# Tutorial de Configuração e Operações Básicas do GitHub Para o Projeto GOT



**Salvador**

**2012**

# INSTALAÇÃO

Primeiro de tudo, baixe e instale os binários do Git para Windows, no link abaixo:

<http://code.google.com/p/msysgit/downloads/detail?name=Git-1.7.9-preview20120201.exe&can=2&q=>

caso esteja utilizando outro SO, procure no site do git o download.

Baixe o .exe e instale tudo default.... 100% next, next, Yes, finish.



Qualquer dúvida, confira no link abaixo:

<http://help.github.com/win-set-up-git/>

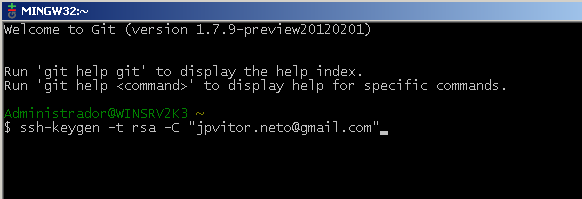
Após instalado, será criado um atalho para o “Git Bash”. É o terminal de comando do Github, o qual traz alguns programas nativos do Unix recompilados para Windows, como “ls”, “cat”, “ssh-keygen”, etc, além de, claro, os executáveis do próprio github, principalmente o executável “git”.

### CONFIGURANDO O GIT HUB PARA ACESSAR O REPOSITÓRIO REMOTO

Abra o GitBash e digite o seguinte comando:

$ ssh-keygen –t rsa –C “seuemail@seuemail.com”

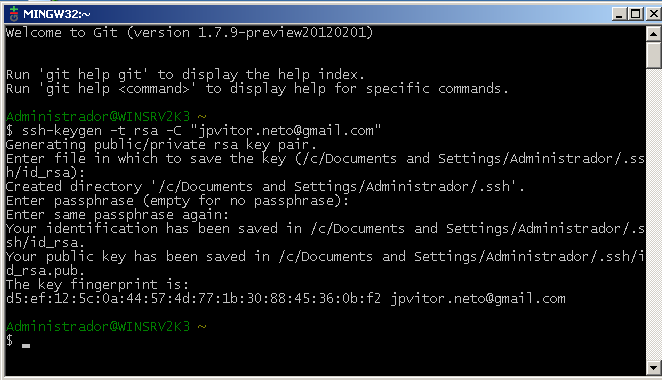
Esse commando é para gerar uma chave pública que será usada para acessar o servidor via ssh e poder realizar modificações no nosso repositório.



Em seguida, o Git vai pergutar a respeito do local e do arquivo onde será gravada a nova chave ssh pública, vamos manter o diretório padrão que é a **pasta raiz do usuário**. Para isso, pressione Enter sem digitar nada.

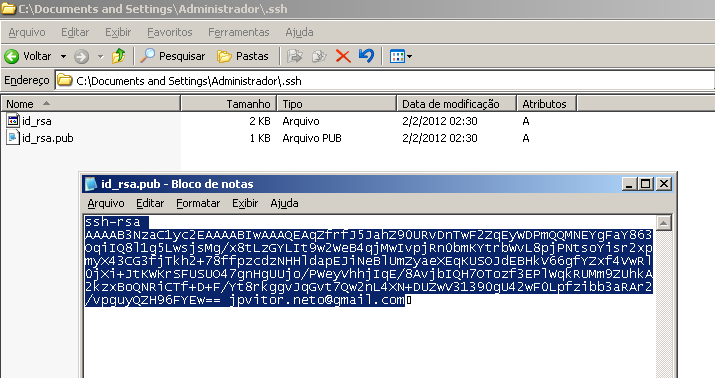
Depois o git vai solicitar que você insira uma senha que será associada à sua chave pública (“Enter passphrase:”). Essa é uma senha SUA, e será vinculada à sua chave ssh.

Após confirmada a senha, sua chave será gerada no diretório escolhido, e aparecerá na tela o seu fingerprint.

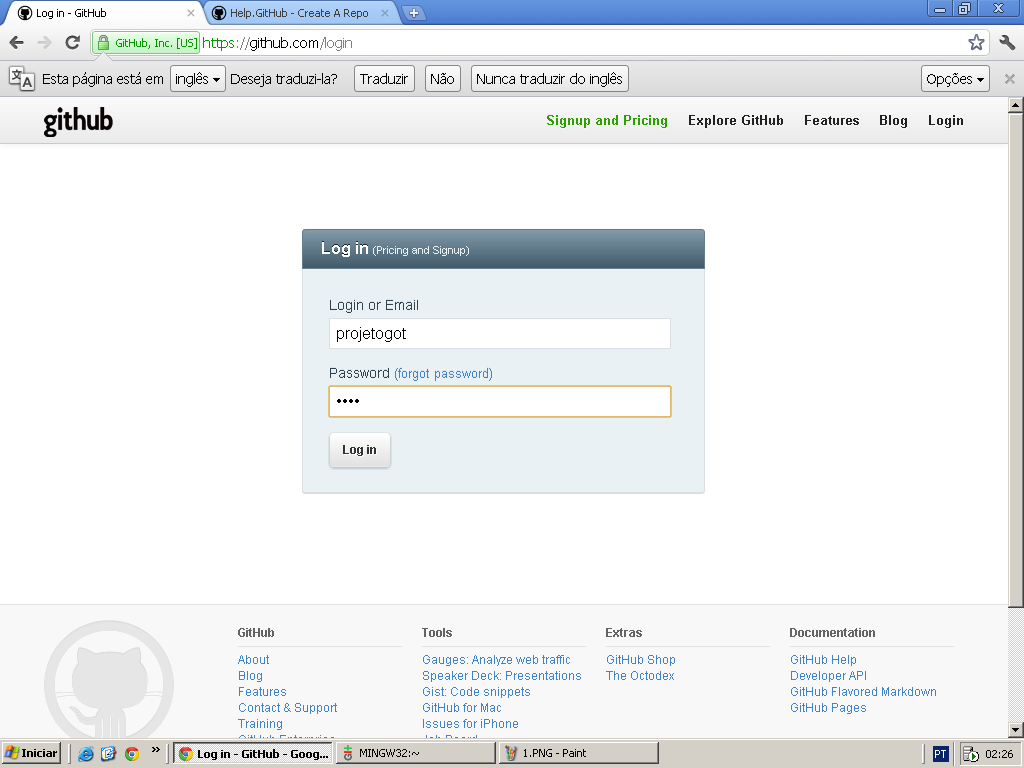


Cada pessoa pode ter uma ou mais chaves públicas de acesso ao repositório, uma para o notebook e outra para o PC, por exemplo.

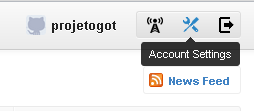
Agora vá no diretório raiz do usuário, no caso do win7 (c:\users\seu\_user), abra o arquivo id\_rsa.pub e copie o seu conteúdo... só não copie aquele caractere maluco que provavelmente é uma quebra de linha do Unix q o Windows n sabe ler... hehehe.



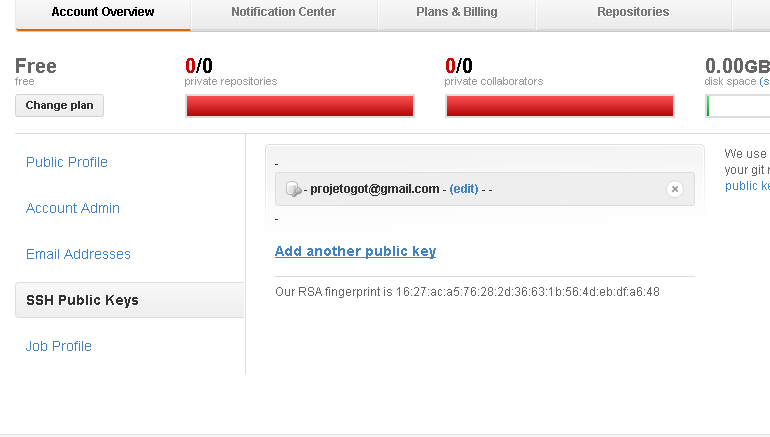
Agora vá até o site do github, e faça login no nosso usuário do projetogot, a senha está no email.

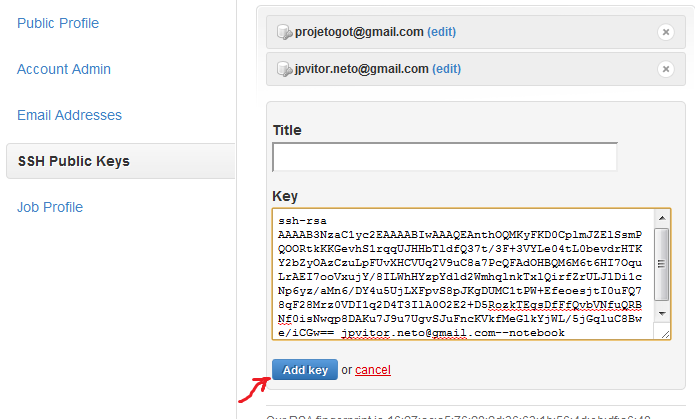


No canto superior direito, acesse as configurações da conta “Account settings”:



Clique em “SSH Public Keys” 🡪 “Add another public key”.



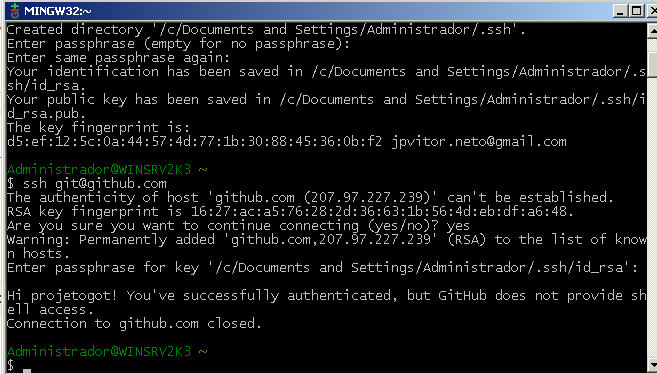


Pronto! Feito isso, nosso repositório já possui uma referência à sua chave ssh pública.

Por último, vamos realizar um acesso SSH ao Git para que o servidor deles seja adicionado à nossa lista de “known hosts”, para isso, no Bash, execute o seguinte comando:

$ ssh git@github.com

Vai aparecer uma mensagen de aviso como mostra abaixo, é isso msm, pode dar um “yes”.

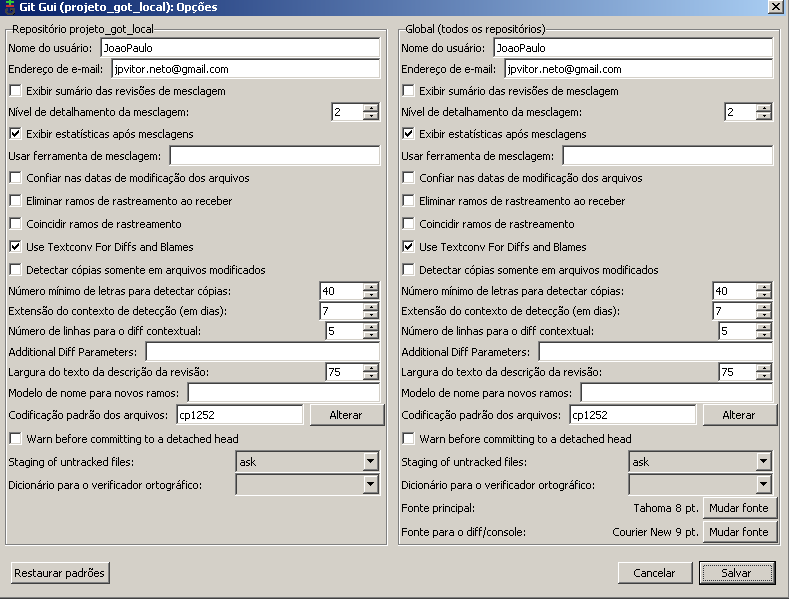
Depois de informar a SUA senha, você receberá a mensagem de boas vindas, como mostra acima: “Hi projetogot! You’ve successfully authenticated... bla bla bla”.

Isso indica que a validação da sua chave está Ok!

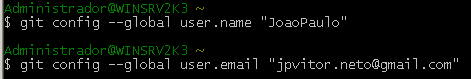
Agora vamos ao próximo passo...

## PERSONALIZAR SEUS DADOS NO GITHUB

Para que os outros usuários vejam quem efetuou qual modificação, é necessário personalizar suas informações no github, para isso, abra a GUI do Git, vá em Editar🡪 Opções, e informe seu nome e email nos dois ambientes (repositório atual e todos os repositórios):



Ou, no Shell, da seguinte forma:



Pronto, agora suas modificações serão devidamente referenciadas.

## CRIAR UMA CÓPIA LOCAL DO REPOSITÓRIO

A partir de agora, vou mostrar os procedimentos no Ambiente Gráfico do Git, o GUI GIT.

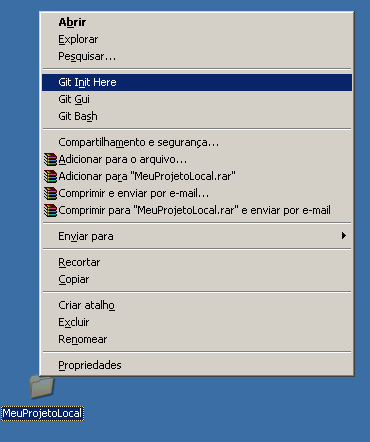
Crie uma nova pasta com um nome qualquer para criar a sua cópia local do repositório.

Depois clique com o botão direito e clique em “Git Init Here”, ou via Shell:

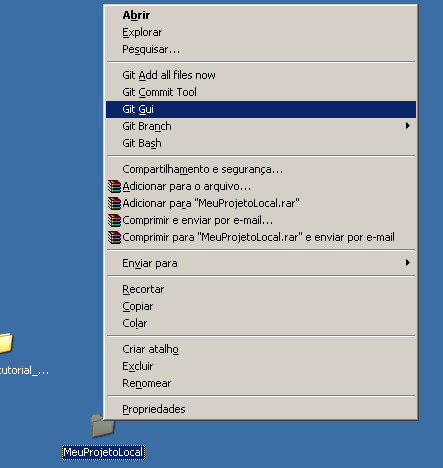
**$ mkdir MinhaPasta**

**$ cd MinhaPasta**

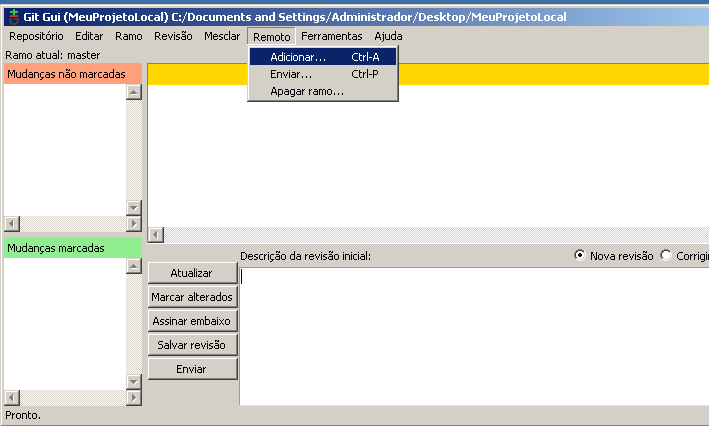
**$ git init**



Você pode acessar o ambiente gráfico clicando na pasta com o botão direito, no menu “Git Gui”

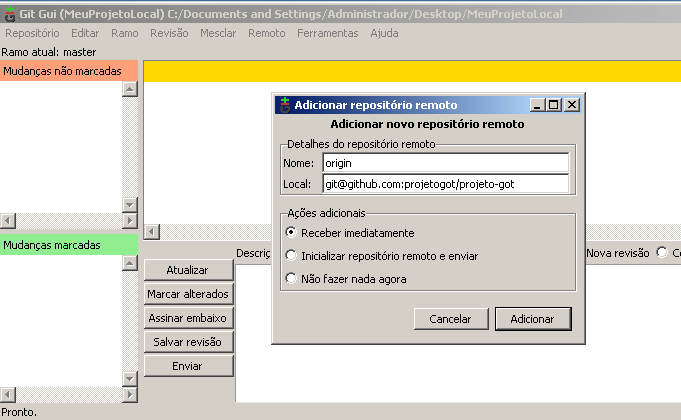


O próximo passo para configurar o repositório é adicionar a ele uma referência ao nosso repositório remoto, no servidor do github, faça o seguinte: clique em “Remoto” 🡪 “Adicionar”



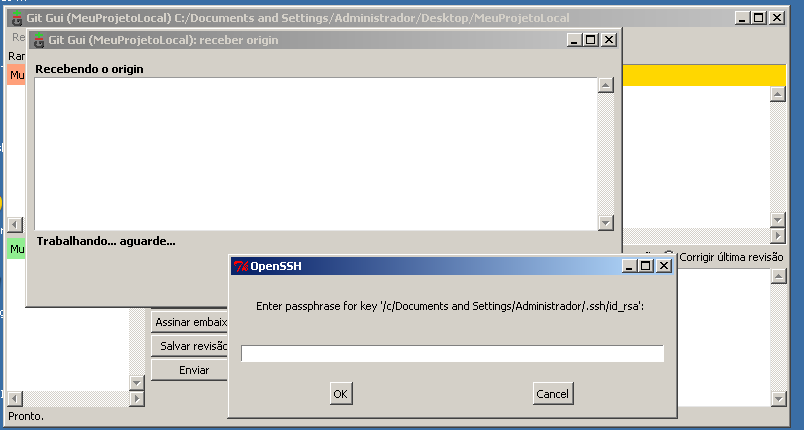
Para o **Nome** do repositório remoto, bote “**origin**”, é um padrão adotado para referenciar o servidor remoto principal.

Para o **Local** do repositório, digite o endereço do nosso repositório, que é: “**git@github.com:projetogot/projeto-got”**, explicando: projetogot é o nosso usuário no gh e projeto-got o nosso repositório deste usuário.



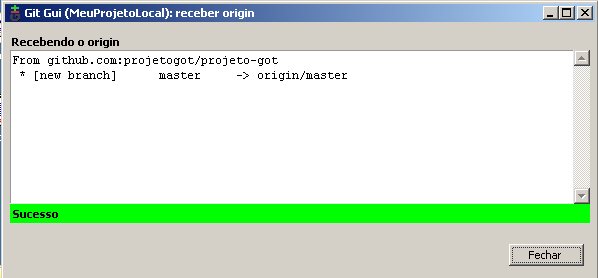
Feito isso deixe marcada a opção “Receber imediatamente” e clique em adicionar...

O github vai solicitar a SUA senha gerada na chave pública:



Confirmada a senha, o git vai adicionar o servidor remoto, e pelo que havia entendido ele já deveria baixar os arquivos remotos para a cópia local, mas não sei que por cargas d’água essa zorra nunca funcionou pra mim... Jájá vou explicar como faz direto no Shell, que esse sim funciona na paz...

Esse procedimento de receber demora um pouco, quando terminar deve aparecer uma msg de Sucesso:



Se no seu PC os arquivos aparecerem no repositório, só alegria, se não aparecer, vou mostrar como faz no Shell... execute o seguinte comando:

$ git pull origin master 🡨 significa, “puxe” da origem para o master

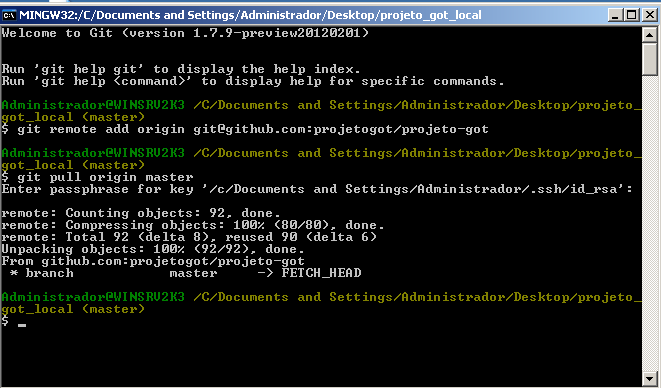
O master é o “branch” padrão de um repositório local. Branches são como “instâncias”, ou cópias do repositório original, onde podemos fazer a miséria que quisermos com nosso projeto e depois decidir se devemos enviar de volta ao servidor ou descartá-los.

Se você não chegou a adicionar a referência ao servidor remoto, lá no GUI, segue o comando para adicioná-lo direto no Git Shell:

$ git remote add origin git@github.com:projetogot/projeto-got

Novamente, insira sua senha e pronto.

Veja a imagem:



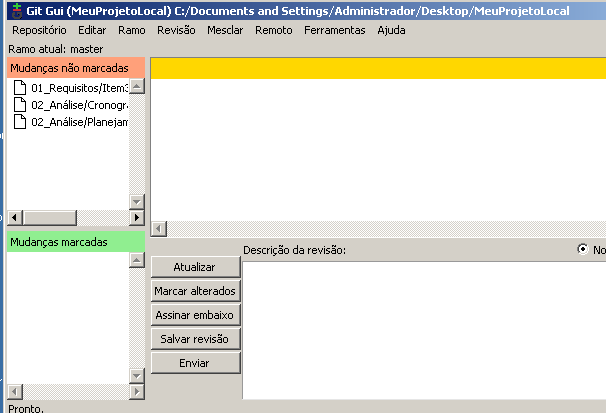
Pronto mans! A partir de agora, o repositório já está linkado ao servidor remoto, e você já pode mexer a vontade no repositório. Quando finalizar suas modificações no projeto, pode adicioná-las ao servidor, como vou mostrar agora.

## ENVIAR SEUS COMMITS PARA O SERVIDOR REMOTO

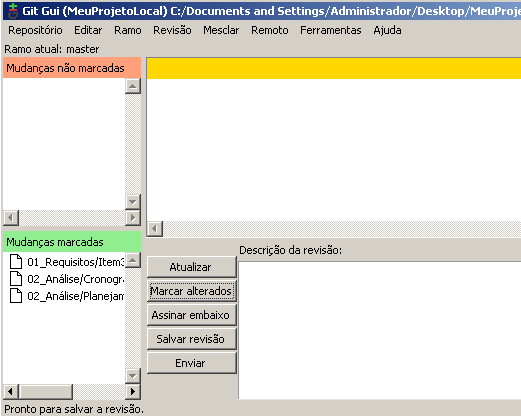
Até agora, nosso repo local já foi inicializado e já possui uma referência remota. O procedimento é o seguinte: as modificações que fizermos na pasta serão aplicadas localmente, até que, quando concluirmos nossas alterações, nos adicionamos (add) os novos arquivos ao próximo “commit”, efetuamos o commit, sempre vinculando um comentário, e por fim enviamos (push) essas alterações para o servidor. Da seguinte forma:

Recebi 3 novos arquivos de Lucas para enviar ao repositório. Coloquei cada um em sua respectiva pasta, Requisitos, Análise, etc... No ambiente gráfico, ao clicar em “Analisar”, o Git verifica que existem 3 novos arquivos em relação ao último commit que foi sincronizado:

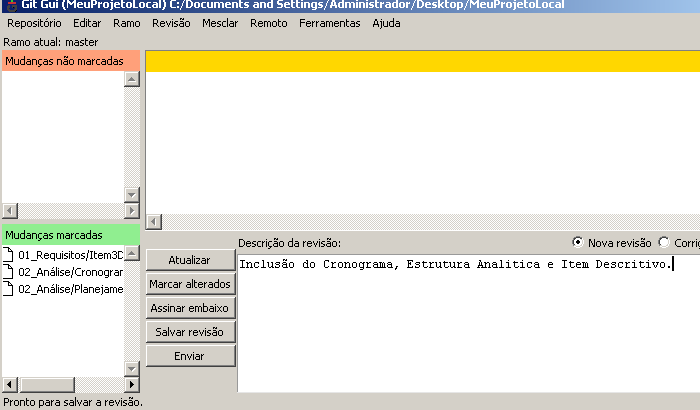
Veja em “Mudanças não marcadas”:



Ao clicar em “marcar alterados”, eles são marcados para “entrar no próximo commit”

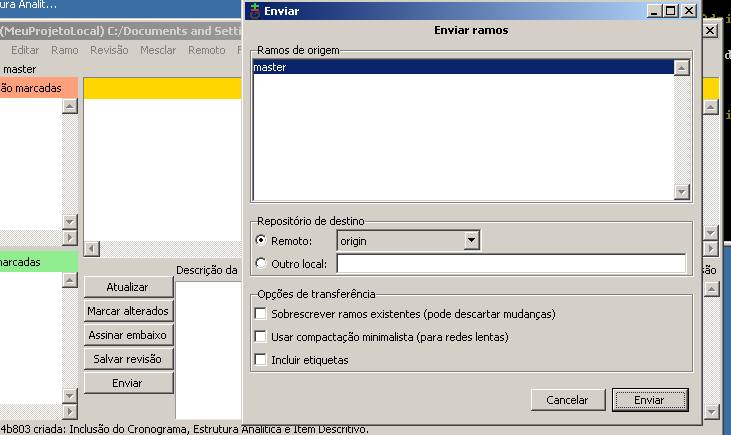


Agora, vamos adicionar um texto de descrição do próximo commit (o que ele chama de revisão, revisão == commit...), algo que seja ÚTIL e INFORMATIVO.

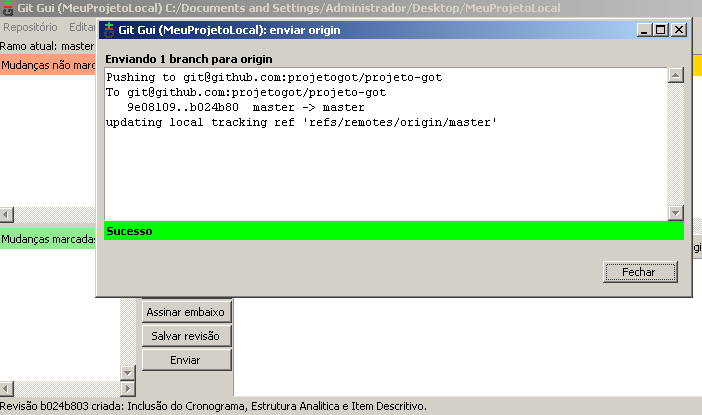


Ao clicarmos em “Salvar Revisão” o Git gera um “Commit” das nossas modificações. Um commit é como uma instância salva de um determinado momento do repositório.

Agora é só mandar “Enviar” (push), para enviar os commits gerados localmente para o servidor!



Insira SUA senha, e aguarde a transferência dos arquivos.



Segue abaixo a sintaxe de comandos no Shell para efetuar estas mesmas operações:

Neste ponto, presume-se que o repositório já foi inicializado e já foi adicionado o remote e que o diretório ativo no Shell é o repositório (cd MeuRepoLocal).

$ git add . 🡨 adiciona todos os arquivos modificados

ou

$ git add arquivo\_xpto 🡨 ou se quiser adicionar um arquivo específico...

$ git commit –m ‘comentario sobre as alterações do commit’ 🡨 commita com coment.

$ git push origin master 🡨 envie para a ‘origin’ o conteúdo do branch ‘master’...

É isso aew, isso tudo eu escrevi com base no que andei lendo nesses últimos 3 dias.

Pode ser que algo aí esteja errado ou sei lá, não sou nenhum expert, qualquer coisa manda um email para:

[jpvitor.neto@gmail.com](mailto:jpvitor.neto@gmail.com)

Valeu mizerês!

# REFERÊNCIAS

<http://help.github.com/>

<http://nathanj.github.com/gitguide/tour.html>

<http://www.ralfebert.de/tutorials/git/>