**CURSO DE GIT 2**

1. **INSTALAÇÃO E CONFIGURAÇÃO**

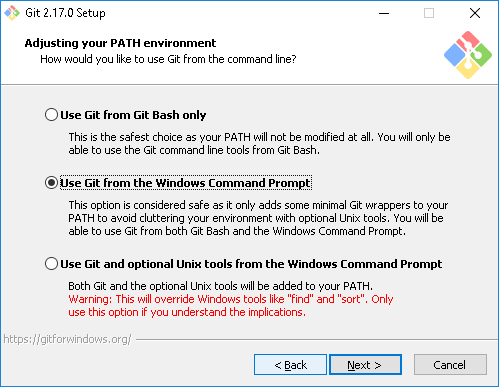
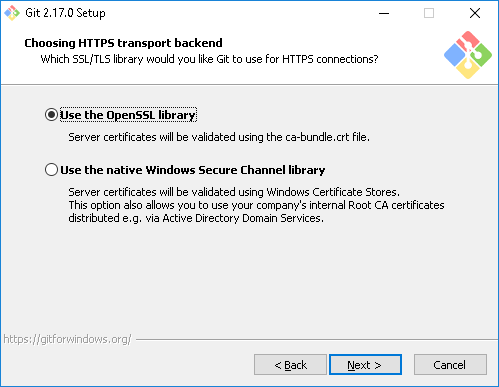
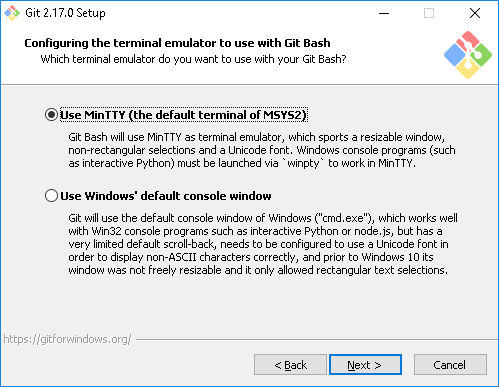
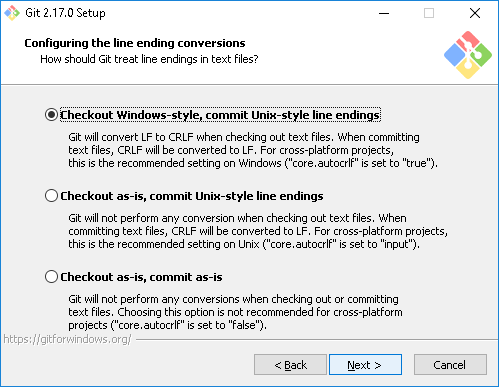
 

Figura 1 - Usa o git também com o prompt do windows.



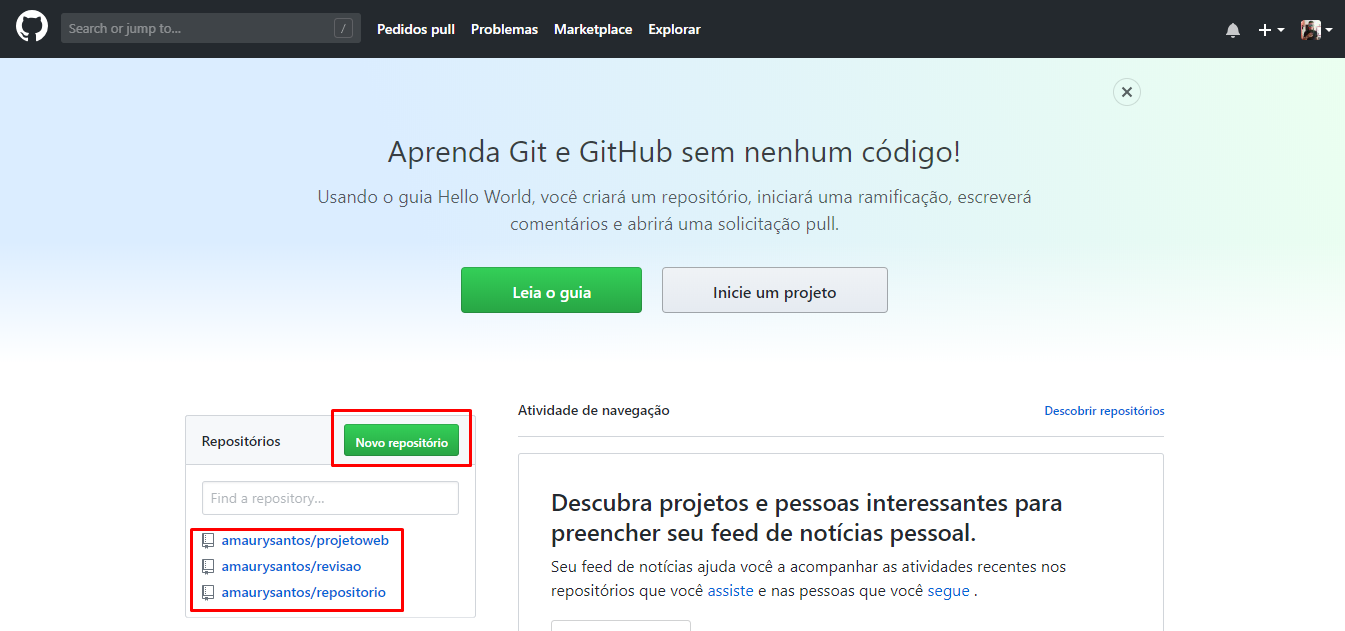
1. **CRIANDO CONTA E REPÓSITORIO REMOTO (GITHUB)**

ACESSE O LINK: <https://github.com/> para criar a conta no github.

O Github é um serviço web que oferece diversas funcionalidades extras aplicadas ao git. Resumindo, você poderá usar gratuitamente o github para hospedar seus projetos pessoais. Além disso, quase todos os projetos/frameworks/bibliotecas sobre desenvolvimento open source estão no github, e você pode acompanhá-los através de novas versões, contribuir informando bugs ou até mesmo enviando código e correções.

Depois de criado a conta, crie também os repositórios remotos, por questões de organização é recomendável que se tenha um repositório remoto para cada projeto separadamente.

Depois de criado a conta sua conta no github, você pode criar repositórios ou trabalhar em um já existente.

Figura 2 - Tela inicial do github.

* 1. **Criando repositório**

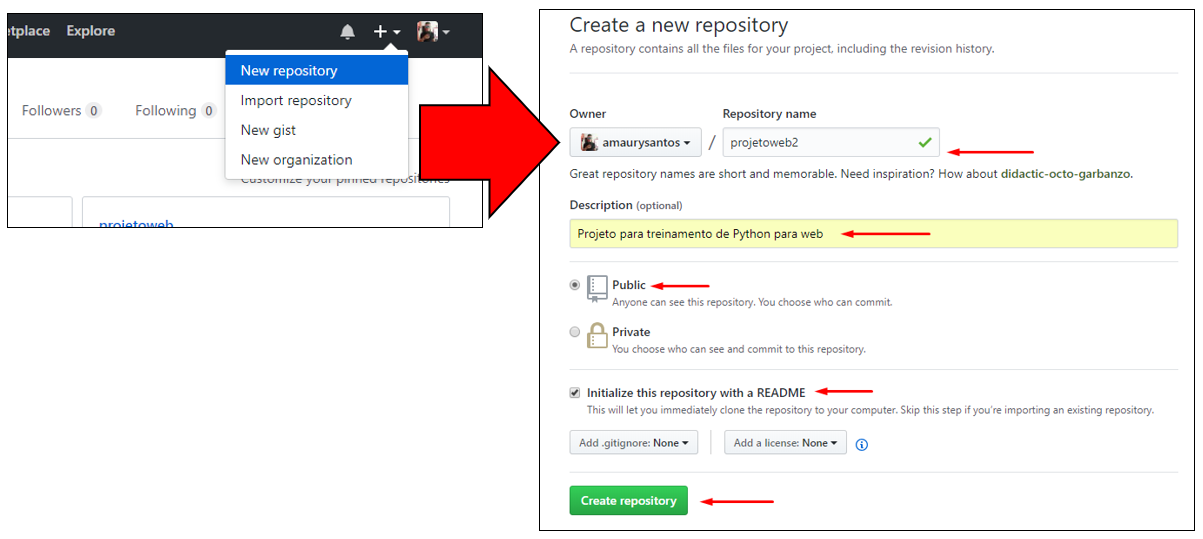
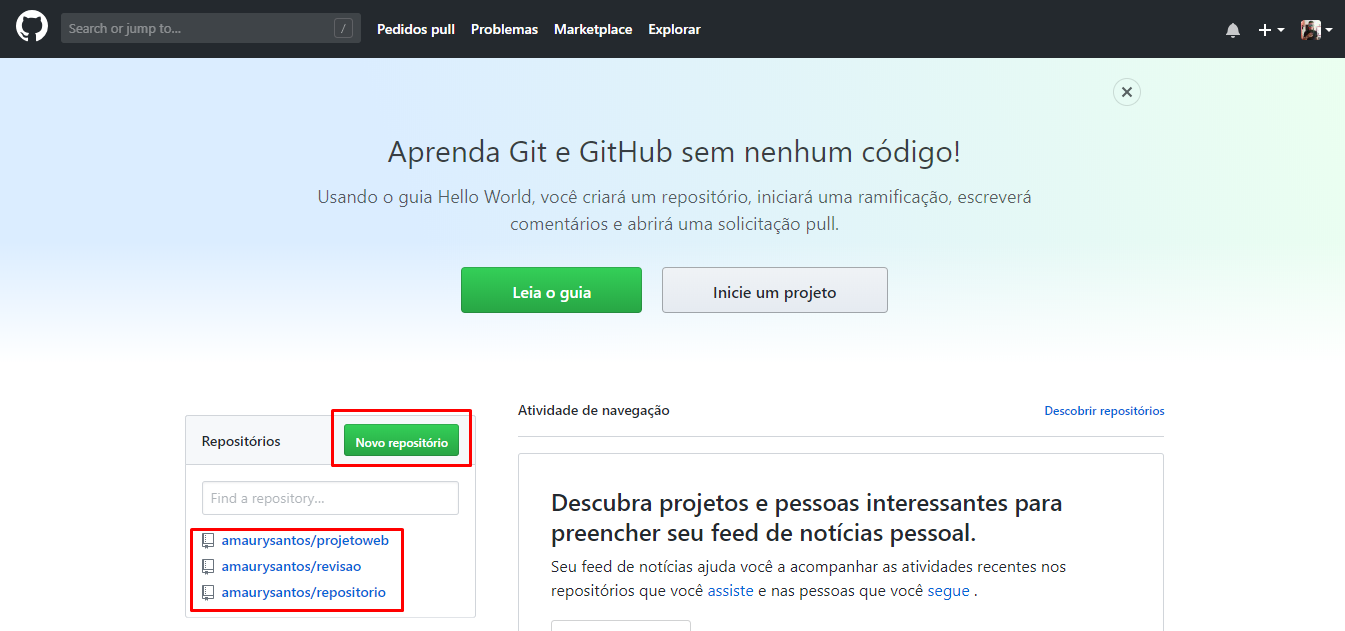


Figura 3 - Criando o repositório.

ou através do botão na tela inicial



Após a criação do repositório, você precisará configurar seu terminal local, para se comunicar com o repositório remoto. As configurações serão o usuário e email, além de uma forma de acesso ao github que será ou via https ou ssh.

1. **CONFIGURANDO SUAS INFORMAÇÕES LOCAL**

Depois de instalar o Git é necessário definir seu username e email. Isso é importante por que os seus commits usarão essas informações para identificar o autor das mudanças. Pois é… se alguém fizer alguma merda no projeto e quebrar todo o sistema, é possível saber quem, quando e qual linha foi o autor do apocalipse.

* 1. **Criando o repositório local no seu computador**

Crie uma pasta no seu computador e atribua um nome que desejar a ela. Essa pasta irá armazenar o seu repositório remoto, que nós iremos clonar (baixar) posteriormente.

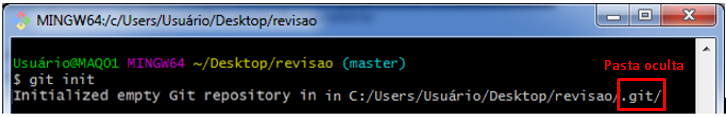
Como estamos iniciando um projeto do zero, desde a criação da conta e dos repositórios, e os mesmos estarão vazios, logo não temos o que clonar ainda, por isso criamos essa pasta local.

* Caso fosse uma conta já existente e com um projeto (repositório remoto) em andamento, nós poderíamos clonar o repositório remoto direto para dentro de algum local de armazenamento do computador, sem a necessidade de criar uma pasta manualmente.
  1. **Configurando o repositório local (GIT INIT)**

A pasta que você acabou de criar no Windows para ser o repositório local, precisa ser configurada para se comunicar com o repositório remoto na web. Após configurá-la, a pasta terá uma subpasta chamada ‘.git’ que estará oculto. Essa pasta é a responsável por todo o gerenciamento do versionador.

O comando utilizado para configurar a pasta local é:

$ git init

Figura 4 - Inicializado repositório vazio.

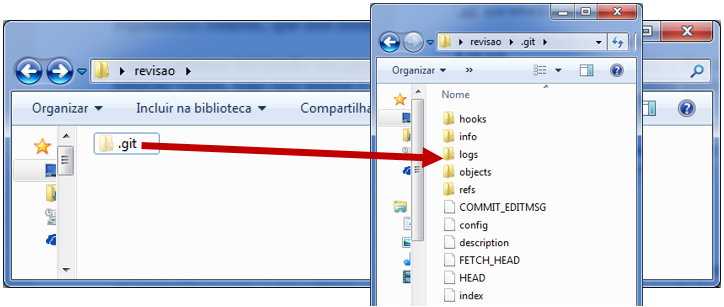


Figura 5 - Pasta local criada e configurada, exibindo a subpasta oculta '.git'.

A pasta ‘revisao’ agora está configurada e pronta para começar a trabalhar com o git apenas localmente.

* 1. **Configurando o repositório local para se comunicar com o repositório remoto no github.**

Nessa etapa nós iremos configurar para que o computador local possa manter a comunicação com o repositório remoto, e esse processo pode ser realizado através de uma comunicação via https ou ssh.

Vamos utilizar a comunicação via ssh, que é uma chave pública, que será a porta de comunicação remota entre a máquina local e o servidor remoto do github.

* 1. **Gerando uma chave pública de comunicação**

URLs SSH fornecem acesso a um repositório Git via SSH, um protocolo seguro. Para usar esses URLs, você deve gerar um par de chaves SSH no seu computador e adicionar a chave pública à sua conta do GitHub.

Antes de gerar a chave de segurança, pode-se verificar se esta já existe, para isso digite no terminal o comando: **ls ~/.ssh.**

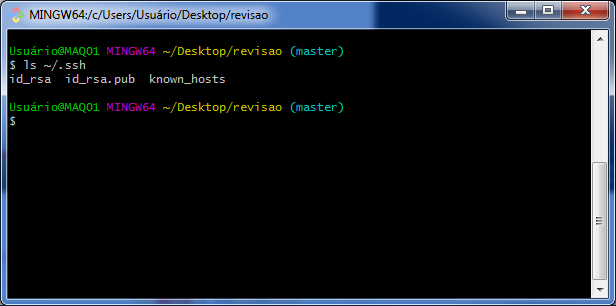


Figura 6 - comando mostrando que já existe a pasta ssh.

O arquivo ssh com a chave pública ficará armazenado no seu computador na pasta de usuário no seguinte caminho ‘**c:\users\nome\_usuário\.ssh**.

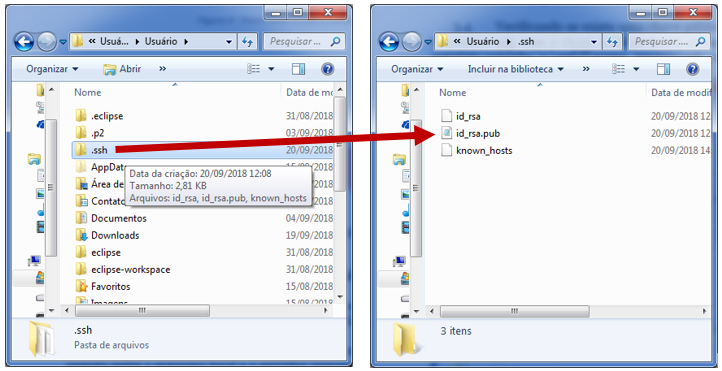


Figura 7 - Pasta ssh à esquerda, e à direita ela aberta com arquivo 'id\_rsa.pub' que contém a chave pública.

No exemplo mostrado, a chave pública já tinha sido gerada, caso contrário, devemos gera-la através do comando: **ssh-keygen**.

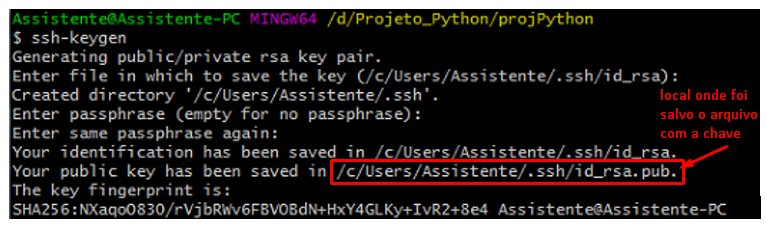


Figura 8 - Informações após o comando de criação da chave SSH.

Após criá-lo, vá até o local onde está o arquivo “id\_rsa.pub” e abra o arquivo com o bloco de notas e copie a chave. Copie todo o arquivo.

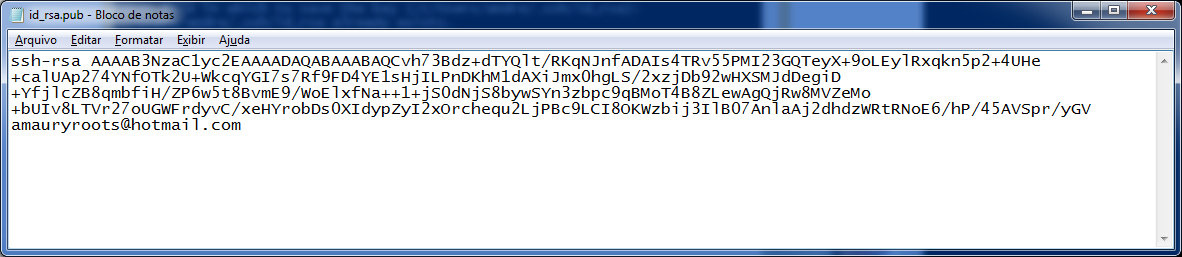


Figura 9 - Arquivo SSH com a chave.

Depois de copiar, vá até a conta no github para cadastrar a chave SSH para permitir a comunicação entre os repositórios local e remoto.

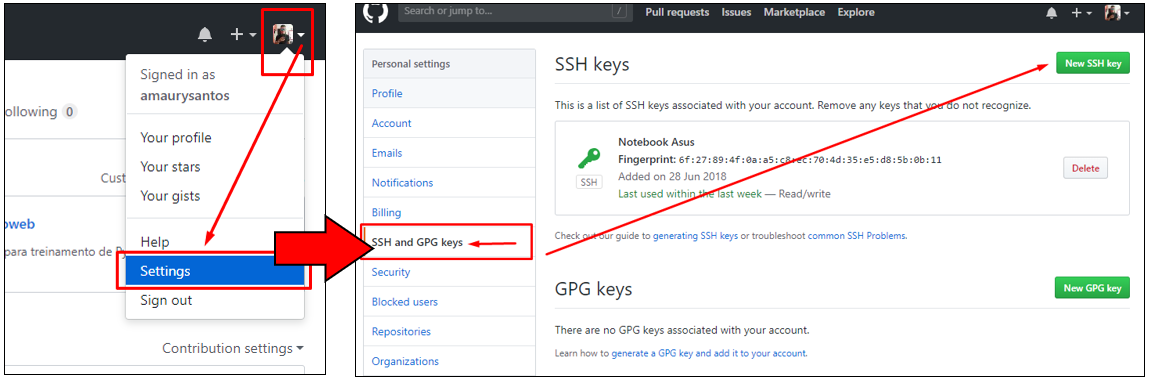


Figura 10 - Passos para adicionar a chave SSH.

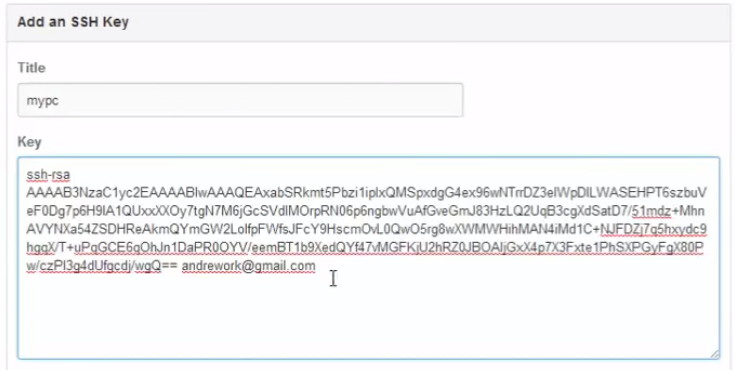


Figura 11 - Atribua um titulo e cole a chave.

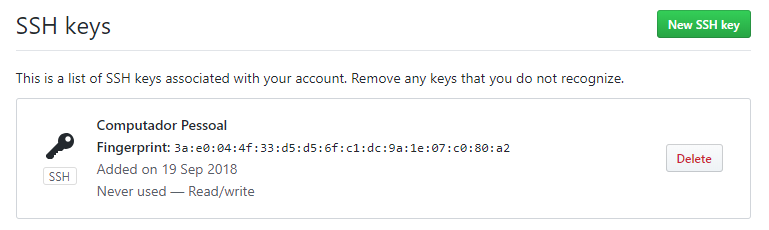
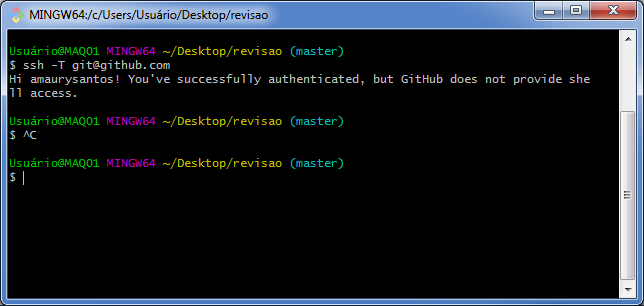


Figura 12 - Chave adicionada.

* 1. **Verificando acesso SSH - Testando a comunicação entre os repositórios**

Execute o comando: $ **ssh -T git@github.ccom**

Aparecerá uma mensagem, digite ‘yes’.



Verifique se a mensagem resultante contém seu nome de usuário. Se você receber uma mensagem de "permissão negada", consulte "Erro: Permissão negada (publickey)".

* 1. **Se der permissão negada (publickey)**

Certifique-se de ter uma chave que está sendo usada dentro da sua conta para o computador que está tentando a conexão, caso contrário, será negada.

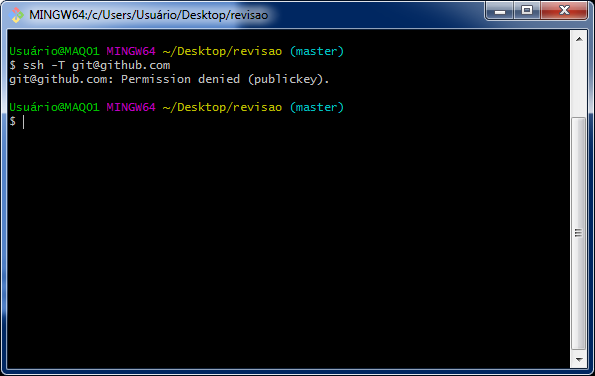


Figura 13 - Permissão negada ao testar a conexão com o repositório remoto.

No momento da criação desse tutorial, essa mensagem de negada ocorreu devido o segundo computador que estava sendo utilizado para realizar os testes, não tinha a chave SSH adicionada na conta do github, por isso a permissão foi negada para qualquer tentativa de conexão remota.

Como solução refiz o item 3.4 desse tutorial, onde ensina como gerar e adiciona a chave SSH de um computador dentro do github.

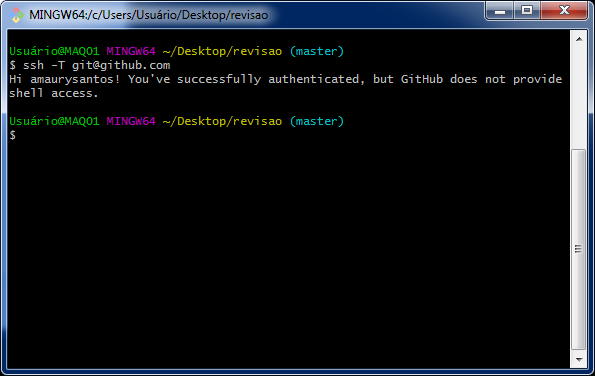


Figura 14 - Depois de adicionado a chave no github, o teste de conexão funcionou perfeitamente.

<https://help.github.com/articles/error-permission-denied-publickey/>

* 1. **Configurando seu nome de usuário e email no git**

O Git usa um nome de usuário para associar commits a uma identidade. O nome de usuário do Git não é o mesmo que o seu nome de usuário do GitHub. Use o comando:

$ git config --global user.name "Amaury Santos"

$ git config --global user.email " amauryroots@hotmail.com "

A partir de agora em todo o seu projeto global, o git irá usar esse cabeçalho a cada commit que for feito.

**Erros**

Depois de configurado usuário e email, e ao tentar realizar algumas atividades como “commit”, “push”, “pull” e apresentar um erro como:

$git commit

\*\*\* Please tell me who you are.

Run

git config --global user.email "you@example.com"

git config --global user.name "Your Name"

to set your account's default identity.

Omit --global to set the identity only in this repository.

Refaça o comando só que omitindo o ‘--global'.

* 1. **Verificando se existe um controlador remoto**

O controlador ou repositório remoto é usado para especificar a conexão do repositório local com o servidor remoto. Isso será útil quando precisarmos efetuar um ‘push,’ ou seja, enviar as modificações locais para o repositório remoto.

Para verificar se há um repositório remoto configurado no seu terminal, execute o comando:

$ **git remote** ou **git remote -v**

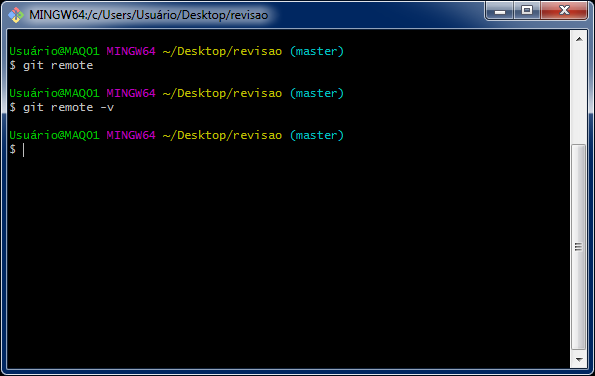
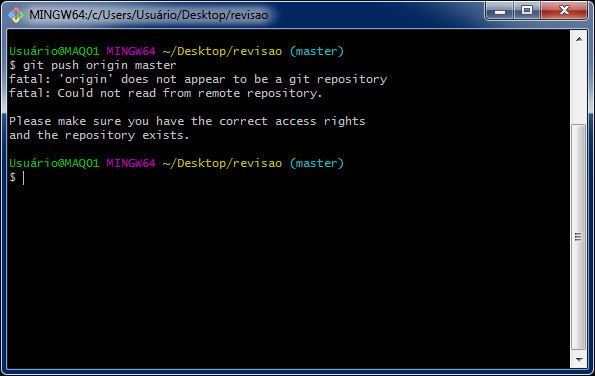


Figura 15 – Terminal sem controle remote configurado.

Como pode-se ver na figura 13, nenhum dos comandos listou repositório remoto.

* 1. **Adicionando um controle remoto**

Por padrão, esse controle remoto se chama ‘**origin’**. Seobservarmos o prompt de comando do terminal no final do nome do repositório local, aparece o nome (master), esse é o nome local.



Para adicionar um novo controle remoto, use o comando ‘git remote add’ no terminal, no diretório em que seu repositório está armazenado.

O comando git remote add leva dois argumentos:

* Um nome remoto, por exemplo, origin
* Uma URL remoto, por exemplo, https://github.com/user/repo.git

Obs.: essa URL é coletada dentro da conta do github, mais especificamente dentro do repositório no botão ‘clone or download’.

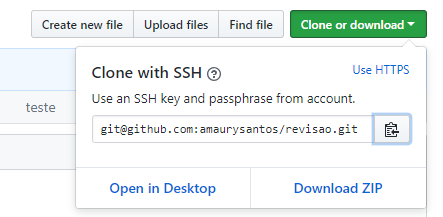


Figura 16 - Local de onde foi retirado a URL.

Exemplo:

$ git remote add **origin** git@github.com:amaurysantos/revisao.git

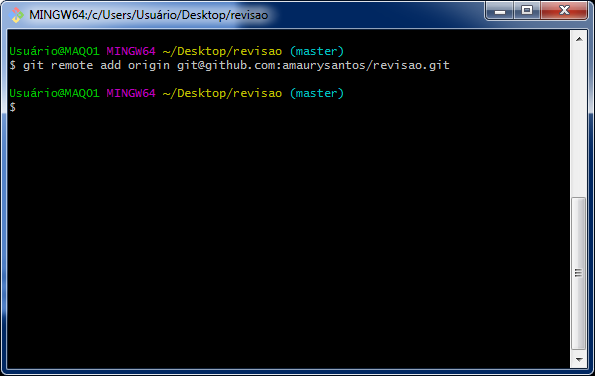


Figura 17 - Executando o comando para adicionar o controle remoto.

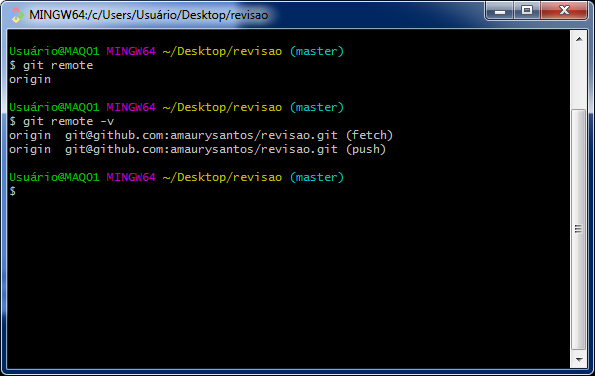


Figura 18 - Terminal com controle remoto configurado após a execução do comando.

1. **VERIFICANDO O STATUS DO REPOSITÓRIO**

O status de um repositório local pode ser: vazio ou não.

Se você criou uma pasta manualmente e depois adicionou um arquivo, o status mostrará que a pasta contém arquivo sem estar sendo monitorado pelo git.

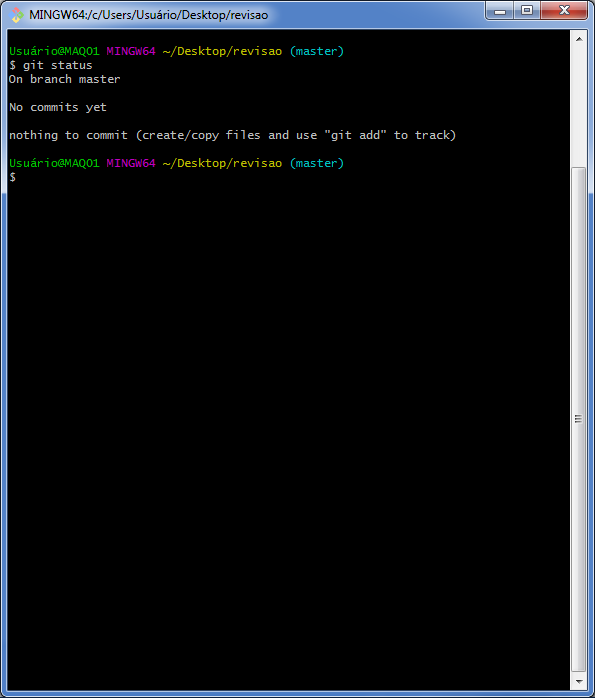


Figura 19 – Status do repositório vazio.

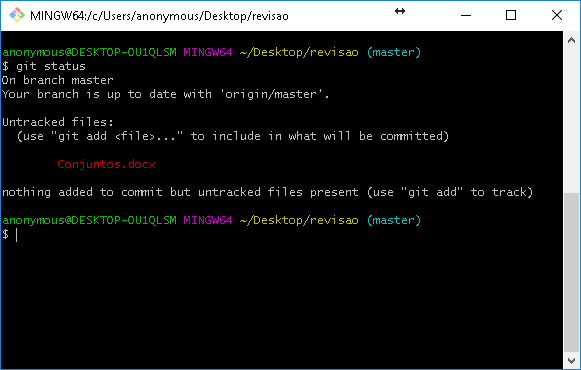


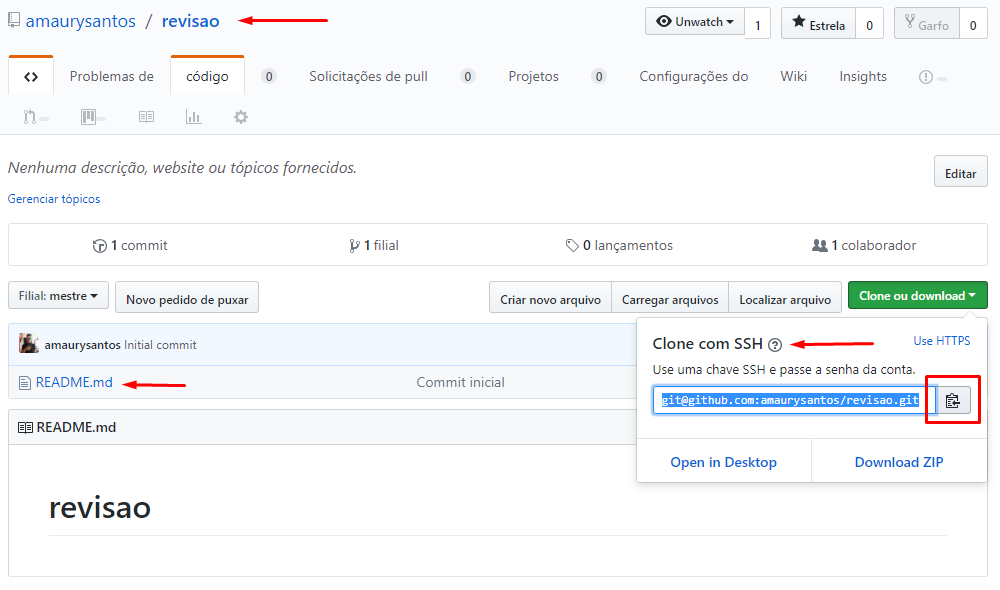
Figure 1 - Status do repositório com um arquivo não monitorado pelo git.

Observe que o arquivo ‘conjuntos’ é exibido na cor vermelho, indicando que um arquivo novo, foi adicionado ao repositório local e que não está sendo monitorado, mas que pode ser adicionado à área de preparação (index), para posteriormente ser comitado para a web.

1. **CLONANDO REPOSITÓRIO PARA MÁQUINA LOCAL**

A tarefa de clonar o repositório é realizada quando se deseja trazer algum projeto (repositório) do github para ser trabalhado localmente.

Entre no repositório, clique no botão ‘**clone or download’**, verifique se está configurado com SSH e depois clique no botão copiar conforme indicado na figura3.

Figura 20 - Tela do repositório recém-criado.

Após copiar o código de clonagem, abra o terminal git bash, e através dele acesse a pasta onde deseja baixar o repositório da web.

Digite o comando ‘**git clone**’ e cole o código do seu repositório conforme a figura 4 e pressione enter.

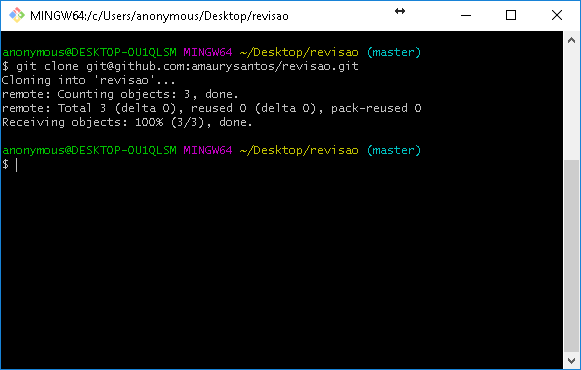


Figura 21 - visualização da pasta com o repositório clonado.

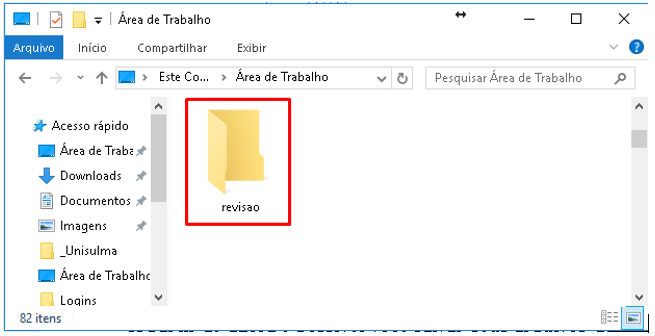


Figura 22 - Pasta do repositório "revisao" criado no github web.

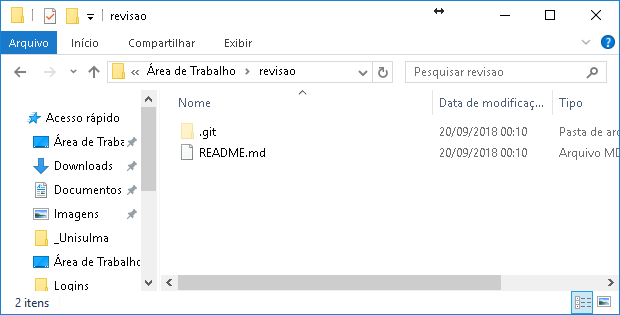


Figura 23 - Conteúdo da Pasta revisao que foi clonada.

A partir de agora é possível você salvar seus arquivos dentro da pasta que foi clonado e posteriormente enviá-los para o github.

Nesse momento a pasta trouxe somente dois arquivos o ‘.git’ que contêm todos os arquivos necessários do seu repositório para funcionar o versionamento e o arquivo ‘readme’ gerado no momento da criação do repositório.

Vamos criar um arquivo dentro da pasta, seja via terminal ou windows, para começarmos a trabalhar com o versionamento.

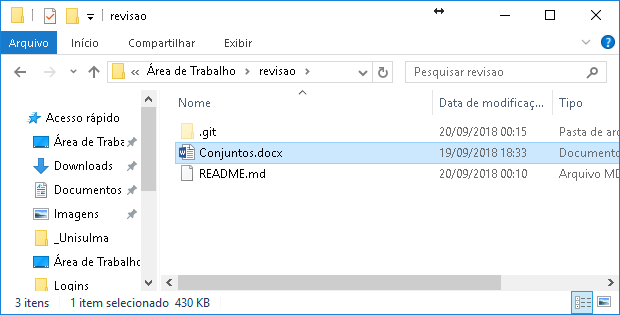


Figura 24 - Criado um arquivo 'conjuntos.docx'

1. **Adicionando arquivo no repositório**

Este comando adiciona o(s) arquivo(s) em um lugar que chamamos de INDEX, que funciona como uma área do git no qual os arquivos possam ser enviados ao Github (repositório web).

$ git add nome\_do\_arquivo [adiciona somente um arquivo especificado]

$ git add \*.txt [adiciona somente os arquivos do tipo ‘txt’]

$ git add . [adiciona todos os arquivos independente do tipo de arquivo]

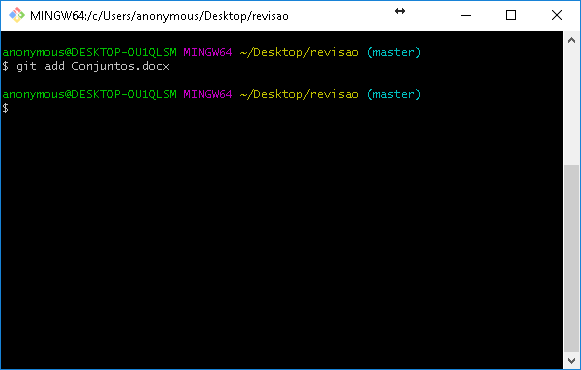


Figura 25 - Adicionando o arquivo 'conjuntos' à área de preparação.

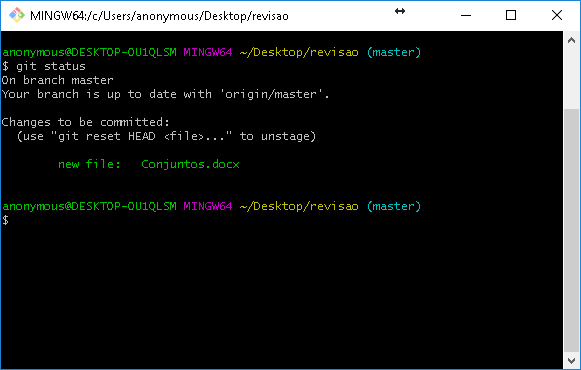
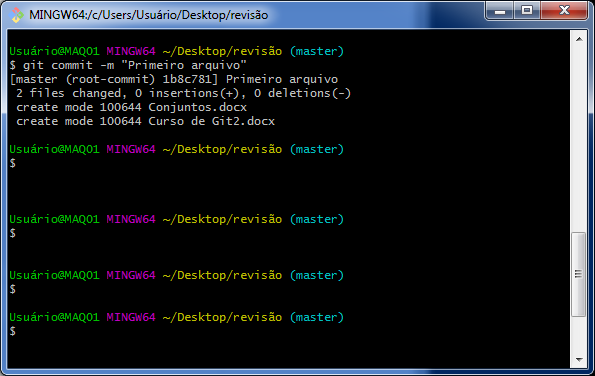


Figura 26 - Status mostrando que um arquivo foi adicionado e está pronto para ser enviado (comitado).

1. **COMITANDO ARQUIVO (COMMIT)**

Commitar é enviar as modificações para a área git directory (repositório local), no momento de executar o comit, devemos obrigatoriamente inserir uma mensagem indicando as alterações realizadas no arquivo. Os commits são todos locais.



Após realizar o commit, agora é possível enviar os arquivos para repositório remoto.

1. **ENVIADO ARQUIVOS AO SERVIDOR (PUSH)**

Para enviar os arquivos para o servidor, você deve fazer um PUSH, para isso você precisa saber o nome do seu servidor remoto. Entre na pasta onde estão os arquivos que deseja enviar ao servidor e execute os comandos abaixo.

Use o comando git remote, para saber o nome do servidor remoto.

PUSH

O remote add é usado para especificar a conexão do repositório local com o servidor remoto. Isso será útil quando precisarmos efetuar um ‘push,’ ou seja, enviar as modificações locais para o repositório remoto.

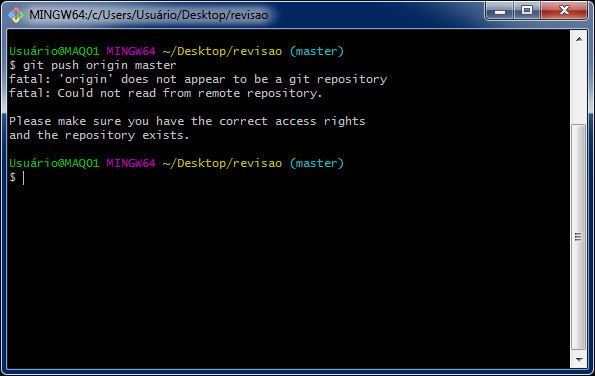


Figura 27 - Mensagem de erro, indicando que não existe um repositório remoto.