



[Criado por Super / Leonardo Claro](#)

Bom galera, eu estava querendo criar esse tutorial a meses para vocês, porém estava sem tempo, por causa de acumulo de serviços na empresa...

A ideia desse tutorial é de passar 1 conhecimento a vocês que vai ser util atualmente com o Mugen , e talvez até futuramente caso queira seguir carreira em desenvolvimento, pois vejo você como futuros desenvolvedores, e tenho certeza que alguns já são e trabalham como analista de desenvolvimento...

O principal foco de eu esta criando esse tutorial, é com intuito de ajudar no desenvolvimento de projetos referentes ao Mugen (jogos completos, chars, stages, screenpack, e etc), podendo ser usado em equipe (que é o principal foco), ou até para sí mesmo, para não perder seus projetos ou poder trabalhar em vários micros com o mesmo projeto, que é até meu caso...

Primeiramente gostaria de explicar um pouco sobre repositório.

A principal função do repositório, é o armazenamento de arquivos em 1 local, com descrição feita pelo desenvolvedor quando efetuar o envio, de onde pode ser recuperado remotamente por outros usuários do projeto, podendo ver a descrição de quem envio o arquivo(s). Todas fabricas de software utilizam repositórios na Internet ou em outras maquinas na própria rede (normalmente no próprio servidor de arquivos da empresa). No nosso iremos utilizar o Google Code, por ter capacidade de 2 gigas inicial(pois vai aumentando), e de o Mugen ser Open Source, vai atender totalmente nossas especificações.

Bom, agora que descrevi um pouco sobre o que é repositório, gostaria de falar ensinar a implementar ele, e repito, ele é totalmente útil para que dificulte a perda de seus projetos por problemas com seu pc.

Peço primeiramente que baixe o Tortoise HG no link abaixo:

<http://tortoisehg.bitbucket.org/download/index.html>

Nós iremos seguir os seguintes passos para implementarmos

- 1 - Criando conta google caso não tenha
- 2 - Criando projeto no google code
- 3 - Configurando "Clone"
- 4 - Utilização



[Criado por Super / Leonardo Claro](#)

## 1 - Criando conta google caso não tenha

Caso tenha conta no google pode pular para o próximo passo, caso não, acesse o endereço abaixo, e siga todos passos nele até efetuar o cadastro completo.

<https://www.google.com/accounts/NewAccount?continue=http%3A%2F%2Fwww.google.com.br%2F&hl=pt-br>

## 2 - Criando projeto no google code

Agora que já temos 1 conta do google, acesse o endereço abaixo para podermos criar 1 repositório no google code, onde será armazenado nossos projetos:

<http://code.google.com/hosting/createProject>



[Criado por Super / Leonardo Claro](#)

## Create a new project

Instantly create your open source hosting project by filling out the form below. For your project, you'll receive:

- Subversion and Mercurial code hosting with 2 gigabytes of storage space
- Download hosting support with 2 gigabytes of storage space
- Integrated source code browsing and code review tools
- An issue tracker and project wiki

[Learn more](#)

**Project name**

Example: my-project-name

**Project summary**

**Description**

**Version control system**

**Source code license**

**Project label(s)**

[add another row](#)

**Create project**



[Criado por Super / Leonardo Claro](#)

Em Project summary, você irá por o nome do seu projeto, em “Description” você irá por a descrição detalhada de seu projeto, em “Version control system” você irá colocar “Mercurial” que é o mais indicado a se usar por ser 1 controle de versão “distribuído” (não desejo entrar muito em detalhes, pois senão sairemos 1 pouco do rumo), em “Source code license” você irá selecionar a licença “Artistic License/GPL” que acho o mais indicado nos projetos de Mugen, em “Project label(s)” você irá colocar palavras chaves referentes ao seu projeto, para que seja localizado nas buscas, depois basta clicar em “Create project”.

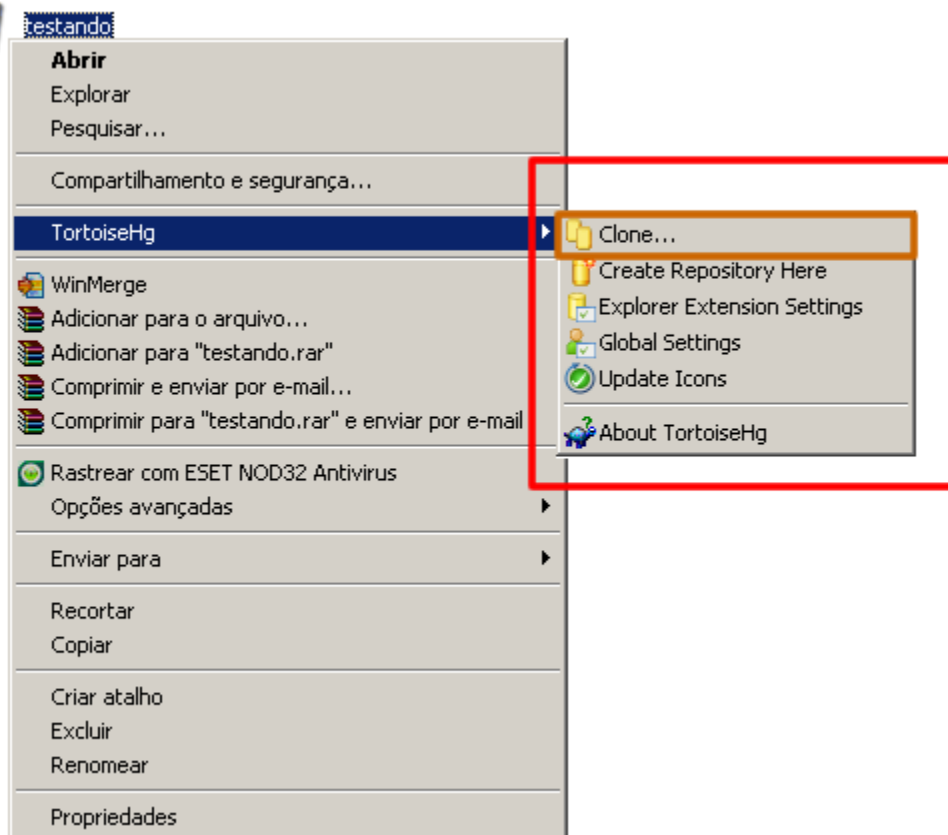
Pronto, você acabou de criar 1 repositório mercurial no google code!!!

### 3 - Configurando “Clone”

Bom, para prosseguirmos agora, é necessário termos o “Tortoise HG” instalado no computador, caso não tenha baixado e instalado e queira instalar, clique no link abaixo para poder pegar a versão mais recente:

<http://tortoisehg.bitbucket.org/download/index.html>

Agora que temos o Tortoise HG instalado corretamente, e clicarmos com o botão direito em qualquer pasta, irá aparecer como na figura abaixo.



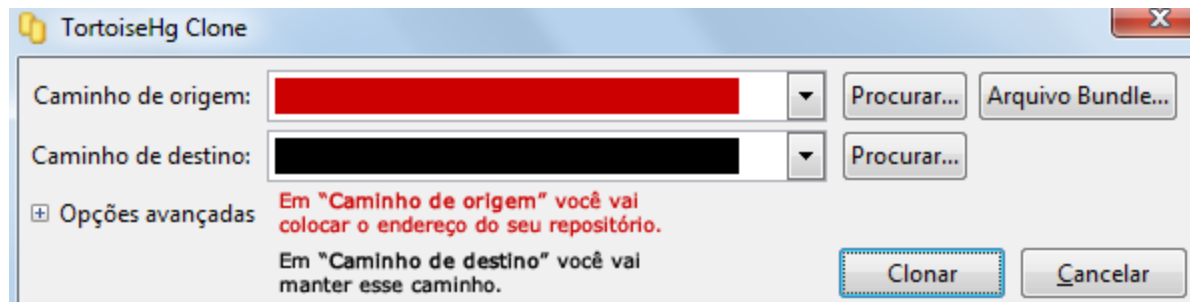
[Criado por Super / Leonardo Claro](#)

Temos agora o Tortoise HG instalado corretamente, poderemos agora prosseguir, você irá agora escolher 1 diretório vazio com o nome do seu projeto, e depois clicar com botão direito do mouse por cima dele, e depois selecionar “Tortoise HG” e clicar em “Clone..” .

Irá aparecer o seguinte:



[Criado por Super / Leonardo Claro](#)



Você irá editar apenas o “Caminho de origem” colocando o caminho do seu repositório no google code que você criou. Para saber qual é, você terá que abrir o repositório que você criou no google code, e irá clicar em "Source", lá vai aparecer o endereço do clone do seu repositório no google code, por exemplo:  
hg clone https://teste.googlecode.com/hg/ teste

Ai você irá pegar apenas o endereço sem o comando e usuário, por exemplo:  
hg clone https://teste.googlecode.com/hg/ teste

você irá pegar apenas o:  
<https://teste.googlecode.com/hg/>

e colocar em “Caminho de origem” e depois clicar em “Clonar”.

Pronto, com isso você criou 1 clone na sua máquina do seu repositório online.

#### 4 - Utilização

Agora vou ensinar o básico para iniciar o uso do seu repositório com o Tortoise Hg.

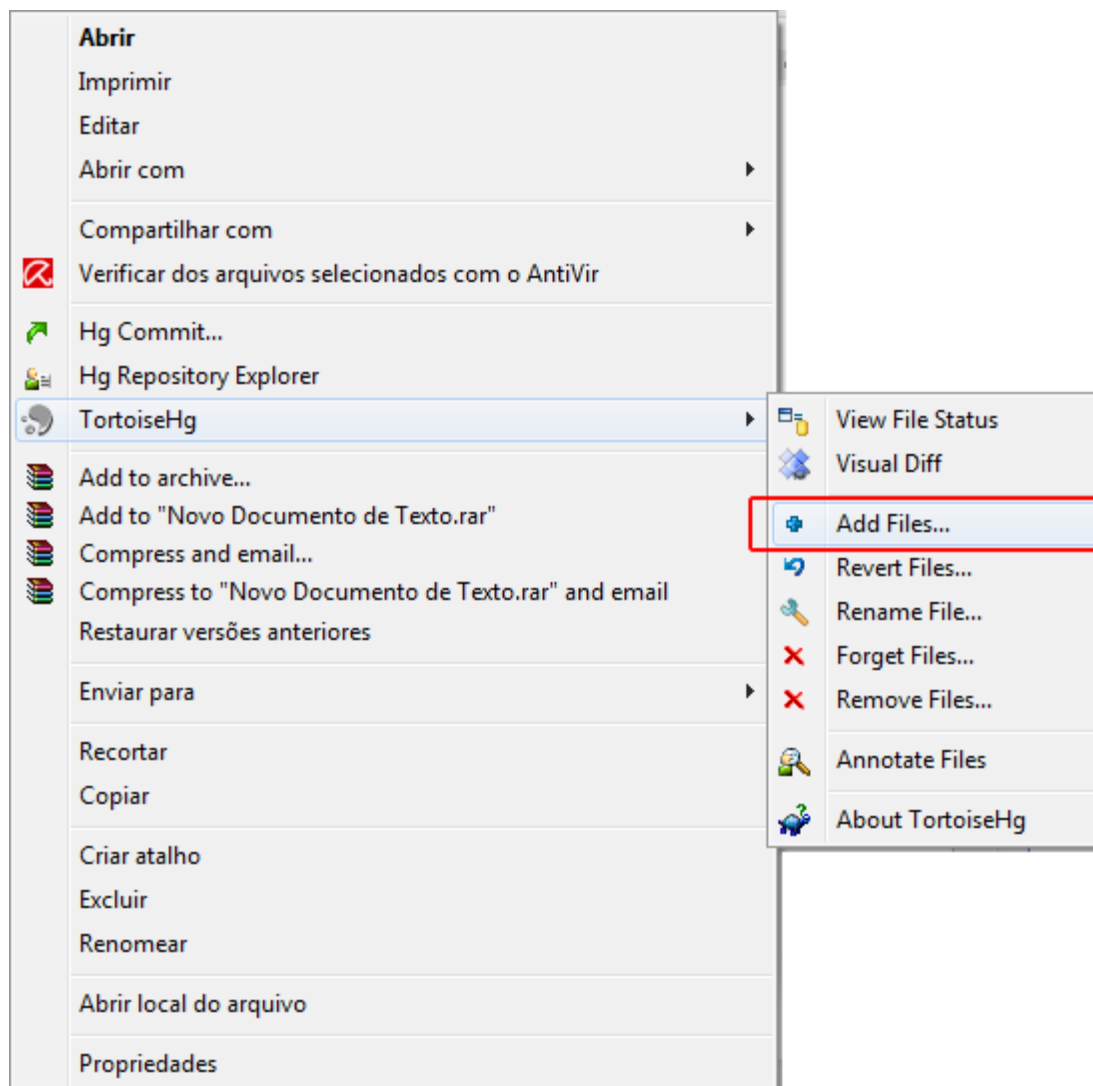


[Criado por 5uper / Leonardo Claro](#)

Peço que crie ou coloque algum arquivo ou projeto completo dentro da pasta que você criou o clone na sua máquina, após colocar o conteúdo, você vai selecionar todo conteúdo que você colocou no diretório que criou o clone, e clicar com botão direito do mouse, vai selecionar “TortoiseHg”, e depois vai clicar em “Add Files...”



[Criado por Super / Leonardo Claro](#)



Caso seja apenas 1 arquivo vai adicionar ele automaticamente, caso não, vai abrir 1 nova janela para você confirmar a seleção,





[Criado por Super / Leonardo Claro](#)

caso a seleção esteja correta, você deverá clicar em “Adicionar” para confirmar o adcionamento dos arquivos, feito isso, o servidor da google, não irá permitir que outro usuário da equipe com clone do projeto em outra maquina, crie o arquivo com o mesmo nome, porém ainda você não enviou ao servidor o arquivo, apenas reservou ele lá...

Agora vamos salvar o arquivo no “Clone” local, você irá clicar com botão direito sobre o diretorio que você colocou como “Clone” e vai clicar em “Hg Commit...”, irá abrir 1 nova janela, lá você terá que selecionar o campo vazio e digitar a descrição que é muito importante para o entendimento da equipe, por isso é sempre necessario que faça uma descrição detalhada de que foi alterado ou adicionado, para que o seu companheiro da equipe, ou você mesmo possa entender o significado dessa “Revisão”, todos Commit que você der, é chamado de nova revisão, a revisão é por arquivo, porém não desejo entrar com muitos detalhes nesse tutorial, qualquer dúvida eu irei responder no forum. Após descrever corretamente, você irá clicar em Commit (Consolidar).

Bom galera, como o repositório “Mercurial” é 1 repositório distribuido, você apenas salvou as alterações localmente na sua máquina, ainda as alterações não foram submetidas ao servidor web (google code), para que as alterações sejam guardadas no nosso repositório web, iremos fazer 1 novo processo muito importante... Você irá clicar com botão direito sobre o diretorio “Clone” seu, e depois em “Hg Repository Explorer”, irá abrir uma nova janela, nela você irá no botão “push”, é a setinha para cima com um traço acima da seta, passando o mouse por cima dela vai aparecer o nome da setinha, com isso irá enviar todas revisões locais que não estão presentes no repositório web (google code).

Agora vocês devem estarem pensando como pegar as alterações(atualizações/revisões) feitas por outro membro da minha equipe, ou por você mesmo que você realizou em outra maquina certo?

Então agora irei ensinar como fazer para receber essas alterações (atualizações/revisões). Você irá clicar no diretório do seu “Clone” com botão direito do mouse, e depois em “Hg Repository Explorer”, após ter feito isso, irá abrir uma nova janela, e você deverá clicar em “Pull”, que é uma setinha para baixo com um traço debaixo dela, com isso você irá pegar todas “alterações(atualizações/revisões)”, realizadas e que não tenha nesse “Clone”, porem para acabar de realizar o processo, você terá que clicar novamente com botão direito do mouse em cima do “Clone” e depois em “update”, ai vai perguntar se quer realmente fazer a atualização, e você clica “Ok”, com isso terá efetivado as alterações no seu projeto, ficando igual ao do repositório web...

Galera caso você deseje criar novo “clone”, com o repositório web que ja contenha arquivos, é muito simples, é só você fazer o “passo 3” desse tutorial que irá descer automaticamente todos arquivos do repositório web.



[Criado por Super / Leonardo Claro](#)

Outras coisa que acho muito importante explicar a vocês, sempre que trabalhar em equipe e for começar a mexer no projeto, aconselho sempre a dar o “Pull” e “update” para ver se não tem novas alterações, para não correr o risco de editar algum arquivo com revisão desatualizada, porque senão quando der o seu “Commit” e “Push”, poderá sobrescrever o trabalho do seu companheiro de equipe, e em casos piores ocasionar “conflitos” no seu repositório, criando 2 margens de tempos, que é bem chatinho de resolver rsrs.

Mais uma coisa, sempre que for mexer em algum arquivo, e estiver trabalhando com o projeto em equipe, você deverá avisar ao seu companheiro, em fabricas de software utilizamos muito o “trac” que é 1 gerenciador de tarefas, que evita 1 pouco de ocorrer o uso de edição do mesmo arquivo, pois sempre quando dividimos tarefas, colocamos tarefas que não terá muitas chances de edição do mesmo arquivo por pessoas diferentes, sem que ambas saibam...

U\_U tem mais outra coisa que acho bastante importante, caso queira trabalhar local sem internet com o repositório, é possível sim em repositório distribuído que é nosso caso (Mercurial), você apenas irá trabalhar com o “Commit” para salvar no repositório local, e “update” para atualizar, aí você deve estar me perguntando para que isso? ou como usaria isso, ou estaria falando que isso é inútil U\_U, porém não é inútil não, pois com isso você terá uma linha do tempo do seu projeto todo, caso você faça algo errado, poderá puxar a revisão do arquivo de qualquer data de commit que você deu, ou seja, terá todos os commits de seu projeto, podendo recuperar o início do seu projeto completo, assim caso você edite algo em algum arquivo, ou ele tenha corrompido, você poderá recuperar qualquer revisão do mesmo... porém aconselho sempre trabalhar com o repositório web para que tenha acesso ao projeto em qualquer lugar...