**FACULDADE DE TECNOLOGIA SENAC GOIÁS**

**Gestão em Tecnologia da Informação**

LogoSenac

Gabriel Zerbine

**Gerência de Configuração**

Prof. Elias

GOIÂNIA,

2017

Sumário

[1. O que é a gerência de configuração? 3](#_Toc478899071)

[2. O que são itens de configuração? 3](#_Toc478899072)

[3. De acordo com o MPS.BR, quais os resultados esperados para o processo de gerência de configuração? 3](#_Toc478899073)

[4. Como se dá o processo de gerência de configuração (exemplifique e indique softwares/ferramentas que auxiliem esse tipo de gerência)? 4](#_Toc478899074)

[5. GitHub 5](#_Toc478899075)

[ O que é e qual sua função? 5](#_Toc478899076)

[ Guia de uso (tutorial passo-a-passo) (Instalação/configuração, criar conta, postar arquivos.... Comandos básicos) 6](#_Toc478899077)

[ Criar uma conta e postar um ou mais documentos (informar o endereço da sua conta no trabalho) 6](#_Toc478899078)

[ Fazer alterações no(s) documento(s) e envia-lo novamente ao repositório (capturar "print" da tela e postar no trabalho) 6](#_Toc478899079)

[Comandos Básicos 13](#_Toc478899080)

[Endereço do Repositório criado 13](#_Toc478899081)

# O que é a gerência de configuração?

A Gerência de Configuração de Software, ou como é conhecida também pelas siglas GCO, é um conjunto de atividades de apoio que permite a absorção ordenada das mudanças inerentes ao desenvolvimento de software, mantendo a integridade e a estabilidade durante a evolução do projeto.

# O que são itens de configuração?

Um item de configuração, que também é conhecido pela sigla IC, é qualquer componente de serviço, elemento de infraestrutura ou outro item que precisa ser gerenciado para garantir a entrega bem-sucedida de serviços.

# De acordo com o MPS.BR, quais os resultados esperados para o processo de gerência de configuração?

GCO1 - Um Sistema de Gerência de Configuração é estabelecido e mantido.

GCO2 - Os itens de configuração são identificados com base em critérios estabelecidos.

GCO3 - Os itens de configuração sujeitos a um controle formal são colocados sob baseline.

GCO4 - A situação dos itens de configuração e das baselines é registrada ao longo do tempo e disponibilizada.

GCO5 - Modificações em itens de configuração são controladas.

GCO6 - O armazenamento, o manuseio e a liberação de itens de configuração e baselines são controlados.

GCO7 - Auditorias de configuração são realizadas objetivamente para assegurar que as baselines e os itens de configuração estejam íntegros, completos e consistentes.

GCO 8 - As informações de itens de configuração são comunicadas às partes interessadas.

# Como se dá o processo de gerência de configuração (exemplifique e indique softwares/ferramentas que auxiliem esse tipo de gerência)?

A gerência de configuração é aplicável para artefatos que requerem um controle mais formal tanto no contexto de trabalhos como no contexto organizacional. Temos vários tipos de artefatos, alguns podem ser armazenados com um simples controle de acesso, outros podem ser versionados sem necessidade de controle formal de mudança. Definir quais são os artefatos que farão parte da analise, faz parte do processo de Gerência de Configuração.

Para a melhor execução do processo de Gerência de Configuração, podemos observar três sistemas principais:

1. Sistema de controle de modificações: esse sistema tem a função de executar a função de controle da configuração de forma sistemática, armazenando todas as informações geradas durante o andamento das solicitações de modificação e relatando essas informações aos envolvidos, assim como estabelecido pela função de contabilização da situação da configuração;

2. Sistema de controle de versões: permite que os itens de configuração sejam identificados, segundo estabelecido pela função de identificação da configuração e que eles evoluam de forma distribuída e concorrente, porém disciplinada. Essa característica é necessária para que diversas solicitações de modificação efetuadas na função de controle da configuração possam ser tratadas em paralelo, sem corromper o sistema de Gerência de Configuração como um todo;

3. Sistema de gerenciamento de construção: automatiza o processo de transformação dos diversos artefatos que compõem um trabalho no sistema de serviço propriamente dito, de forma aderente à função de gerenciamento de liberação e entrega. Além disso, esse sistema estrutura as baselines selecionadas para liberação, conforme necessário, para a execução da função de avaliação e revisão da configuração.

Para auxiliar e garantir a execução das atividades do processo Gerência de Configuração, uma organização pode definir uma equipe de Gerência de Configuração, normalmente única, abrangendo toda a organização. Além dessa equipe, pode ser definido o Comitê de Controle da Configuração (CCC), que é um grupo de pessoas responsável por avaliar e aprovar ou rejeitar modificações propostas em itens de configuração, e certificar que as modificações aprovadas foram implementadas. Esse grupo pode ser definido por trabalho e ter tamanho variado, dependendo de suas necessidades. Possíveis membros desse grupo são: líder e alguns membros chave da equipe de Gerência de Configuração, líder do trabalho, representantes da equipe de Garantia da Qualidade, representantes da equipe de marketing e representantes do cliente. Este processo é sincronizado com o processo de Gerência de Mudanças (GMU), quando a organização estiver no Nível E de maturidade do MR-MPS-SV.

Algumas ferramentas que podem ajudar nessa Gerência de Configuração de Software são: Subversion, Trac, Hudson, Selenium, Checkstyle, Mantis, Jenkins e o GitHub, que trataremos no ponto seguinte.

# GitHub

## O que é e qual sua função?

É uma ferramenta que foi lançada em 2008 e que é usada para hospedar os projetos dos desenvolvedores. Hoje, ela é a ferramenta de hospedagem preferida dos desenvolvedores por ter muitas vantagens.

Uma vantagem dele é funcionar muito bem com o Git. Ele não é a mesma coisa que o GitHub! O Git é um sistema de controle de versão distribuído. Isso quer dizer que por ele, o desenvolvedor pode, a cada passo no desenvolvimento do projeto, ir criando versões e se por acaso uma delas der problema, ele terá a possibilidade de pegar a versão anterior. O GitHub consegue pegar todas as informações geradas pelo Git e armazená-las da mesma forma que o Git, isso quer dizer do ponto de vista histórico do projeto.

Outra vantagem é o compartilhamento de projetos, que foi onde os desenvolvedores viram uma vantagem enorme. Lá eles conseguem encontrar projetos parecidos com os seus e reutilizar códigos de outros.

Como o GitHub ficou famoso nessa área, ele se tornou um portfólio para os desenvolvedores, um local onde eles podem demonstrar como eles programam.

Ele se tornou uma rede social dessa área. Ele possibilita que um desenvolvedor siga o perfil de outro, receba notificações e ainda é possível enviar mensagens diretas para um determinado perfil, o que facilita a troca de experiências, além claro, de fazer novas amizades.

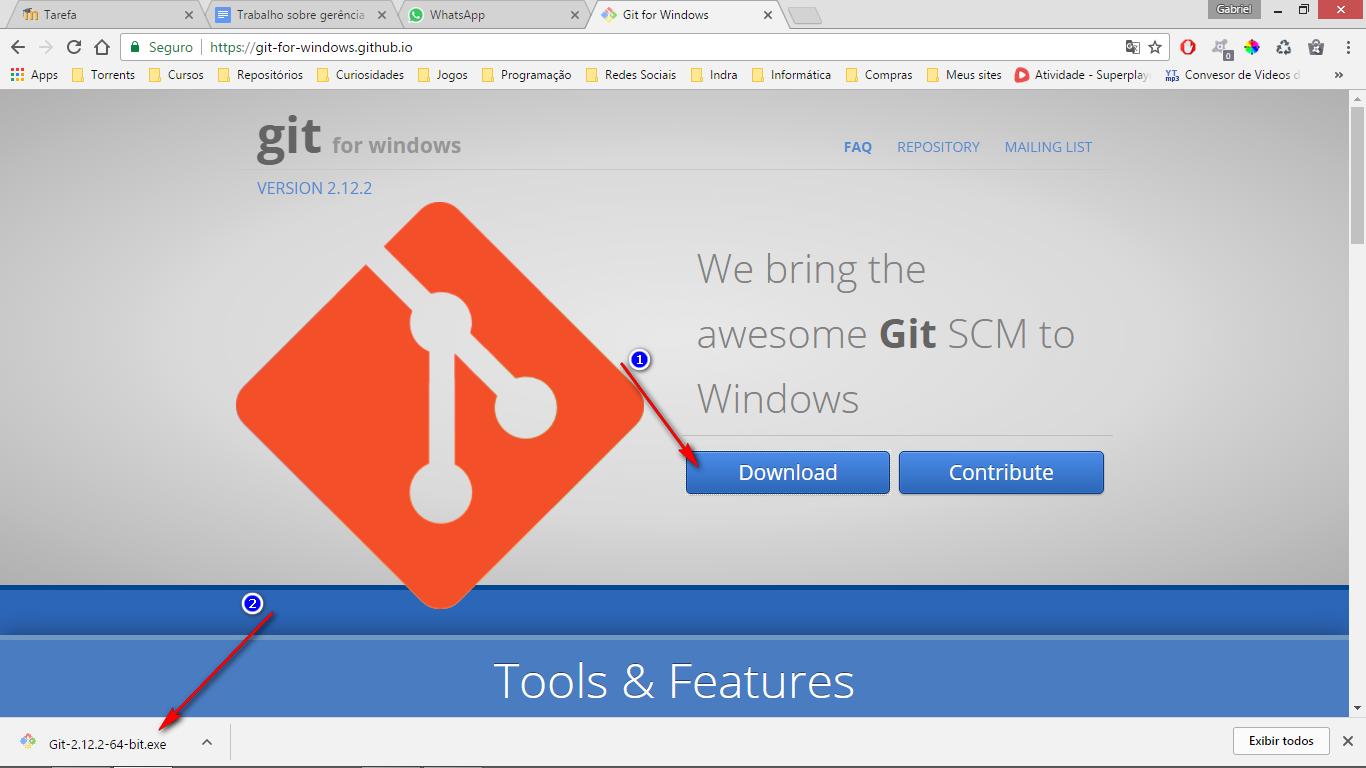
Através do GitHub os usuários ainda podem aprender mais pela observação dos demais projetos, bem como a contribuição dos demais participantes.

## Guia de uso (tutorial passo-a-passo) (Instalação/configuração, criar conta, postar arquivos.... Comandos básicos)

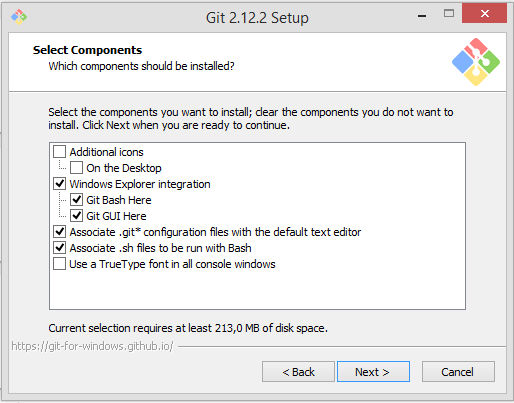
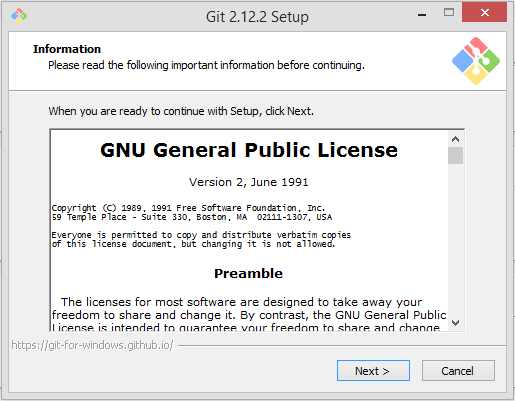
## Criar uma conta e postar um ou mais documentos (informar o endereço da sua conta no trabalho)

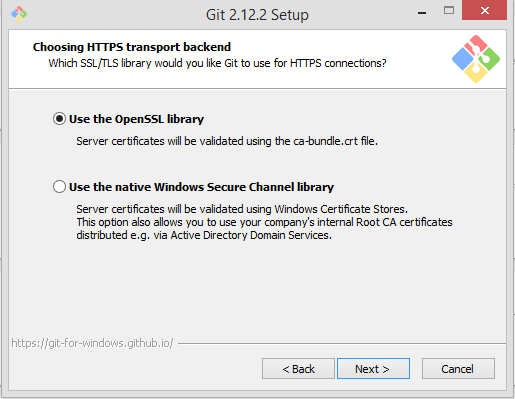
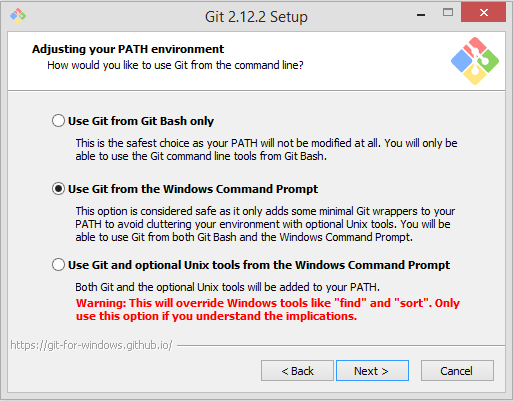
## Fazer alterações no(s) documento(s) e envia-lo novamente ao repositório (capturar "print" da tela e postar no trabalho)

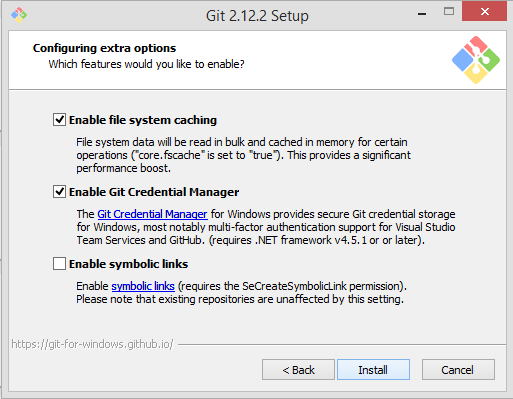
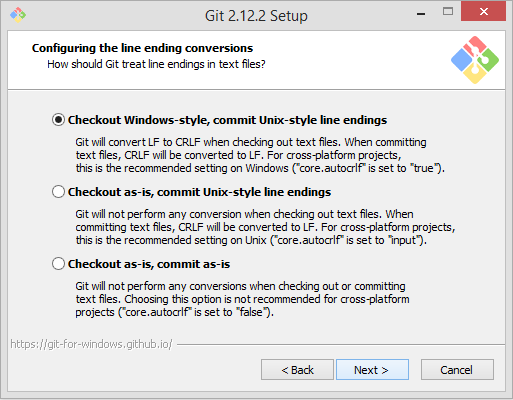
Primeiramente será necessário instalar o Git. Para isso deve-se baixar o executável do programa no site <http://msysgit.github.com>



Depois que o download terminar, clicar no executável para começar a instalação. O processo é simples!

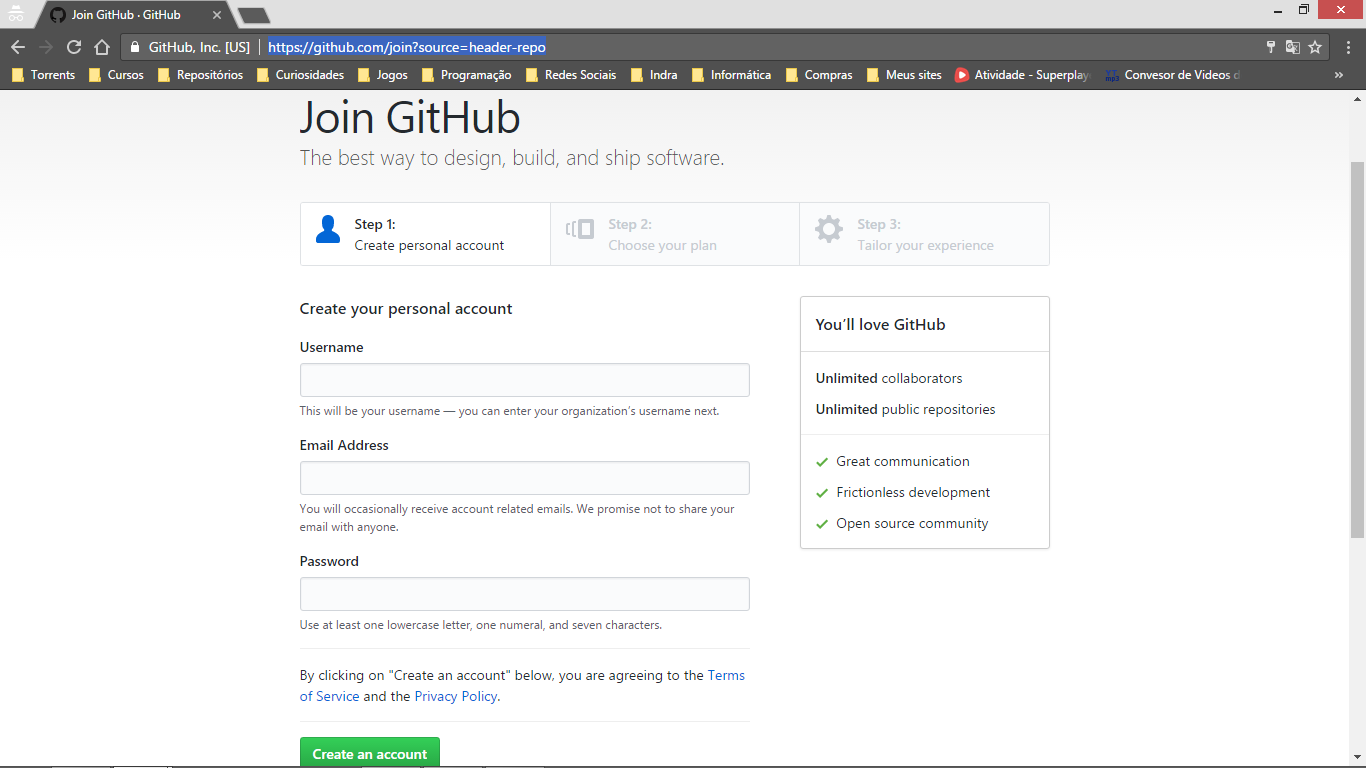




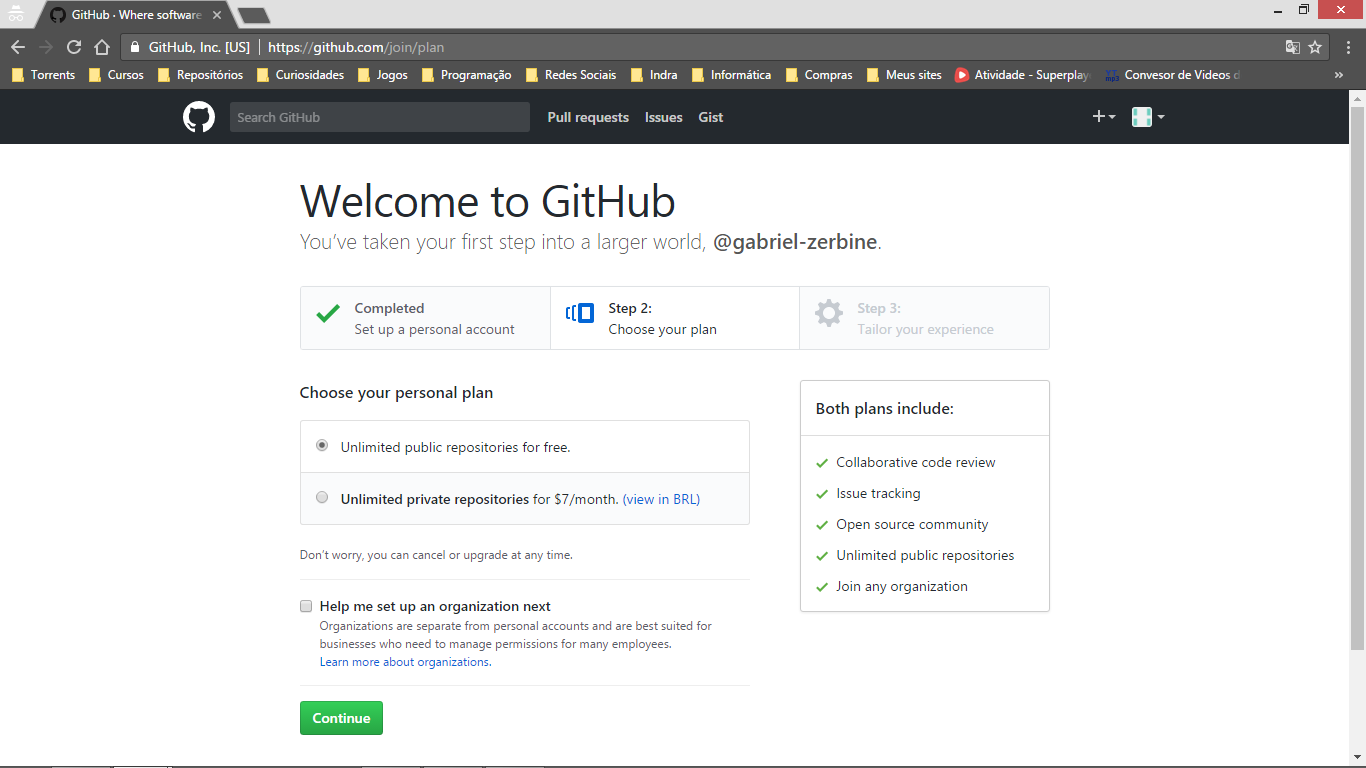


Uma dica, no terceiro print, o git permite de usar os seus comandos na linha de comando do Windows. É só deixar a opção que eu escolhi.

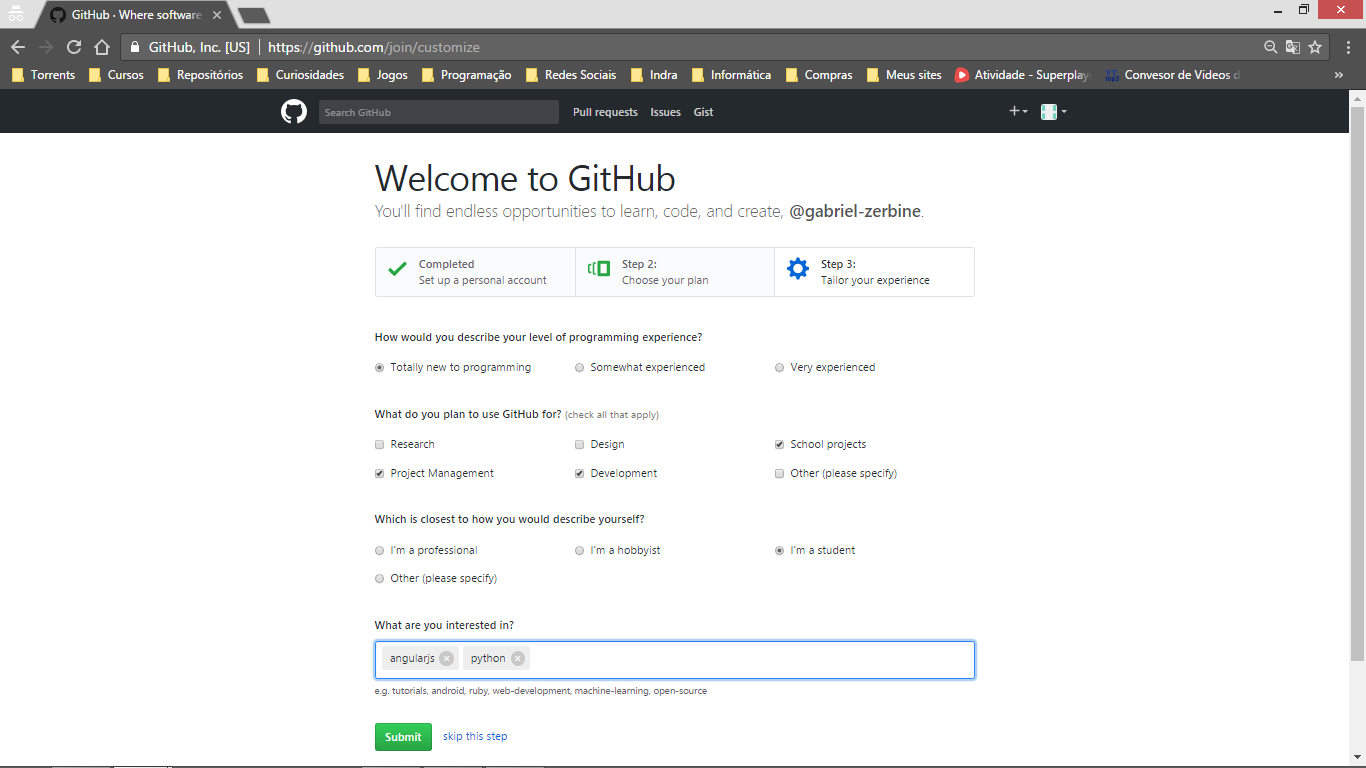
Agora temos que criar uma conta no GitHub. Para a criação, devemos acessar o site <https://github.com/join?source=header-repo> e preencher o primeiro formulário e depois clicar em “Create an account”:



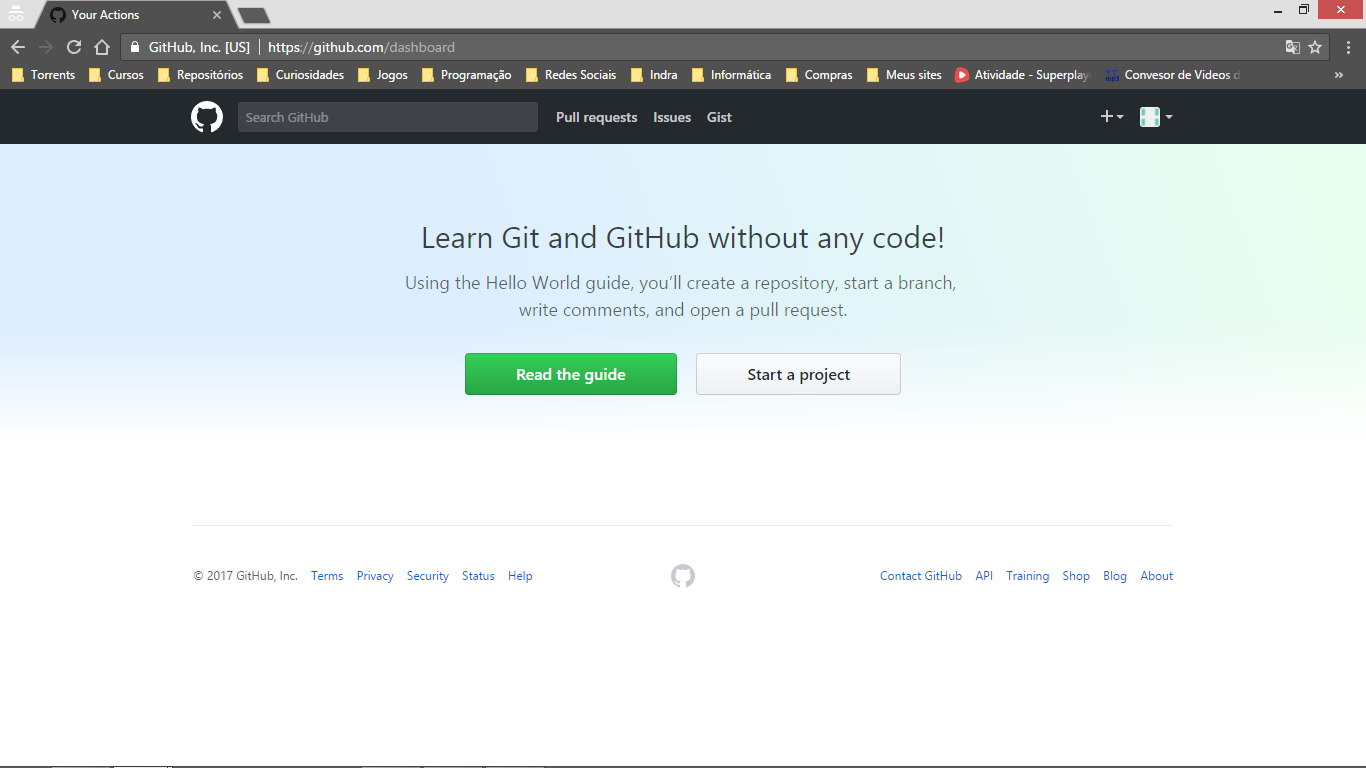
Depois ele te pedirá para escolher o plano. Tem o gratuito e o pago que custa $7 por mês.

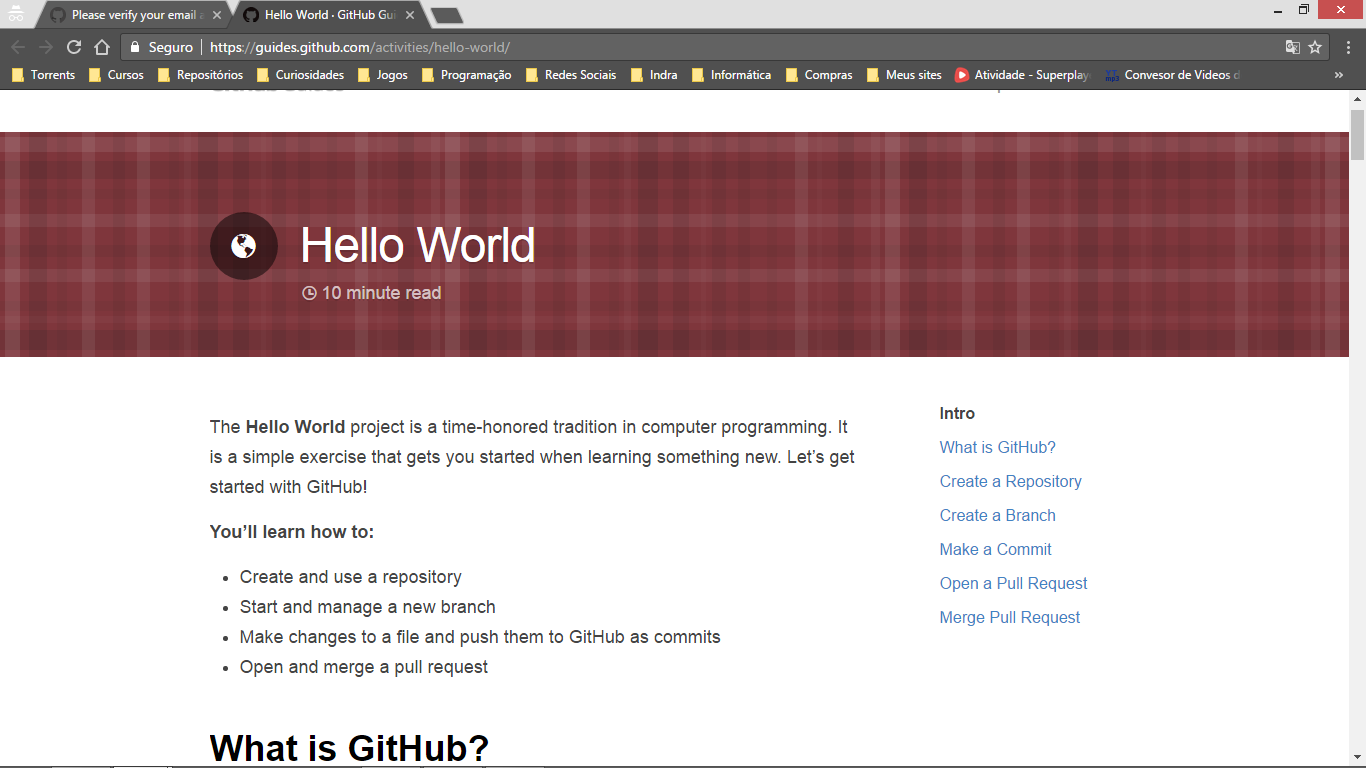


Ele te fará um questionamento para saber o que te interessa, é opcional.

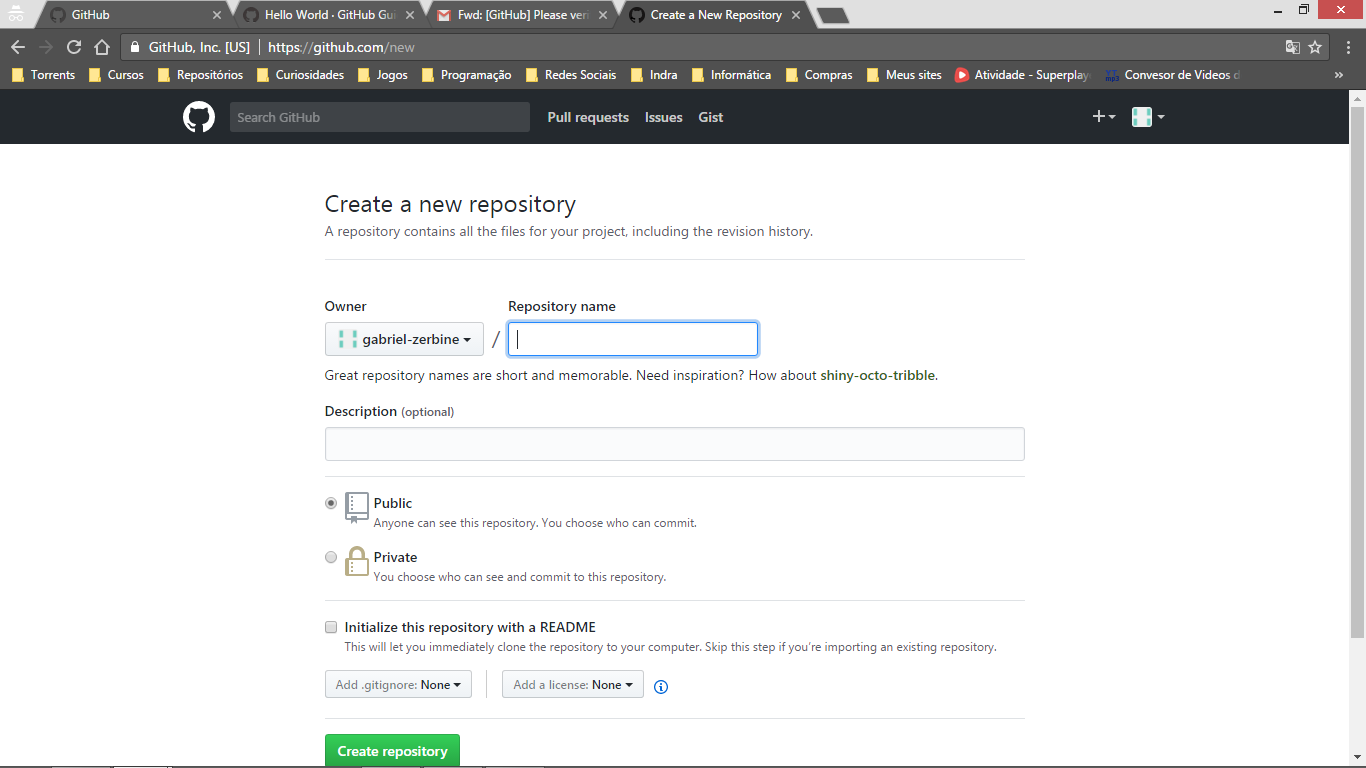


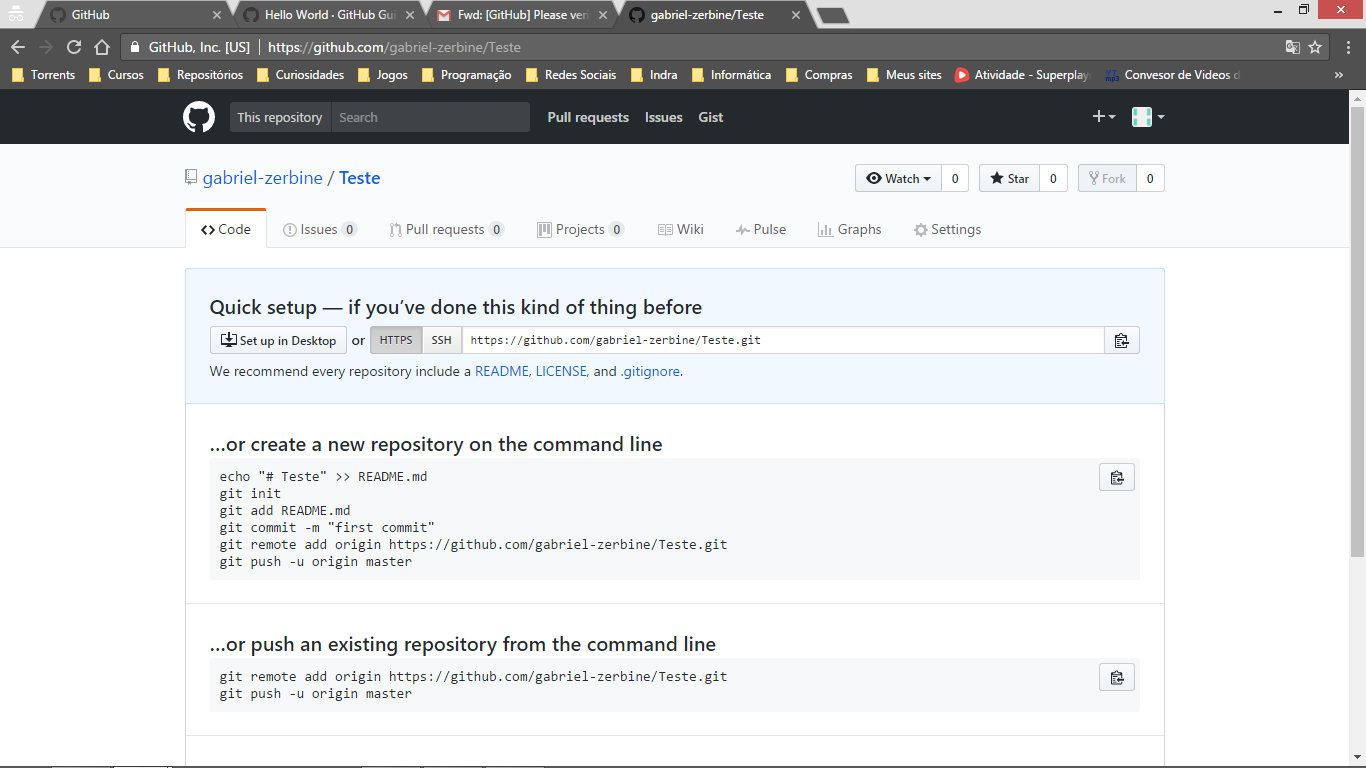
Pronto! Conta criada! Link da conta criada: <https://github.com/gabriel-zerbine> .

O site vai te guiar à um link onde ele vai te perguntar se você quer ver o guia ou já partir para criar um projeto:

Guia

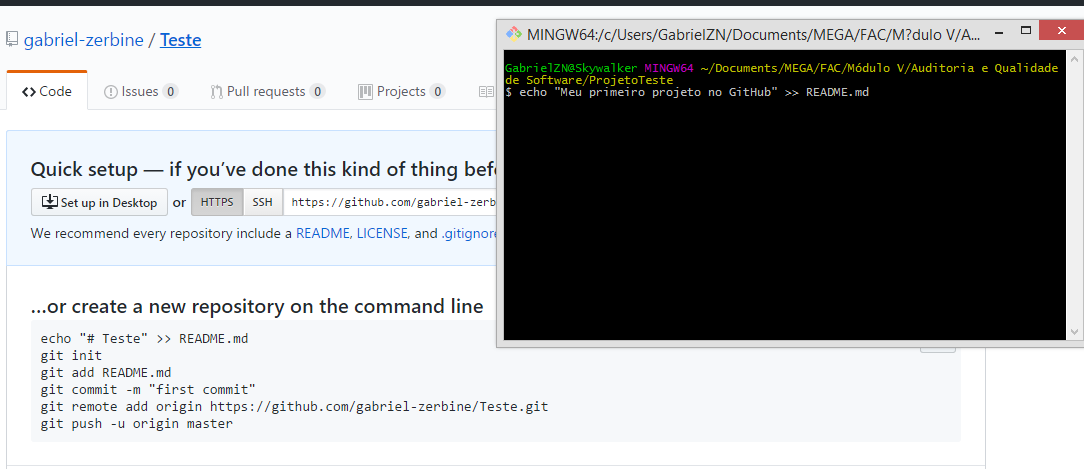
Agora se você quiser partir para um projeto, ele vai te pedir para confirmar o e-mail. Depois de verificar seu e-mail, clique no “Start a Project”. Ele vai apresentar uma página onde você tem que colocar o nome do repositório/projeto no primeiro campo, depois uma descrição do projeto se quiser e clicar no “Create Repository”.





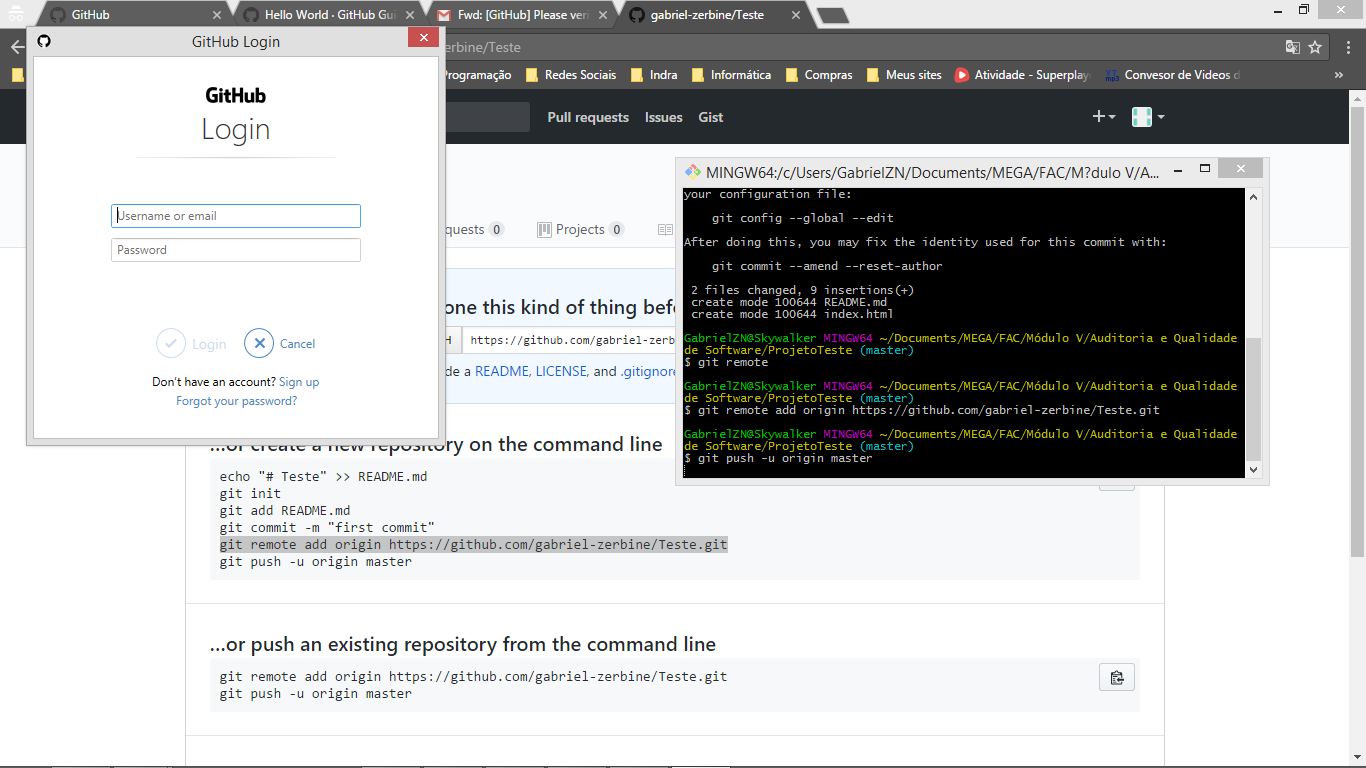
Pronto, agora seu repositório está pronto. Agora vá ao diretório onde se encontra o seu projeto (pode ser pelo cmd do Windows). No diretório do projeto, clique com o botão direito em qualquer lugar para abrir o Git Bash Here, que é uma linha de comando do Git.

Agora é só seguir os comandos que o GitHub te instrui a utilizar.

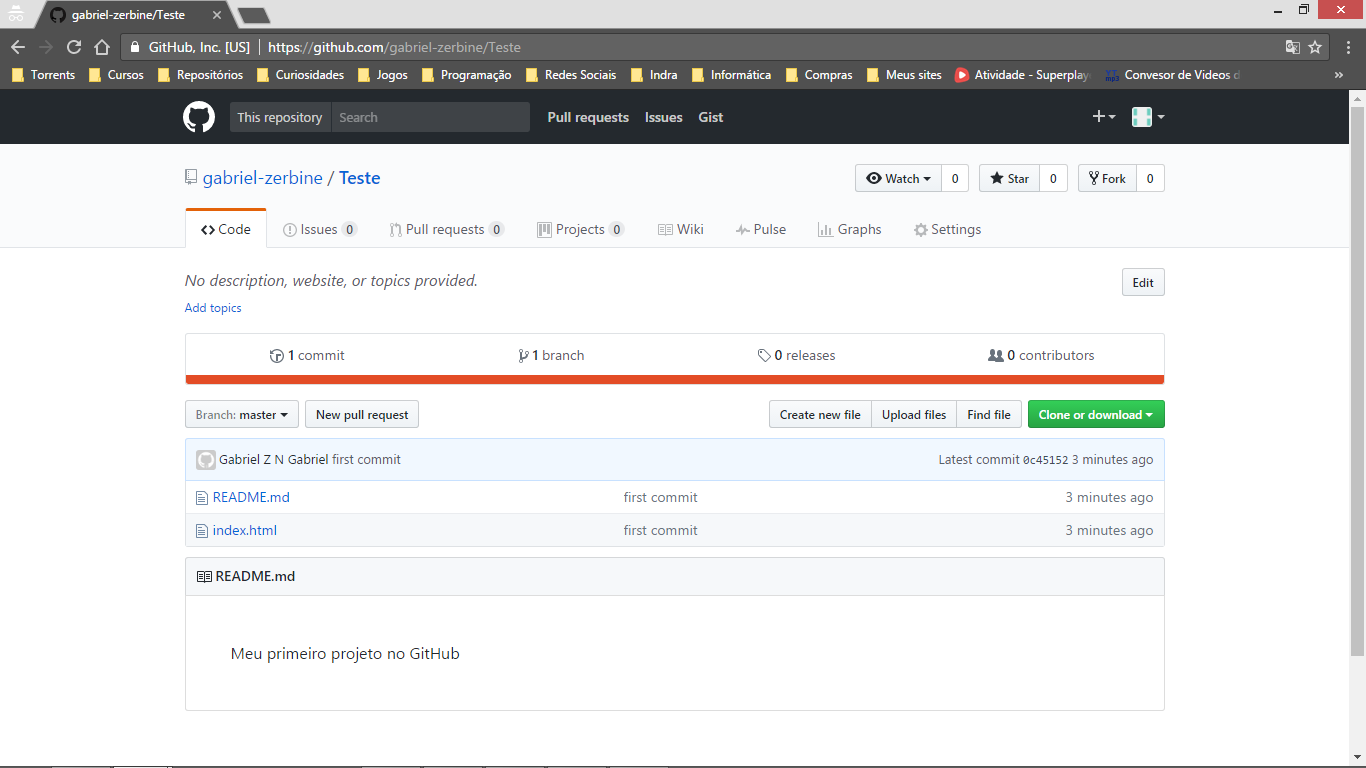


Lembrando que quando você for usar o comando ‘git add ..........’ utilizar desse jeito ‘git add \*’ porque ele adiciona no repositório todos os arquivos que estão na pasta onde você está.

Quando você der o último comando, depois de um tempo vai aparecer um pop-up para você logar.



Agora se você der um refresh na tela do browser, você verá que a tela mudou. Ele te mostra os arquivos que subiram.



## Comandos Básicos

git init 🡪 em um diretório que ainda não tem Git, esse comando inicial o Git no diretório.

git add 🡪 adiciona arquivos na fila para subir no GitHub.

git commit –m “...” 🡪 comentários das alterações realizadas.

git remote add origin <servidor> 🡪 conecta seu repositório local ao repositório no GitHub.

git push 🡪 envia as alterações para o repositório remoto.

git pull 🡪 atualiza seu projeto local.

git merge 🡪 o Git tenta fazer o merge automático entre duas branchs diferentes. Os conflitos é responsabilidade do usuário de fazer.

## Endereço do Repositório criado

<https://github.com/gabriel-zerbine>