

# git e github: aula prática

```
$ echo `whoami`
```

```
Narumi Abe
```

# Clonando o repositório

mostrar o site do repositório

<https://github.com/naruminho/aula-git>

Mostrar a página do github

<https://naruminho.github.io/aula-git/>

Configurando o github

Estas etapas são feitas somente uma vez

# Criando uma chave pública

Existem diversos métodos para se autenticar com o git. Vamos usar o método das chaves. Usuários de sistemas bons provavelmente já possuem chaves públicas e privadas.

Usuários de windows devem criar a chave pública e a chave privada.

# Criando as chaves

Entre no prompt de comando e digite o comando abaixo e pressione [ENTER].

```
ssh-keygen -t rsa -C "seuemail@dominio.com"
```

Aperte [ENTER] novamente para confirmar.

Digite uma frase e pressione [ENTER], depois digite a mesma frase e pressione [ENTER] novamente.

# Adicionando a chave pública

```
cd .ssh [enter]  
cat id_rsa.pub [enter]
```

**cd .ssh** significa entrar na pasta **.ssh**  
**cat id\_rsa.pub** mostra o conteúdo do arquivo (nossa chave pública)  
Copiar o conteúdo que será exibido. Depois iremos colar no site.

Obs 1: Use **type** ao invés de **cat** no windows.

Obs 2: Não ficarei falando que é preciso apertar o [ENTER]. O computador não adivinhará sozinho quando você terminar de digitar.

# Adicionando uma chave pública

No github, vá em configurações do usuário.

Na tela que irá se abrir, siga as etapas e cole o conteúdo copiado do slide anterior.

The screenshot shows the GitHub 'SSH keys' settings page. On the left is a sidebar with 'Personal settings' and various options. 'SSH and GPG keys' is highlighted with a red box and a circled '1'. The main content area has two sections: 'SSH keys' and 'GPG keys'. In the 'SSH keys' section, a 'New SSH key' button is highlighted with a red box and a circled '2'. Below it, a key named 'my key' is listed with its fingerprint and a 'Delete' button. In the 'GPG keys' section, a 'New GPG key' button is visible. To the right of the 'GPG keys' section, there is a form with a 'Title' field containing '3 qualquer nome' (circled '3') and a 'Key' field containing '4 cole a chave pública aqui' (circled '4'). At the bottom right, an 'Add SSH key' button is visible.


**Personal settings**

- Profile
- Account
- Emails
- Notifications
- Billing
- SSH and GPG keys** (1)
- Security
- Blocked users
- Repositories
- Organizations
- Saved replies
- Applications
- Developer settings

## SSH keys

(2) **New SSH key**

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

 **my key**  
**Fingerprint:** c5:8a:9f:df:86:ff:86:c1:3b:4e:ae:1e:ba:d3:49:e9  
Added on May 13, 2018  
Never used — Read/write **Delete**

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](#).

**Title** (3) qualquer nome

**Key** (4) cole a chave pública aqui

**Add SSH key**

# Configurando o usuário

Substitua com seus dados.

```
git config --global user.name "John Smith"
```

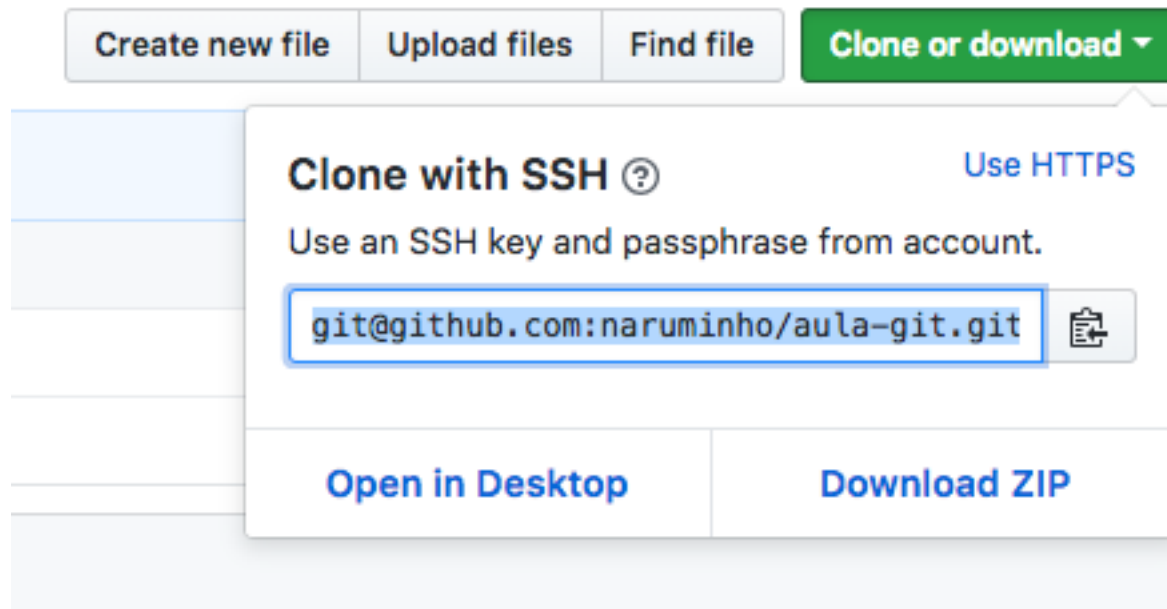
```
git config --global user.email "jsmith@upenn.edu"
```



Começando um projeto

# Clonando o repositório

Primeiramente, acessamos o endereço do repositório que desejamos usar e copiamos o link.



# Clonando o repositório

Adicione uma pasta onde ficará seu código. Entre na pasta desejada usando o comando

```
cd nome_da_pasta  
git init  
git remote add origin git@github.com:naruminho/aula-git.git  
git pull origin master
```

ou

```
git clone https://github.com/naruminho/aula-git
```

# Criando um novo branch

```
git branch minhabranch  
git checkout minhabranch
```

O comando acima cria uma branch chamada **minhabranch**. Em seguida, diz para o git usar a **minhabranch** como a branch atual.

Recomenda-se sempre criar novas branches e depois fazer o **merge**. Não editamos a branch **master** diretamente, pois nela fica o código estável.

# Alterando o código

Podemos agora fazer as alterações no código normalmente.  
Use o comando abaixo para verificar as alterações:

```
git status
```

# Comitando nosso código

```
git add -all  
git commit -m "mensagem"
```

Os comandos acima adicionam os arquivos alterados e comitam as alterações. a opção **-m "mensagem"** descreve as alterações feitas.

# Enviando para o servidor

```
git push origin minhabranch
```

O comando acima envia as alterações para a branch minhabranch.

# Recebendo alterações do servidor

```
git pull origin minhabranch
```

Tão importante quanto push nossos códigos sempre que eles forem alterados, é pull os códigos atualizados pelo resto do time. Use o comando acima para sincronizar os arquivos.



# Comandos úteis

Verificar se existem alterações não comitadas

```
git status
```

Ver qual branch está sendo usado

```
git branch
```

Muda para outra branch

```
git checkout outrabranch
```

Comita todas as alterações e adiciona um comentário

```
git commit -a -m "commentario"
```

# Comandos úteis

Mostra a lista de alterações

```
git log
```

Mostra as duas últimas alterações

```
git log -2
```