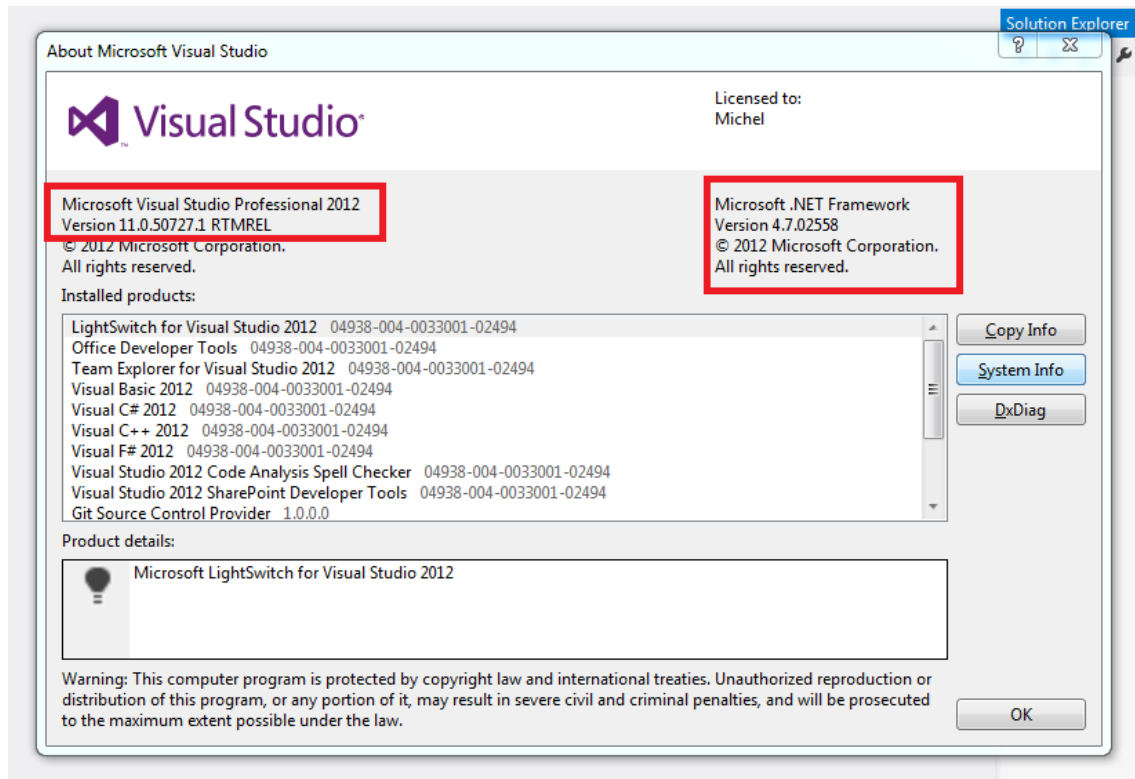
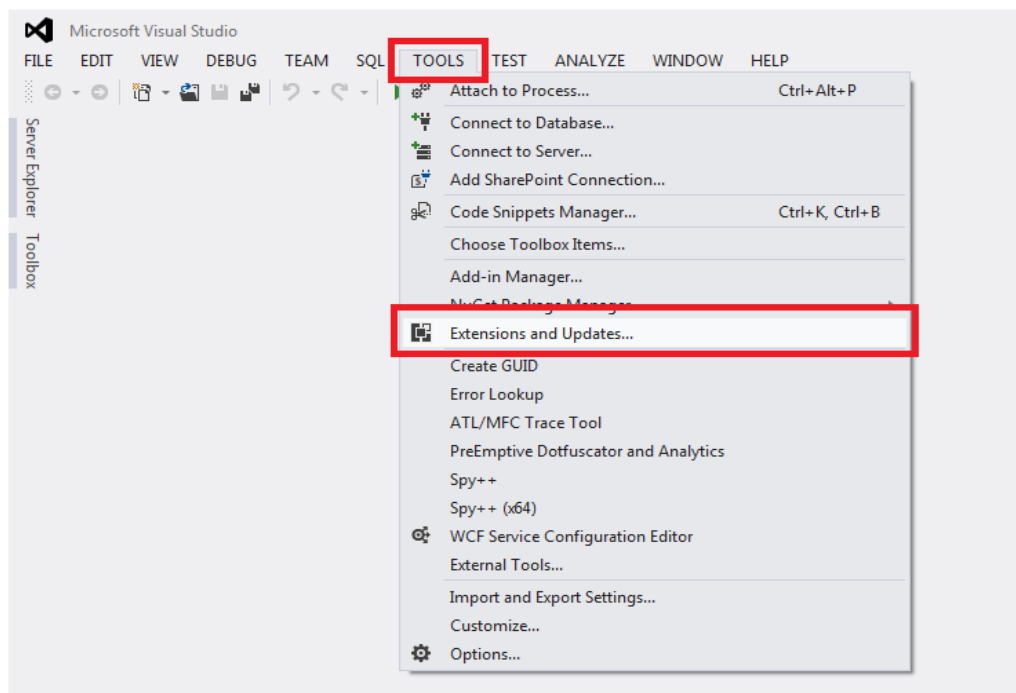


Esse tutorial descreve os passos necessários para integrar o Visual Studio 2012 com o Git e como fazer o upload de um projeto em um repositório do GitHub. Tive a ideia de criar um passo a passo diante da dificuldade que tive de encontrar um material “mastigado” para essa versão do Visual Studio, pois é muito mais fácil encontrar material para as versões atuais.

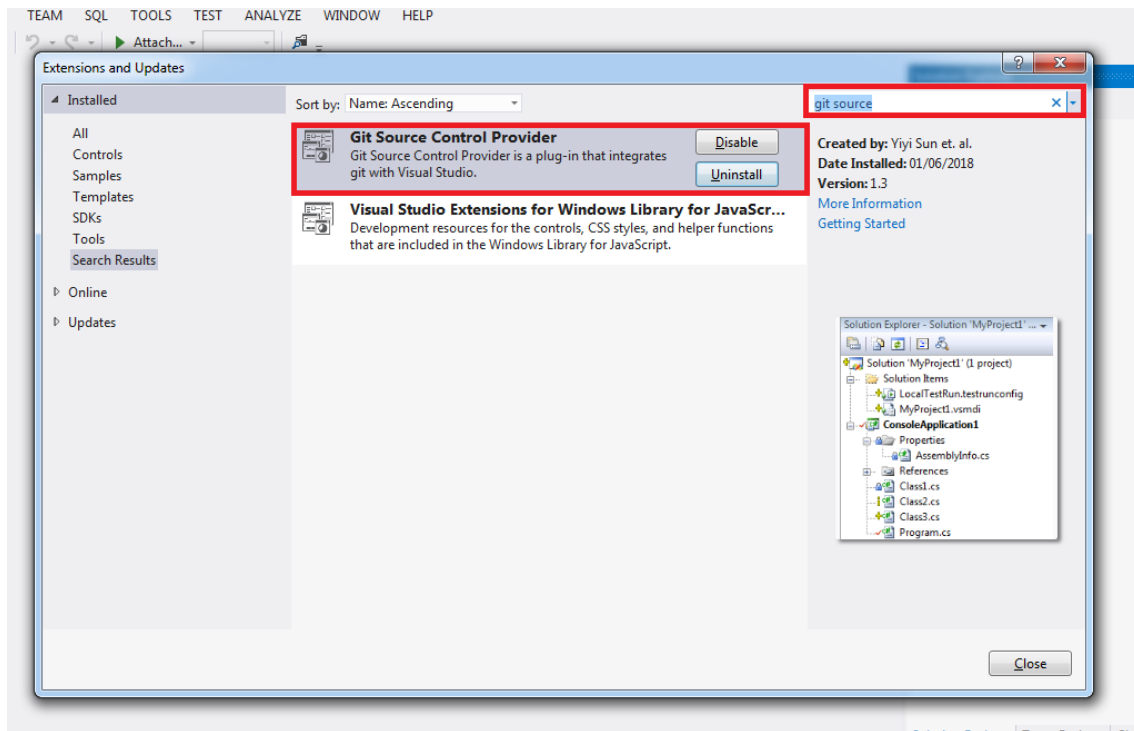
Se você chegou até aqui, tenho certeza que o Visual Studio 2012 já está instalado em sua máquina. Abaixo segue a versão que utilizei nesse tutorial.



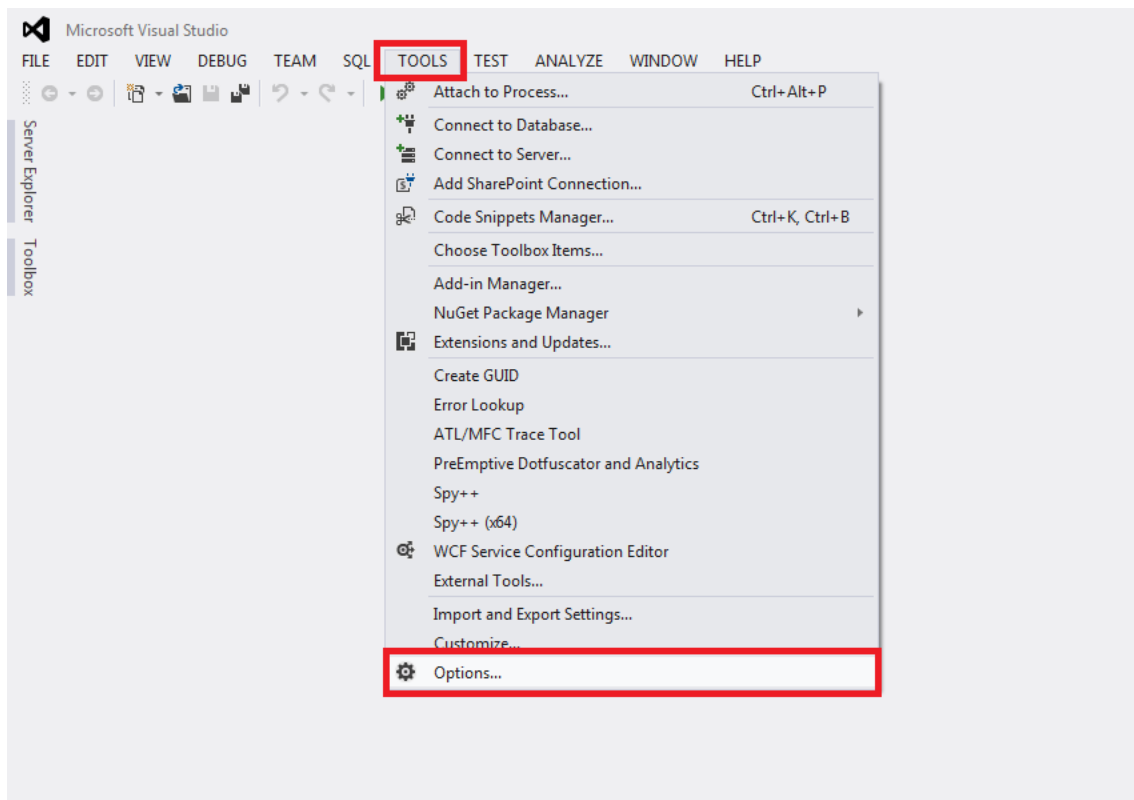
O próximo passo é instalar o plug-in que integra o Git com o Visual Studio.



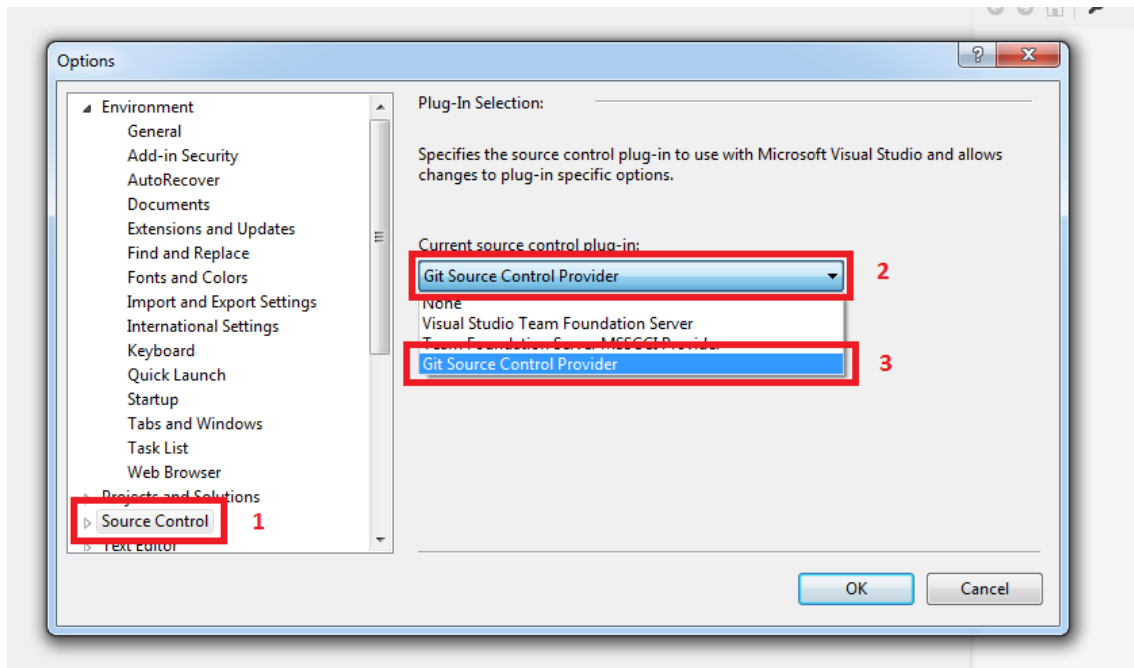
Na janela seguinte existe um campo de busca onde facilita a localização do plug-in que será utilizado, basta digitar **git source** e selecionar a opção **Git Source Control Provider**, conforme consta abaixo. Como no meu computador já está instalado, aparece apenas as opções de desabilitar e desinstalar.



Após concluir a instalação, clique em **tools**, depois em **options...**

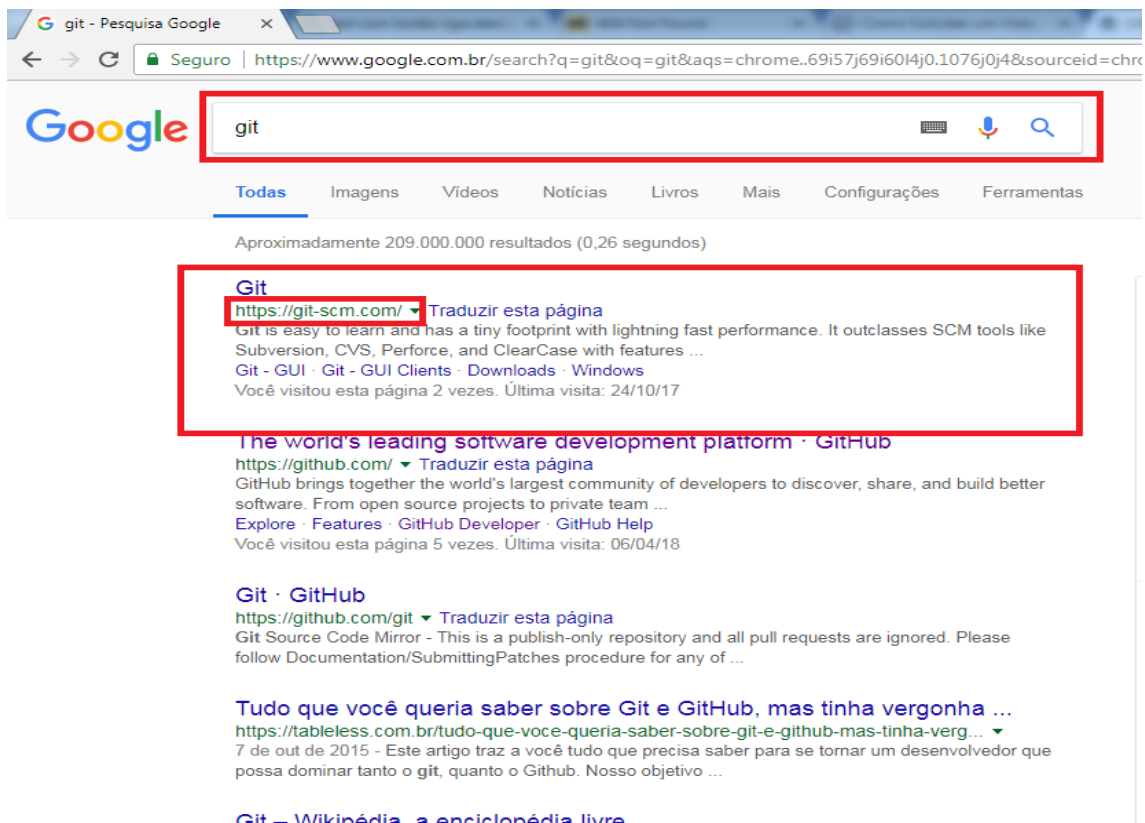


Na janela seguinte clique em **source control** (opção 1), depois clique na seta da combo box (opção 2), selecione a opção **Git Source Control Provider** e por fim clique em OK.

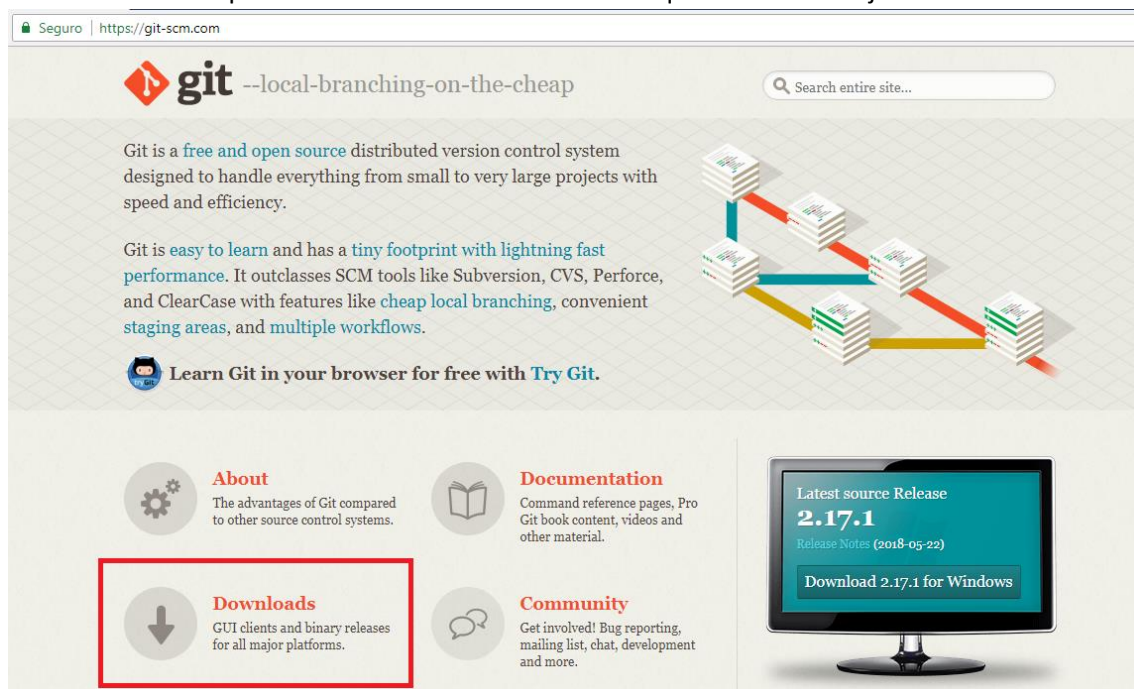


Após seguir os passos acima, feche o Visual Studio e vamos instalar o Git na máquina. Eu sei que algumas pessoas não gostam de fazer coisas através de linhas de comando, mas foi o modo mais prático que encontrei para subir projetos.

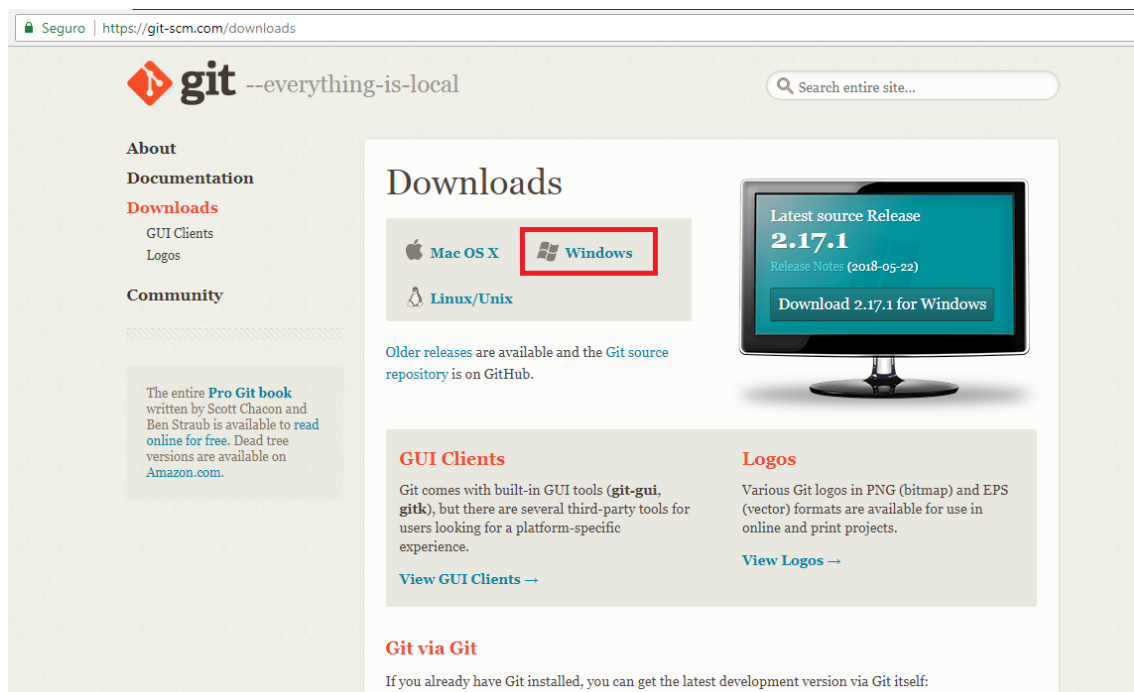
No site do Google digite **git** e algumas opções vão aparecer na pesquisa. O site que precisa ser acessado está destacado abaixo. <https://git-scm.com/>



Dentro do site do Git procure pelo campo de downloads, digo “procure” pois esses sites mudam muito e o print desse tutorial ser diferente de quando você esteja lendo.



Como o tutorial segue a instalação para Windows, eu selecionei o sistema operacional correspondente ao meu.

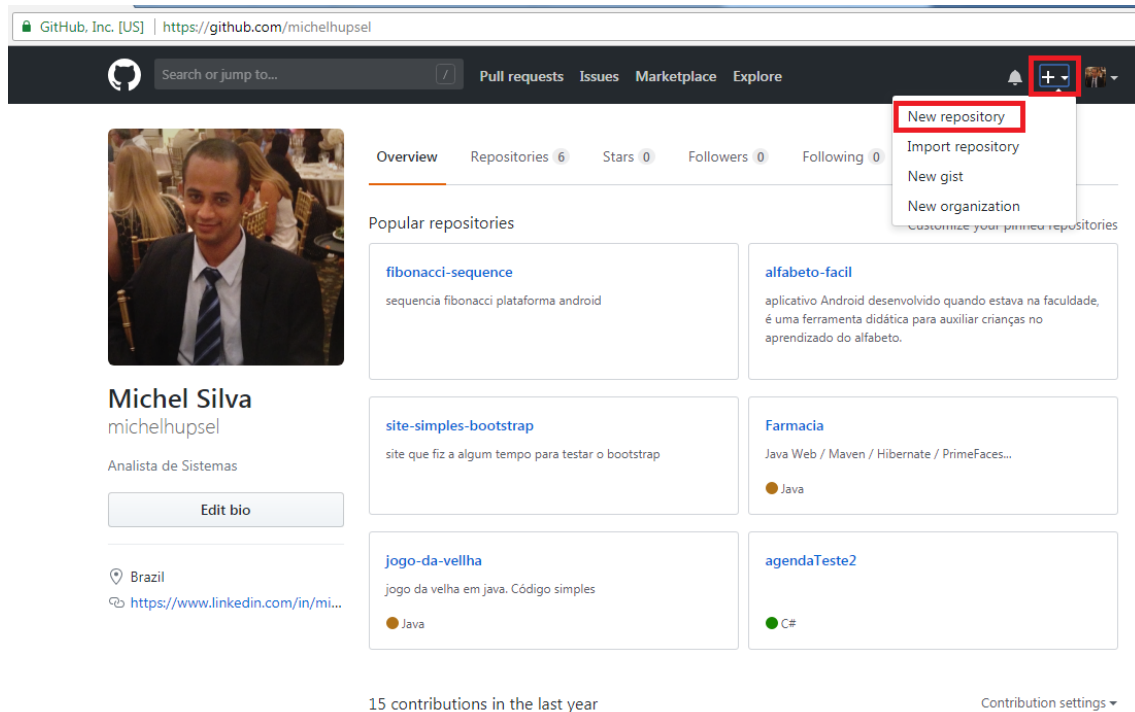


Como o processo de instalação do git é bem simples, não é relevante colocar os prints aqui. Posso dizer a você que é o tradicional *next-next-finish*. Deixa as opções default selecionadas que não terá problemas.

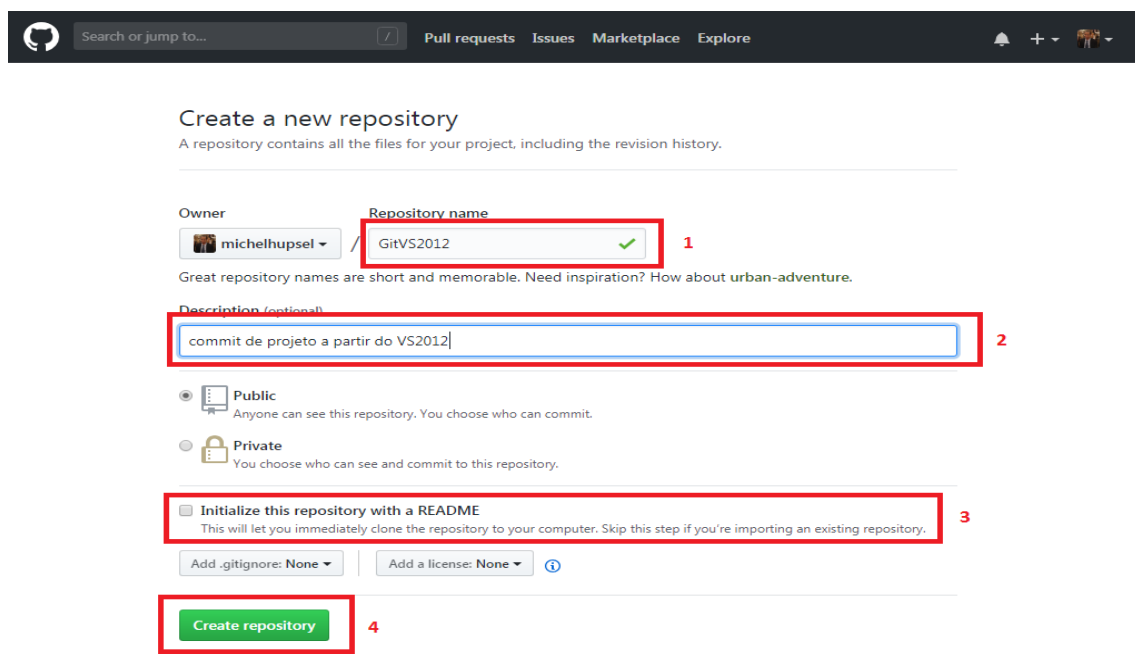
Agora que já instalamos o plug-in necessário e o git no computador, vamos a parte que interessa. Como fazer o upload de um projeto no GitHub utilizando o Visual Studio 2012. É necessário que

you have an account on GitHub and that you create a repository for the project you want to upload, according to the prints below.

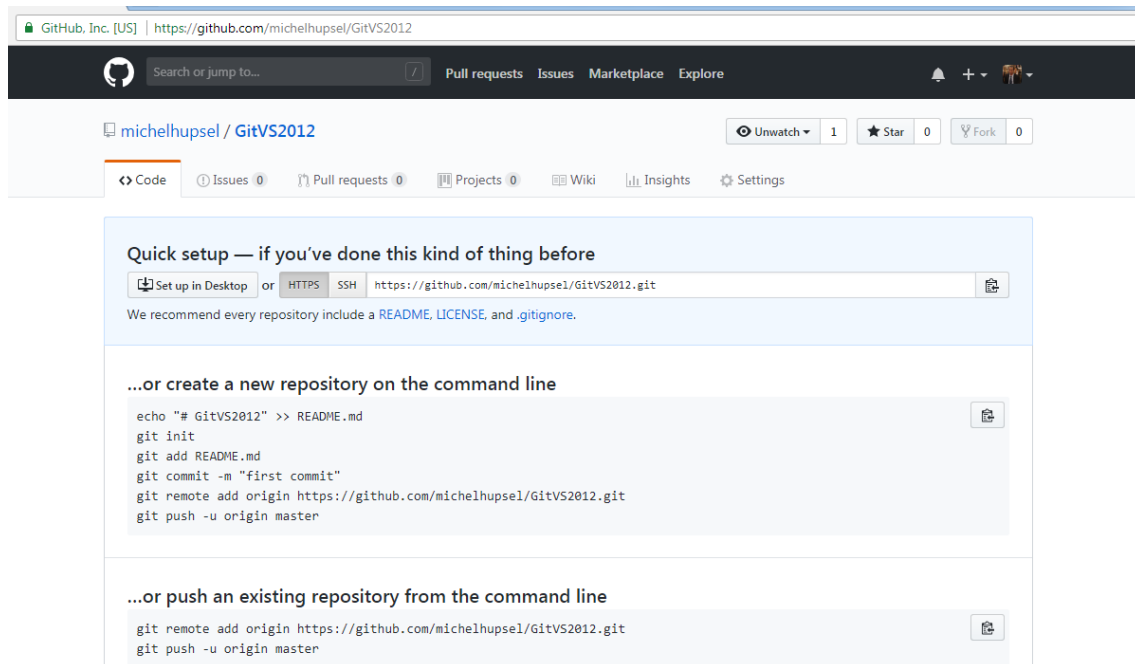
When logging in to your GitHub account, you must click on the **+** in the top right field and select the option **New repository**.



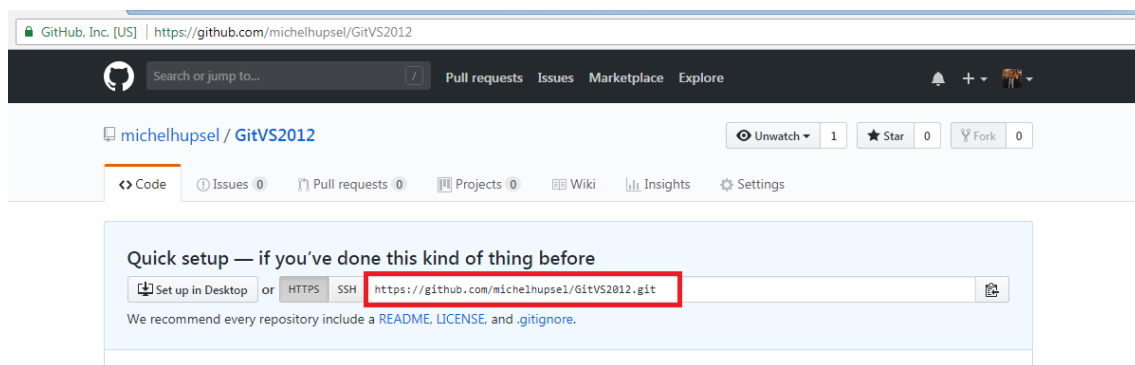
In the next window the steps are very simple. In the first option you must name the repository, in the second you must describe the project briefly, the third option is a point that complicates the project commit while testing and for this reason I suggest that you not mark this option and finally in the fourth option click on **create repository**.



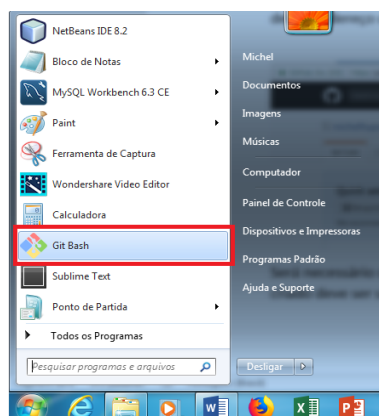
Se você seguiu os passos exatamente como citados acima, a sua tela deve ser parecida com essa. Nomeie o seu projeto como quiser, não precisa ser exatamente o mesmo nome que coloquei acima.



O que você vai precisar para fazer upload do seu projeto é apenas essa parte. Você vai precisar desse endereço um pouco mais afrente.

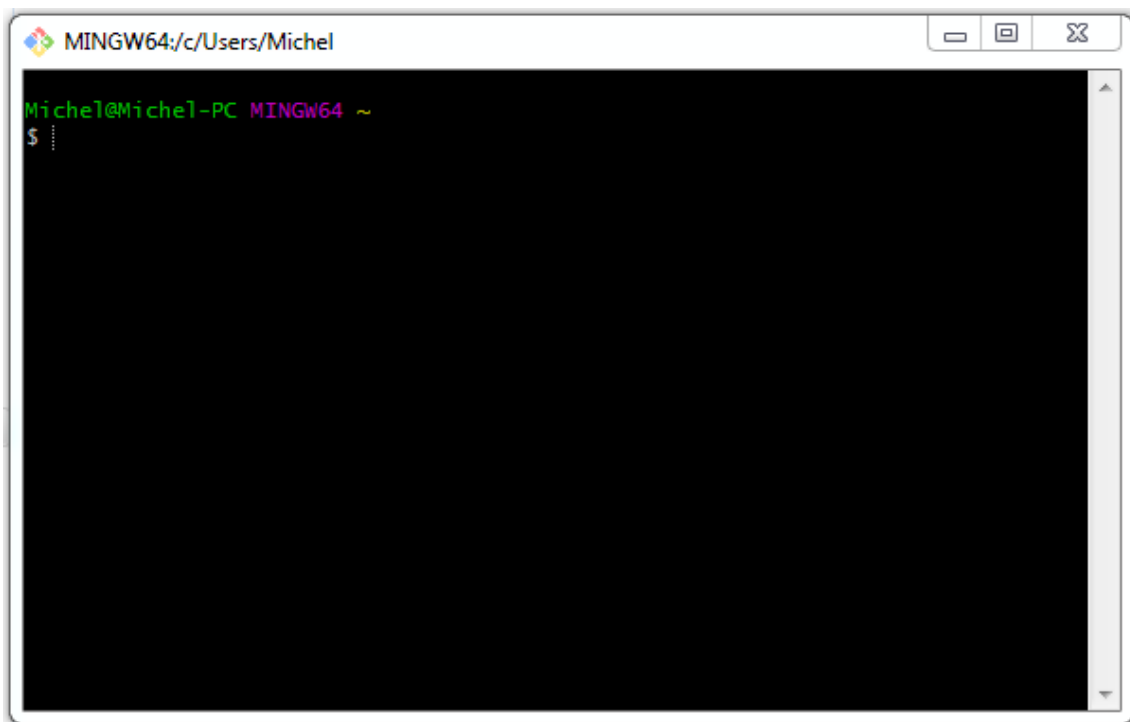


Será necessário criar uma pasta para que o projeto seja clonado e será a mesma que o projeto criado deve ser salvo. Os prints vão seguir em ordem para que seja fácil compreender.



Se você instalou o Git conforme mencionei acima, vá até o menu iniciar do seu computador e procure pelo **Git Bash**. Vá abrir um terminal tipo o DOS, mas não se assuste pois são comandos bem simples. Se der alguma mensagem de erro, verifique se digitou corretamente ou então vai para o Google e tenta resolver.

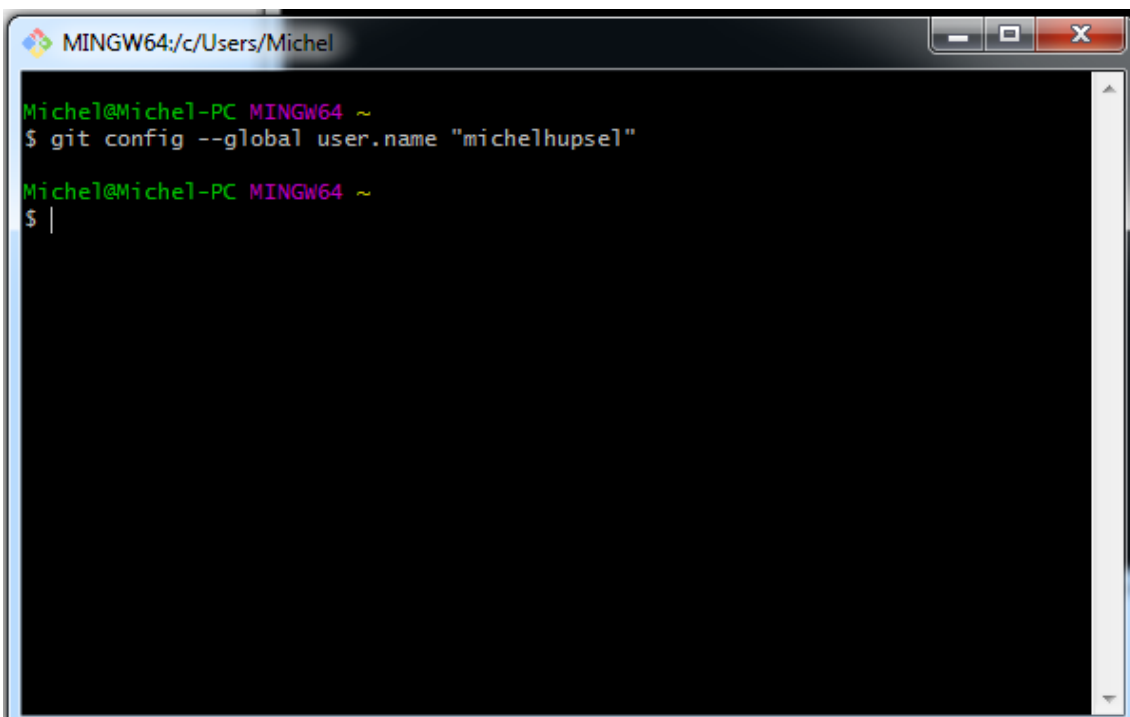
Essa é a tela que deve aparecer para você.



```
MINGW64:/c/Users/Michel

Michel@Michel-PC MINGW64 ~
$
```

O primeiro passo é informar o seu nome de usuário através do comando **git config --global user.name "usuarioGit"**. Entre as aspas você insere o seu nome de usuário no GitHub, feito isso pressione enter e a tela deve aparecer como a de baixo.

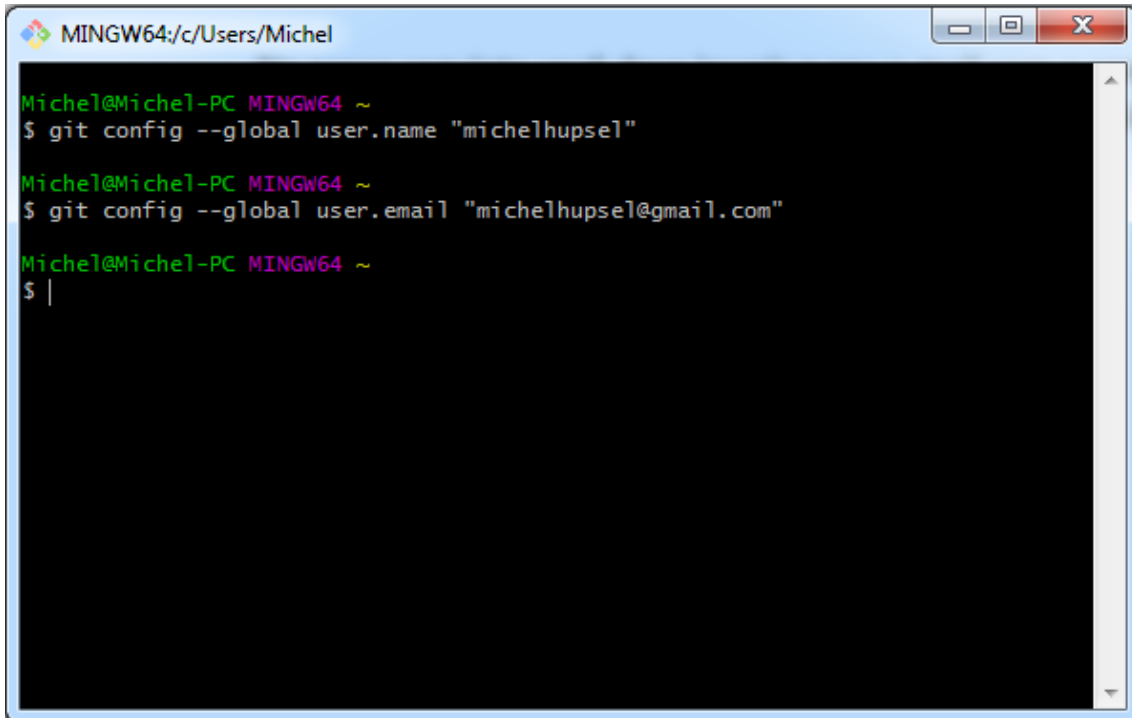


```
MINGW64:/c/Users/Michel

Michel@Michel-PC MINGW64 ~
$ git config --global user.name "michelhupsel"

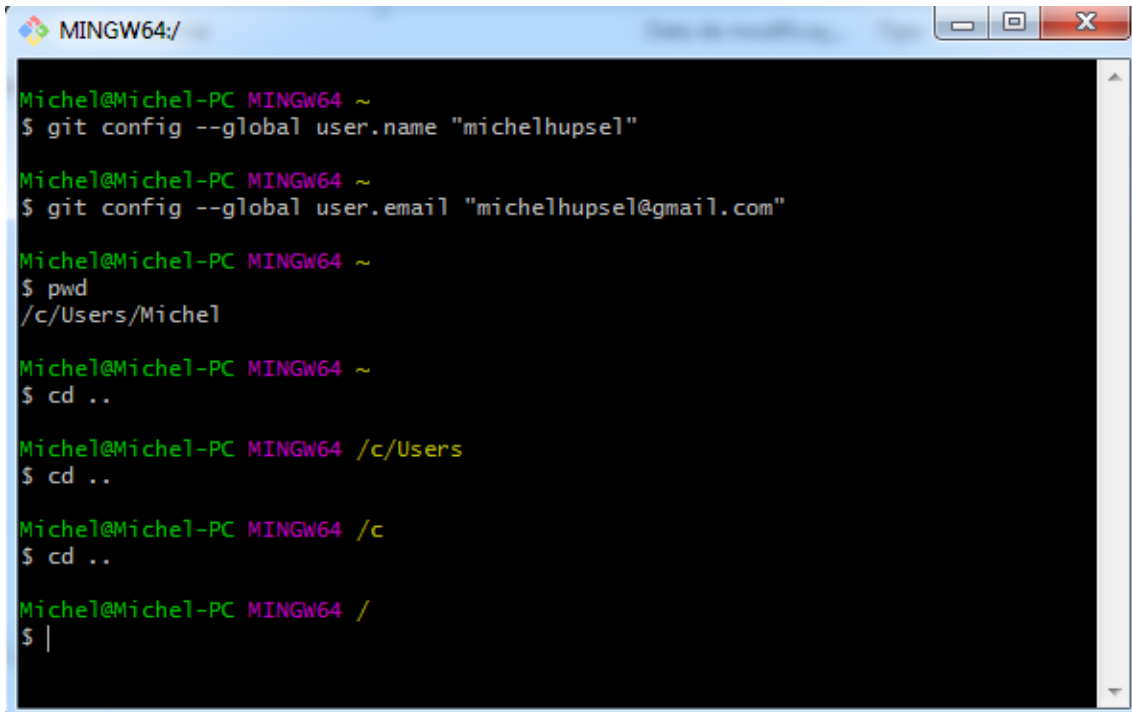
Michel@Michel-PC MINGW64 ~
$ |
```

No passo seguinte você deve inserir o seu e-mail com o comando **git config --global user.email "seuEmail@aqui"**. Entre as aspas digite o seu e-mail do GitHub, feito isso pressione enter e a tela deve aparecer como a de baixo.

A screenshot of a Windows terminal window titled "MINGW64:/c/Users/Michel". The prompt is "Michel@Michel-PC MINGW64 ~". The user enters two commands: "git config --global user.name \"michelhupsel\"" and "git config --global user.email \"michelhupsel@gmail.com\"". The prompt returns to "\$ |".

```
MINGW64:/c/Users/Michel
Michel@Michel-PC MINGW64 ~
$ git config --global user.name "michelhupsel"
Michel@Michel-PC MINGW64 ~
$ git config --global user.email "michelhupsel@gmail.com"
Michel@Michel-PC MINGW64 ~
$ |
```

No passo seguinte é necessário indicar a pasta onde o repositório será clonado. Digite **pwd** para saber em qual pasta você está e digite **cd ..** até voltar na pasta raiz, conforme abaixo.

A screenshot of a Windows terminal window titled "MINGW64:/". The prompt is "Michel@Michel-PC MINGW64 ~". The user enters "pwd", which returns "/c/Users/Michel". Then the user enters "cd .." three times, moving up the directory hierarchy from "/c/Users/Michel" to "/c" and finally to the root "/" directory.

```
MINGW64:/
Michel@Michel-PC MINGW64 ~
$ git config --global user.name "michelhupsel"
Michel@Michel-PC MINGW64 ~
$ git config --global user.email "michelhupsel@gmail.com"
Michel@Michel-PC MINGW64 ~
$ pwd
/c/Users/Michel
Michel@Michel-PC MINGW64 ~
$ cd ..
Michel@Michel-PC MINGW64 /c/Users
$ cd ..
Michel@Michel-PC MINGW64 /c
$ cd ..
Michel@Michel-PC MINGW64 /
$ |
```

Após voltar para a pasta raiz, você deve inserir o endereço onde criou a pasta. No meu exemplo a pasta foi criada em C:\GitTesteVS2012 (disco local C e o nome da pasta é GitTesteVS2012), mas para setar o endereço o comando vai ser **cd /c/GitTesteVS2012**. Observe que existe uma diferença na barra e que o diretório não tem os dois pontos. Após digitar o comando e apertar

enter, o endereço que foi setado deve aparecer em amarelo de acordo com o que foi feito no print.

A terminal window titled 'MINGW64:/c/GitTesteVS2012' with a black background and green text. The user 'Michel' is at 'Michel-PC' using 'MINGW64'. The commands and output are: 'git config --global user.email "michelhupsel@gmail.com"', 'pwd' returns '/c/Users/Michel', 'cd ..' moves to '/c/Users', another 'cd ..' moves to '/c', a third 'cd ..' moves to '/', 'pwd' returns '/', 'cd /c/GitTesteVS2012' is entered and highlighted with a red box, and the prompt changes to '/c/GitTesteVS2012', which is also highlighted with a red box.

```
MINGW64:/c/GitTesteVS2012
$ git config --global user.email "michelhupsel@gmail.com"

Michel@Michel-PC MINGW64 ~
$ pwd
/c/Users/Michel

Michel@Michel-PC MINGW64 ~
$ cd ..

Michel@Michel-PC MINGW64 /c/Users
$ cd ..

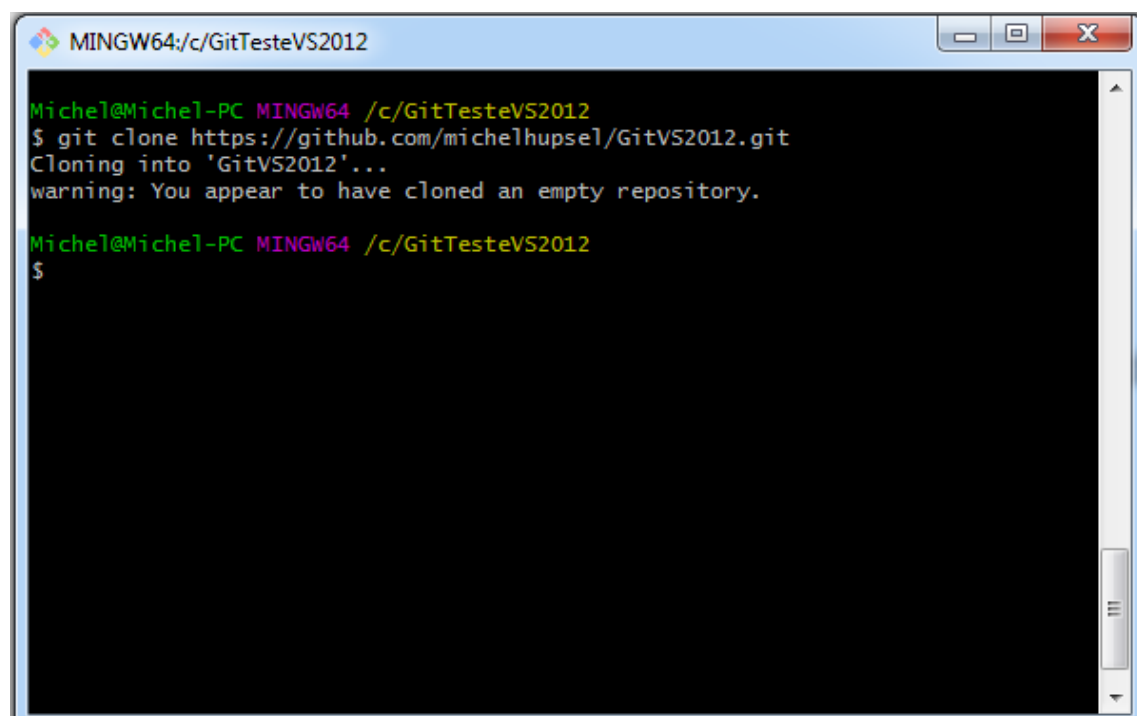
Michel@Michel-PC MINGW64 /c
$ cd ..

Michel@Michel-PC MINGW64 /
$ pwd
/

Michel@Michel-PC MINGW64 /
$ cd /c/GitTesteVS2012

Michel@Michel-PC MINGW64 /c/GitTesteVS2012
$
```

Feito isso, vamos precisar daquele endereço que foi gerado quando criamos o repositório lá no GitHub. Vamos usar o comando `git clone https://github.com/michelhupsel/GitVS2012.git`. Após digitar o comando pressione enter. Observe que no meu exemplo está sendo exibida uma mensagem dizendo que aparentemente clonei um repositório vazio, isso significa que funcionou.

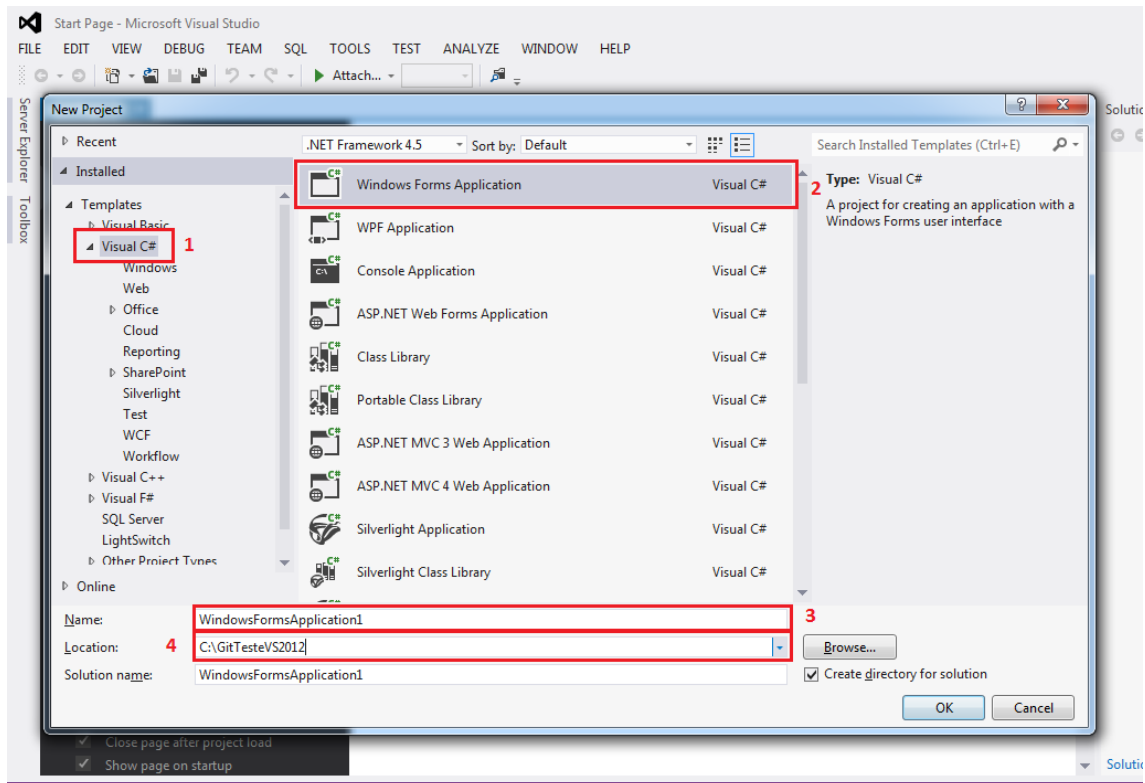
A terminal window titled 'MINGW64:/c/GitTesteVS2012' with a black background and green text. The user 'Michel' is at 'Michel-PC' using 'MINGW64'. The commands and output are: 'git clone https://github.com/michelhupsel/GitVS2012.git', which outputs 'Cloning into 'GitVS2012'...' and 'warning: You appear to have cloned an empty repository.', and the prompt returns to '\$'.

```
MINGW64:/c/GitTesteVS2012
Michel@Michel-PC MINGW64 /c/GitTesteVS2012
$ git clone https://github.com/michelhupsel/GitVS2012.git
Cloning into 'GitVS2012'...
warning: You appear to have cloned an empty repository.

Michel@Michel-PC MINGW64 /c/GitTesteVS2012
$
```

Vamos manter a tela do git aberta e abrir o Visual Studio para criar um novo projeto e colocar ele no GitHub, dentro do repositório que foi clonado.

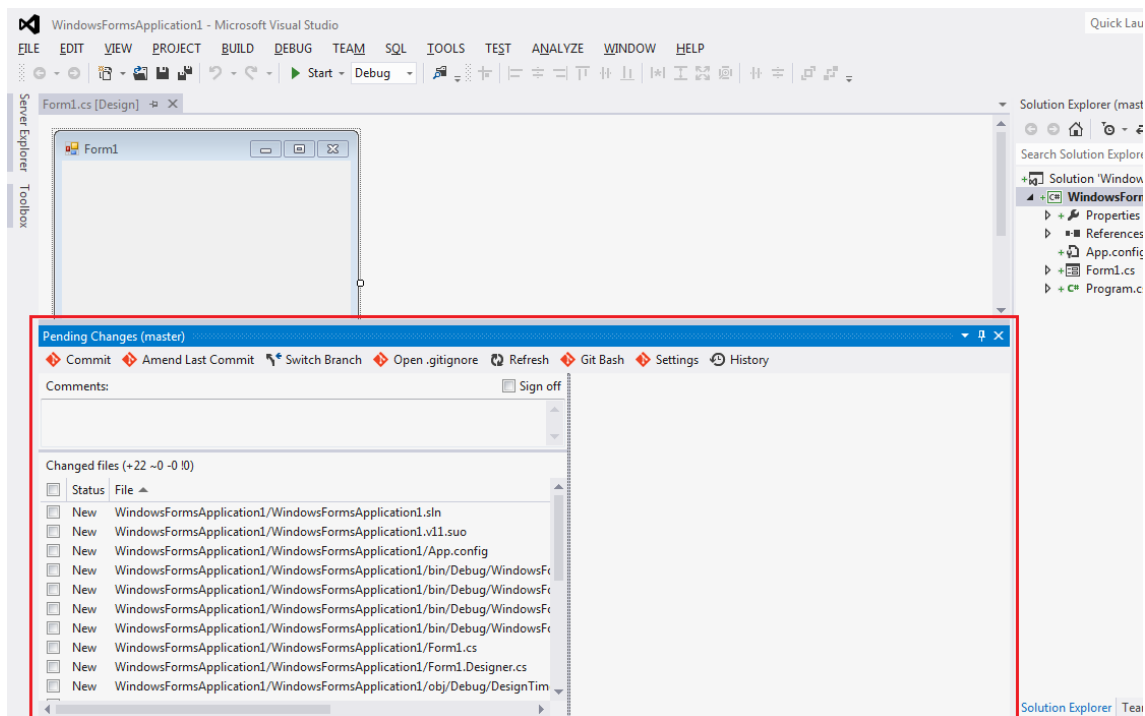
Com o seu Visual Studio aberto clique em **File >> New Project** e na janela que abriu você pode selecionar o que eu utilizei como exemplo, a parte mais importante do processo é setar exatamente a mesma pasta onde o seu Git foi clonado, conforme fiz no número 4 do exemplo abaixo.



Alguns passos atrás, antes de fechar o Visual Studio, a opção **Git Source Control Provider** foi selecionada e por conta disso a sua tela deve aparecer como a minha no projeto. Mas antes você deve digitar o comando que inicia o git, caso contrário não será possível selecionar os arquivos e fazer o commit. O comando digitado no Git Bash é **git init**.

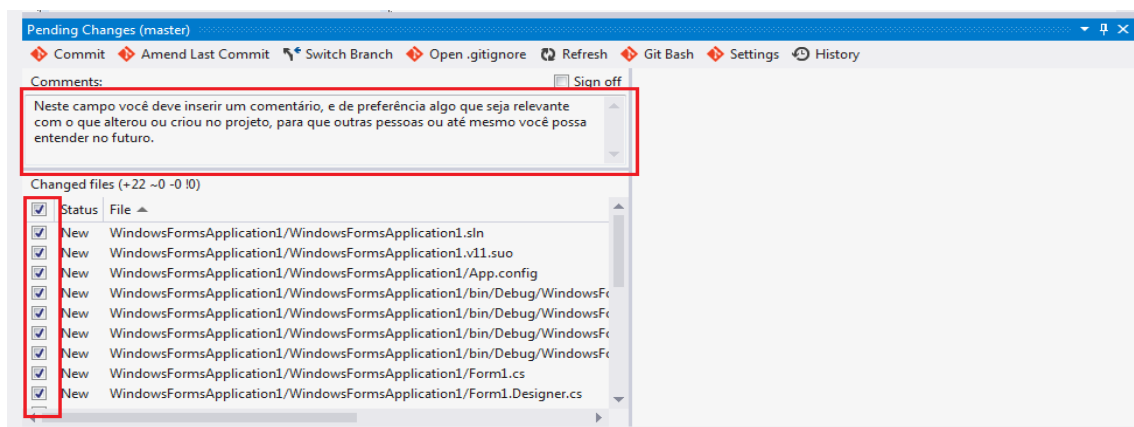
```
MINGW64:/c/GitTesteVS2012
Michel@Michel-PC MINGW64 /c/GitTesteVS2012
$ git clone https://github.com/michelhupsel/GitVS2012.git
Cloning into 'GitVS2012'...
warning: You appear to have cloned an empty repository.
Michel@Michel-PC MINGW64 /c/GitTesteVS2012
$ git init
Initialized empty Git repository in C:/GitTesteVS2012/.git/
Michel@Michel-PC MINGW64 /c/GitTesteVS2012 (master)
$
```

Após o comando de inicialização, o seu projeto vai se parecer assim.

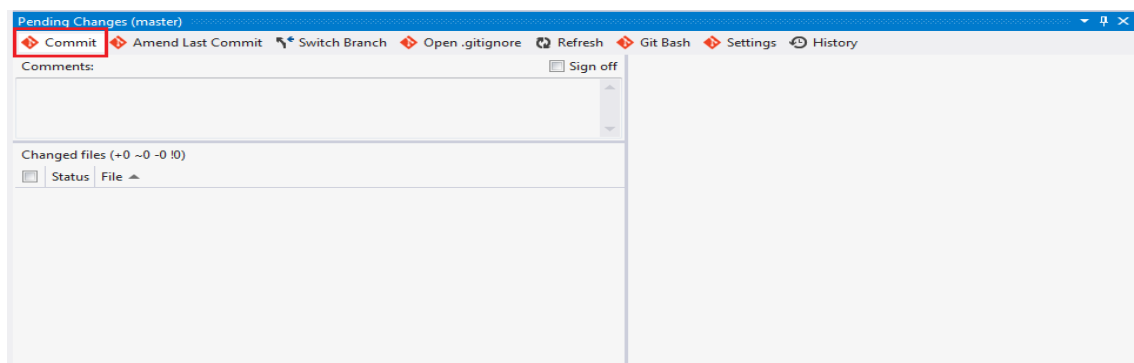


A partir daqui já é possível fazer o commit do seu projeto no GitHub. Vou fazer uma sequência de prints do que deve ser feito para concluir esse tutorial.

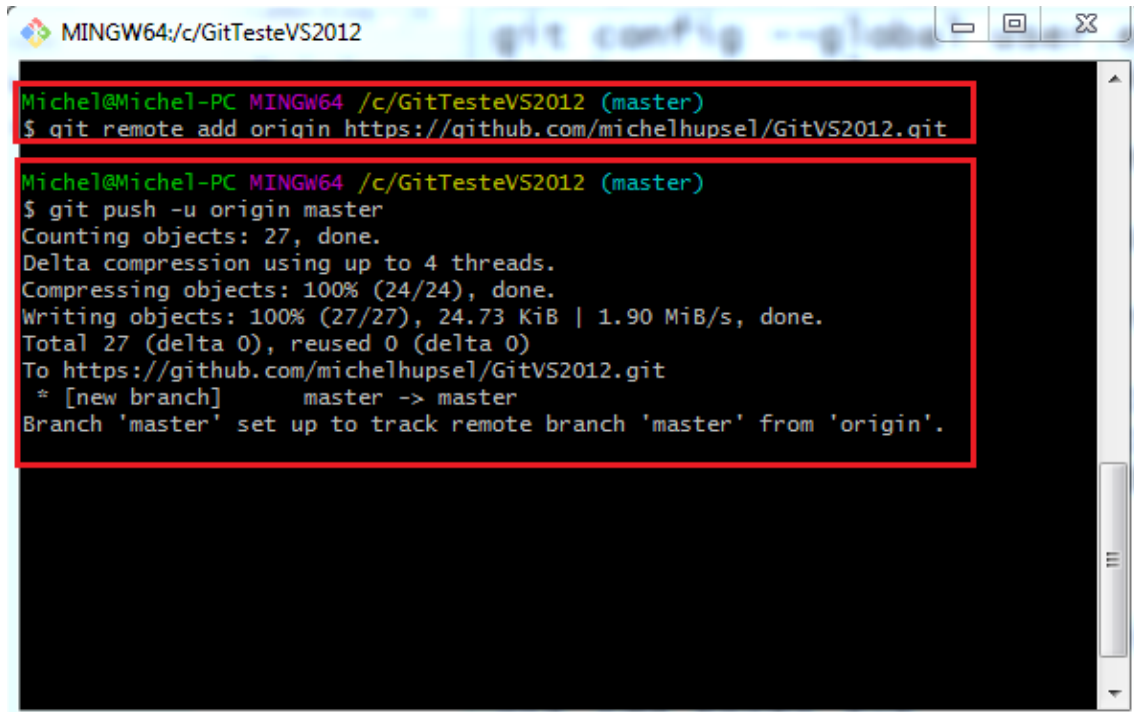
No campo **Comments** você deve preencher com algo que seja pertinente as alterações realizadas. Na coluna onde ficam as **combo box** você pode selecionar o que deseja commitar.



No passo seguinte você vai pressionar em Commit e as informações devem apenas desaparecer, se tudo estiver de acordo é claro.



Agora precisamos finalizar utilizando o Git Bash mais uma vez. O primeiro comando utilizado informa qual o endereço do repositório que deseja copiar os arquivos, no meu caso é o **git remote add origin https://github.com/michelhupsel/GitVS2012.git**. E o próximo comando é responsável por confirmar o commit dos arquivos no repositório remoto **git push -u origin master**.

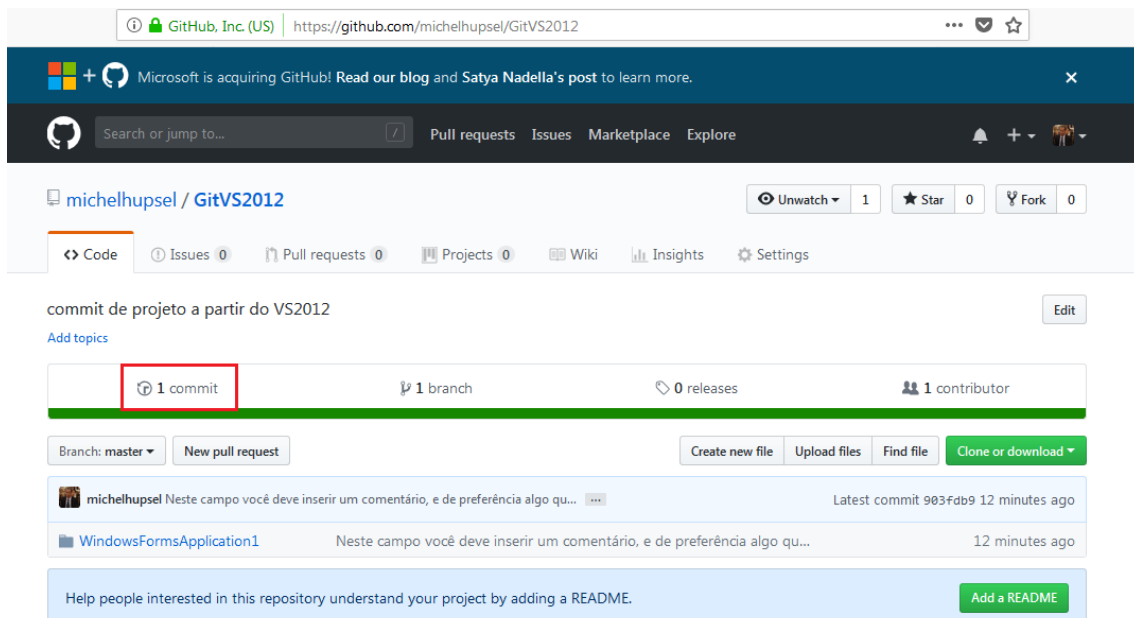


```
MINGW64:/c/GitTesteVS2012

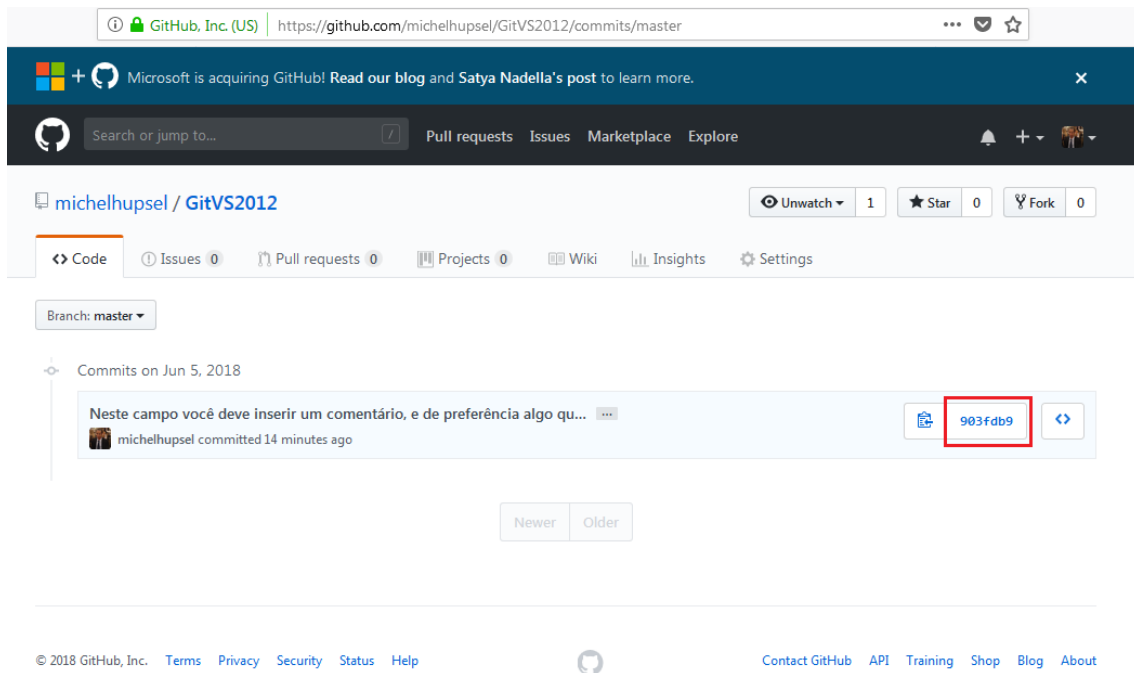
Michel@Michel-PC MINGW64 /c/GitTesteVS2012 (master)
$ git remote add origin https://github.com/michelhupsel/GitVS2012.git

Michel@Michel-PC MINGW64 /c/GitTesteVS2012 (master)
$ git push -u origin master
Counting objects: 27, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (24/24), done.
Writing objects: 100% (27/27), 24.73 KiB | 1.90 MiB/s, done.
Total 27 (delta 0), reused 0 (delta 0)
To https://github.com/michelhupsel/GitVS2012.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

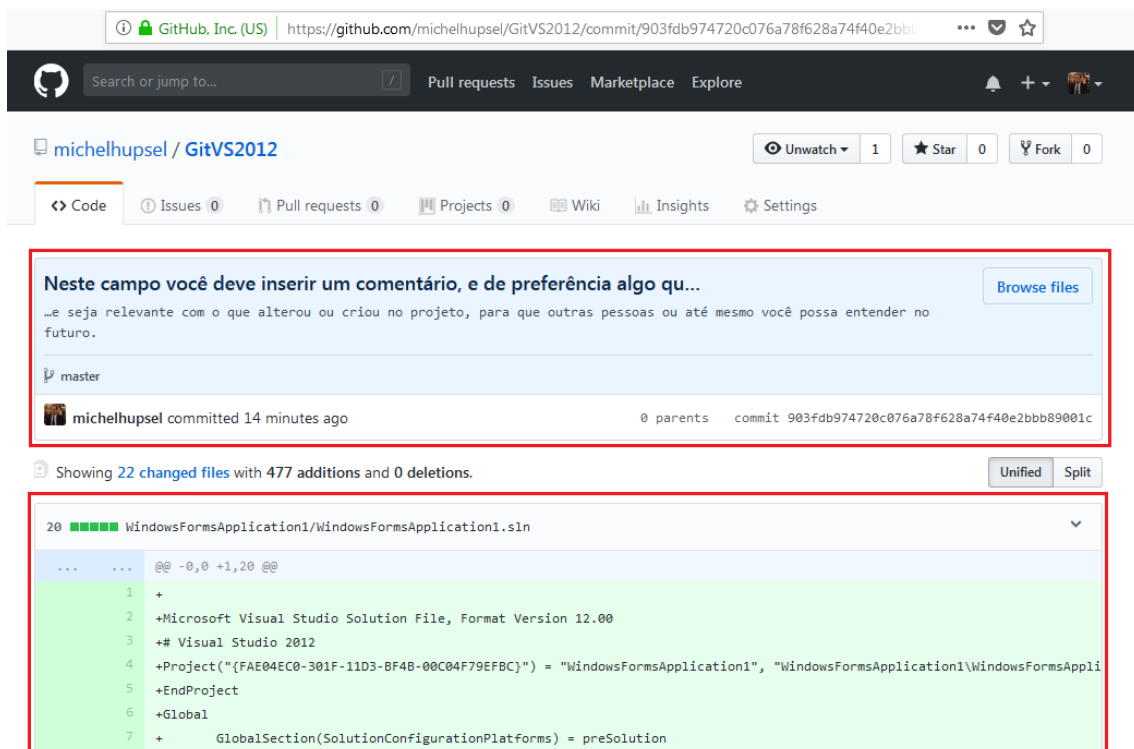
Para confirmar que todo esse processo teve sucesso, vamos voltar ao repositório criado no GitHub para verificar se houve alguma alteração.



O commit foi feito e para mais detalhes clique em commit e selecione a opção destacada em vermelho.



Na tela seguinte você pode observar que no primeiro campo está o comentário feito no commit e mais abaixo as alterações realizadas no código.



Espero que esse tutorial possa ajudar aos que tiveram a mesma dúvida que eu. Caso queira contribuir, vou deixar disponível todos os prints, o texto no formato .docx (word) para edição e no formato PDF. Caso tenha utilizado algum termo técnico inadequado, sinta-se à vontade para corrigir. Obrigado pelo interesse no material.