

Controle de Versão com Git

Versão 1.0

HISTÓRICO DE VERSÕES

<i>Versão</i>	<i>Data</i>	<i>Autor da Versão</i>	<i>Alterações da Versão</i>
1.0	11/11/19	Pedro Akira Danno Lima	Documento original Controle de Versão com Git.

RESUMO DO DOCUMENTO

<i>Descrição:</i>	Este documento descreve os processos de instalação, configuração e uso do Git.
<i>Local de Publicação:</i>	DIOPE/SUPCD/CDOPC/CDMSS
<i>Validade da Versão:</i>	03/12/2019
<i>Baseado no Modelo de Publicação Versão 1.0</i>	

Sumário

Instalando o Git no Linux.....	5
Verificando a instalação.....	7
Configurando o Git.....	8
Gerando chave SSH.....	11
Adicionando a chave SSH no Git Corporativo.....	14
Copiando o conteúdo da Chave Pública.....	14
Cadastrando a Chave Pública no Git Corporativo.....	15
Adicionando a chave SSH no GitHub.....	17
Criação de repositório GitHub.....	20
Criar arquivo README.md.....	23

Sobre o controle de versão

O que é "controle de versão" e por que você deveria se importar? O controle de versão é um sistema que registra alterações em um arquivo ou conjunto de arquivos ao longo do tempo, para que você possa recuperar versões específicas mais tarde. Para os exemplos deste documento, você usará o software de código fonte como os arquivos controlados pela versão, embora, na realidade, você possa fazer isso com quase qualquer tipo de arquivo em um computador.

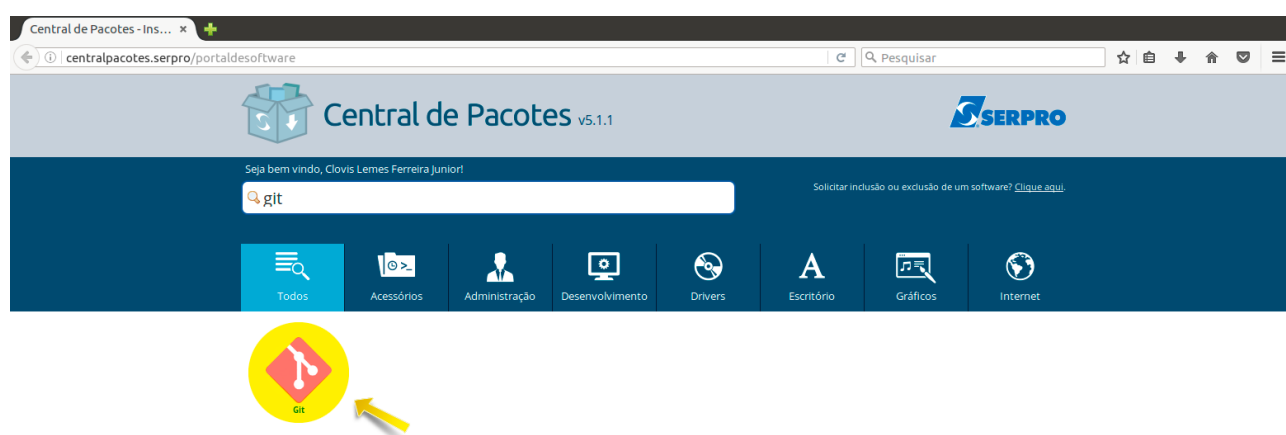
Se você é um designer gráfico ou da web e deseja manter todas as versões de uma imagem ou layout (o que você certamente desejaria), o VCS (Version Control System) é uma coisa muito sábia de se usar. Ele permite reverter os arquivos selecionados de volta para um estado anterior, reverter todo o projeto para um estado anterior, comparar as alterações ao longo do tempo, ver quem modificou pela última vez algo que pode estar causando um problema, quem apresentou um problema, quando e mais. O uso de um VCS geralmente também significa que, se você estragar tudo ou perder arquivos, poderá recuperar facilmente.

Instalando Git

Instalando o Git no Linux

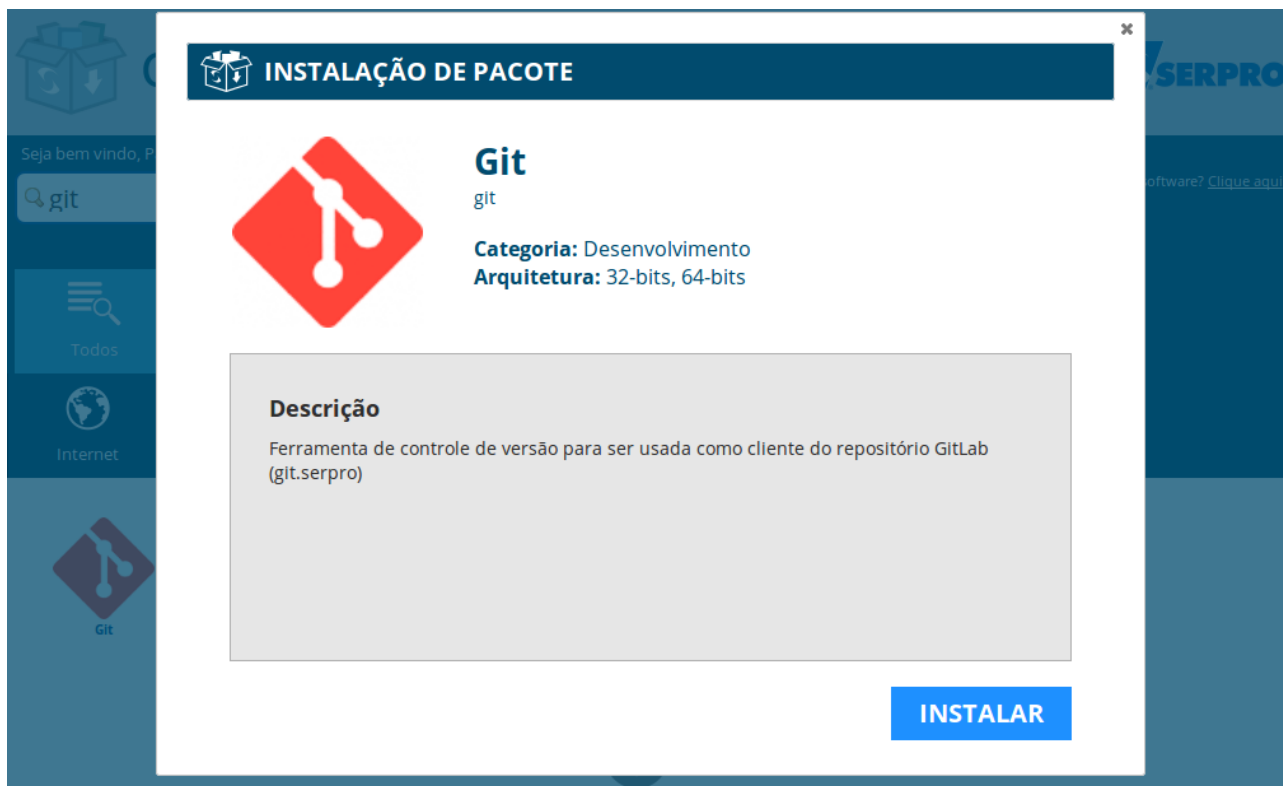
Passo 1.

Acessar [Central de Pacotes do SERPRO](http://centralpacotes.serpro.gov.br/portalsoftware) e buscar pelo termo "git".



Passo 2.

Em seguida, basta clicar sobre o ícone do "Git" e clicar no botão "Instalar".

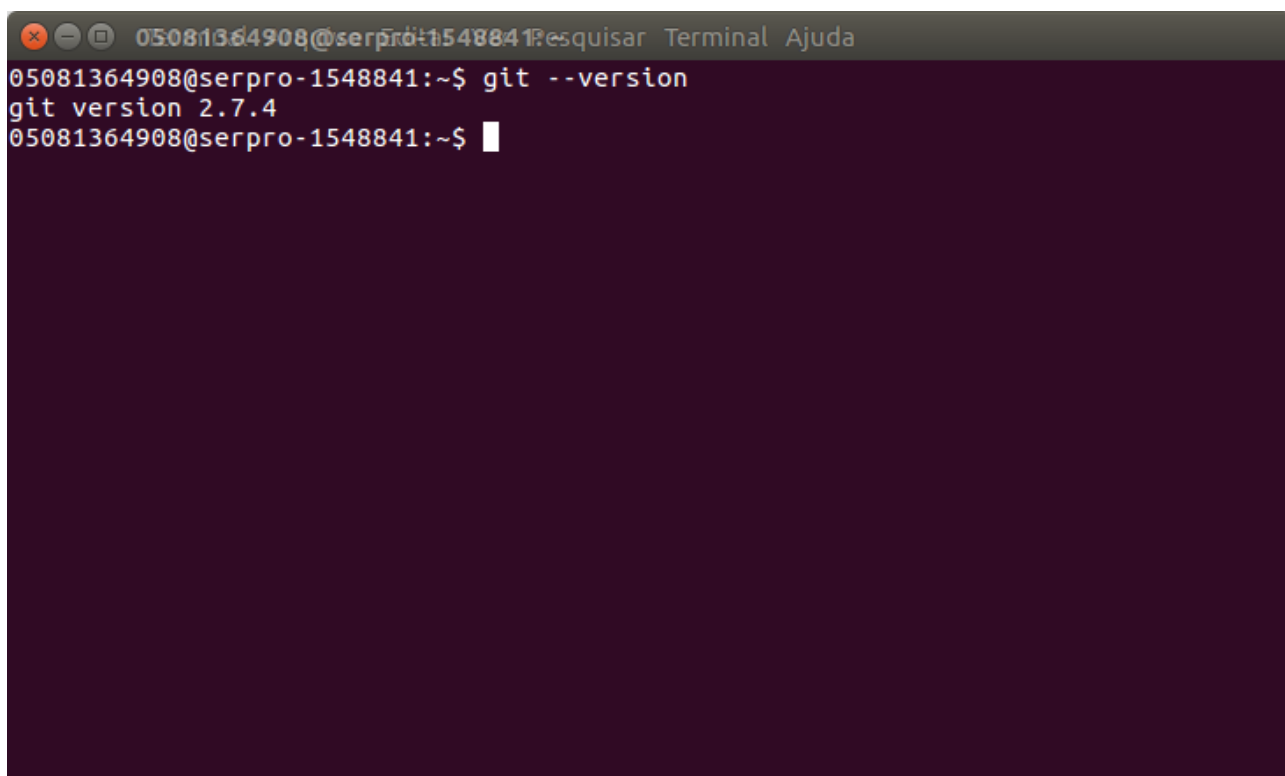


Verificando a instalação

Após a instalação do pacote Git, no terminal de comandos, digite o seguinte comando:

```
$ git --version
```

Se o pacote Git estiver instalado corretamente, você visualizará a informação sobre a versão instalada.



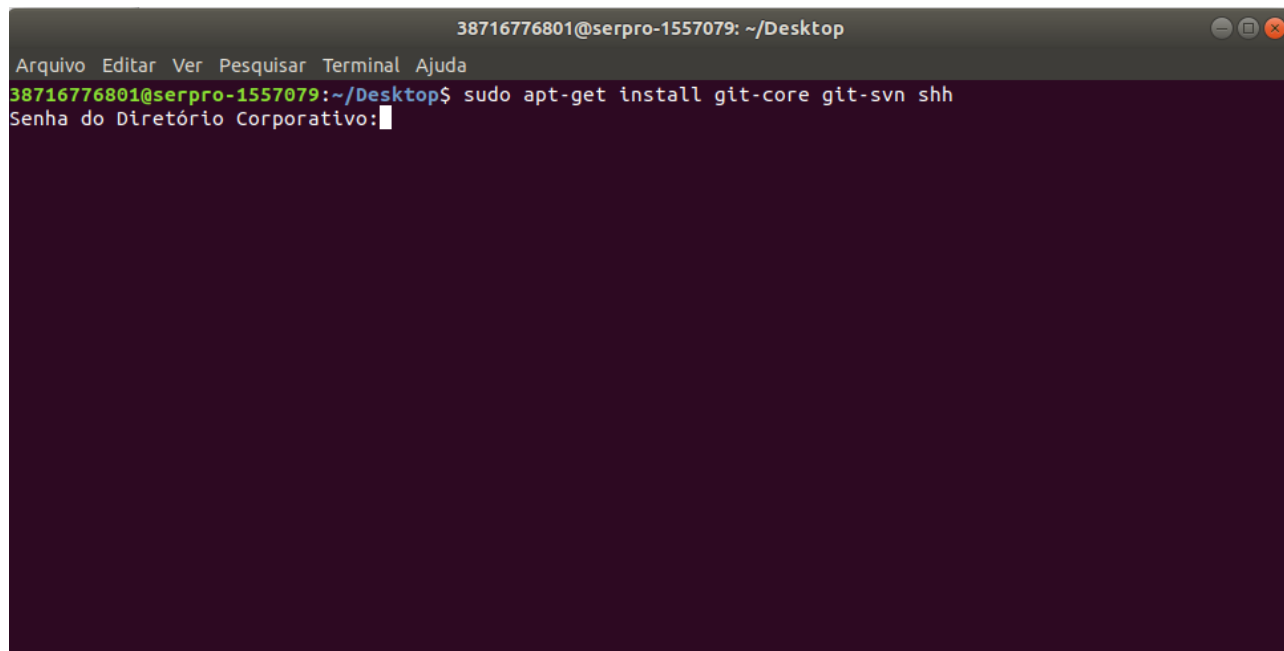
```
05081364908@serpro-1548841:~$ git --version
git version 2.7.4
05081364908@serpro-1548841:~$
```

Instalação por via de comando:

Instalação no ambiente Linux-Ubuntu:

No terminal de comandos, digite o seguinte comando:

```
$ sudo apt-get install git-core git-svn ssh
```

A screenshot of a terminal window titled "38716776801@serpro-1557079: ~/Desktop". The window has a menu bar with "Arquivo", "Editar", "Ver", "Pesquisar", "Terminal", and "Ajuda". The terminal shows the command "38716776801@serpro-1557079:~/Desktop\$ sudo apt-get install git-core git-svn ssh" being entered. Below the command, the prompt "Senha do Diretório Corporativo:" is visible with a cursor. The terminal background is dark purple.

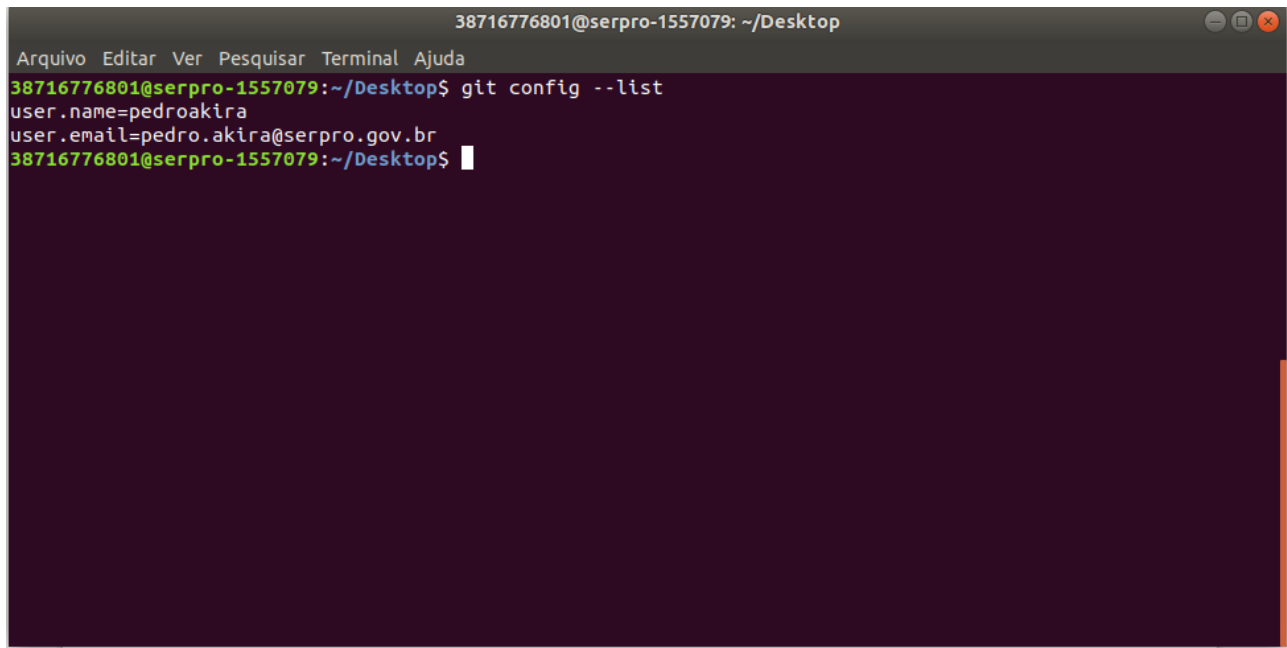
Configurando o Git

Após a instalação do Git na estação de trabalho, é necessário realizar a configuração básica para informar ao Git seu nome e email. Essas informações serão necessárias no momento de realizarmos as operações de *commit*.

Caso já tenha o Git instalado em sua estação de trabalho e já tenha configurado seu nome e email no Git mas tem dúvida das configurações atuais, você pode executar o seguinte comando:

Retorne ao terminal e insira o comando a seguir:

`$ git config --list`



```
38716776801@serpro-1557079: ~/Desktop
Arquivo Editar Ver Pesquisar Terminal Ajuda
38716776801@serpro-1557079:~/Desktop$ git config --list
user.name=pedroakira
user.email=pedro.akira@serpro.gov.br
38716776801@serpro-1557079:~/Desktop$
```

Observe o resultado do comando e caso os campos "user.name" e/ou "user.email" estejam vazios ou incorretos execute a instrução abaixo para atualizar as informações

Retorne ao terminal e insira os dois comandos a seguir:

`$ git config --global user.name "<seu nome>"`

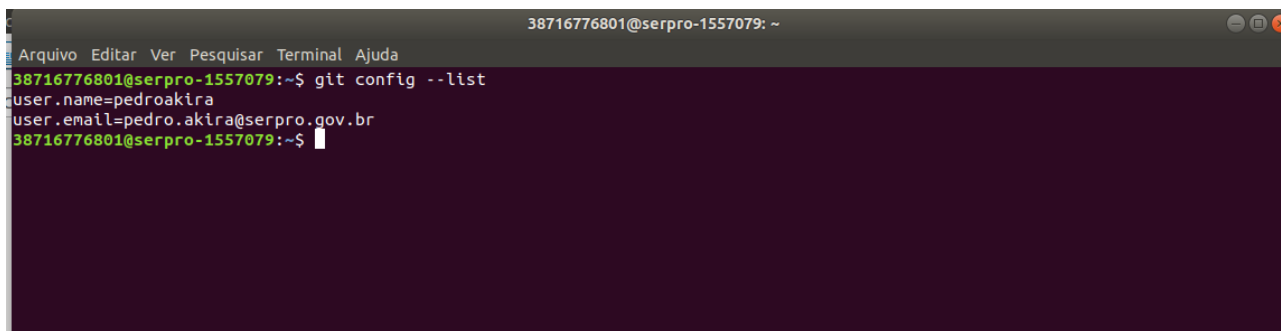
`$ git config --global user.email "<seu email corporativo>"`



```
38716776801@serpro-1557079: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
38716776801@serpro-1557079:~$ git config --global user.name "pedroakira"
38716776801@serpro-1557079:~$ git config --global user.email "pedro.akira@serpro.gov.br"
38716776801@serpro-1557079:~$
```

Para confirmar as alterações execute o comando de listagem de configurações citado anteriormente:

`git config --list`

A screenshot of a terminal window with a dark background. The title bar at the top reads '38716776801@serpro-1557079: ~'. The menu bar includes 'Arquivo', 'Editar', 'Ver', 'Pesquisar', 'Terminal', and 'Ajuda'. The terminal shows the command 'git config --list' being executed, with the following output: 'user.name=pedroakira' and 'user.email=pedro.akira@serpro.gov.br'. The prompt '38716776801@serpro-1557079:~\$' is visible at the bottom of the output.

Na listagem apresentada você deverá visualizar as configurações do "user.name" e "user.email" caso não esteja encontrando verifique se não houve nenhum tipo de erro ao digitar os comandos anteriores.

Gerando chave SSH

Vários servidores Git autenticam usando chaves públicas SSH. Para fornecer uma chave pública, cada usuário no seu sistema deve gerar uma se eles ainda não a possuem. Este processo é similar entre os vários sistemas operacionais. Primeiro, você deve checar para ter certeza que você ainda não possui uma chave. Por padrão, as chaves SSH de um usuário são armazenadas no diretório `~/.ssh`. Você pode facilmente verificar se você tem uma chave indo para esse diretório e listando o seu conteúdo:

```
$ cd ~/.ssh
$ ls
authorized_keys2  id_dsa      known_hosts
config           id_dsa.pub
```



```
38716776801@serpro-1557079: ~/.ssh
Arquivo Editar Ver Pesquisar Terminal Ajuda
38716776801@serpro-1557079:/$ cd ~/.ssh
38716776801@serpro-1557079:~/.ssh$ ls
config id_rsa id_rsa-gitcorporativo-treinamento id_rsa-gitcorporativo-treinamento.pub id_rsa.pub known_hosts
38716776801@serpro-1557079:~/.ssh$
```

Você está procurando por um par de arquivos chamados *algo* e *algo.pub*, onde *algo* é normalmente `id_dsa` ou `id_rsa`. O arquivo `.pub` é a sua chave pública, e o outro arquivo é a sua chave privada. Se você não tem estes arquivos (ou não tem nem mesmo o diretório `.ssh`), você pode criá-los executando um programa chamado `ssh-keygen`, que é fornecido com o pacote SSH em sistemas Linux/Mac e vem com o pacote MSysGit no Windows:

Gerar chave ssh:

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/schacon/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/schacon/.ssh/id_rsa.
Your public key has been saved in /Users/schacon/.ssh/id_rsa.pub.
The key fingerprint is:
43:c5:5b:5f:b1:f1:50:43:ad:20:a6:92:6a:1f:9a:3a schacon@agadorlaptop.local
```

A screenshot of a terminal window titled '38716776801@serpro-1557079: ~/.ssh'. The terminal shows the command 'ssh-keygen' being executed. The output is: 'Generating public/private rsa key pair.' followed by 'Enter file in which to save the key (/home/38716776801/.ssh/id_rsa):'. The cursor is positioned at the end of the prompt line. The terminal has a dark background and a menu bar at the top with options: 'Arquivo', 'Editar', 'Ver', 'Pesquisar', 'Terminal', 'Ajuda'.

Primeiro ele confirma onde você quer salvar a chave (.ssh/id_rsa), e então pergunta duas vezes por uma senha, que você pode deixar em branco se você não quiser digitar uma senha quando usar a chave.

Agora, cada usuário que executar o comando acima precisa enviar a chave pública para você ou para o administrador do seu servidor Git (assumindo que você está usando um servidor SSH cuja configuração necessita de chaves públicas). Tudo o que eles precisam fazer é copiar o conteúdo do arquivo .pub e enviar para você via e-mail. As chaves públicas são parecidas com isso.

```
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIWAAAQEAKlOUpkDHrfHY17SbrmTIpNLTGK9Tjom/BWDSU
GP1+nafz1HDTYw7hdI4yZ5ew18JH4JW9jbhUFRviQzM7x1ELEVF4h91FX5QVkbPppSwg0cda3
Pbv7kOdJ/MTyBlWxFCR+HAo3FXRitBqxiX1nKhXpHAZsMciLq8V6RjsNAQwdsdMFvSlVK/7XA
t3FaoJoAsncM1Q9x5+3V0Ww68/eIFmb1zuUF1jQJKprX88XypNDvjYNby6vw/Pb0rwert/En
mZ+AW40ZPnTPI89ZPmVMLuayrD2cE86Z/il8b+gw3r3+1nKatmIkjn2so1d01QraTlMqVSsbx
NrRFi9wrf+M7Q== schacon@agadorlaptop.local
```

```
38716776801@serpro-1557079: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
38716776801@serpro-1557079:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAIHKd16/omM9zICjG1/oeS1A7rlyZl/h/Pbuczu0H2HHAaFY41TPIW5YA3CoGh6mfW6PQXVMUaHcFpH2biFRntj+yevXfNVzhkmcMR9BVQlVy
gzFVzIwdJp0BgogHhxiJhvd/GLInmYmgCCrXoy5MqWdHj1z3NdY2wGxD1adLbyxI3kCkqeUCxshtEdnjnh6J46SL4sJx8Jp5tDG4tA0Aa/p29lvvrikr8Zz+Clz/LsL/8JBrXAsVUV2lqRTssMsR0K
3+P3ozV0Gpx6vtIdN+UQLIHnn1wTfc598CWTGc22juAJMN8WPentJAbBRRlTF4uqw8EKfthVYUMv9Z 38716776801@serpro-1557079
38716776801@serpro-1557079:~$
```

Adicionando a chave SSH no Git Corporativo

Nesta etapa, deveremos informar ao Git Corporativo a sua chave pública para que ele possa identificá-lo como usuário quando houver alguma interação, seja por meio de comandos via terminal, Eclipse ou alguma outra ferramenta escolhida.

Copiando o conteúdo da Chave Pública

É necessário copiar o conteúdo da Chave Pública criada nos passos anteriores, caso não lembre veja os tópicos "Gerando chave SSH usando o Terminal" ou "Gerando chave SSH usando o Eclipse".

Copie o conteúdo do arquivo localizado em "`~/.ssh/id_rsa-gitcorporativo-treinamento.pub`", se você estiver utilizando um sistema operacional como o Linux ou "`C:\Users\05081364908\.ssh\id_rsa-gitcorporativo-treinamento.pub`" no caso do Windows com o seu editor de preferência.

No exemplo abaixo estamos utilizando o comando "cat" para realizar a tarefa:

cat ~/.ssh/id_rsa-gitcorporativo-treinamento.pub

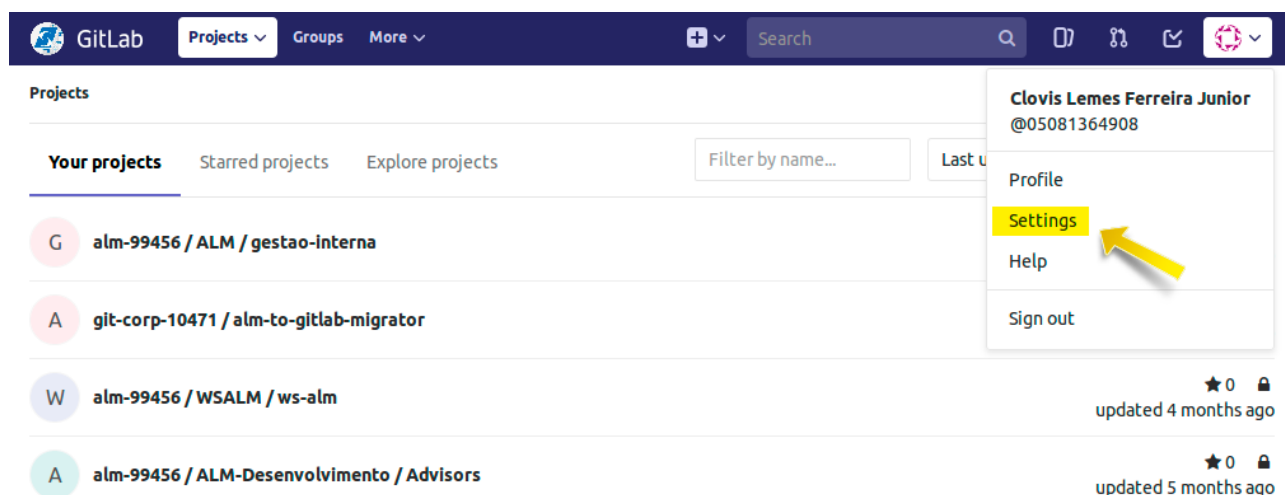
```
05081364908@serpro-1548841: ~  
05081364908@serpro-1548841:~$ cat ~/.ssh/id_rsa-gitcorporativo-treinamento.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAQDvz21Bwrl2f48dCeZ9wufP0rrKYrod8Bn3Q+qsnYAO7PkXNY0yxI0yzzwX5cTm0D1mr2Kqy+r1wQbBsQ0K64lJSSkNQLGAqwMuIrEoW  
Jsl/5PjQMRuAhfPKKe92zY83/Av8TQIvsjnlBa81IA1nQvM4D3c9aQ2s9MhmMoxqJfpeRJ/GemWKZ3YUsHuZ5Qo1beTfIZZ2rg+pzUMnFL8tqgJLRc7TEvoKs8enTdY5ftvaentTsV4/R  
FkJv8B/0sg1f3/46cCgY5wN0r3smOk28+VesK6smthL80LH3IFA0/zJoTAbFM10rLnHNTmbts6apxo6vNhFC27vxUYFHxPVaxFJwsvmJYH31Nv1bnwGnXPcXP4lxjvTCLsqwObrMeek2  
f8bmlUY8BS5EH6bs76o2hlQFSX543f9Gd2pp1E0MCJ1AdvP76acBVk+etmF8lvcczUNwnX3m1s6CrTk6ock2MZ+lfI273Yb4DlT8NqeBQGbcJehy7m5qubfQUCBKNL8YmrkpyvHuQWcd0t  
PdegNjx3v7W4Ge0MTHpW2dGPKlvDY/8zt0RcYyJtcsqsQBUR+2sQUQdCMEQ/D/6T8bemZ17qseZ+5d55GL66fQM4Pka8KPzFEb+7QAyeLd1/kzc10C4Q9o5jfn4VLgt30WIqsVdoUUUKko  
BEXZCkHJniYw== clovts.ferreira@serpro.gov.br  
05081364908@serpro-1548841:~$
```

O conteúdo a ser copiado é da chave pública e não da chave privada. Você deve checar se o arquivo a ser copiado no comando apresentado tem a extensão .pub

Cadastrando a Chave Pública no Git Corporativo

Acesse o [ambiente de Homologação do Git Corporativo](#)

1. Realize o login, caso não esteja logado
2. Acesse o menu de configurações/settings do seu usuário, que se encontra no menu direito superior conforme imagem a seguir:



Na tela de configurações, acesse o menu "SSH Keys". Nessa tela, deveremos inserir a chave SSH para que o Git Corporativo possa reconhecer seu usuário quando você interagir com a ferramenta.

Com o conteúdo da Chave Pública já copiada para a memória, cole na caixa com o rótulo "**Key**" do Git Corporativo.

O campo "**Title**" deverá ser preenchido automaticamente pelo Git Corporativo. Caso o campo fique vazio, insira um nome para sua chave e clique no botão "**Add key**".

GitLab Projects Groups More

User Settings > SSH Keys

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

Before you can add an SSH key you need to [generate it](#).

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGC4viA1a1bmjmxMnjXboal5wx9erbtgwz+uWIW5LIMTYFZ4QhsaFt+BAi4Bdmw0y5kXm0zt+Y5oTDY0xP
q5qePWEOGtNGKQImLrYUvQOMRvbtK54QnwYmxE3hnQXhAvAYk79owj07H5426f86JMK+JflrXlJrA52vEaaJwGE+Kw==
clovis.ferreira@serpro.gov.br
```

Title

clovis.ferreira@serpro.gov.br

Add key

Your SSH keys (0)

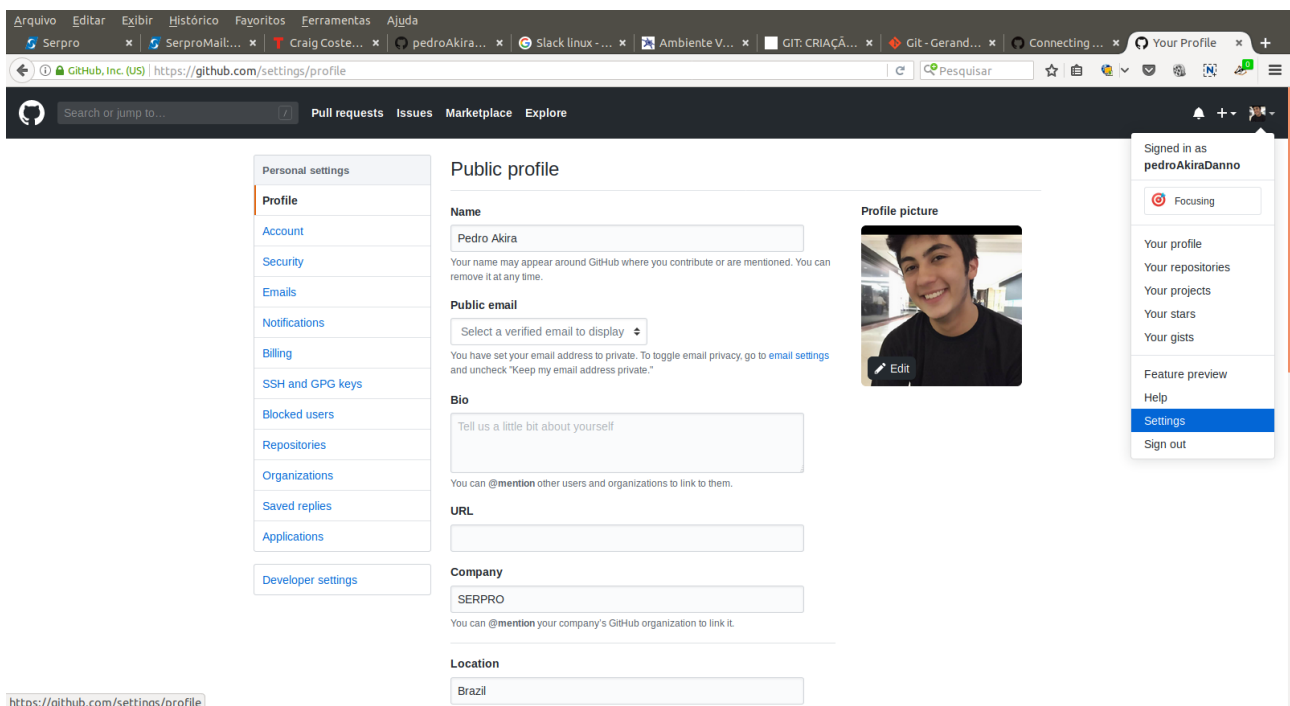
There are no SSH keys with access to your account.

É possível cadastrar mais de uma chave SSH no Git Corporativo. Isso se torna útil quando você tem mais de uma máquina/servidor que precise realizar tarefas no Git Corporativo. Dessa forma, cada máquina/servidor usa sua própria chave SSH.

Adicionando a chave SSH no GitHub

Acesse o <https://github.com/>

3. Realize o login, caso não esteja logado
4. Acesse o menu de configurações/settings do seu usuário, que se encontra no menu direito superior conforme imagem a seguir:



Na tela de configurações, acesse o menu "SSH and GPG keys".

The screenshot shows the GitHub 'SSH and GPG keys' settings page. On the left, a sidebar lists various settings categories, with 'SSH and GPG keys' selected. The main area is titled 'SSH keys' and contains a list of three existing SSH keys. Each key entry displays a key icon, the key title, the public key fingerprint, the date it was added, and a 'Delete' button. A 'New SSH key' button is located in the top right corner of the SSH keys section. Below this, the 'GPG keys' section is shown, featuring a 'New GPG key' button and a message indicating that no GPG keys are currently associated with the account.

Nessa tela, deveremos inserir a chave SSH para que o GitHub possa reconhecer seu usuário quando você interagir com a ferramenta, clicando em "New SSH key".

Com o conteúdo da Chave Pública já copiada para a memória, cole na caixa com o rótulo "**Key**" do Git Corporativo.

O campo " Title" deverá ser preenchido automaticamente pelo Git Corporativo. Caso o campo fique vazio, insira um nome para sua chave e clique no botão " Add SSH key".

GitHub, Inc. (US) | <https://github.com/settings/ssh/new> | Pesquisar

Search or jump to... [2] Pull requests Issues Marketplace Explore

Personal settings

Profile

Account

Security

Emails

Notifications

Billing

SSH and GPG keys

Blocked users

Repositories

Organizations

Saved replies

Applications

Developer settings

SSH keys / Add new

Title

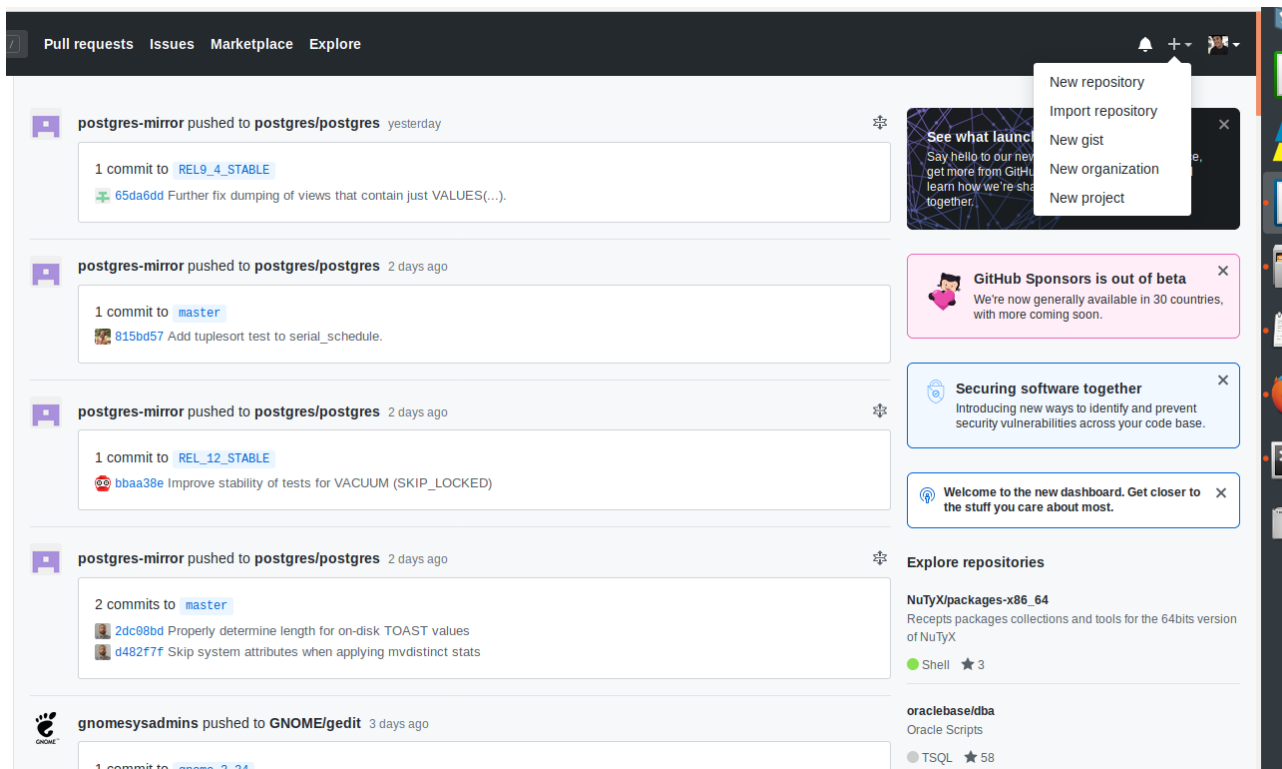
Key

Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Add SSH key

Criação de repositório GitHub

Para iniciar a criação de um novo projeto. Você poderá criar seu novo projeto clicando no botão "New repository".



Será aberta uma nova tela para criação do projeto.

Coloque o nome do repositório no campo Repository name*

Após o preenchimento desses campos básicos, clique no botão "Create repository".

https://github.com/new

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner **Repository name ***

pedroAkiraDanno / teste ✓

Great repository names are short and memorable. Need inspiration? How about **turbo-doodle**?

Description (optional)

repositorio teste, para tutorial

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

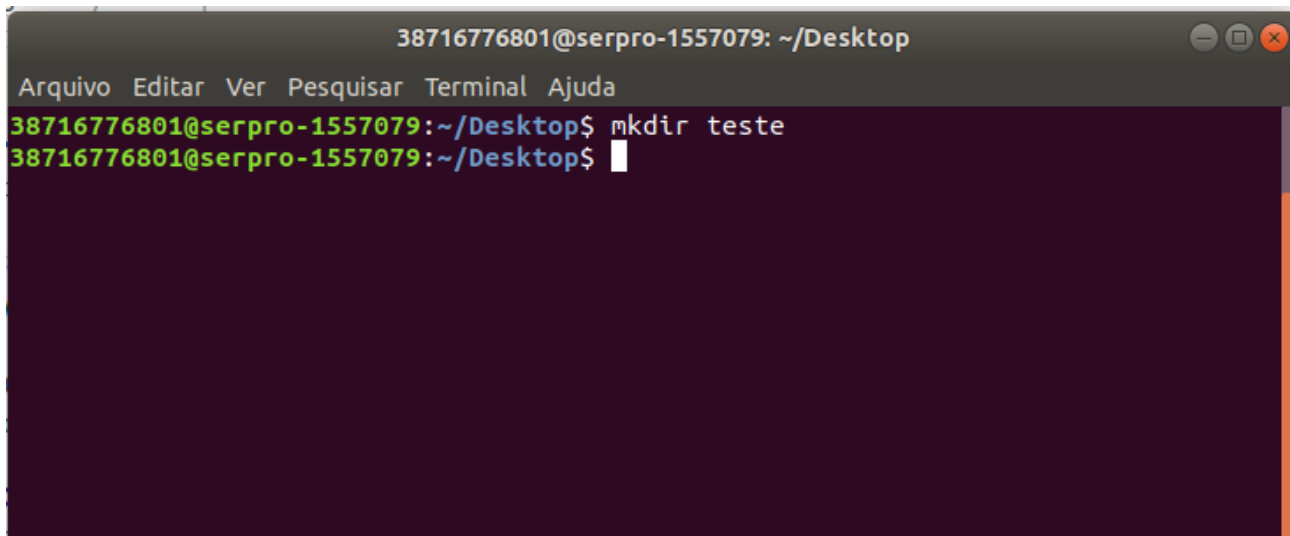
Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

Criando repositório local

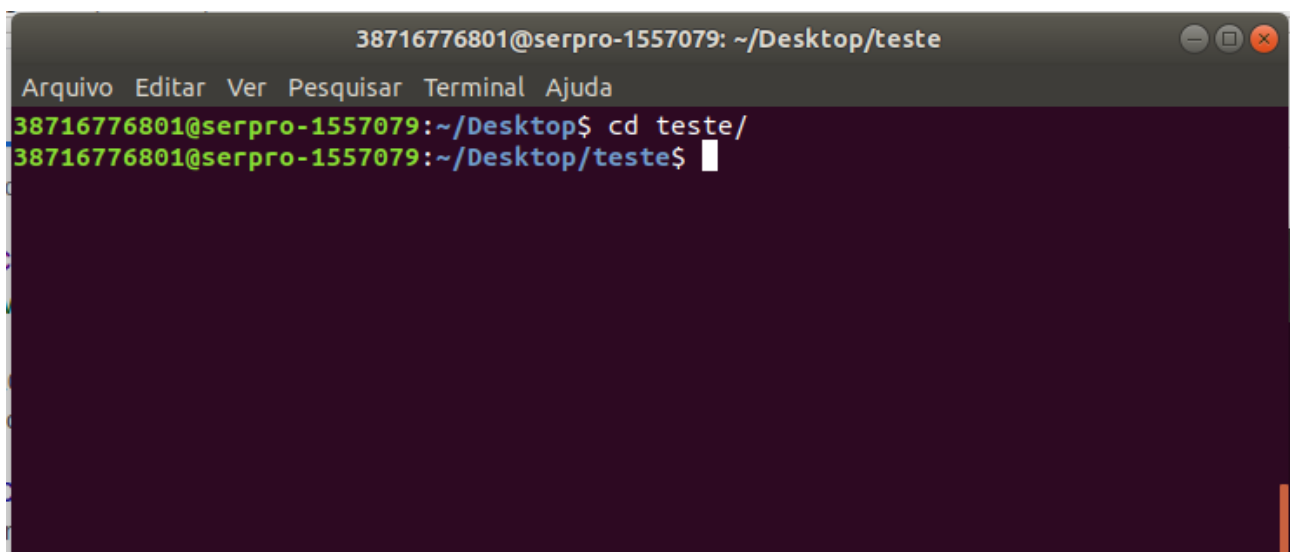
Criar diretorio teste

```
$ mkdir teste
```



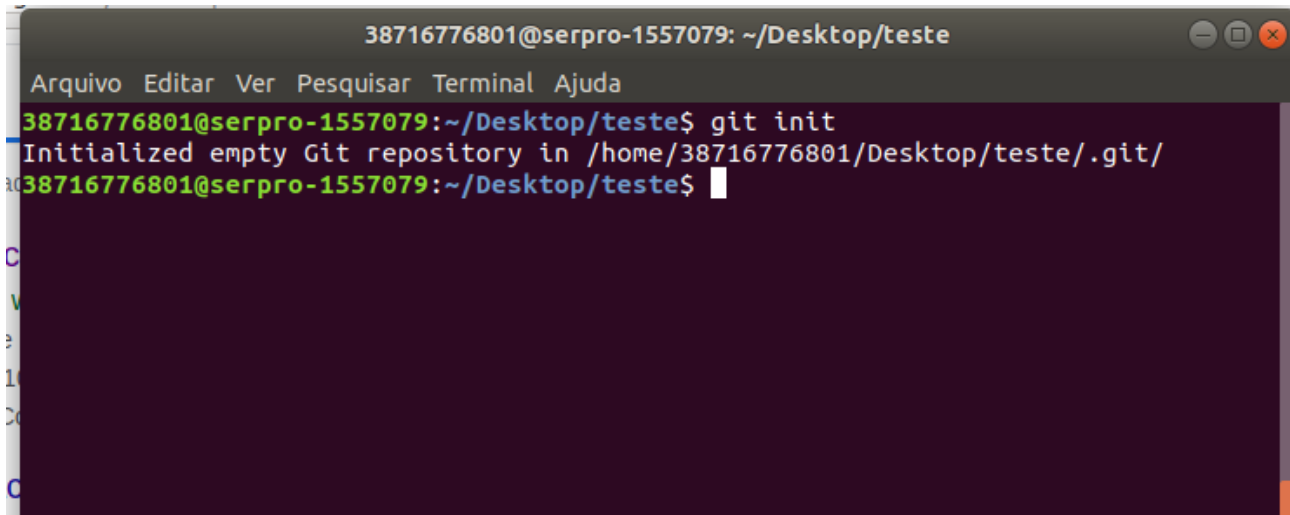
A terminal window titled '38716776801@serpro-1557079: ~/Desktop'. The menu bar includes 'Arquivo', 'Editar', 'Ver', 'Pesquisar', 'Terminal', and 'Ajuda'. The terminal shows the command 'mkdir teste' being executed successfully, with the prompt changing to '38716776801@serpro-1557079:~/Desktop\$'.

Digitar no terminal cd teste/



A terminal window titled '38716776801@serpro-1557079: ~/Desktop/teste'. The menu bar includes 'Arquivo', 'Editar', 'Ver', 'Pesquisar', 'Terminal', and 'Ajuda'. The terminal shows the command 'cd teste/' being executed successfully, with the prompt changing to '38716776801@serpro-1557079:~/Desktop/teste\$'.

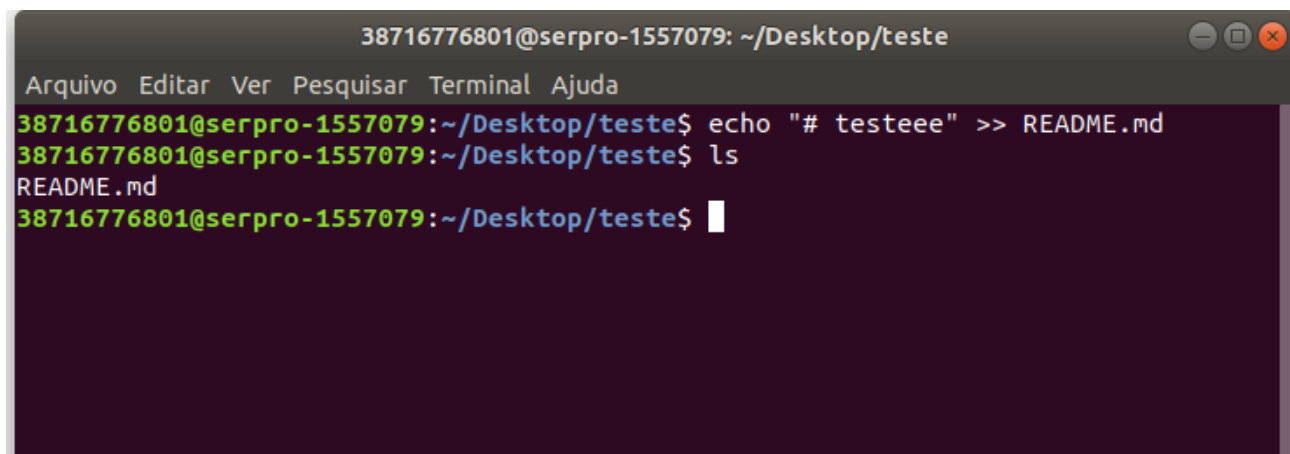
Digitar : git init, para criação de arquivos .git

A terminal window titled '38716776801@serpro-1557079: ~/Desktop/teste' with a menu bar 'Arquivo Editar Ver Pesquisar Terminal Ajuda'. The terminal shows the command 'git init' being executed, resulting in the message 'Initialized empty Git repository in /home/38716776801/Desktop/teste/.git/'. The prompt is now '38716776801@serpro-1557079:~/Desktop/teste\$' with a cursor.

```
38716776801@serpro-1557079: ~/Desktop/teste
Arquivo Editar Ver Pesquisar Terminal Ajuda
38716776801@serpro-1557079:~/Desktop/teste$ git init
Initialized empty Git repository in /home/38716776801/Desktop/teste/.git/
38716776801@serpro-1557079:~/Desktop/teste$
```

Criar aquivo README.md

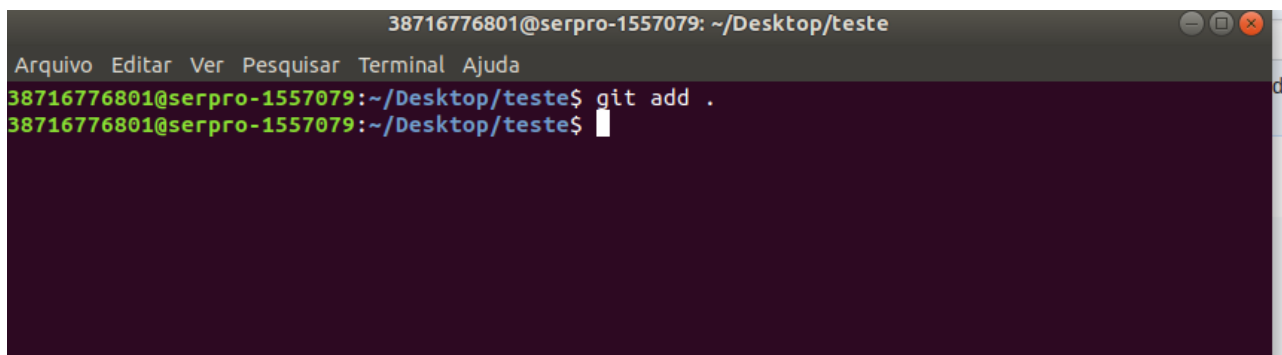
\$ echo "# teste" >> README.md

A terminal window titled '38716776801@serpro-1557079: ~/Desktop/teste' with a menu bar 'Arquivo Editar Ver Pesquisar Terminal Ajuda'. The terminal shows the command 'echo "# teste" >> README.md' being executed, followed by 'ls' which lists 'README.md'. The prompt is now '38716776801@serpro-1557079:~/Desktop/teste\$' with a cursor.

```
38716776801@serpro-1557079: ~/Desktop/teste
Arquivo Editar Ver Pesquisar Terminal Ajuda
38716776801@serpro-1557079:~/Desktop/teste$ echo "# teste" >> README.md
38716776801@serpro-1557079:~/Desktop/teste$ ls
README.md
38716776801@serpro-1557079:~/Desktop/teste$
```

Adicionado mudanças

\$ git add .



```
38716776801@serpro-1557079: ~/Desktop/teste
Arquivo Editar Ver Pesquisar Terminal Ajuda
38716776801@serpro-1557079:~/Desktop/teste$ git add .
38716776801@serpro-1557079:~/Desktop/teste$
```

Fazendo primeiro commit

\$ git commit -m "first commit"



```
38716776801@serpro-1557079: ~/Desktop/teste
Arquivo Editar Ver Pesquisar Terminal Ajuda
38716776801@serpro-1557079:~/Desktop/teste$ git commit -m "first commit"
[master (root-commit) e22c5b0] first commit
1 file changed, 2 insertions(+)
create mode 100644 README.md
38716776801@serpro-1557079:~/Desktop/teste$
```


Adicionado repositório local com repositório gitHub

git remote add origin https://github.com/pedroAkiraDanno/testeeee.git

```
38716776801@serpro-1557079: ~/Desktop/teste
Arquivo Editar Ver Pesquisar Terminal Ajuda
38716776801@serpro-1557079:~/Desktop/teste$ git remote add origin https://github.com/pedroAkiraDanno/testeeee.git
```

Enviando para repositório remoto gitHub, digitar usuário e senha.

\$ git push -u origin master

```
38716776801@serpro-1557079: ~/Desktop/teste
Arquivo Editar Ver Pesquisar Terminal Ajuda
38716776801@serpro-1557079:~/Desktop/teste$ git push -u origin master
Username for 'https://github.com': pedroAkiraDanno
Password for 'https://pedroAkiraDanno@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 224 bytes | 224.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/pedroAkiraDanno/testeeee.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
38716776801@serpro-1557079:~/Desktop/teste$
```

Referências Bibliográficas

[1] Version Control with Git, O'Reilly books, 2009.

site <https://cds.cern.ch/record/1190773/files/9780596520120_TOC.pdf>

[2] Pro Git book, 2014

site <<https://git-scm.com/book/en/v2>>

[3] GIT: PRINCIPAIS FUNCIONALIDADES DO GIT CORPORATIVO - CURSO - A DISTÂNCIA -
V. 1 - De 02/09/2019 a 30/09/2019

site <<https://moodle.ead.serpro.gov.br/>>

[4] Gerando Sua Chave Pública SSH

site <<https://git-scm.com/book/pt-br/v1/Git-no-Servidor-Gerando-Sua-Chave-P%C3%BAblica-SSH>>

[5] Criação de chaves SSH em vários sistemas operacionais

site <<https://help.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh>>

FICHA TÉCNICA

SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS

Diretora Presidente
Caio Mario Paes de Andrade

Diretor de Operações
Iran Martins Porto Júnior

Superintendente de Operações – SUPOP
Gilberto de Oliveira Netto

**Departamento de Gestão dos Serviços de Rede de Acesso e Comunicação –
SUPOP/OPGTC**
Weldson Queiroz de Lima

Divisão de Tecnologias de Acesso e Comunicação – SUPOP/OPGTC/OPTAC
Régison Rodrigo Martins

Rede de Suporte Técnico de
Paulo Roberto dos Santos Arruda

Elaboração
Pedro Akira Danno Lima – DIOPE/ CDOPC/CDMSS/CDOP1

Colaboração
Luciana Candida Souza Machado - DIOPE/SUPCD/CDOPC/CDMSS/CDOP1

Revisão da Versão
Luciana Candida Souza Machado - DIOPE/SUPCD/CDOPC/CDMSS/CDOP1

Versão 1.0
Novembro / 2019