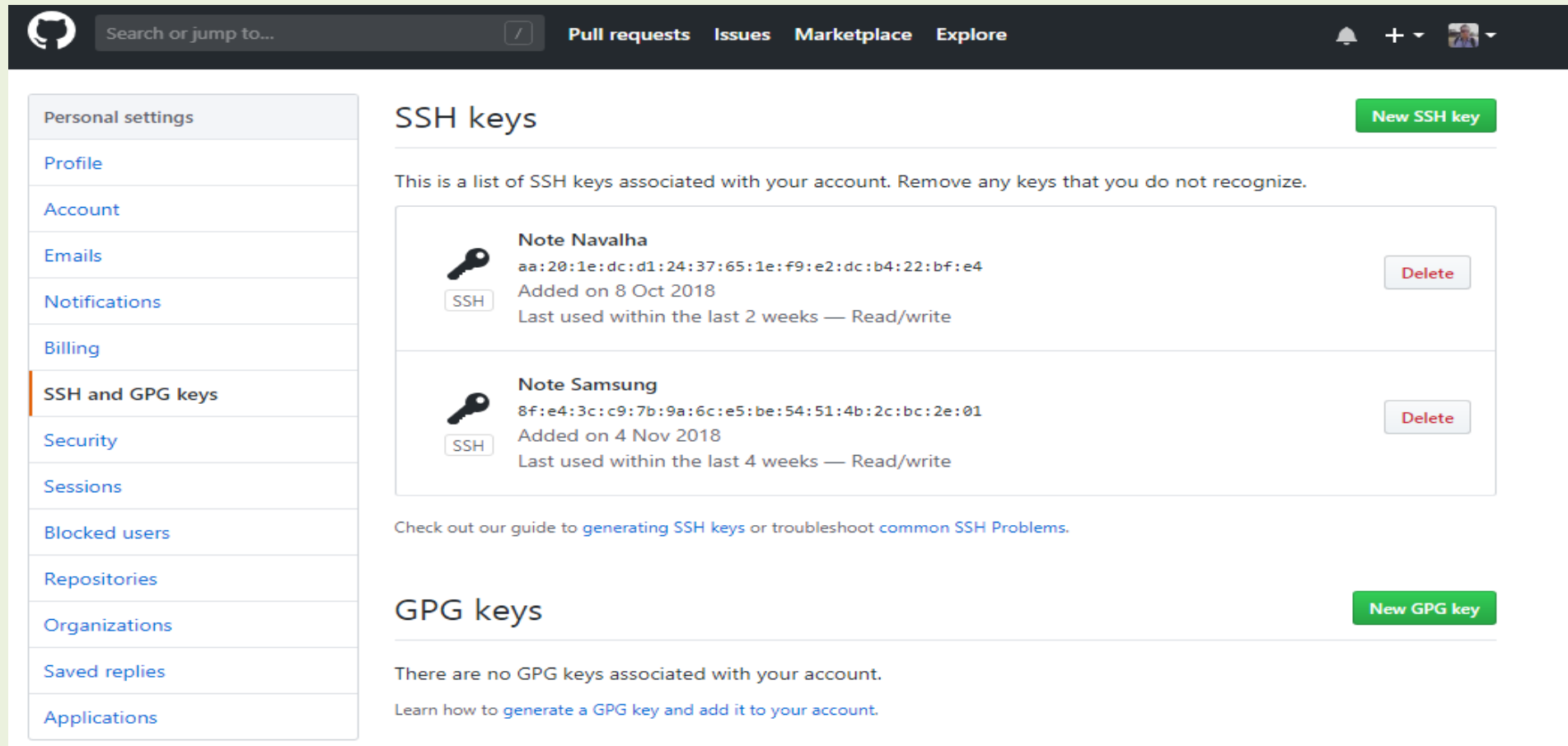
c:\projetos
λ |
```

The screenshot shows a Windows Command Prompt window titled "cmd" with a blue header bar. The window displays the output of the `ssh-keygen` command. The user is in the `c:\projetos` directory. The command `ssh-keygen -t rsa -b 4096 -C "igors.d@hotmail.com"` is executed. The output shows the generation of an RSA key pair, the file names `/c:/Users/Igor Santos/.ssh/id_rsa` and `/c:/Users/Igor Santos/.ssh/id_rsa.pub`, the key fingerprint, and a randomart image for the RSA key. The randomart image is a square grid of characters representing the key's fingerprint. The window's taskbar at the bottom shows the "cmd.exe" process, a search bar, and several system icons. A yellow key icon with "SSH" written on it is located in the bottom right corner of the image.

# Gerando Chave SSH (ssh key)

you need to access the file `id_rsa.pub`, copy all  
Your content and insert it on the keys page in gitlab



To access the keys page go to **Settings** and then **SSH Key**



The screenshot shows the GitLab user interface. At the top is a dark navigation bar with the GitLab logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. Below this is a sidebar with a list of settings: Personal settings, Profile, Account, Emails, Notifications, Billing, SSH and GPG keys (highlighted with an orange bar), Security, Sessions, Blocked users, Repositories, Organizations, Saved replies, and Applications. The main content area is titled 'SSH keys' and includes a 'New SSH key' button. It contains a list of two SSH keys: 'Note Navalha' and 'Note Samsung'. Each key entry shows a key icon, the key name, the public key text, the date added, and the last used status. A 'Delete' button is present for each key. Below the list is a link to a guide on generating SSH keys. At the bottom of the main area is a section for 'GPG keys' with a 'New GPG key' button and a message stating there are no GPG keys associated with the account, along with a link to learn how to generate one.

SSH keys New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

	<b>Note Navalha</b> aa:20:1e:dc:d1:24:37:65:1e:f9:e2:dc:b4:22:bf:e4 Added on 8 Oct 2018 Last used within the last 2 weeks — Read/write	<span>Delete</span>
	<b>Note Samsung</b> 8f:e4:3c:c9:7b:9a:6c:e5:be:54:51:4b:2c:bc:2e:01 Added on 4 Nov 2018 Last used within the last 4 weeks — Read/write	<span>Delete</span>

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

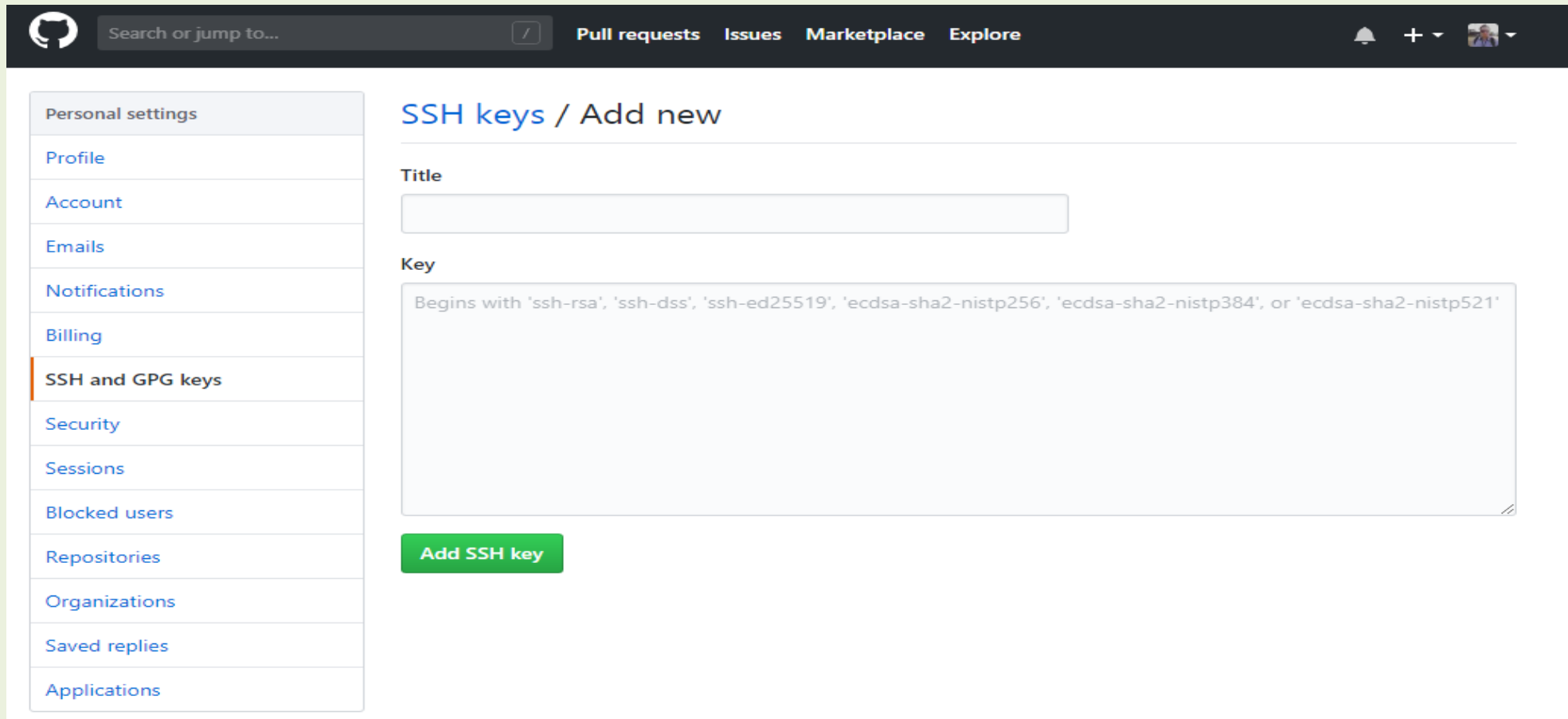
GPG keys New GPG key

There are no GPG keys associated with your account.  
Learn how to [generate a GPG key and add it to your account](#).

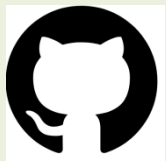


# Gerando Chave SSH (ssh key)

Basta inserir uma identificação para a sua chave e a “key”



The screenshot shows the GitHub interface for adding a new SSH key. On the left is a sidebar with navigation links: Personal settings, Profile, Account, Emails, Notifications, Billing, SSH and GPG keys (highlighted with an orange bar), Security, Sessions, Blocked users, Repositories, Organizations, Saved replies, and Applications. The main content area is titled 'SSH keys / Add new'. It contains a 'Title' text input field and a 'Key' text area. The 'Key' area has a placeholder text: 'Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521''. Below the key input is a green button labeled 'Add SSH key'.



# Obtenha um Repositório

crie uma cópia de trabalho em um repositório local  
executando o comando

```
git clone /caminho/para/o/repositório
```

# Fluxo de Trabalho

seus repositórios locais consistem em três "árvores" mantidas pelo git.  
a primeira delas é sua **Working Directory** que contém os arquivos vigentes.  
a segunda **Index** que funciona como uma área temporária e finalmente  
a **Head** que aponta para o último **commit** (confirmação) que você fez.



# Fluxo de Trabalho

Você pode propor mudanças (adicioná-las ao **Index**) usando

```
git add <arquivo>
```

```
git add . / git add *
```

Este é o primeiro passo no fluxo de trabalho básico do git.  
Para realmente confirmar estas mudanças  
(isto é, fazer um *commit*), use

```
git commit -m "comentários das alterações"
```

Agora o arquivo é enviado para o **HEAD**,  
mas ainda não para o repositório remoto.



# Enviando Alterações

Suas alterações agora estão no **HEAD** da sua cópia de trabalho local.  
Para enviar estas alterações ao seu repositório remoto, execute

```
git push origin master
```

Altere *master* para qualquer ramo (*branch*) desejado,  
enviando suas alterações para ele.

# Ramificando

*Branches* ("ramos") são utilizados para desenvolver funcionalidades isoladas umas das outras. O branch *master* é o branch "padrão" quando você cria um repositório. Use outros branches para desenvolver e mescle-os (*merge*) o branchmaster após a conclusão.



# Ramificando

crie um novo branch chamado "funcionalidade\_x" e selecione-o usando

```
git checkout -b funcionalidade_x
```

retorne para o master usando

```
git checkout master
```

e remova o branch da seguinte forma

```
git branch -d funcionalidade_x
```

um branch *não está disponível a outros* a menos que você envie o branch para seu repositório remoto

```
git push origin <funcionalidade_x>
```

# Atualizar & Mesclar

para atualizar seu repositório local com a mais nova versão, execute

```
git pull
```

na sua pasta de trabalho para *obter e fazer merge* (mesclar) alterações remotas.  
para fazer merge de um outro branch ao seu branch ativo (ex. master), use

```
git merge <branch>
```

em ambos os casos o git tenta fazer o merge das alterações automaticamente.

Infelizmente, isto nem sempre é possível e resulta em *conflitos*.

Você é responsável por fazer o merge estes *conflitos* manualmente editando os arquivos exibidos pelo git. Depois de alterar, você precisa marcá-los como merged com

```
git add <arquivo>
```

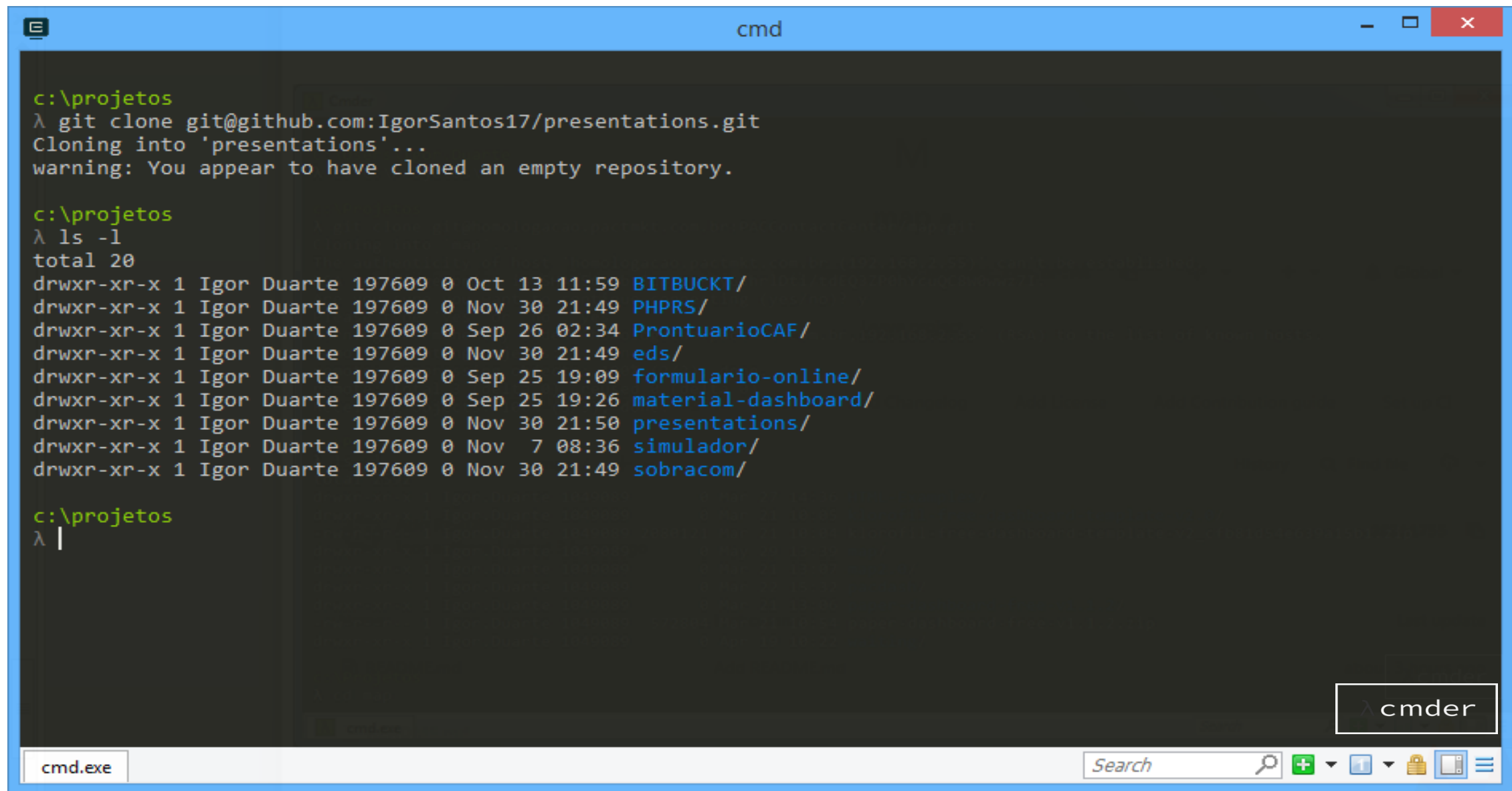
antes de fazer o merge das alterações, você pode também pré-visualizá-as usando

```
git diff <branch origem> <branch destino>
```



git

Git na prática



```
c:\projetos
λ git clone git@github.com:IgorSantos17/presentations.git
Cloning into 'presentations'...
warning: You appear to have cloned an empty repository.

c:\projetos
λ ls -l
total 20
drwxr-xr-x 1 Igor Duarte 197609 0 Oct 13 11:59 BITBUCKET/
drwxr-xr-x 1 Igor Duarte 197609 0 Nov 30 21:49 PHPRS/
drwxr-xr-x 1 Igor Duarte 197609 0 Sep 26 02:34 ProntuarioCAF/
drwxr-xr-x 1 Igor Duarte 197609 0 Nov 30 21:49 eds/
drwxr-xr-x 1 Igor Duarte 197609 0 Sep 25 19:09 formulario-online/
drwxr-xr-x 1 Igor Duarte 197609 0 Sep 25 19:26 material-dashboard/
drwxr-xr-x 1 Igor Duarte 197609 0 Nov 30 21:50 presentations/
drwxr-xr-x 1 Igor Duarte 197609 0 Nov 7 08:36 simulador/
drwxr-xr-x 1 Igor Duarte 197609 0 Nov 30 21:49 sobracom/

c:\projetos
λ |
```

cmd.exe

Search

cmdr

```
c:\projetos
λ cd presentations\

c:\projetos\presentations (master)
λ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    "24-07-18-PHPRS_Certifica\303\247\303\243o_PHP.pptx"

nothing added to commit but untracked files present (use "git add" to track)
c:\projetos\presentations (master)
λ git add .

c:\projetos\presentations (master)
λ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   "24-07-18-PHPRS_Certifica\303\247\303\243o_PHP.pptx"

c:\projetos\presentations (master)
λ git commit -m "add apresentacao do meetup PHPRS - Certificacoes"
[master (root-commit) 69afdfc] add apresentacao do meetup PHPRS - Certificacoes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "24-07-18-PHPRS_Certifica\303\247\303\243o_PHP.pptx"

c:\projetos\presentations (master)
λ |
```

cmd.exe

Search

```
c:\projetos\presentations (master)
λ git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)
nothing to commit, working tree clean

c:\projetos\presentations (master)
λ git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 5.49 MiB | 297.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/IgorSantos17/presentations/pull/new/master
remote:
To github.com:IgorSantos17/presentations.git
 * [new branch]      master -> master

c:\projetos\presentations (master)
λ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean

c:\projetos\presentations (master)
λ |
```

cmd.exe

Search

cmd.exe



[Pull requests](#)[Issues](#)[Marketplace](#)[Explore](#)[IgorSantos17 / presentations](#)[Unwatch](#)

1

[★ Star](#)

0

[Fork](#)

0

[Code](#)[Issues 0](#)[Pull requests 0](#)[Projects 0](#)[Wiki](#)[Insights](#)[Settings](#)

Apresentações, Palestras e Talks por aí

[Edit](#)[Manage topics](#)[3 commits](#)[1 branch](#)[0 releases](#)[1 contributor](#)[Branch: master](#)[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download](#)[IgorSantos17 update readme](#)[Latest commit 816a0f2 just now](#)[24-07-18-PHPRS\\_Certificação\\_PHP.pptx](#)[add apresentacao do meetup PHPRS - Certificacoes](#)[27 minutes ago](#)[README.md](#)[update readme](#)[just now](#)[README.md](#)

# Talks por Aí by Igor Santos

Nesse repositório estão todas palestras, apresentações e Talks que já fiz por aí.



# Onde Me Achar



[twitter.com/IgorSantoos17](https://twitter.com/IgorSantoos17)



[linkedin.com/in/igorsantoos](https://linkedin.com/in/igorsantoos)



[github.com/IgorSantos17](https://github.com/IgorSantos17)



[medium.com/@igorsantos17](https://medium.com/@igorsantos17)





#junteseamanada