

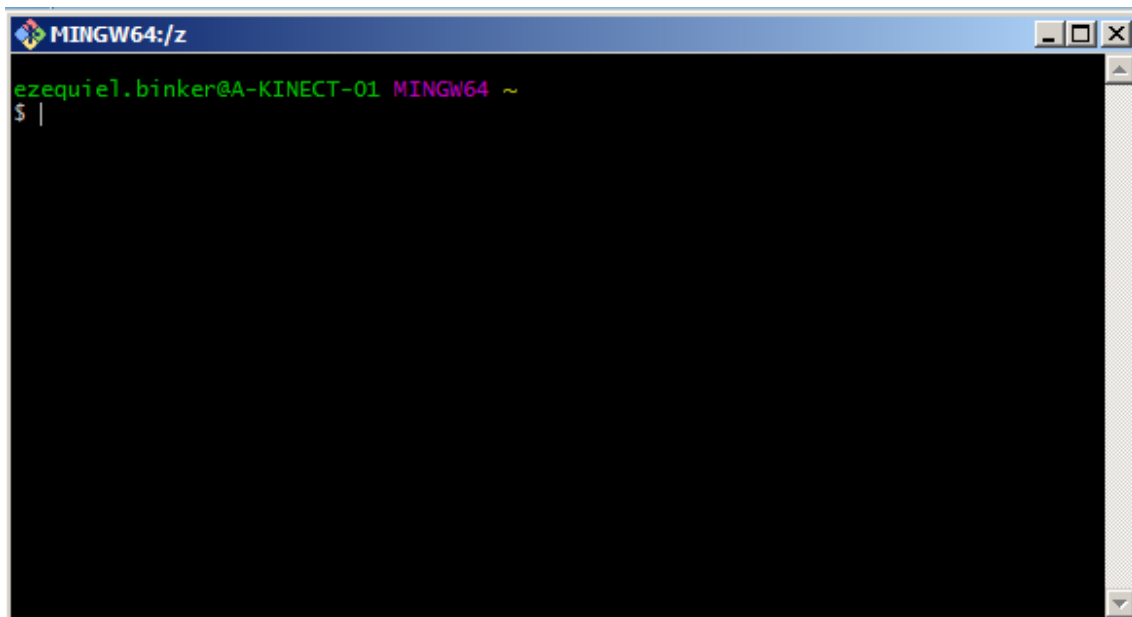
Cómo utilizar Git Bash

Git Bash es una consola para controlar el versionado de nuestros proyectos.

¿En qué nos beneficia?

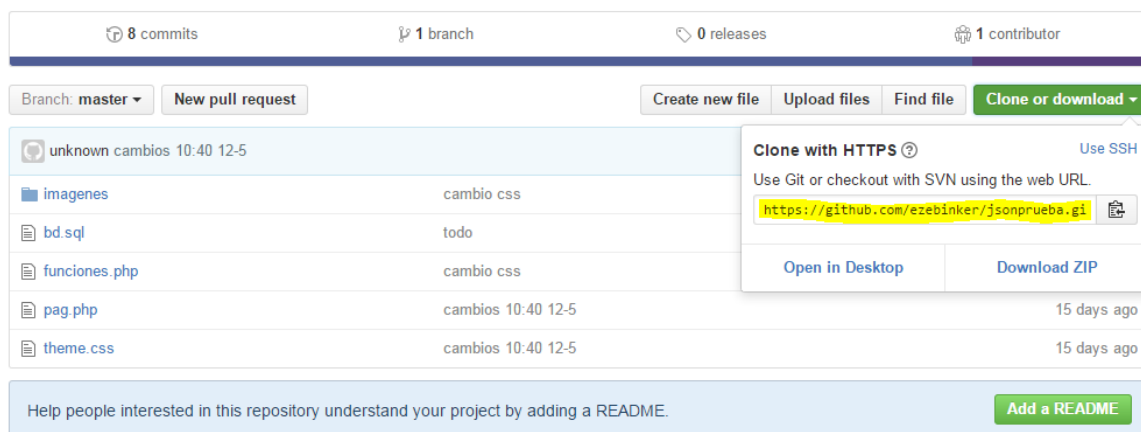
- Es una plataforma muy sencilla para poder actualizar nuestros proyectos en la nube de **Github**
- Es veloz y no tiene interfaz gráfica

De manera similar a esta, aparece Git Bash apenas lo iniciamos:



¿Cómo clonar nuestro proyecto?

Clonar es obtener una copia de un proyecto que se encuentra en la nube. Veremos cuando iniciamos sesión en Github.com, que en el botón Clone or download aparece una URL de nuestro proyecto de Github.



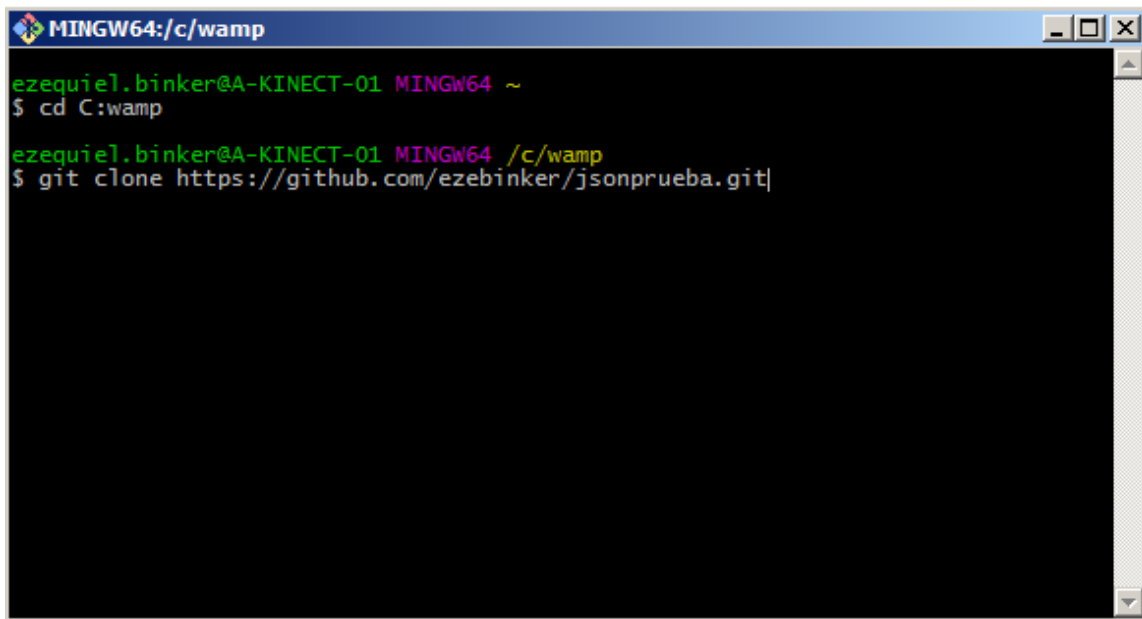
Tutorial Git Bash

Copiamos la dirección que figura en la imagen.

Luego, nos dirigimos a Git Bash, y mediante SHIFT+INSERT, pegamos la dirección copiada en la consola, luego de escribir “git clone”.

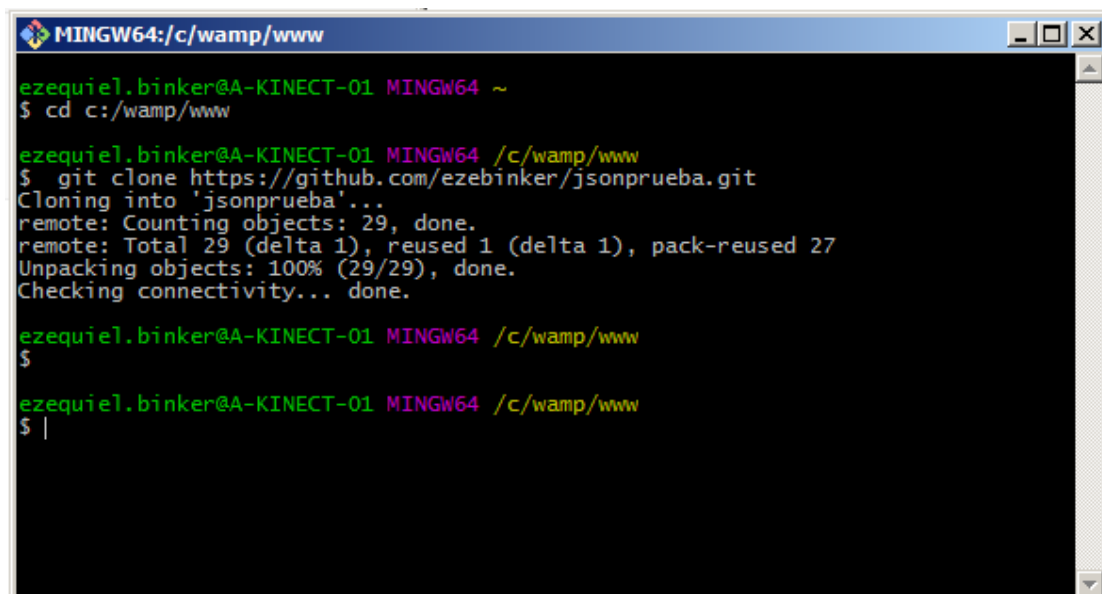
Recuerden que previamente, deben estar posicionados en la carpeta en la cual quieren clonar el proyecto.

HINT: Usen cd.. para ir atrás en un directorio y cd nombredirectorio para avanzar de directorio.



```
MINGW64:/c/wamp
ezequiel.binker@A-KINECT-01 MINGW64 ~
$ cd C:wamp
ezequiel.binker@A-KINECT-01 MINGW64 /c/wamp
$ git clone https://github.com/ezebinker/jsonprueba.git
```

Luego apretamos Enter y procede a clonar el proyecto en la carpeta elegida.



```
MINGW64:/c/wamp/www
ezequiel.binker@A-KINECT-01 MINGW64 ~
$ cd c:/wamp/www
ezequiel.binker@A-KINECT-01 MINGW64 /c/wamp/www
$ git clone https://github.com/ezebinker/jsonprueba.git
Cloning into 'jsonprueba'...
remote: Counting objects: 29, done.
remote: Total 29 (delta 1), reused 1 (delta 1), pack-reused 27
Unpacking objects: 100% (29/29), done.
Checking connectivity... done.
ezequiel.binker@A-KINECT-01 MINGW64 /c/wamp/www
$
ezequiel.binker@A-KINECT-01 MINGW64 /c/wamp/www
$ |
```

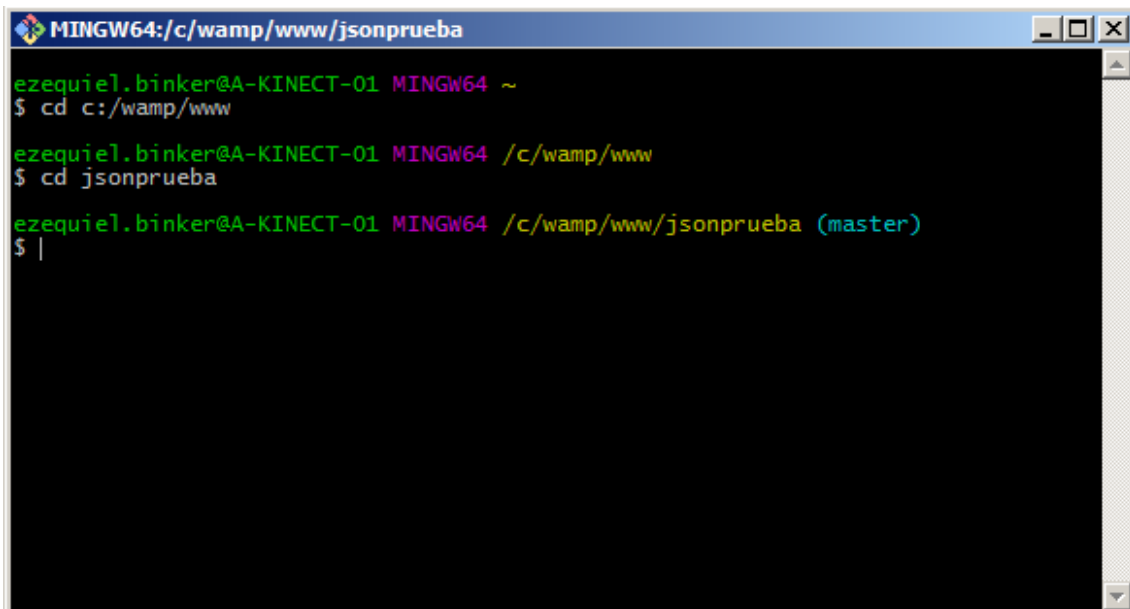
Aclaración: Preferentemente clonar el proyecto en una carpeta del disco C

Tutorial Git Bash

Perfecto! Nuestro proyecto ya está clonado en la carpeta deseada.

¿Cómo obtener la última versión de nuestro proyecto, si ya tengo el branch master en mi PC?

Primero, me posiciono sobre el **master** (la carpeta general clonada)



```
MINGW64:/c:/wamp/www/jsonprueba
ezequiel.binker@A-KINECT-01 MINGW64 ~
$ cd c:/wamp/www
ezequiel.binker@A-KINECT-01 MINGW64 /c:/wamp/www
$ cd jsonprueba
ezequiel.binker@A-KINECT-01 MINGW64 /c:/wamp/www/jsonprueba (master)
$ |
```

Luego, voy a escribir “**git pull**”, para poder actualizar a la última versión.

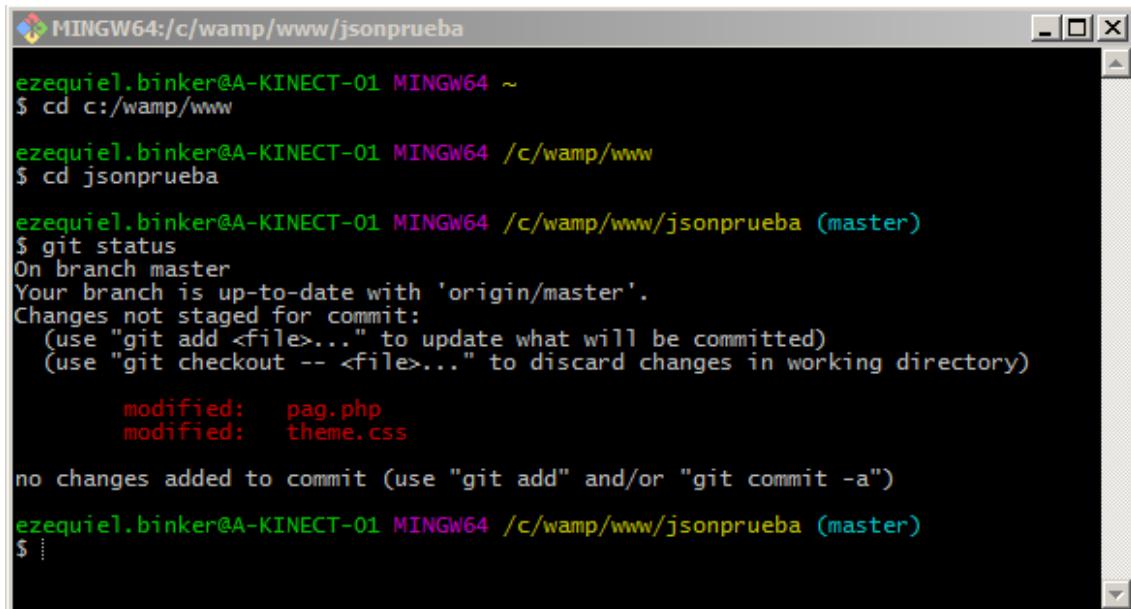


```
MINGW64:/c:/wamp/www/jsonprueba
ezequiel.binker@A-KINECT-01 MINGW64 ~
$ cd c:/wamp/www
ezequiel.binker@A-KINECT-01 MINGW64 /c:/wamp/www
$ cd jsonprueba
ezequiel.binker@A-KINECT-01 MINGW64 /c:/wamp/www/jsonprueba (master)
$ git pull
```

Listo! Ya tengo la última versión de la nube en mi PC.

¿Cómo ver si mi proyecto tiene alguna modificación en los archivos?

Utilizaremos “**git status**” para ver si hay alguna diferencia entre el código que se encuentra en nuestra PC con respecto al código que existe en la nube.



```
MINGW64:/c/wamp/www/jsonprueba
ezequiel.binker@A-KINECT-01 MINGW64 ~
$ cd c:/wamp/www

ezequiel.binker@A-KINECT-01 MINGW64 /c/wamp/www
$ cd jsonprueba

ezequiel.binker@A-KINECT-01 MINGW64 /c/wamp/www/jsonprueba (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   pag.php
        modified:   theme.css

no changes added to commit (use "git add" and/or "git commit -a")
ezequiel.binker@A-KINECT-01 MINGW64 /c/wamp/www/jsonprueba (master)
$
```

Como indica la imagen, Git Bash me alerta que pag.php y theme.css fueron modificados con respecto a su versión online.

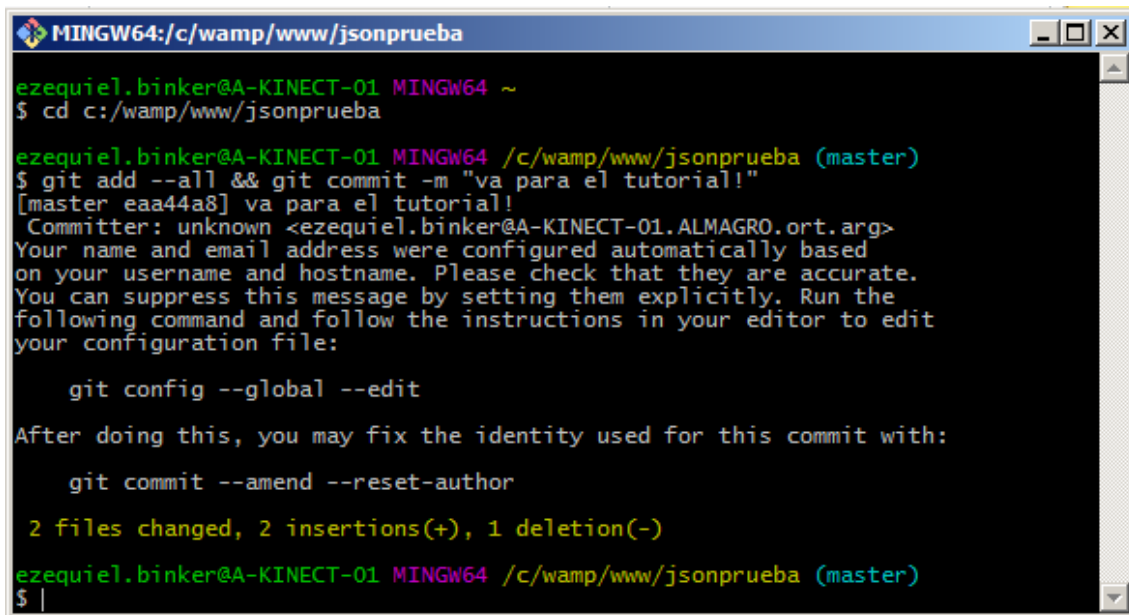
¿Cómo sincronizar los cambios hechos con la nube?

Nuestro objetivo será hacer “**push**” para subir los datos a mi repositorio Git.

Puedo elegir entre actualizar todo mi proyecto, o actualizar algunos archivos. En este tutorial veremos como actualizar todo.

La sentencia es “**git add --all && git commit -m "comment" ”**

Significa: Agrega todos los archivos para commitear, y commitea. Donde dice comment, voy a detallar qué es lo que estoy actualizando, más que nada por un tema organizativo del equipo.



```
MINGW64:/c:/wamp/www/jsonprueba
ezequiel.binker@A-KINECT-01 MINGW64 ~
$ cd c:/wamp/www/jsonprueba

ezequiel.binker@A-KINECT-01 MINGW64 /c:/wamp/www/jsonprueba (master)
$ git add --all && git commit -m "va para el tutorial!"
[master eaa44a8] va para el tutorial!
Committer: unknown <ezequiel.binker@A-KINECT-01.ALMAGRO.ort.arg>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

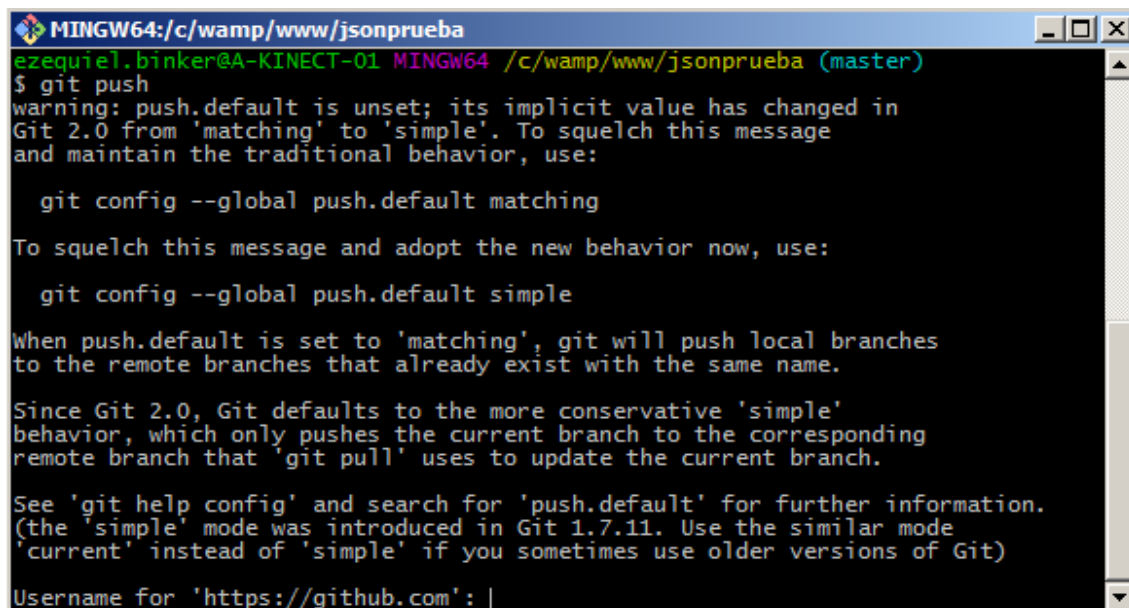
After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

2 files changed, 2 insertions(+), 1 deletion(-)
ezequiel.binker@A-KINECT-01 MINGW64 /c:/wamp/www/jsonprueba (master)
$ |
```

Luego de esto, escribo la palabra mágica “**push**”, obviamente luego de escribir git.

Me va a pedir obviamente, los datos de log in de Github, para cerciorarse que soy yo el que quiere actualizar mi repositorio y no cualquier persona. (Recuerden que los repositorios por lo general son públicos)



```
MINGW64:/c:/wamp/www/jsonprueba
ezequiel.binker@A-KINECT-01 MINGW64 /c:/wamp/www/jsonprueba (master)
$ git push
warning: push.default is unset; its implicit value has changed in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the traditional behavior, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': |
```

Ingreso mi nombre de usuario y contraseña y presiono enter.

```
MINGW64:/c:/wamp/www/jsonprueba

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': ezebinker
Password for 'https://ezebinker@github.com':
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 404 bytes | 0 bytes/s, done.
Total 4 (delta 3), reused 0 (delta 0)
To https://github.com/ezebinker/jsonprueba.git
   393cb93..eaa44a8  master -> master

ezequiel.binker@A-KINECT-01 MINGW64 /c:/wamp/www/jsonprueba (master)
$ |
```

Finalmente, ya todos mis cambios fueron subidos a la nube de Github!!