Guía rápida de Git en modo gráfico

Hipólito Guzmán Miranda
Universidad de Sevilla

Contenido

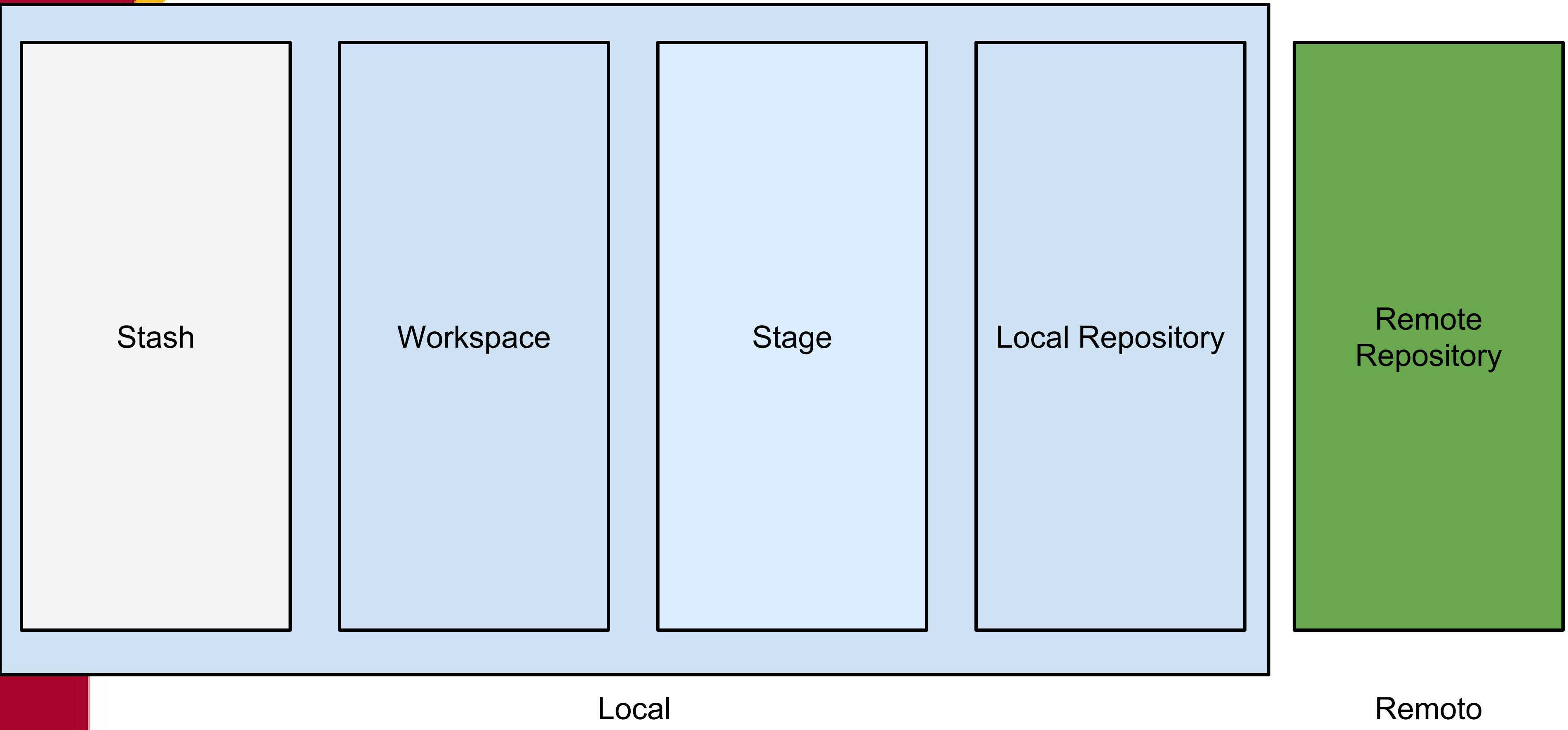
- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout

Contenido

- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout

Áreas de git

Áreas



Stash

Workspace

Stage

Local Repository

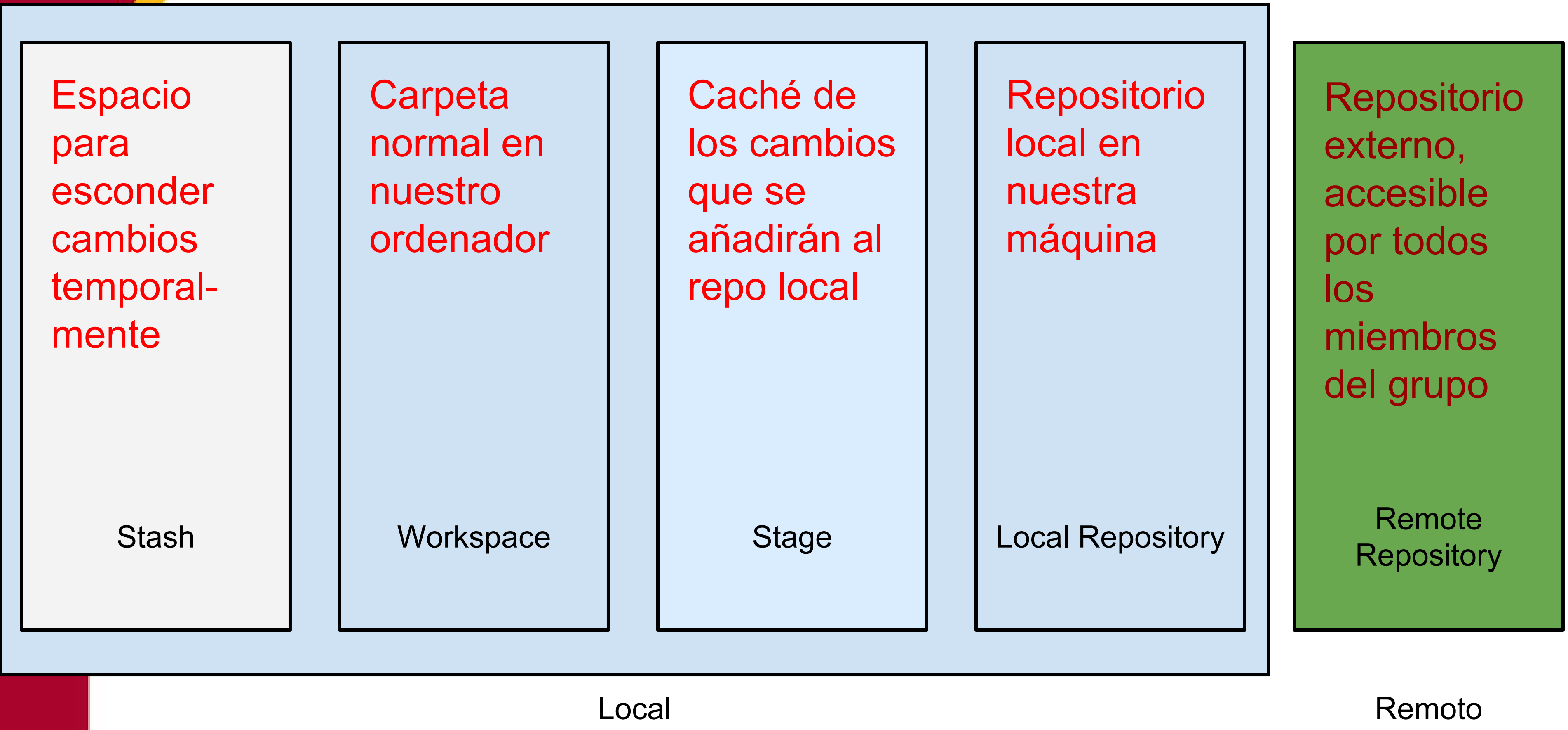
Remote Repository

Local

Remoto

Áreas de git

Descripción



Espacio para esconder cambios temporal-mente

Stash

Carpeta normal en nuestro ordenador

Workspace

Caché de los cambios que se añadirán al repo local

Stage

Repositorio local en nuestra máquina

Local Repository

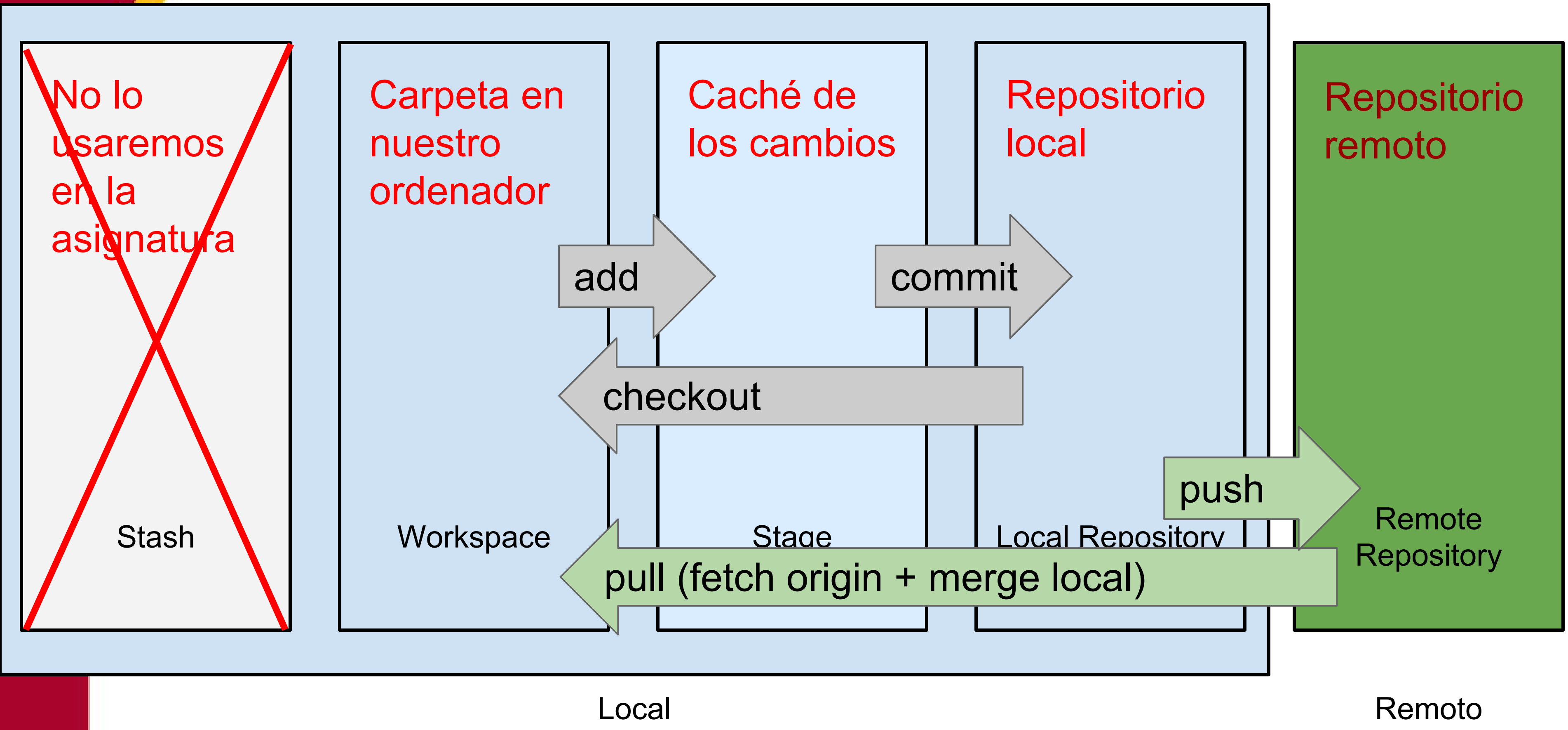
Repositorio externo, accesible por todos los miembros del grupo

Remote Repository

Local

Remoto

Comandos

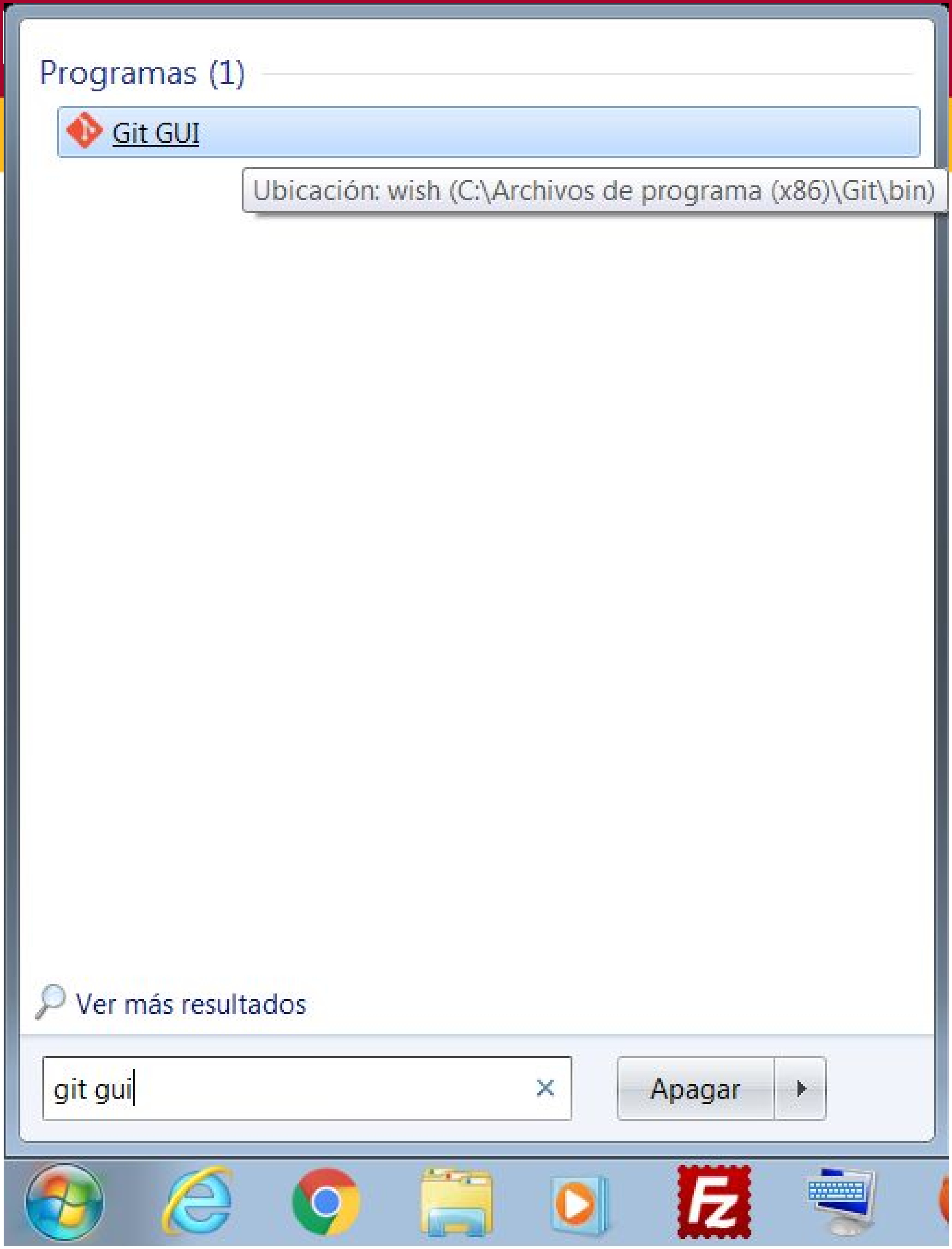

No lo usaremos en la asignatura
Stash
Carpeta en nuestro ordenador
Workspace
Caché de los cambios
Stage
Repositorio local
Local Repository
Repositorio remoto
Remote Repository
add
commit
checkout
push
pull (fetch origin + merge local)
Local
Remoto

Contenido

- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout

C

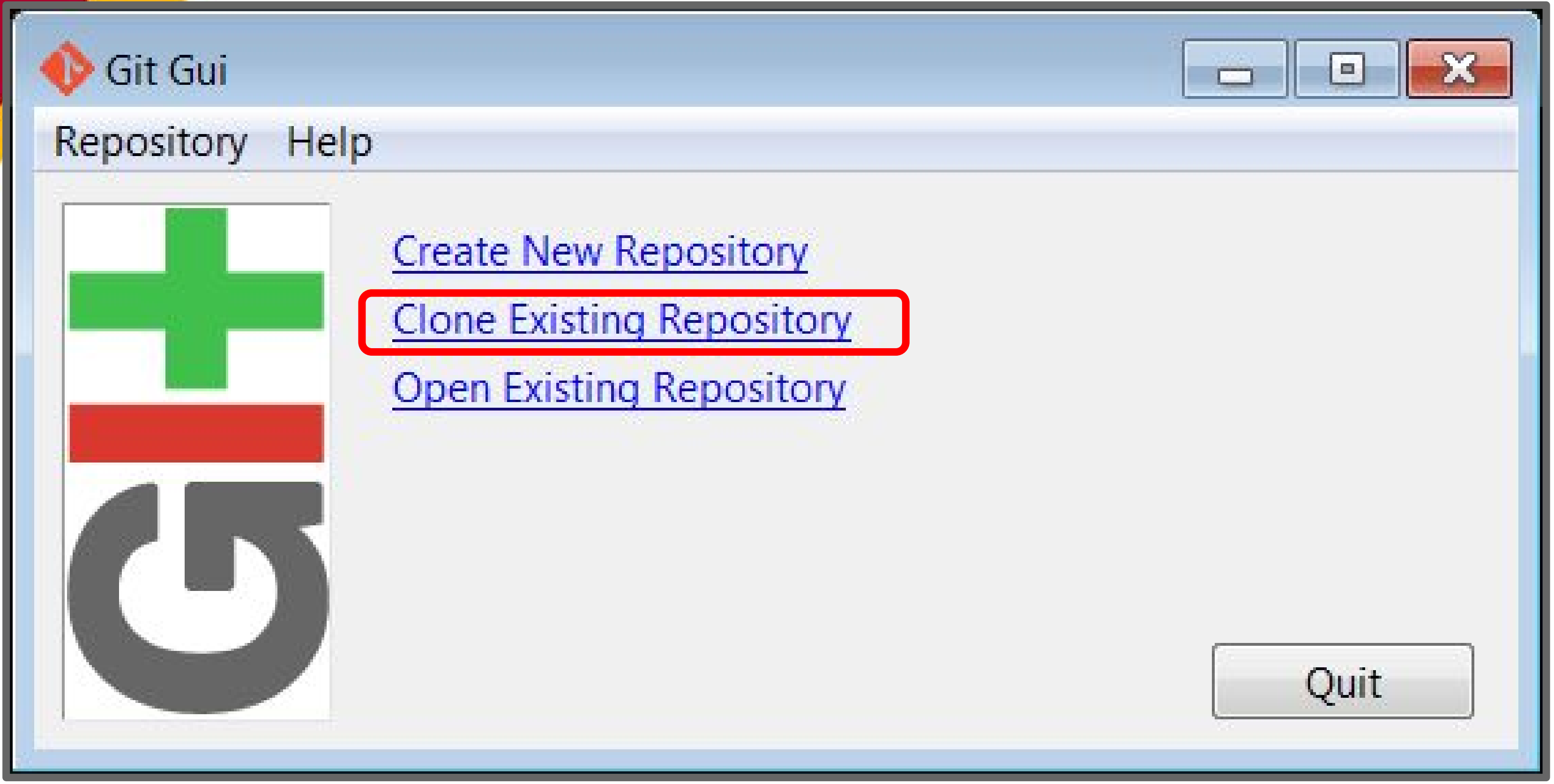
Abrimos Git GUI



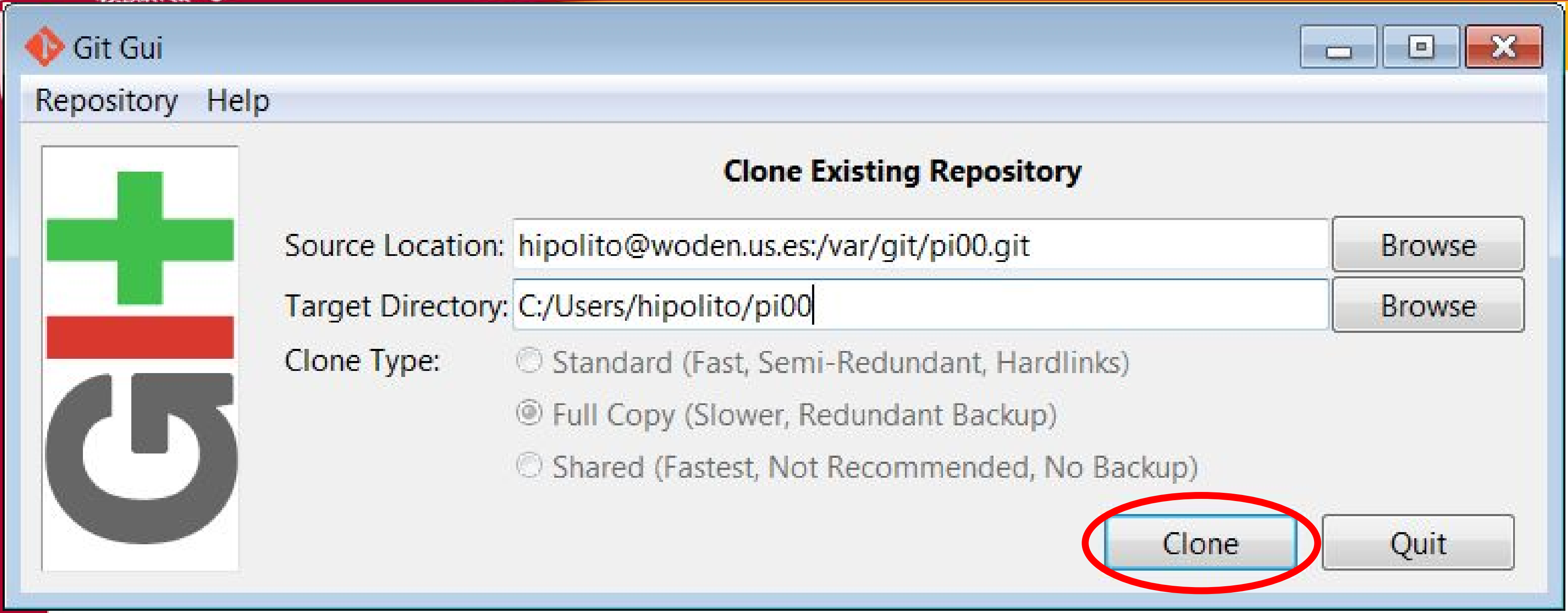
Ubicación: wish (C:\Archivos de programa (x86)\Git\bin)

git gui

Clonar el repo



Clonar el repo



Clone Existing Repository

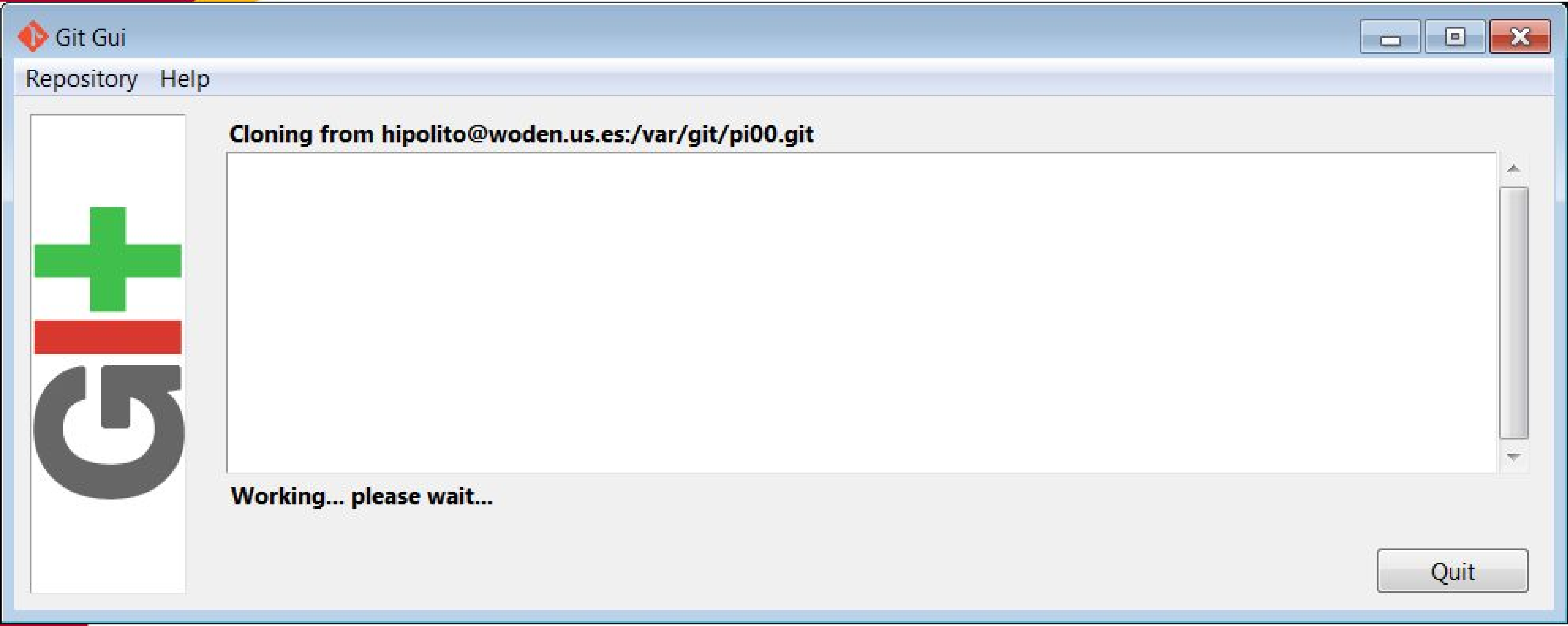
Source Location: hipolito@woden.us.es:/var/git/pi00.git

Target Directory: C:/Users/hipolito/pi00

Clone Type:
○ Standard (Fast, Semi-Redundant, Hardlinks)
● Full Copy (Slower, Redundant Backup)
○ Shared (Fastest, Not Recommended, No Backup)

- Source es `usuario@servidor:ruta_al_repo`
- Cambia `hipolito` por tu usuario y `pi00.git` por el repositorio de tu grupo
- El target directory será creado al clonar el repositorio

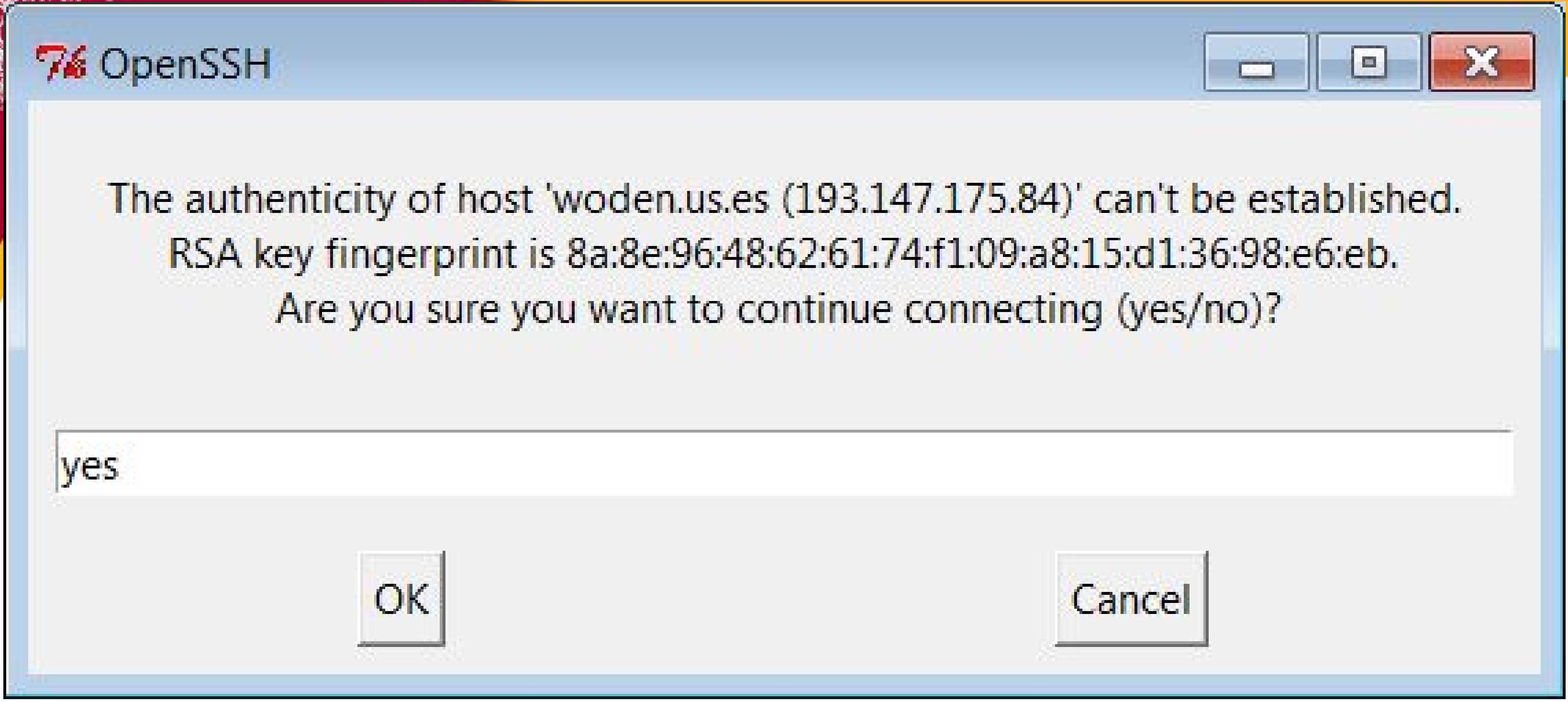
Clonar el repo



Cloning from hipolito@woden.us.es:/var/git/pi00.git

Working... please wait...

Clonar el repo

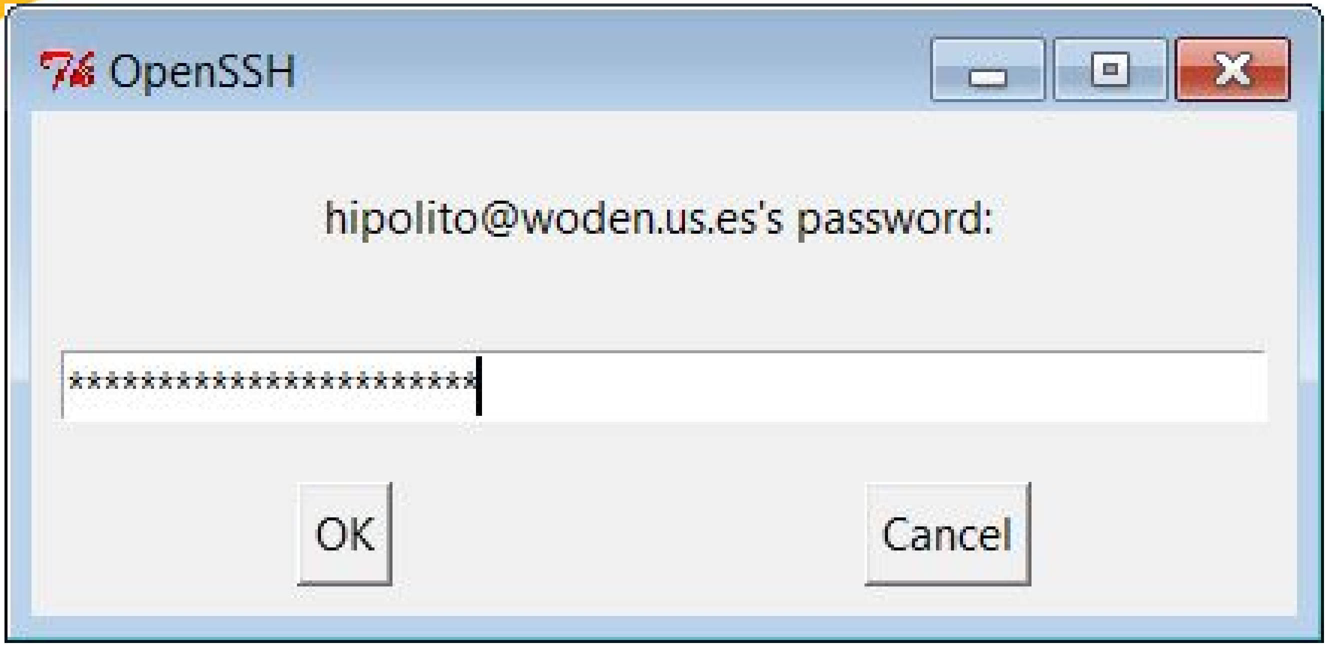


The authenticity of host 'woden.us.es (193.147.175.84)' can't be established.
RSA key fingerprint is 8a:8e:96:48:62:61:74:f1:09:a8:15:d1:36:98:e6:eb.
Are you sure you want to continue connecting (yes/no)?

yes

Este mensaje sólo lo veremos la primera vez que conectemos con un servidor (en nuestro caso, el servidor es woden.us.es)

Clonar el repo

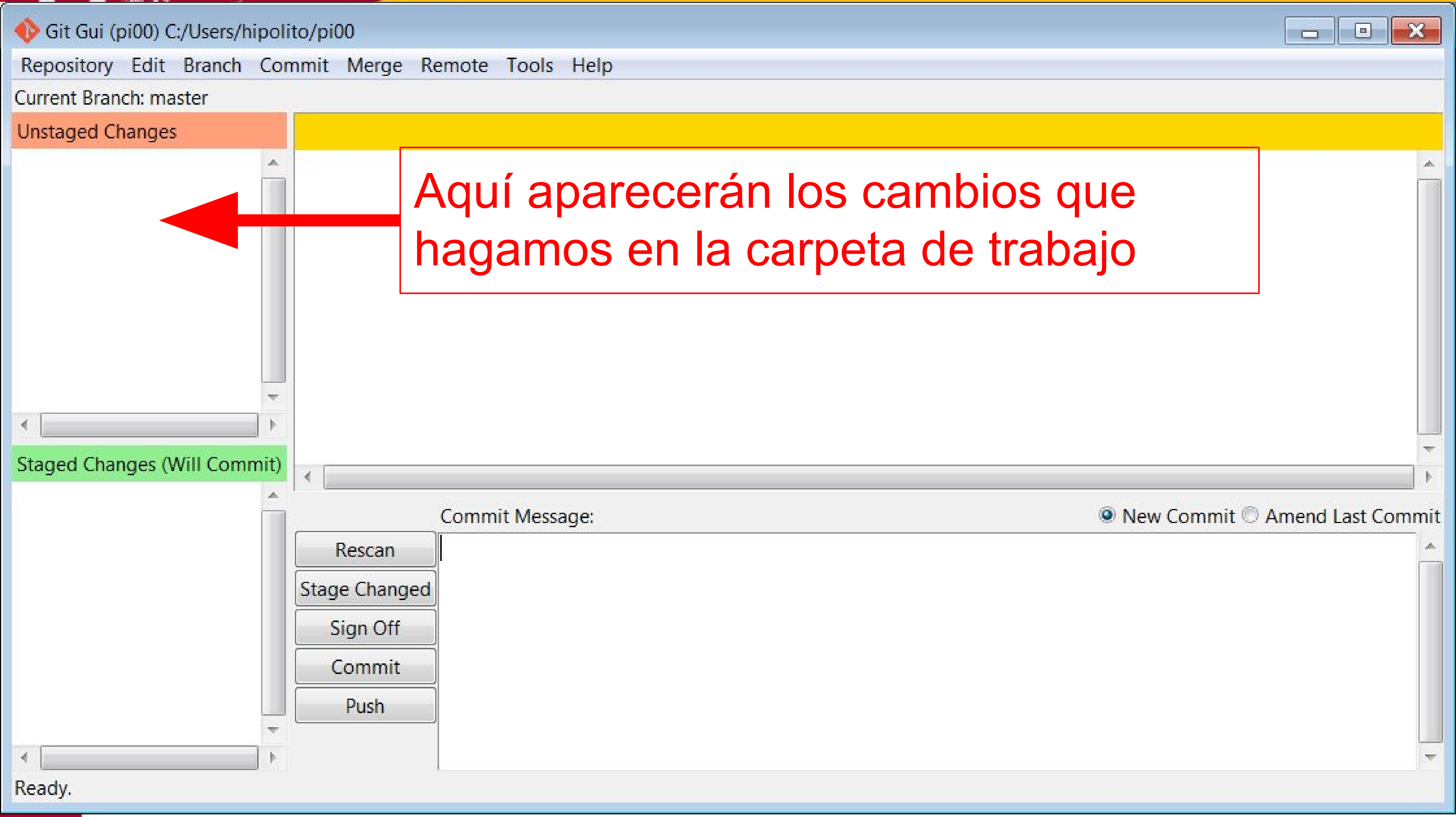


OpenSSH

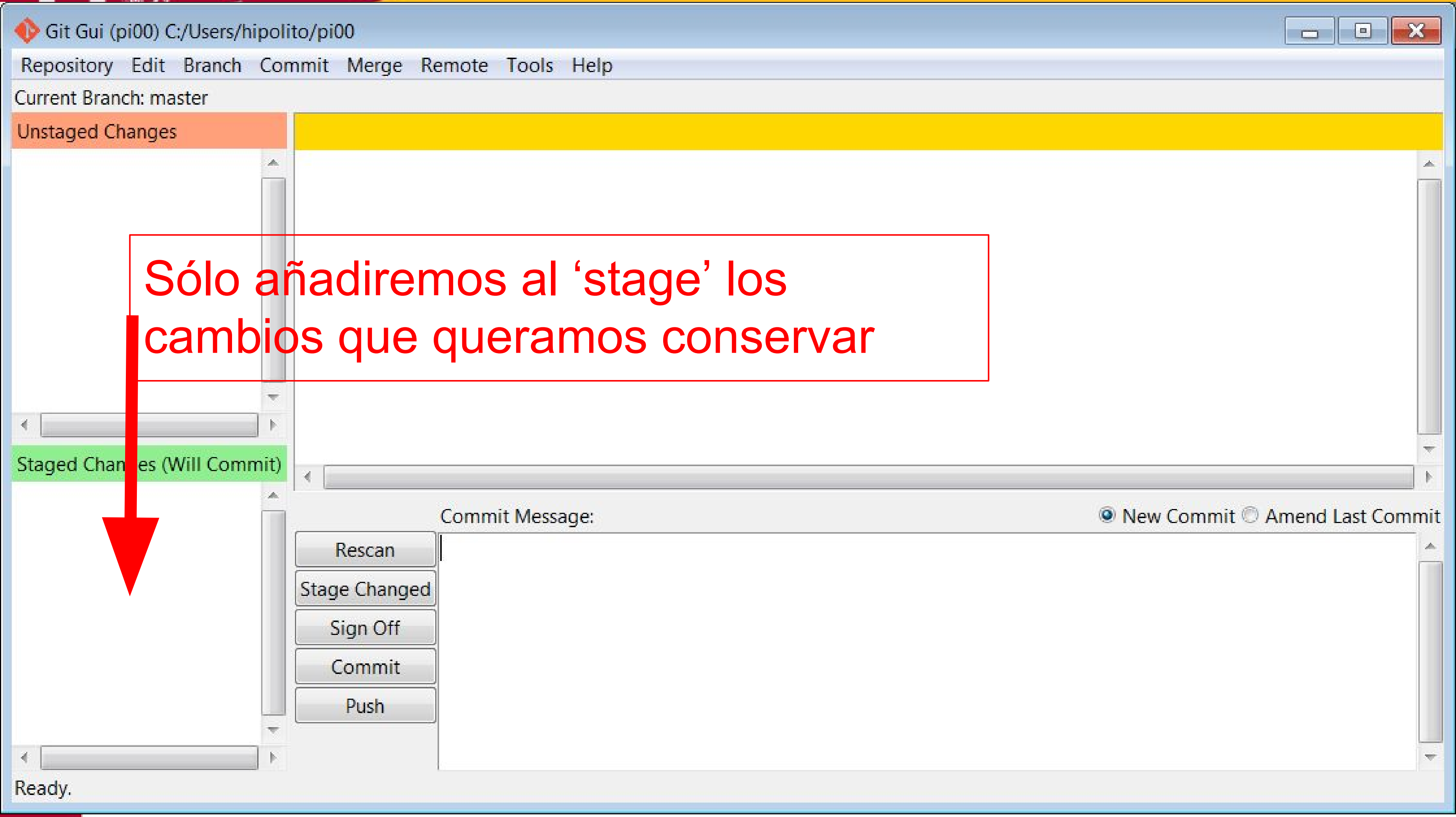
hipolito@woden.us.es's password:

****

Siempre que hagamos una operación con el repositorio remoto nos pedirá el password



Aquí aparecerán los cambios que hagamos en la carpeta de trabajo



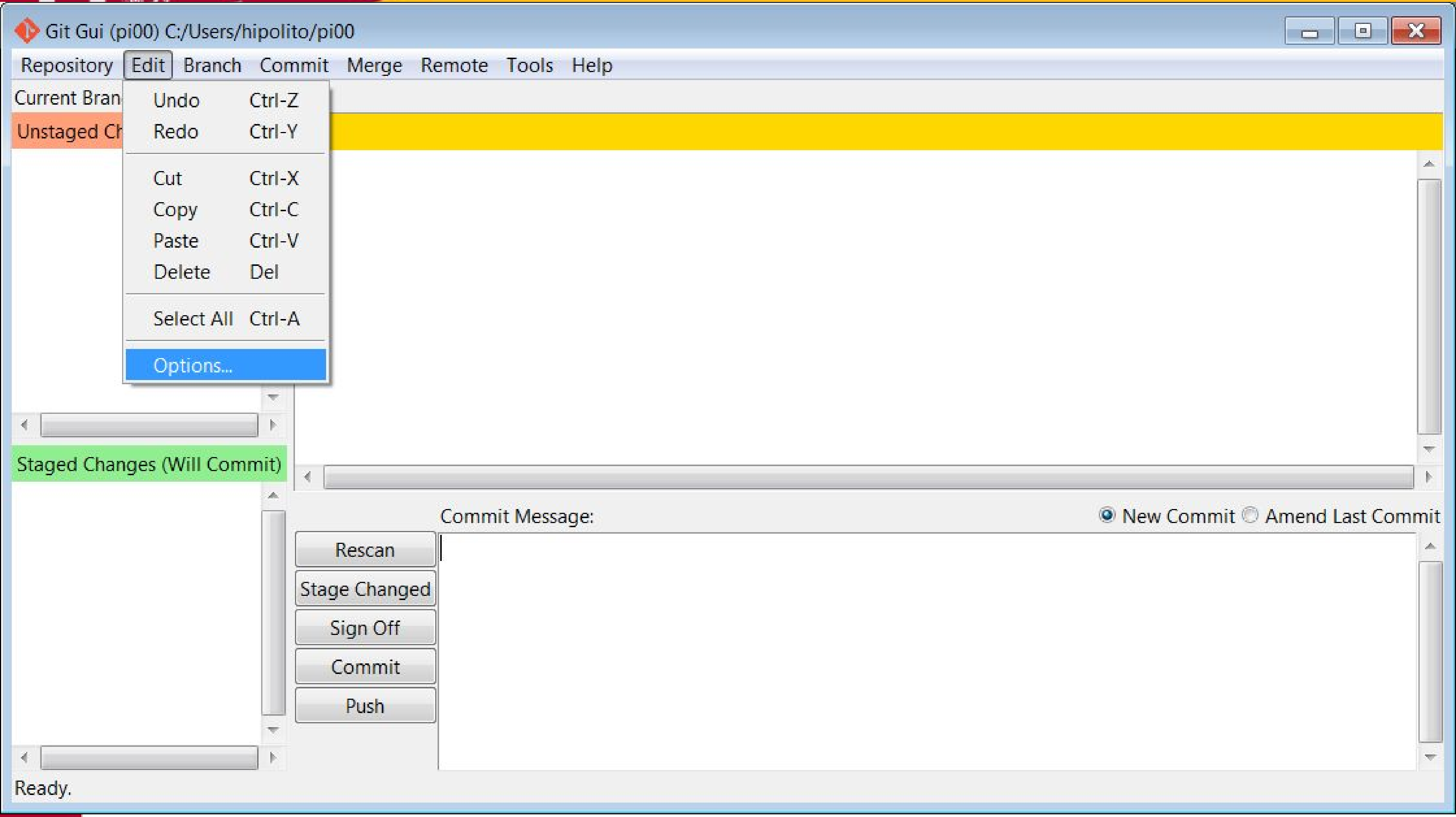
● New Commit ○ Amend Last Commit

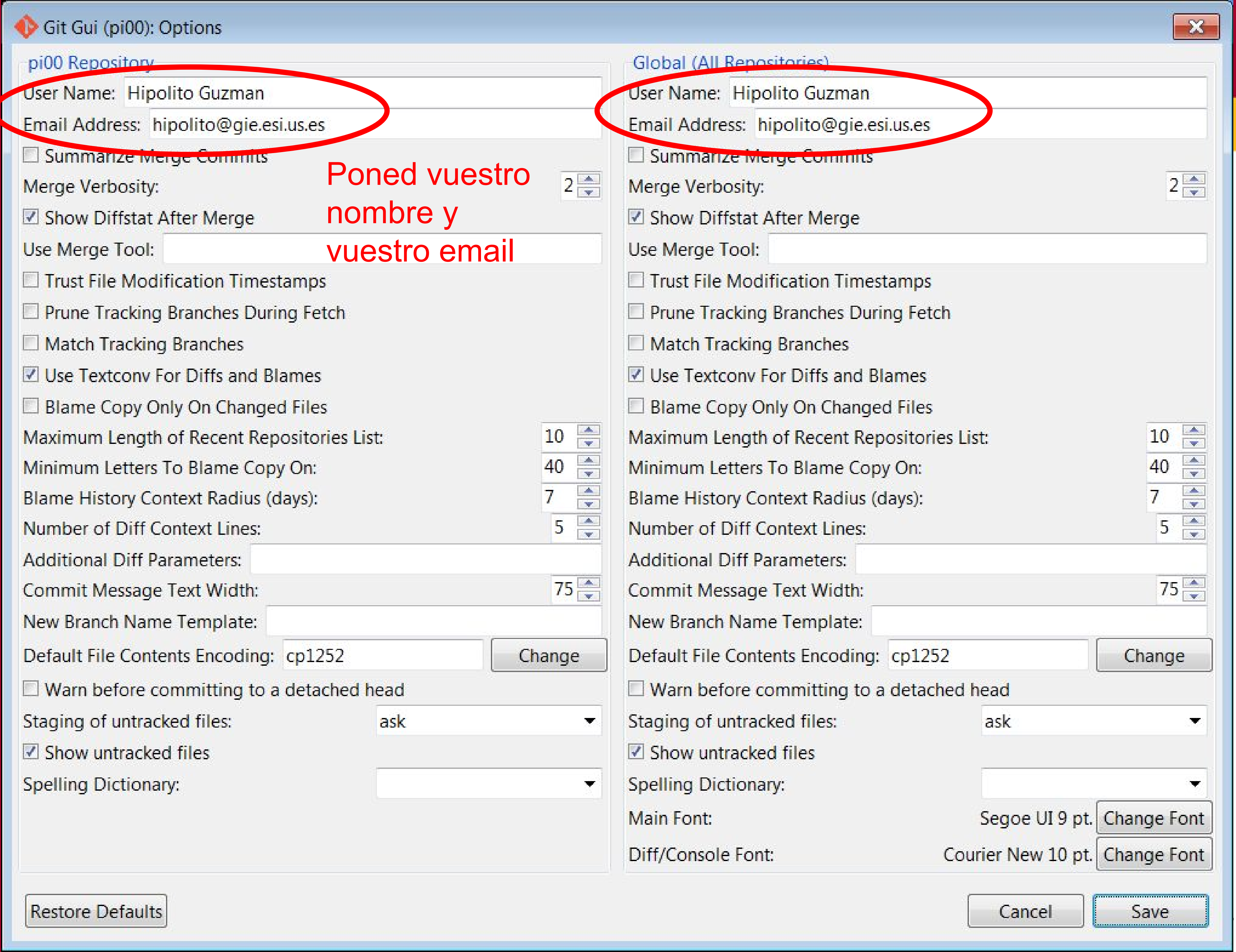
Sólo añadiremos al ‘stage’ los cambios que queramos conservar

Contenido

- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout

Configuramos git





Poned vuestro nombre y vuestro email

pi00 Repository

User Name: Hipolito Guzman

Email Address: hipolito@gie.esi.us.es

□ Summarize Merge Commits

Merge Verbosity: 2

☑ Show Diffstat After Merge

Use Merge Tool:

□ Trust File Modification Timestamps

□ Prune Tracking Branches During Fetch

□ Match Tracking Branches

☑ Use Textconv For Diffs and Blames

□ Blame Copy Only On Changed Files

Maximum Length of Recent Repositories List: 10

Minimum Letters To Blame Copy On: 40

Blame History Context Radius (days): 7

Number of Diff Context Lines: 5

Additional Diff Parameters:

Commit Message Text Width: 75

New Branch Name Template:

Default File Contents Encoding: cp1252 Change

□ Warn before committing to a detached head

Staging of untracked files: ask

☑ Show untracked files

Spelling Dictionary:

Global (All Repositories)

User Name: Hipolito Guzman

Email Address: hipolito@gie.esi.us.es

□ Summarize Merge Commits

Merge Verbosity: 2

☑ Show Diffstat After Merge

Use Merge Tool:

□ Trust File Modification Timestamps

□ Prune Tracking Branches During Fetch

□ Match Tracking Branches

☑ Use Textconv For Diffs and Blames

□ Blame Copy Only On Changed Files

Maximum Length of Recent Repositories List: 10

Minimum Letters To Blame Copy On: 40

Blame History Context Radius (days): 7

Number of Diff Context Lines: 5

Additional Diff Parameters:

Commit Message Text Width: 75

New Branch Name Template:

Default File Contents Encoding: cp1252 Change

□ Warn before committing to a detached head

Staging of untracked files: ask

☑ Show untracked files

Spelling Dictionary:

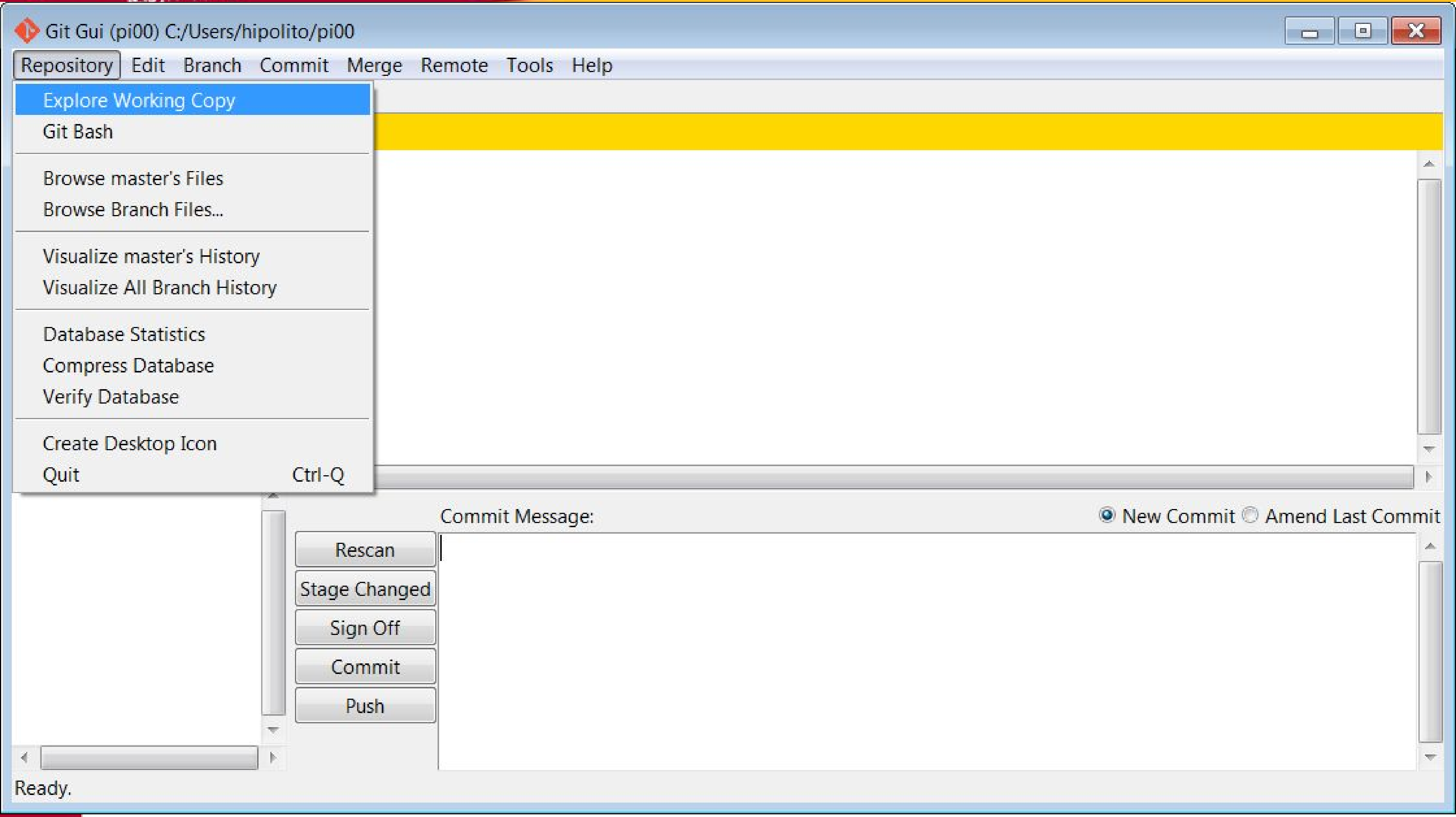
Main Font: Segoe UI 9 pt. Change Font

Diff/Console Font: Courier New 10 pt. Change Font

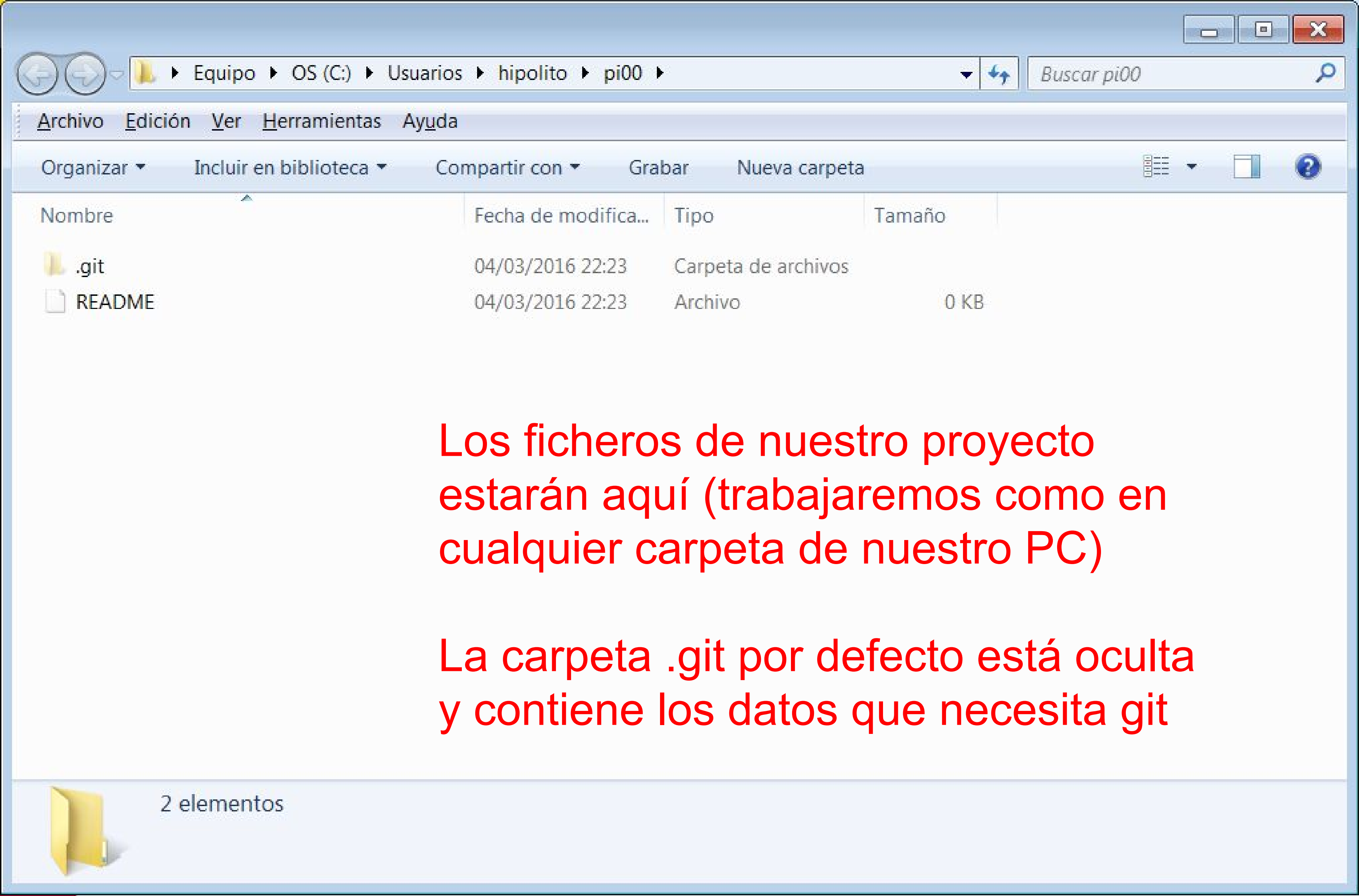
Contenido

- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout

Añadir ficheros al stage



Abrimos la carpeta del repo

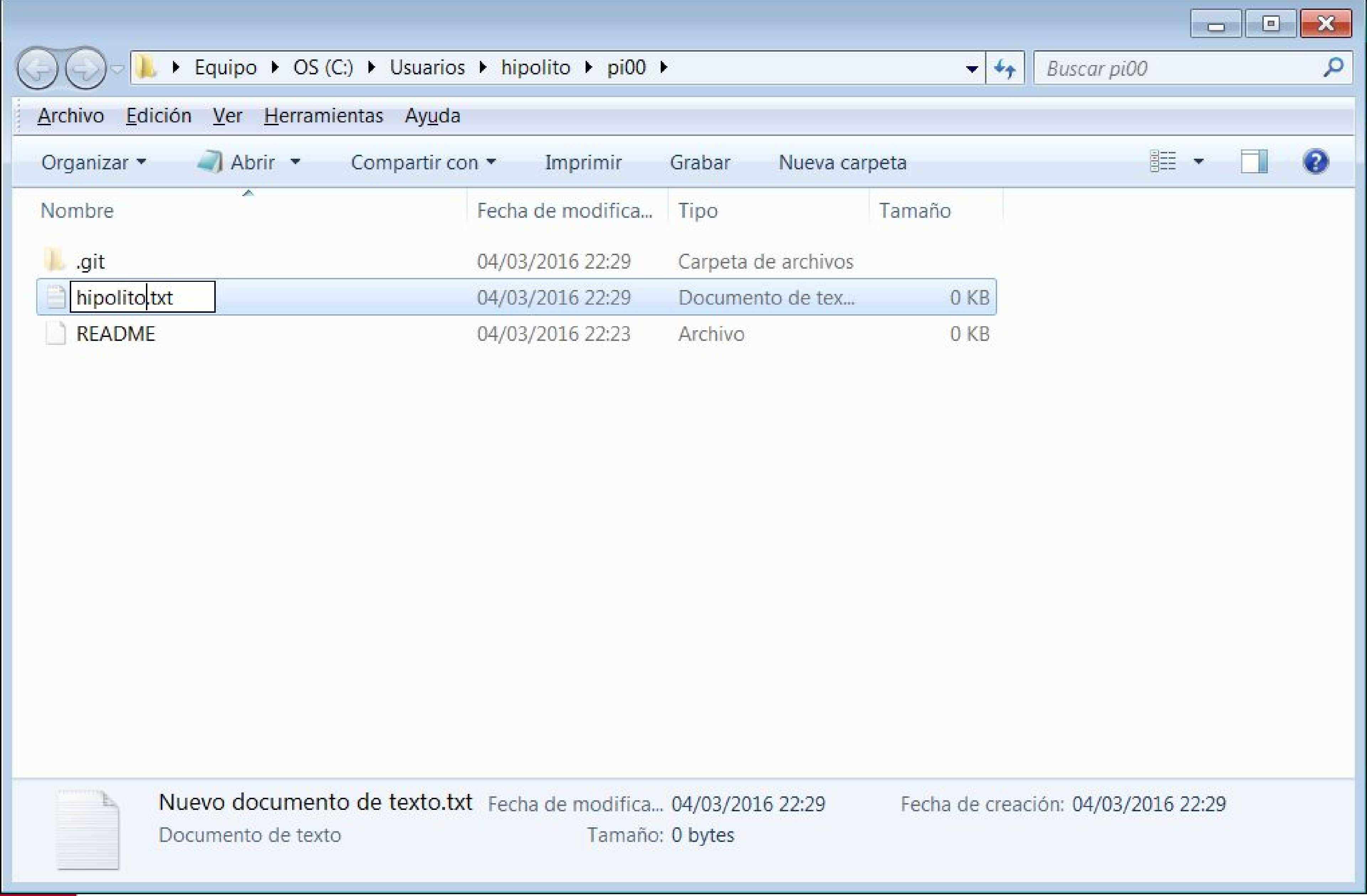


Nombre	Fecha de modifica...	Tipo	Tamaño
.git	04/03/2016 22:23	Carpeta de archivos	
README	04/03/2016 22:23	Archivo	0 KB

Los ficheros de nuestro proyecto estarán aquí (trabajaremos como en cualquier carpeta de nuestro PC)

La carpeta .git por defecto está oculta y contiene los datos que necesita git

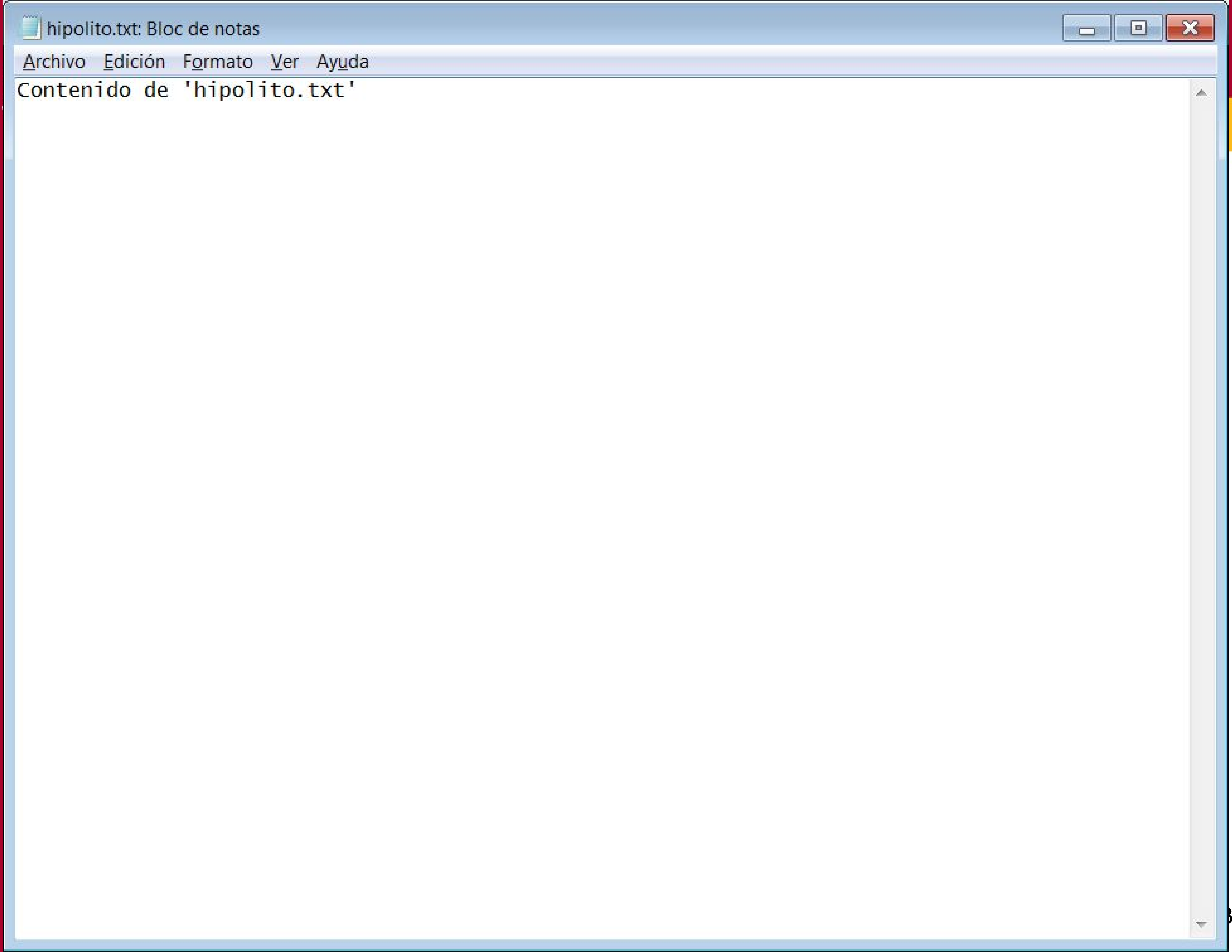
Workspace es una carpeta más de nuestro PC



Nombre	Fecha de modifica...	Tipo	Tamaño
.git	04/03/2016 22:29	Carpeta de archivos	
hipolito.txt	04/03/2016 22:29	Documento de tex...	0 KB
README	04/03/2016 22:23	Archivo	0 KB

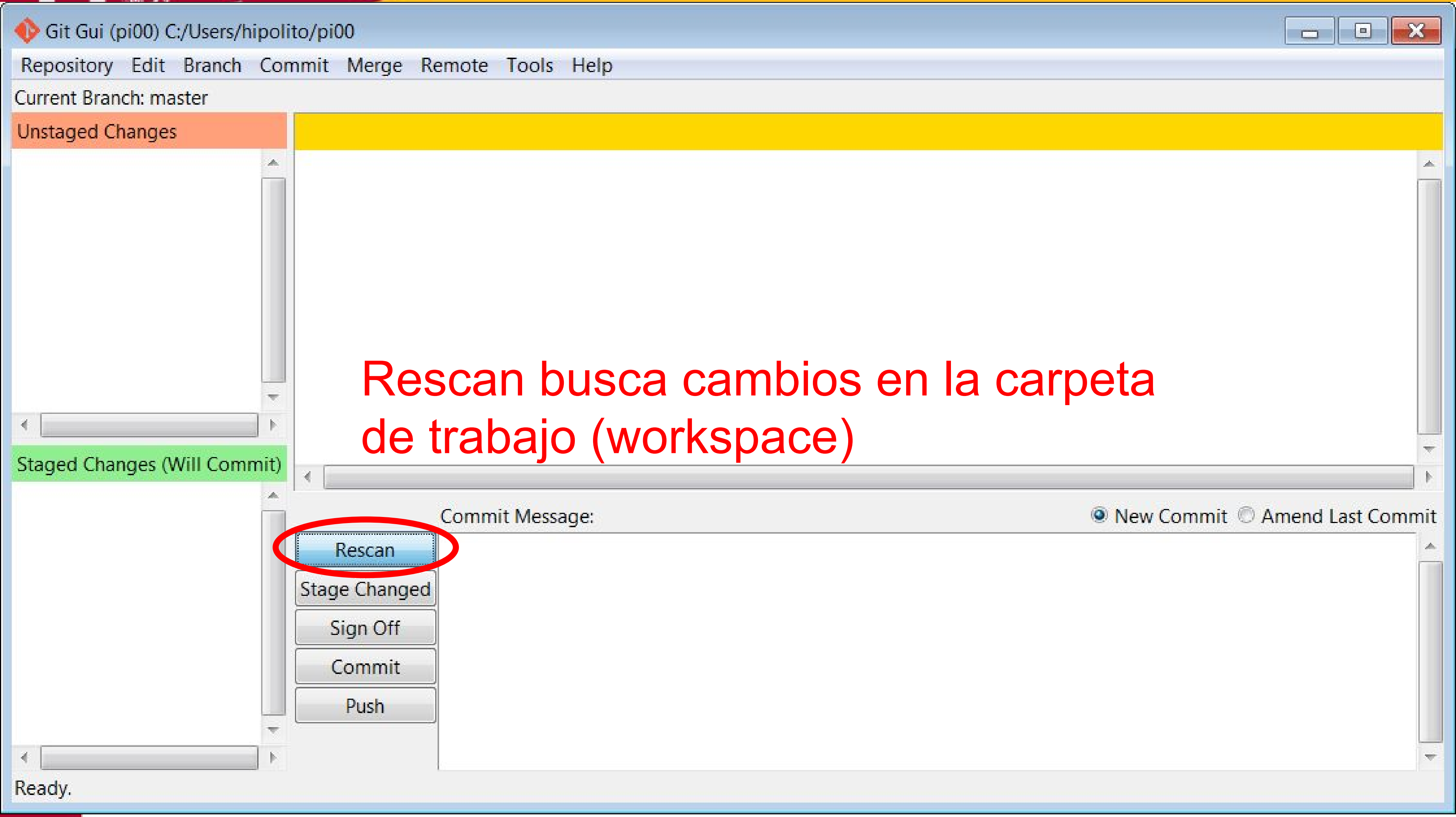
Nuevo documento de texto.txt
Documento de texto
Fecha de modifica... 04/03/2016 22:29
Tamaño: 0 bytes
Fecha de creación: 04/03/2016 22:29

Creamos un fichero con nuestro nombre



```
Contenido de 'hipolito.txt'
```

Añadir ficheros al stage



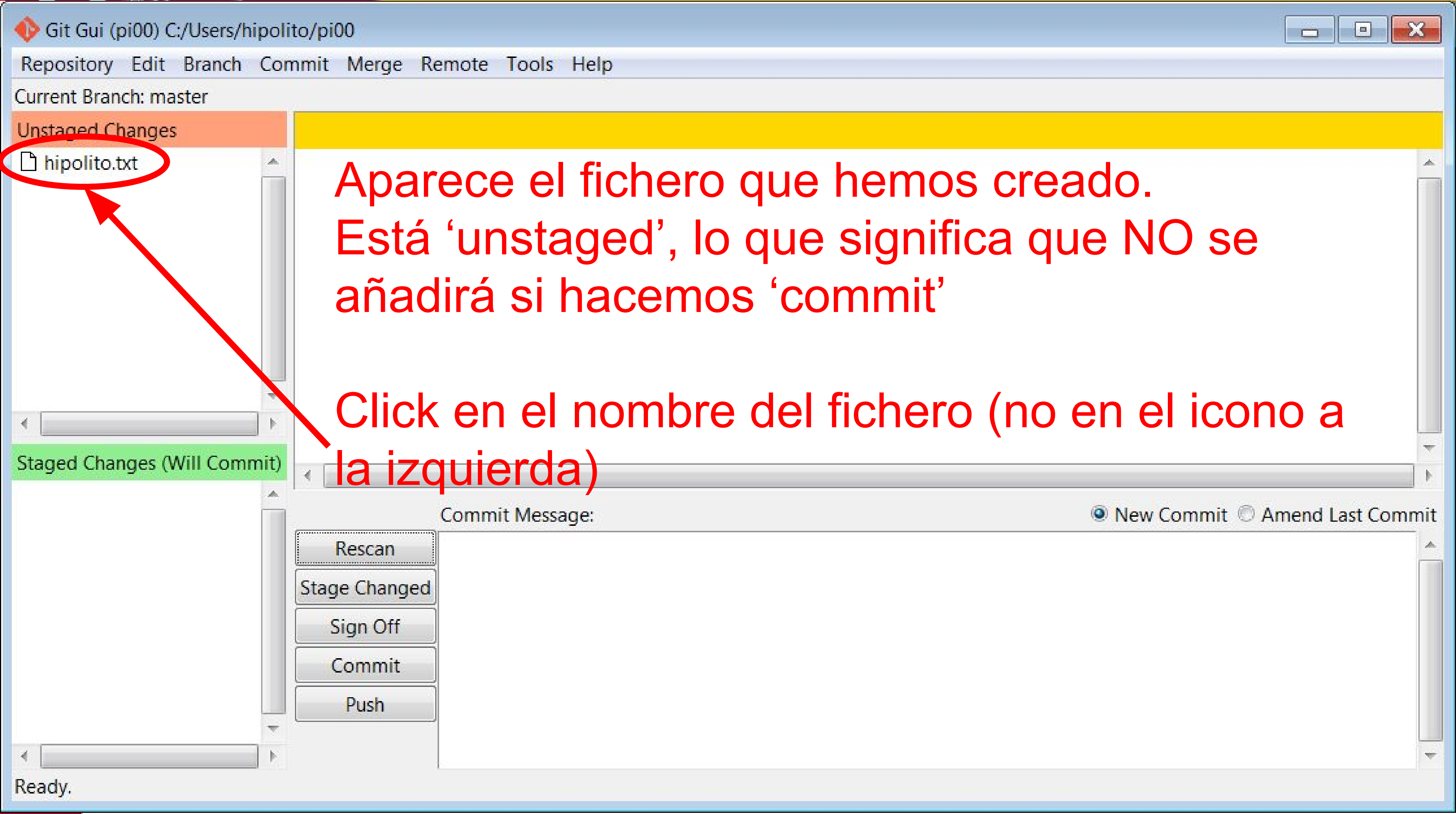
Current Branch: master

Commit Message: ● New Commit ○ Amend Last Commit

Ready.

Rescan busca cambios en la carpeta de trabajo (workspace)

Añadir ficheros al stage



Current Branch: master

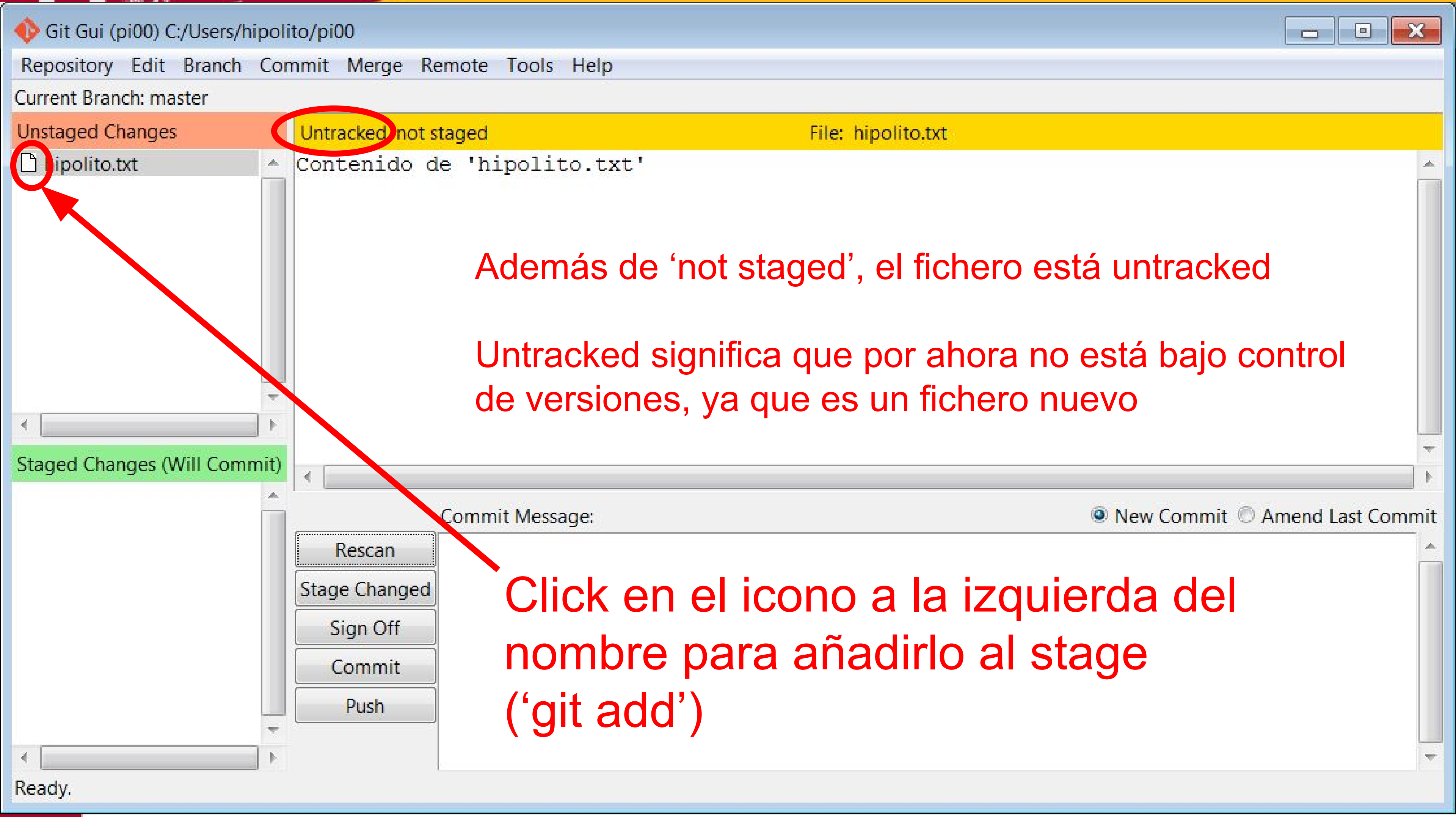
● New Commit ○ Amend Last Commit

Ready.

Aparece el fichero que hemos creado.
Está ‘unstaged’, lo que significa que NO se añadirá si hacemos ‘commit’

Click en el nombre del fichero (no en el icono a la izquierda)

Añadir ficheros al stage



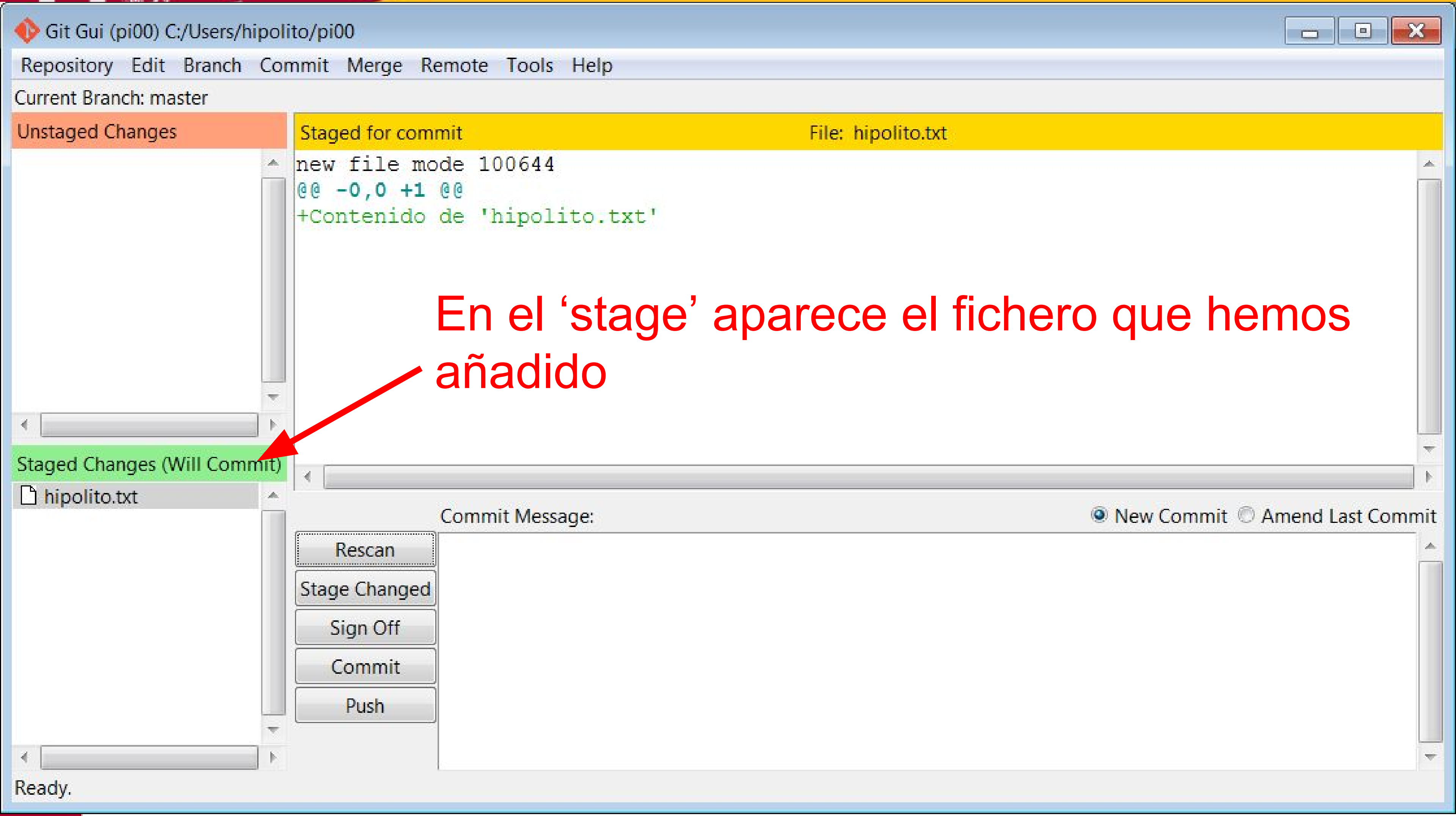
```
Contenido de 'hipolito.txt'
```

Además de ‘not staged’, el fichero está untracked

Untracked significa que por ahora no está bajo control de versiones, ya que es un fichero nuevo

Click en el icono a la izquierda del nombre para añadirlo al stage (‘git add’)

Añadir ficheros al stage



Git Gui (pi00) C:/Users/hipolito/pi00

Current Branch: master

Staged for commit File: hipolito.txt

```
new file mode 100644
@@ -0,0 +1 @@
+Contenido de 'hipolito.txt'
```

Staged Changes (Will Commit)

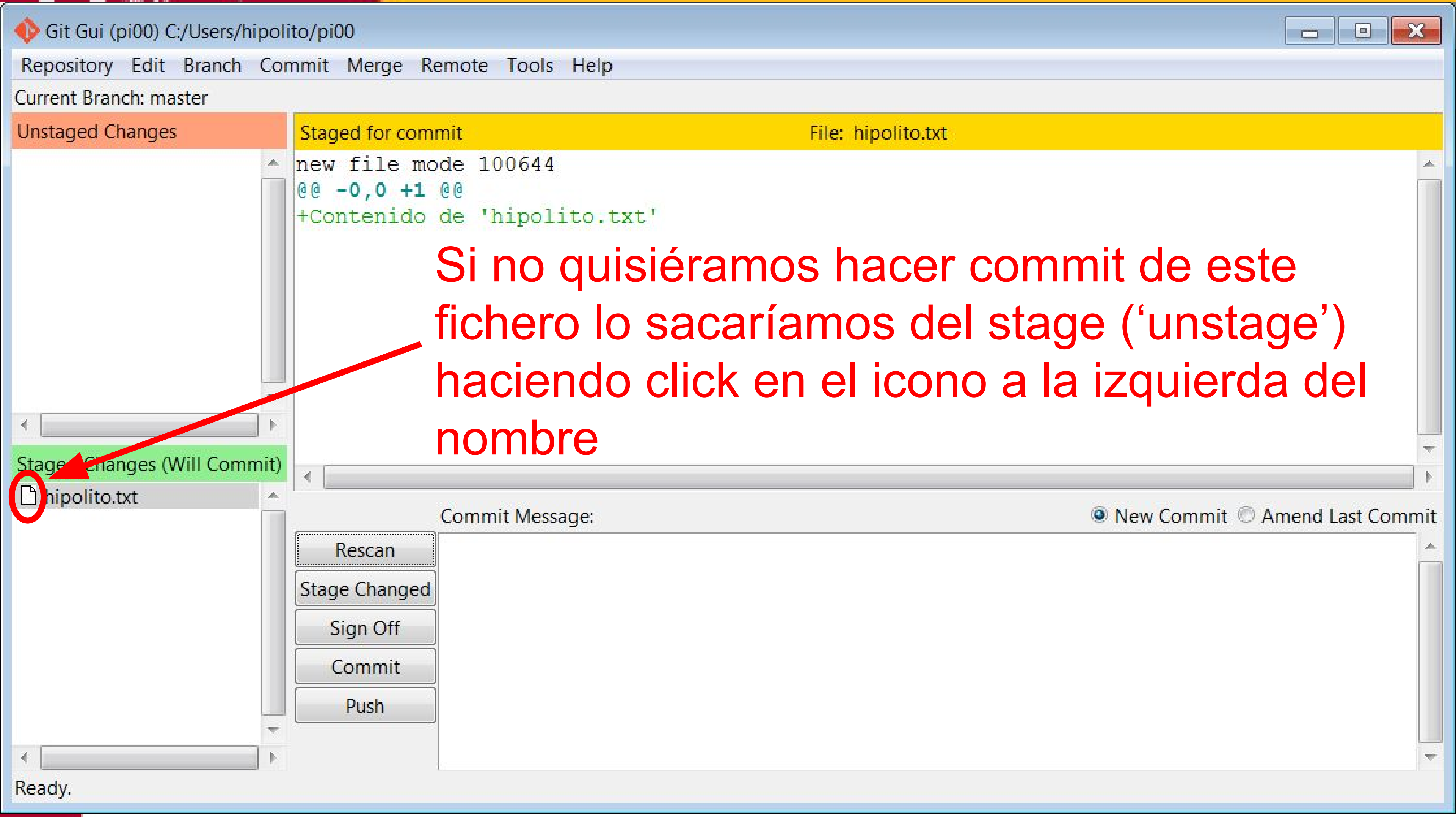
hipolito.txt

Commit Message: ● New Commit ○ Amend Last Commit

Ready.

En el ‘stage’ aparece el fichero que hemos añadido

Añadir ficheros al stage



Git Gui (pi00) C:/Users/hipolito/pi00
Repository Edit Branch Commit Merge Remote Tools Help
Unstaged Changes
Staged for commit
Staged Changes (Will Commit)
Rescan
Stage Changed
Sign Off
Commit
Push

Current Branch: master

File: hipolito.txt

```
new file mode 100644
@@ -0,0 +1 @@
+Contenido de 'hipolito.txt'
```

hipolito.txt

Commit Message: ● New Commit ○ Amend Last Commit

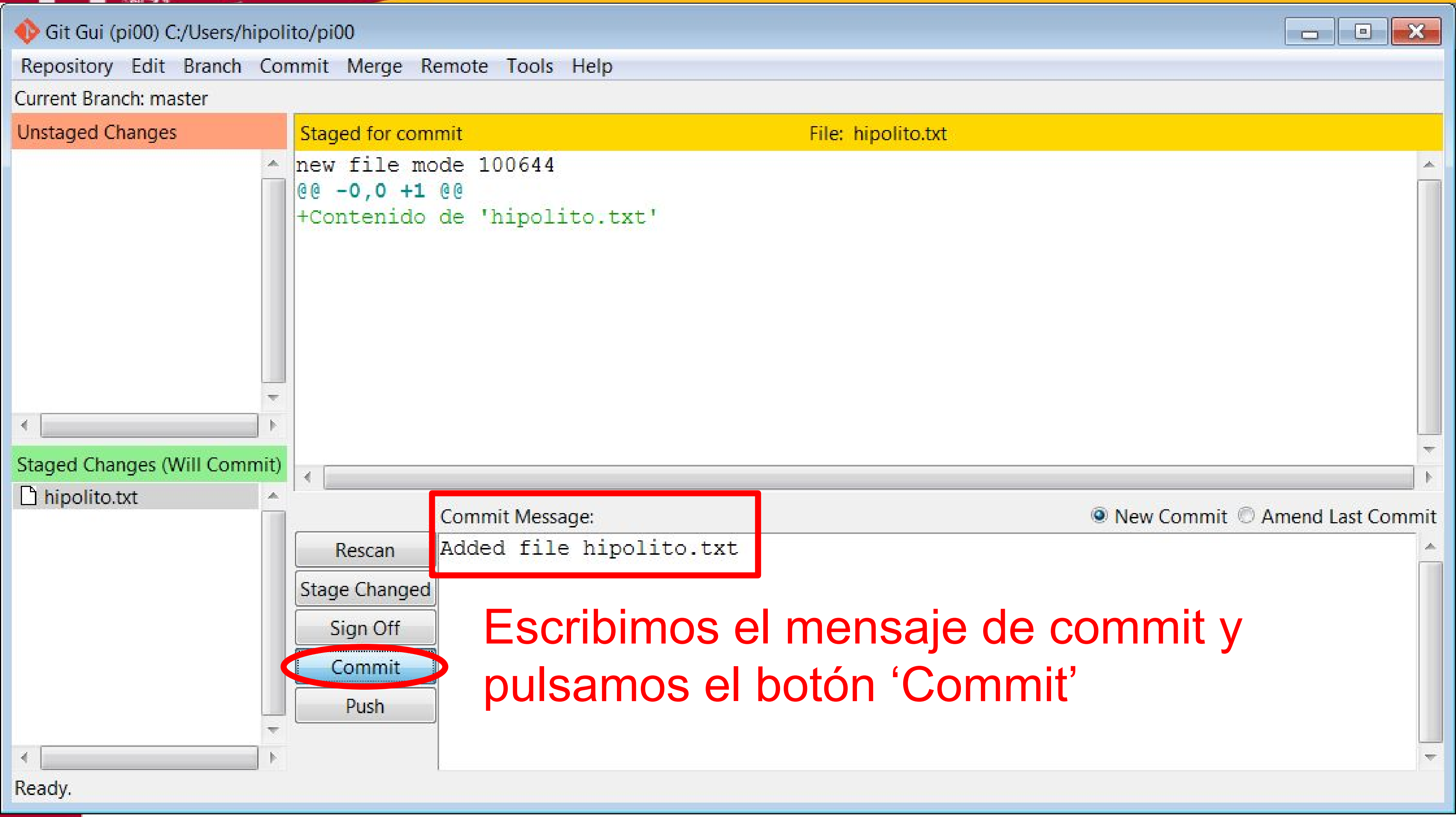
Ready.

Si no quisiéramos hacer commit de este fichero lo sacaríamos del stage (‘unstage’) haciendo click en el icono a la izquierda del nombre

Contenido

- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout

Creando commits



Current Branch: master

Staged Changes (Will Commit): hipolito.txt

Staged for commit File: hipolito.txt

```
new file mode 100644
@@ -0,0 +1 @@
+Contenido de 'hipolito.txt'
```

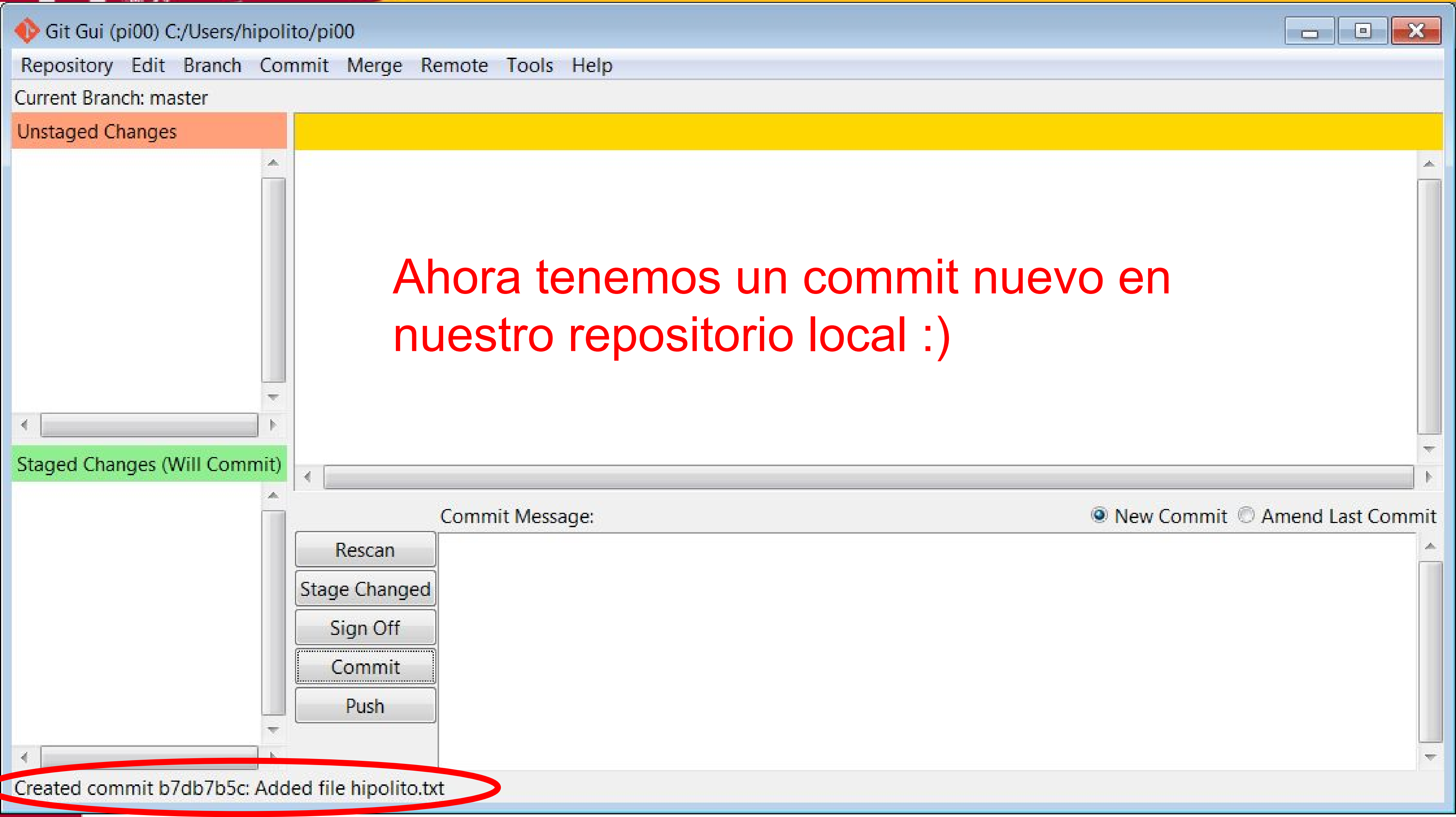
Commit Message: ● New Commit ○ Amend Last Commit

```
Added file hipolito.txt
```

Ready.

Escribimos el mensaje de commit y pulsamos el botón ‘Commit’

Creando commits



Current Branch: master

Commit Message:

● New Commit ○ Amend Last Commit

```
Created commit b7db7b5c: Added file hipolito.txt
```

Ahora tenemos un commit nuevo en nuestro repositorio local :)

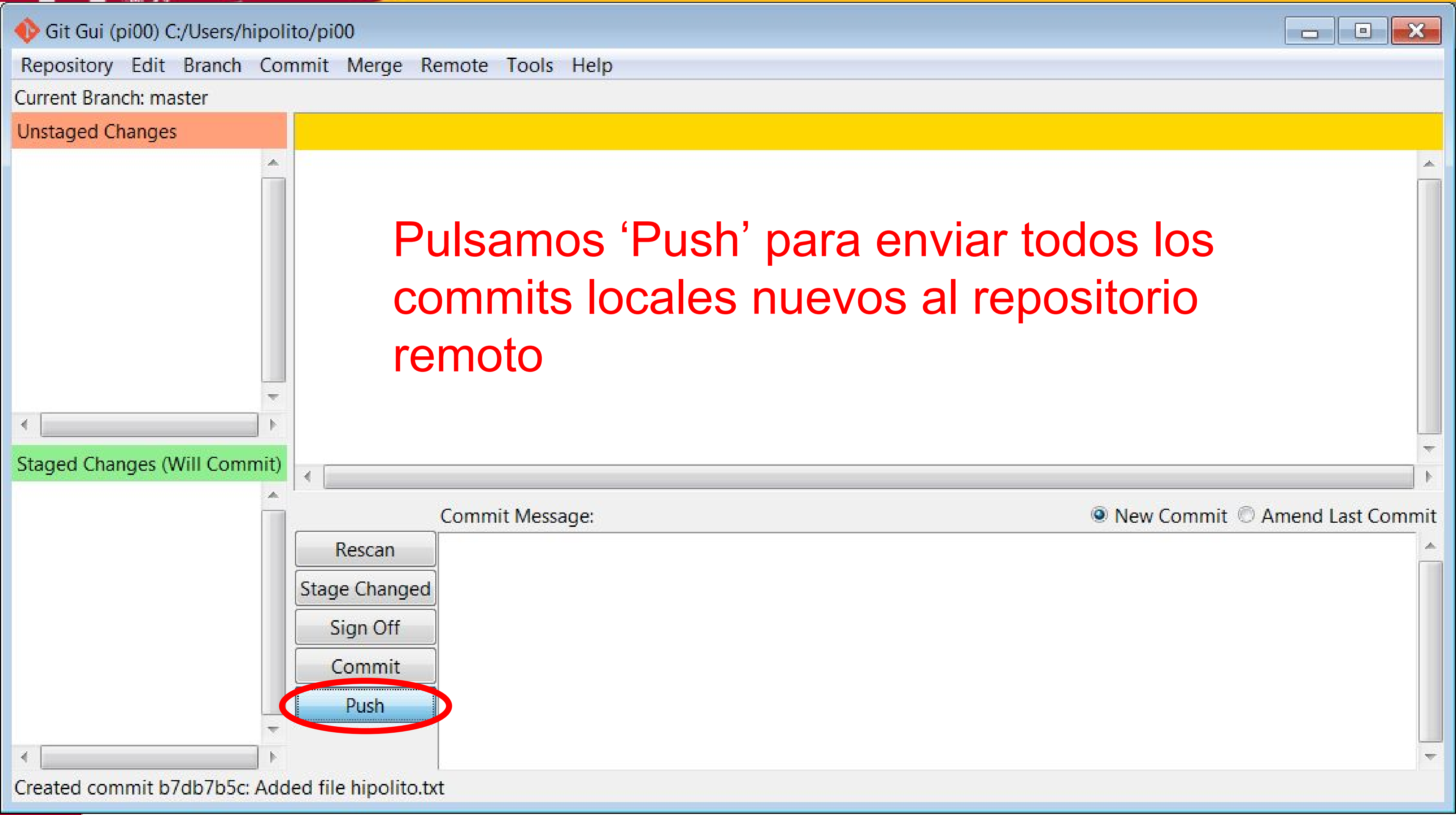
Formato correcto de un mensaje de commit

Aunque es texto plano y podemos hacerlo como queramos, lo recomendado es:

- Primera línea:
  - Resumen conciso de los cambios
  - Máximo 50 caracteres
- Si añadimos una descripción más extensa:
  - Dejamos la segunda línea en blanco
  - Tercera línea y posteriores:
    - Descripción más detallada
    - máximo 72 caracteres

Contenido

- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout



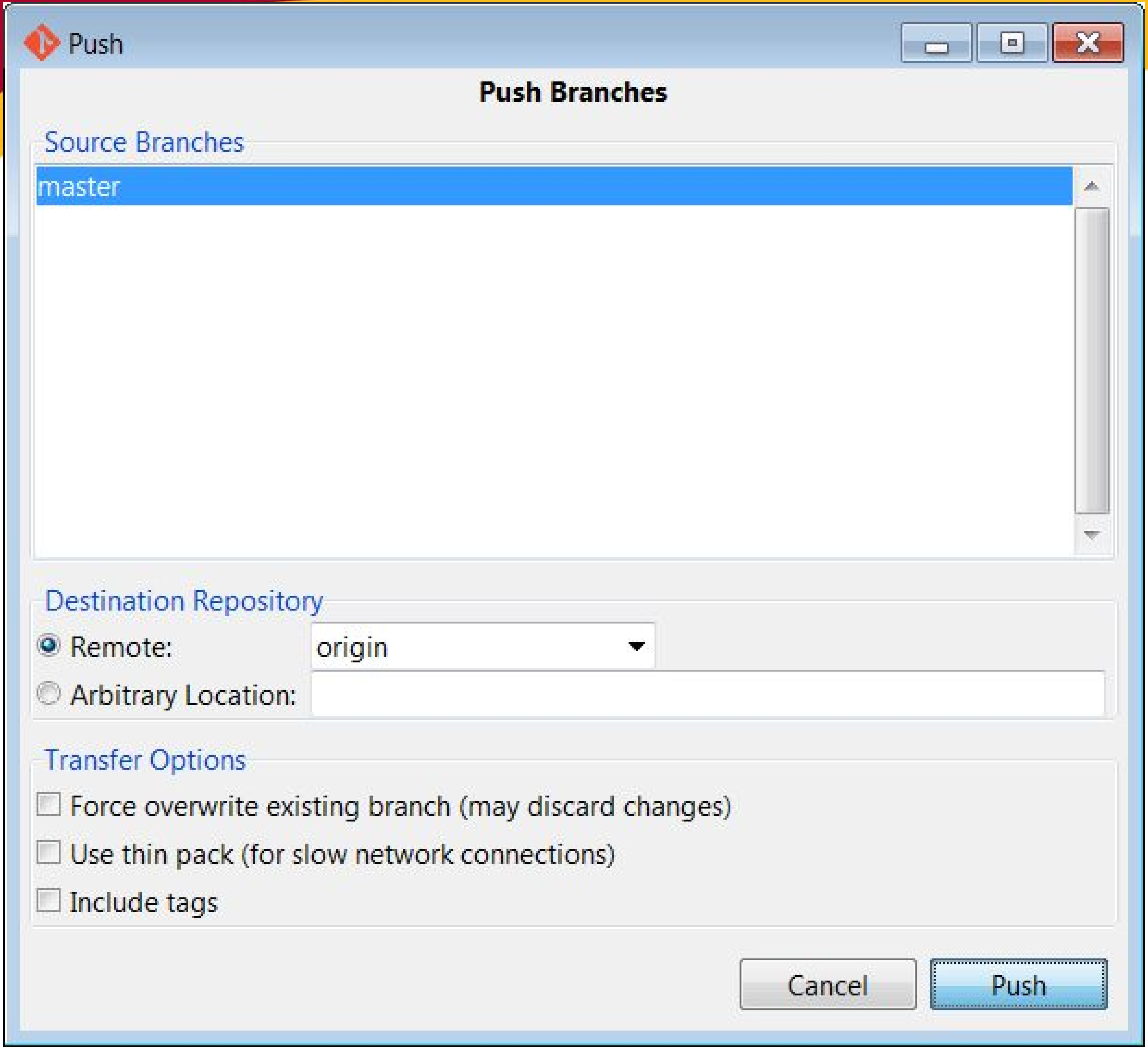
Current Branch: master

Commit Message: ● New Commit ○ Amend Last Commit

Created commit b7db7b5c: Added file hipolito.txt

Pulsamos ‘Push’ para enviar todos los commits locales nuevos al repositorio remoto

Push



Push Branches

Source Branches

master

Destination Repository

● Remote: origin

○ Arbitrary Location:

Transfer Options

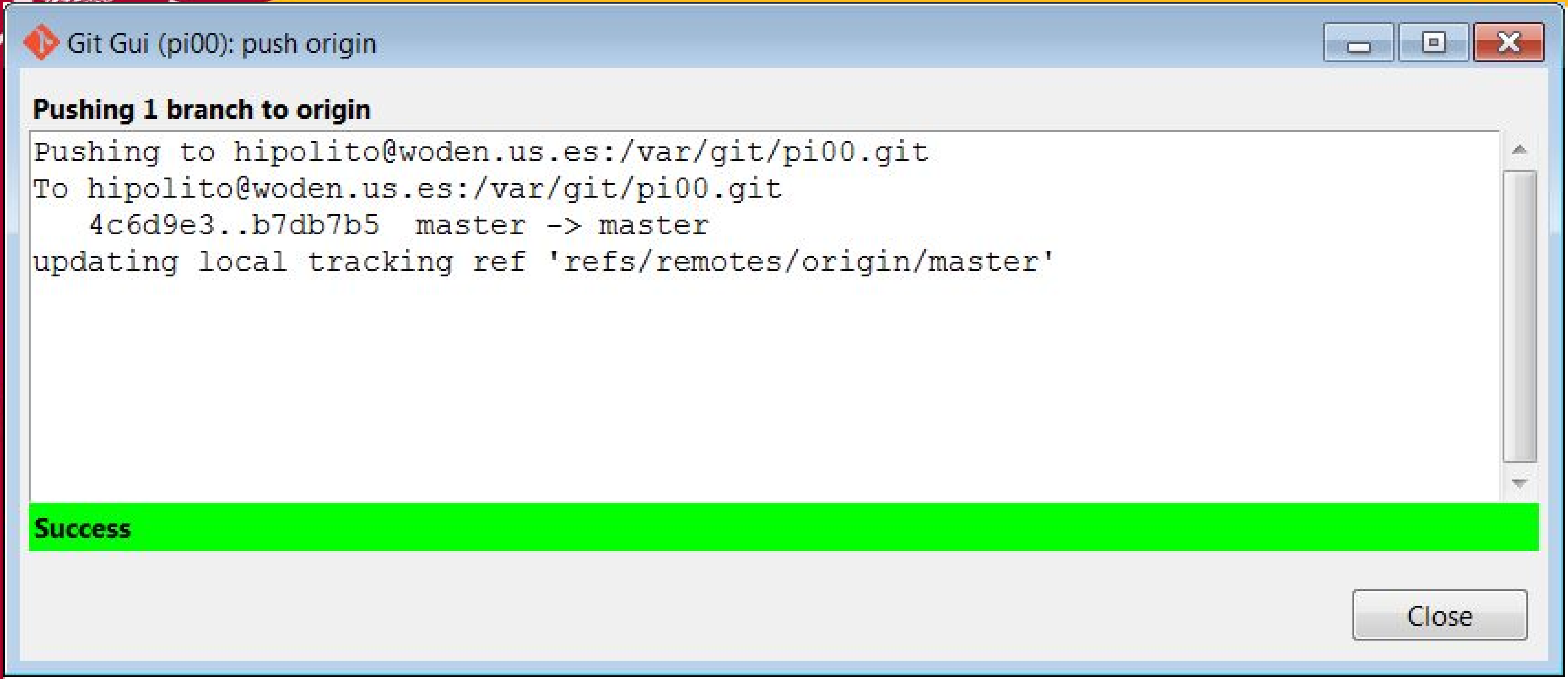
□ Force overwrite existing branch (may discard changes)

□ Use thin pack (for slow network connections)

□ Include tags

Por defecto ‘pusheamos’ a la rama `master`

Push



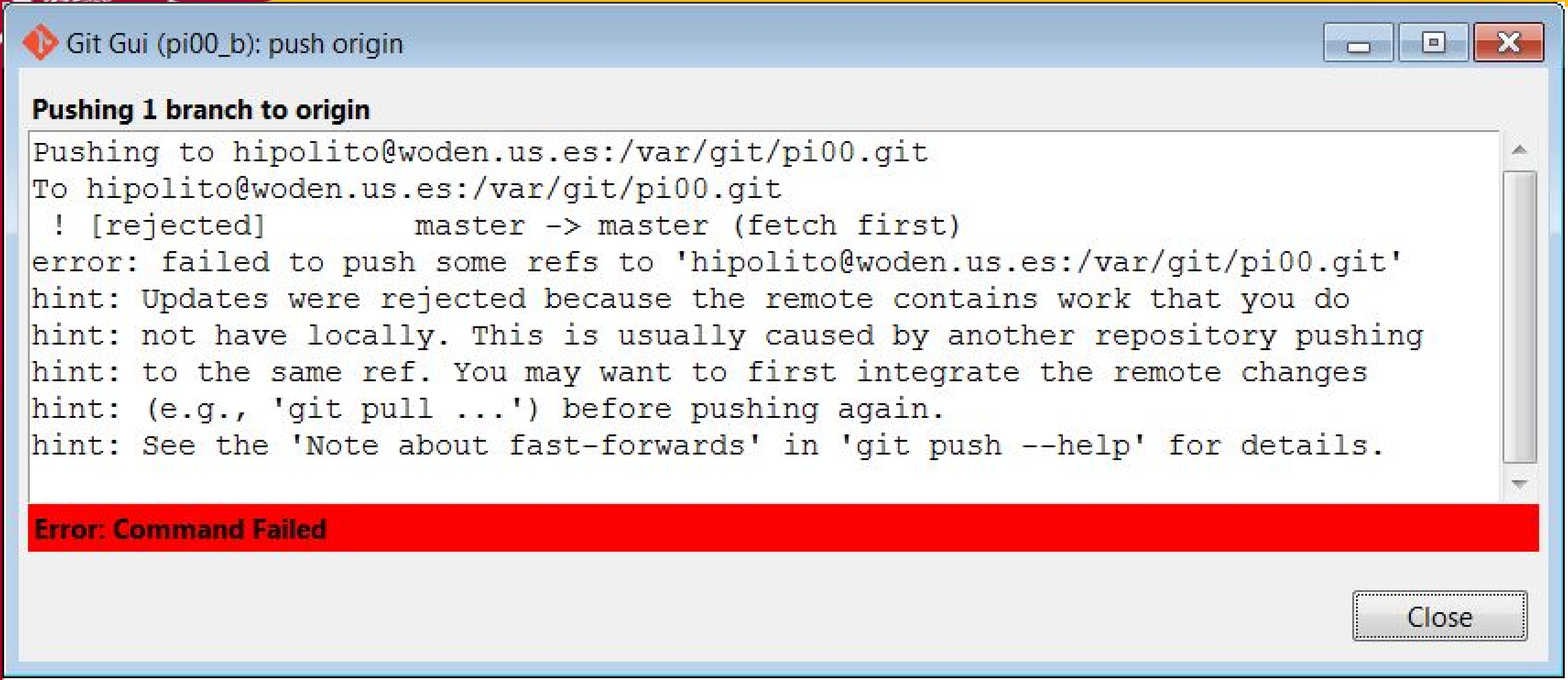
Pushing 1 branch to origin

```
Pushing to hipolito@woden.us.es:/var/git/pi00.git
To hipolito@woden.us.es:/var/git/pi00.git
   4c6d9e3..b7db7b5  master -> master
updating local tracking ref 'refs/remotes/origin/master'
```

Success

Normalmente todo va bien...

Push



Pushing 1 branch to origin

```
Pushing to hipolito@woden.us.es:/var/git/pi00.git
To hipolito@woden.us.es:/var/git/pi00.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'hipolito@woden.us.es:/var/git/pi00.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

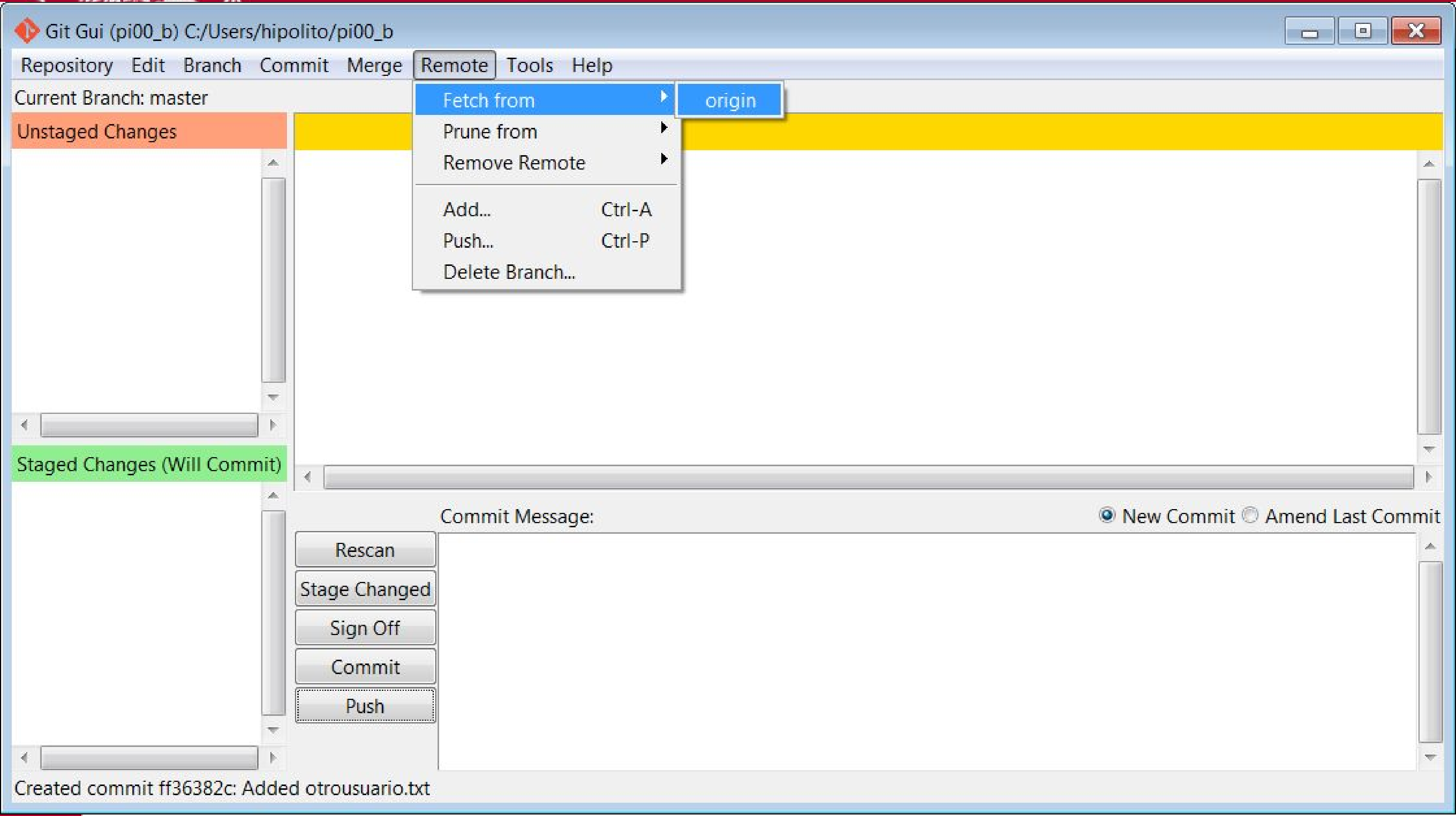
Error: Command Failed

Si nos da este error, es porque otro compañero ha hecho push antes que nosotros: debemos actualizar nuestro repo local (fetch + merge) antes de hacer push

Contenido

- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout

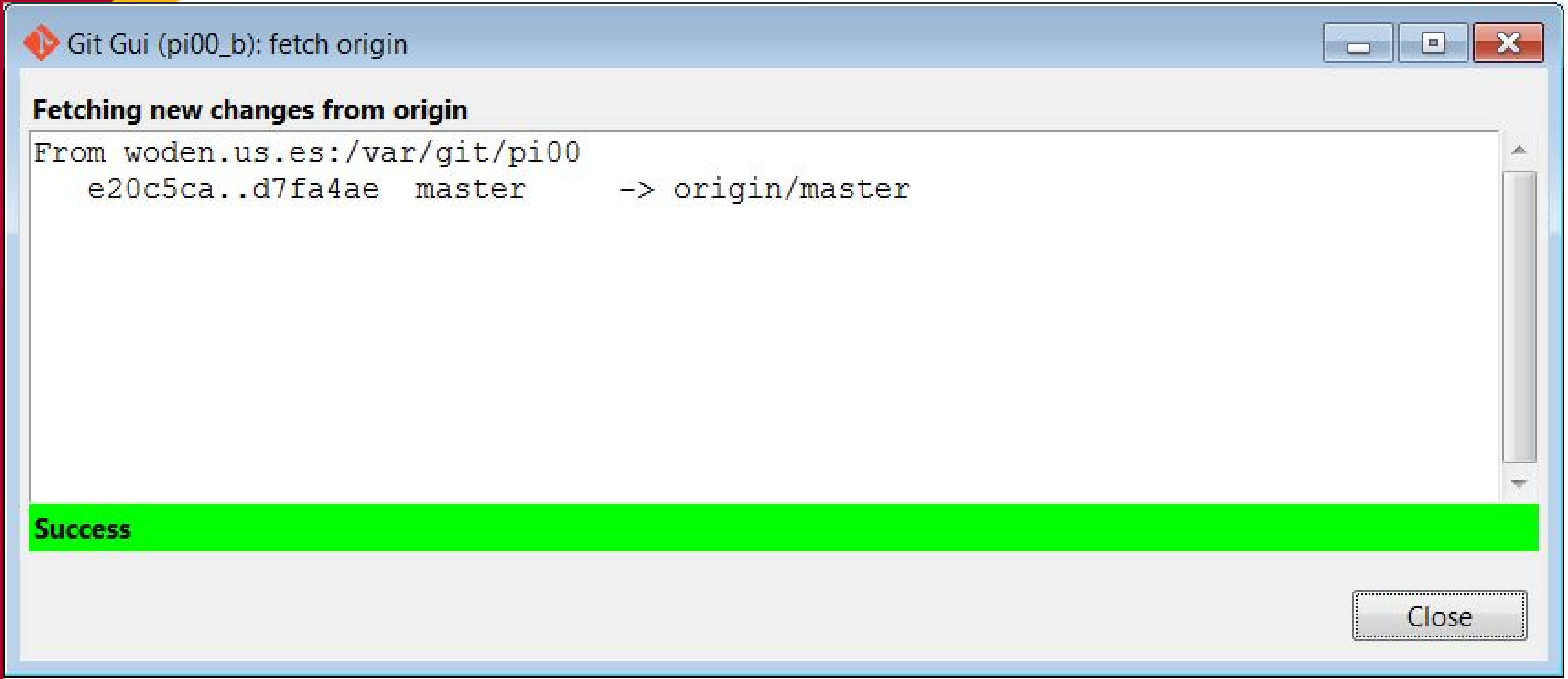
Pull: fetch + merge



Created commit ff36382c: Added otrousuario.txt

Fetch from origin descarga los cambios nuevos que haya en el repo remoto

Pull: fetch + merge



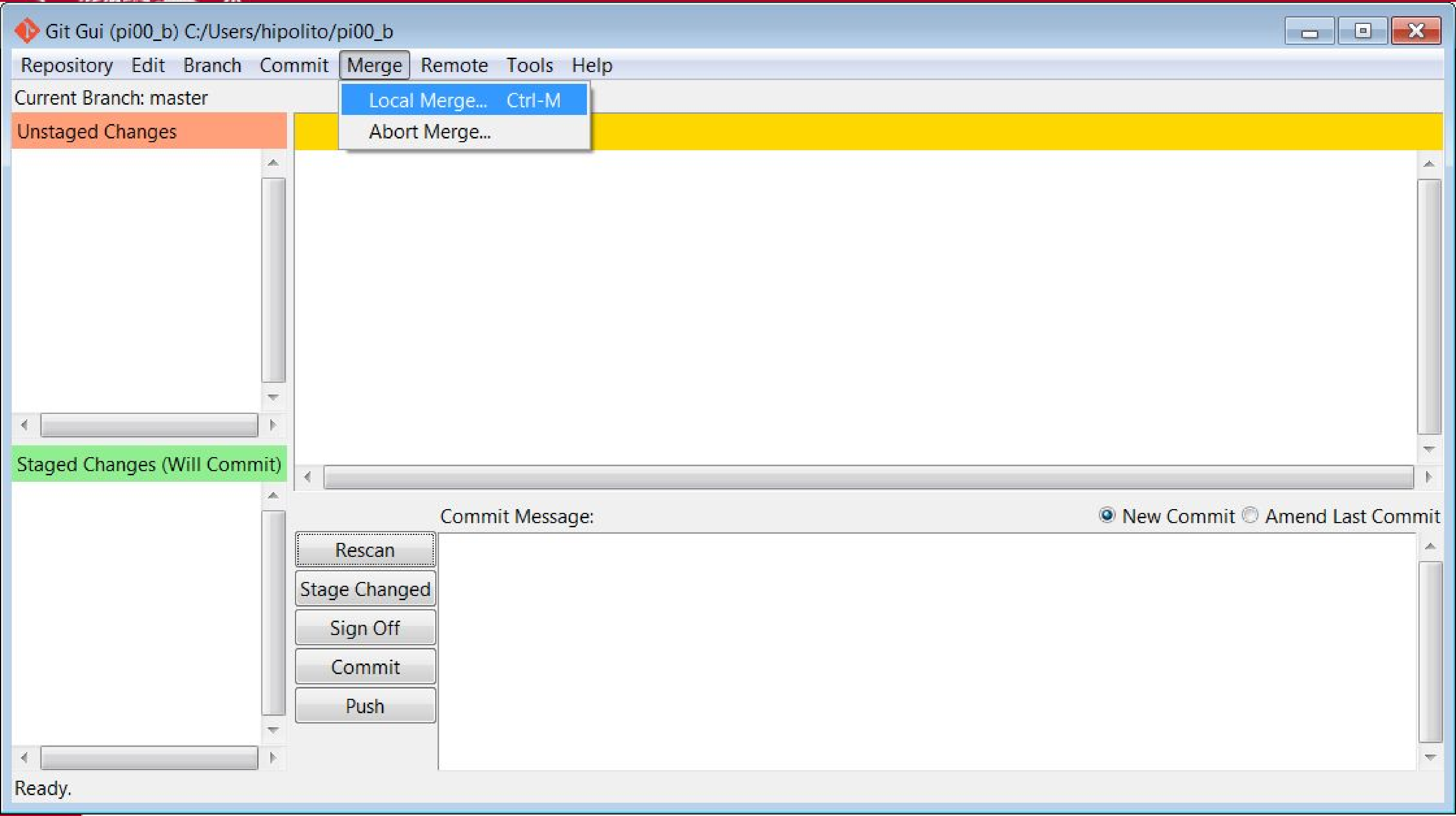
Fetching new changes from origin

```
From woden.us.es:/var/git/pi00
   e20c5ca..d7fa4ae  master     -> origin/master
```

Success

(Nota: si al hacer fetch este cuadro de diálogo no contiene texto, es que no hay cambios nuevos, por lo que no será necesario un merge)

Pull: fetch + merge



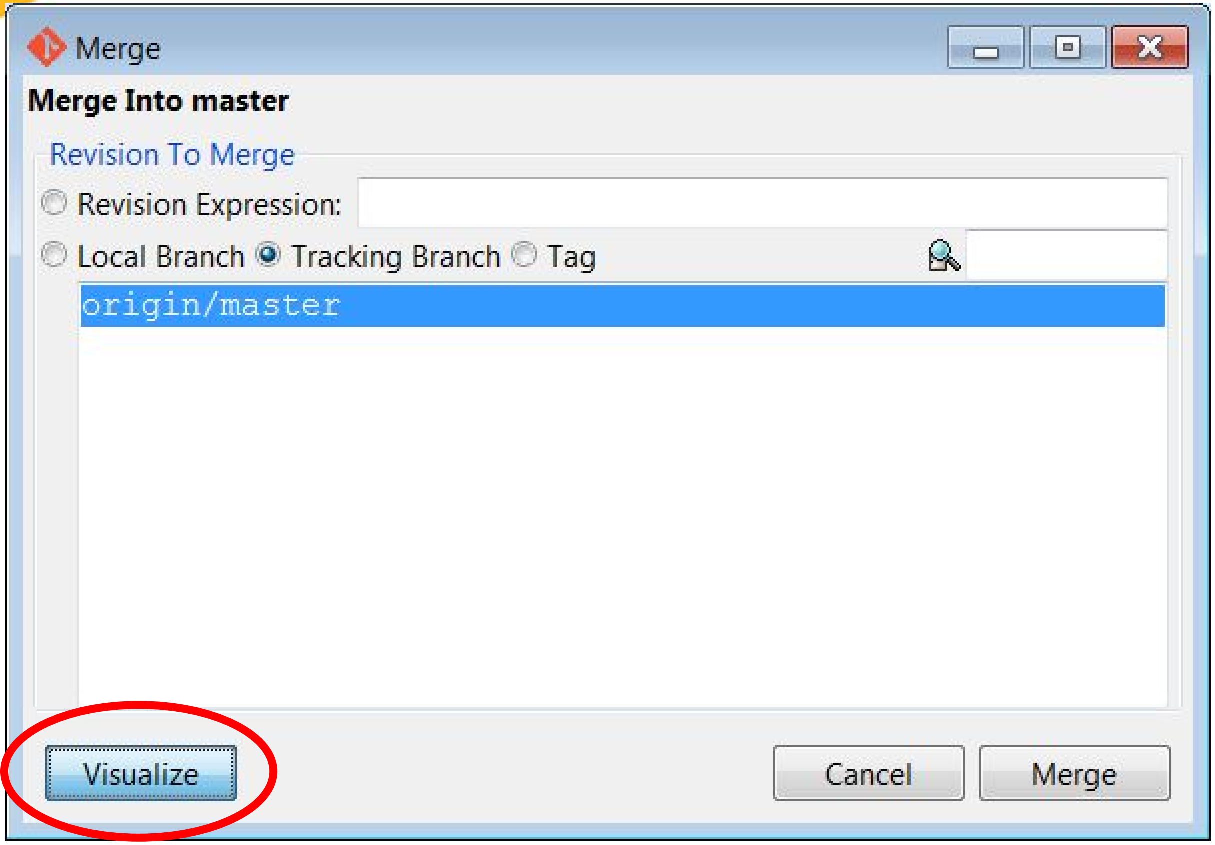
Current Branch: master

● New Commit ○ Amend Last Commit

Ready.

Local merge fusiona los cambios nuevos en el repo local y el workspace

Pull: fetch + merge



Merge

Merge Into master

Revision To Merge

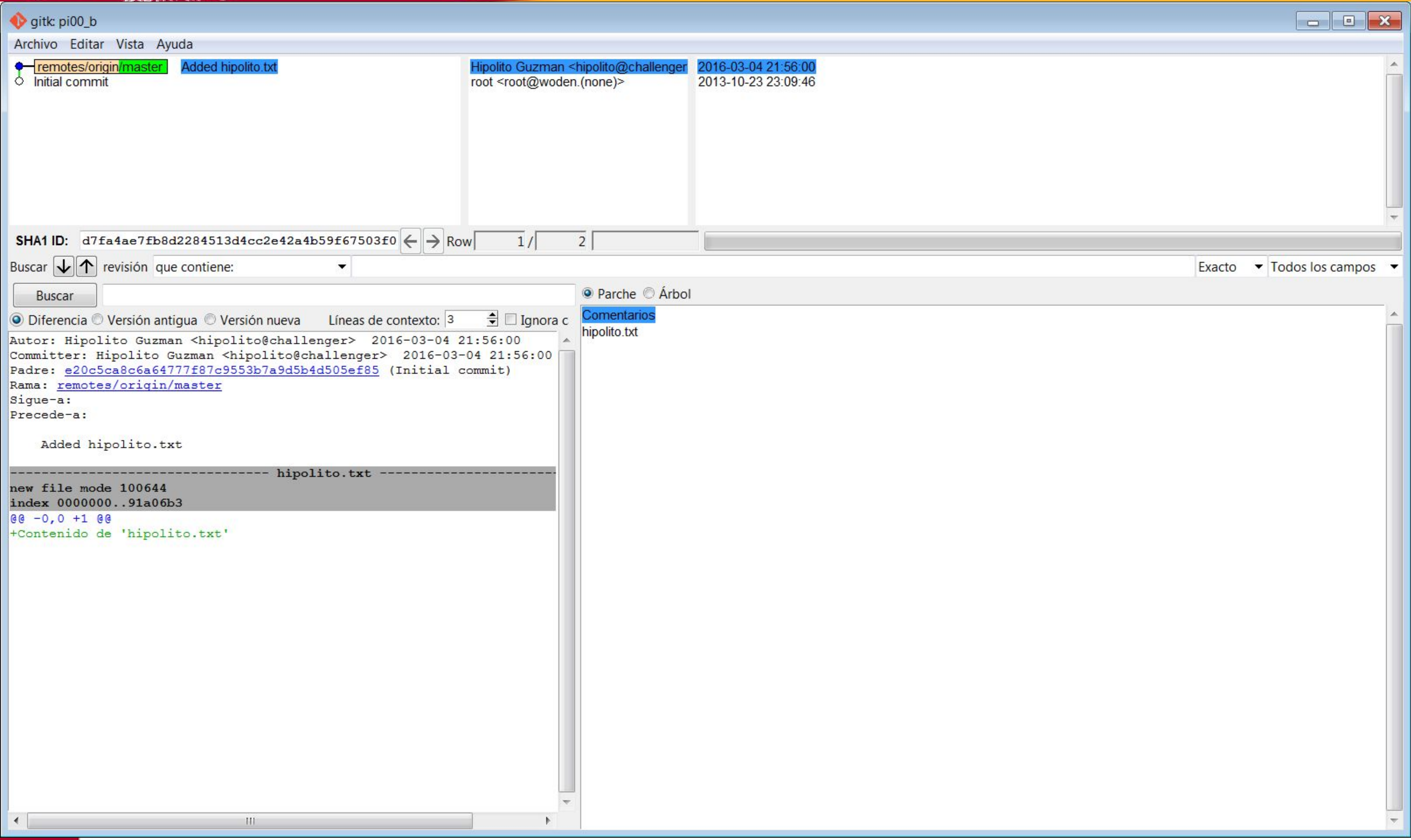
○ Revision Expression:

○ Local Branch ● Tracking Branch ○ Tag

origin/master

Podemos ver los cambios antes de fusionar

Pull: fetch + merge


gitk: pi00_b
Archivo Editar Vista Ayuda

remotes/origin/master Added hipolito.txt | Hipolito Guzman <hipolito@challenger | 2016-03-04 21:56:00
Initial commit | root <root@woden.(none)> | 2013-10-23 23:09:46

SHA1 ID: d7fa4ae7fb8d2284513d4cc2e42a4b59f67503f0 Row 1 / 2

Buscar ↓ ↑ revisión que contiene:
Exacto
Todos los campos
Buscar
● Parche ○ Árbol
● Diferencia ○ Versión antigua ○ Versión nueva Líneas de contexto: 3 □ Ignora c

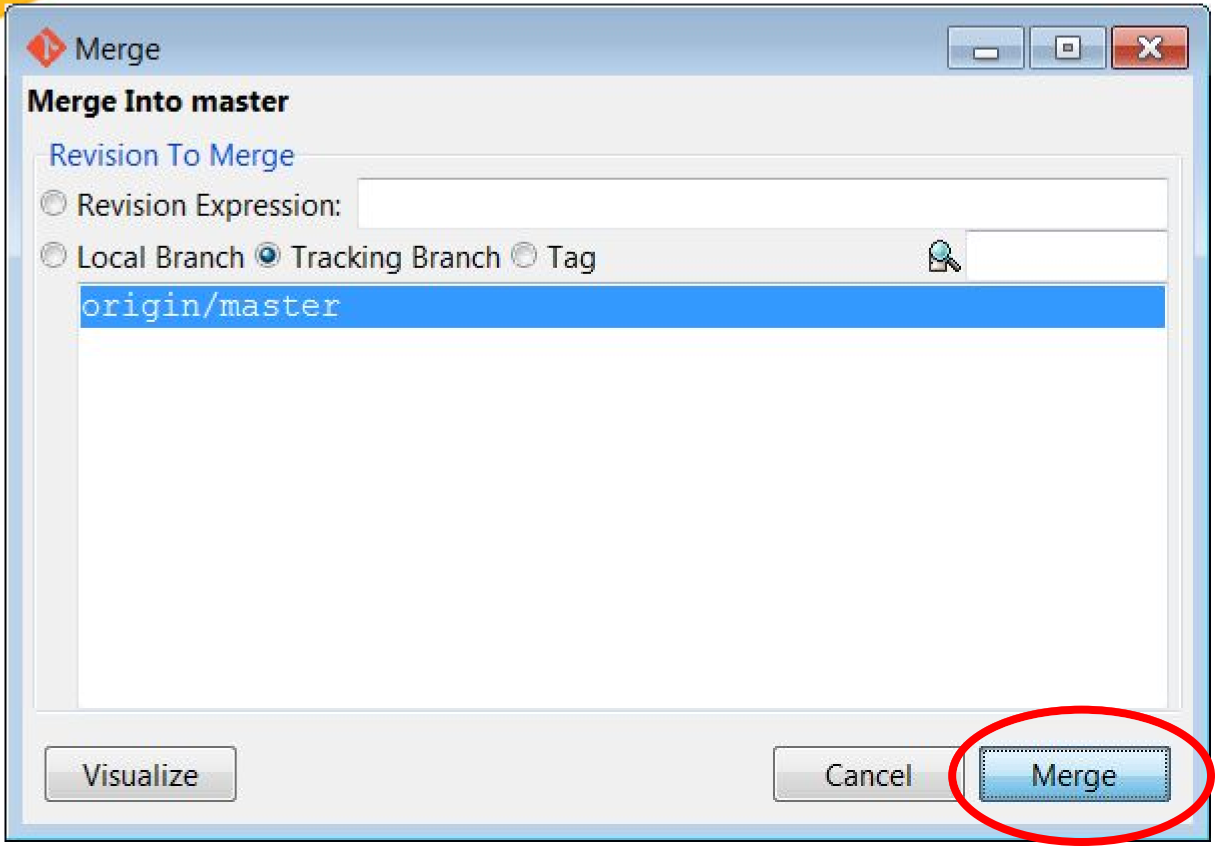
```
Autor: Hipolito Guzman <hipolito@challenger>  2016-03-04 21:56:00
Committer: Hipolito Guzman <hipolito@challenger>  2016-03-04 21:56:00
Padre: e20c5ca8c6a64777f87c9553b7a9d5b4d505ef85 (Initial commit)
Rama: remotes/origin/master
Sigue-a:
Precede-a:

    Added hipolito.txt

--------------------------------- hipolito.txt ----------------------
new file mode 100644
index 0000000..91a06b3
@@ -0,0 +1 @@
+Contenido de 'hipolito.txt'
```

Comentarios
hipolito.txt

Pull: fetch + merge



Merge Into master

Revision To Merge

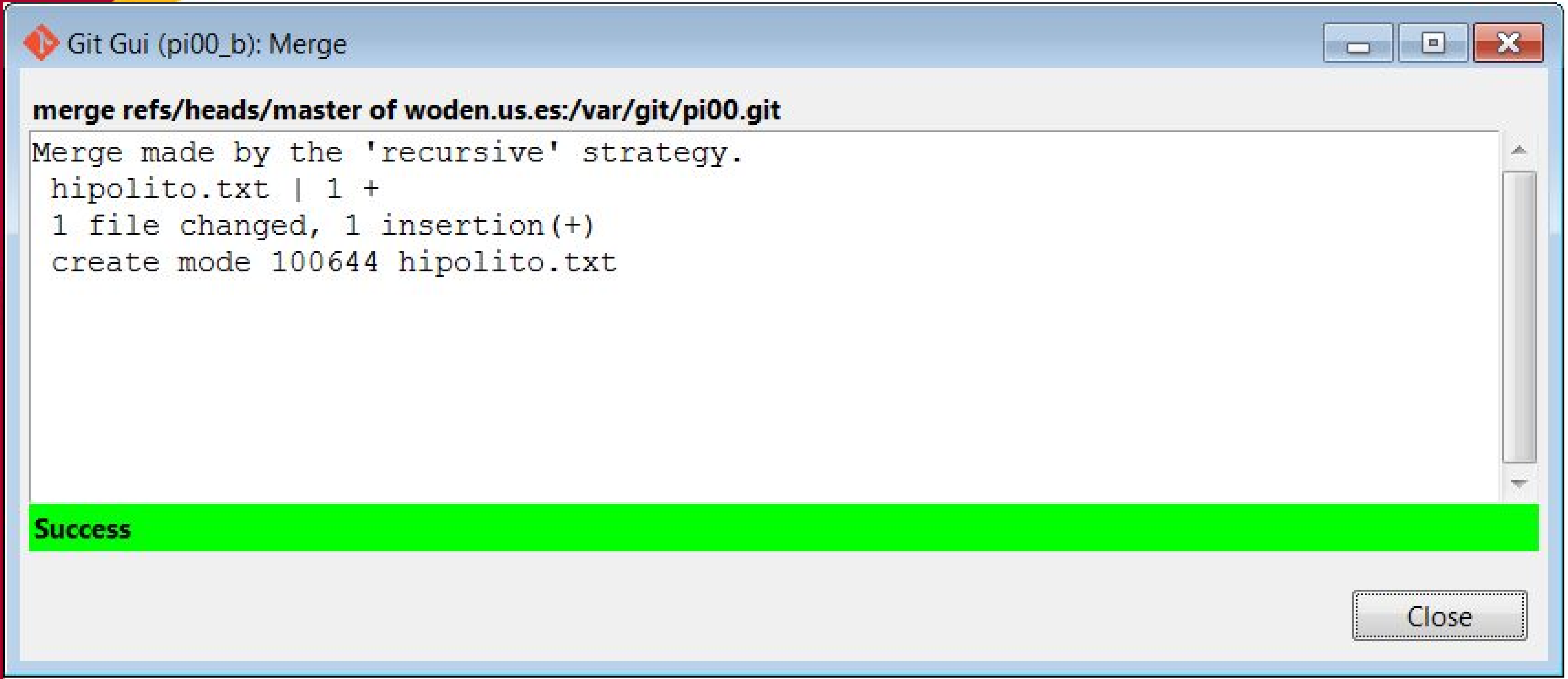
○ Revision Expression:

○ Local Branch ● Tracking Branch ○ Tag

origin/master

Pulsamos 'Merge'

Pull: fetch + merge



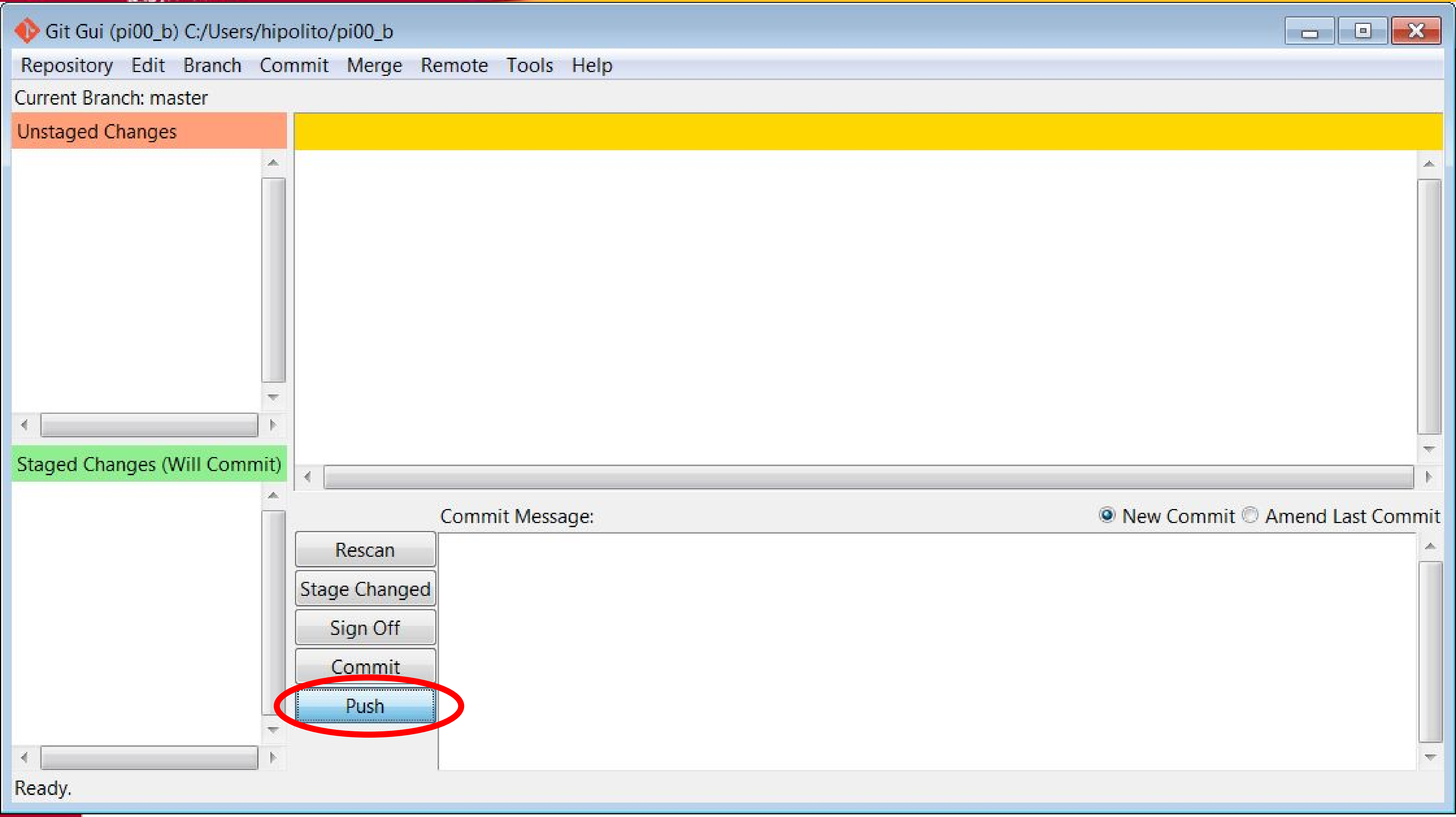
merge refs/heads/master of woden.us.es:/var/git/pi00.git

```
Merge made by the 'recursive' strategy.
 hipolito.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 hipolito.txt
```

Success

Merge completado con éxito

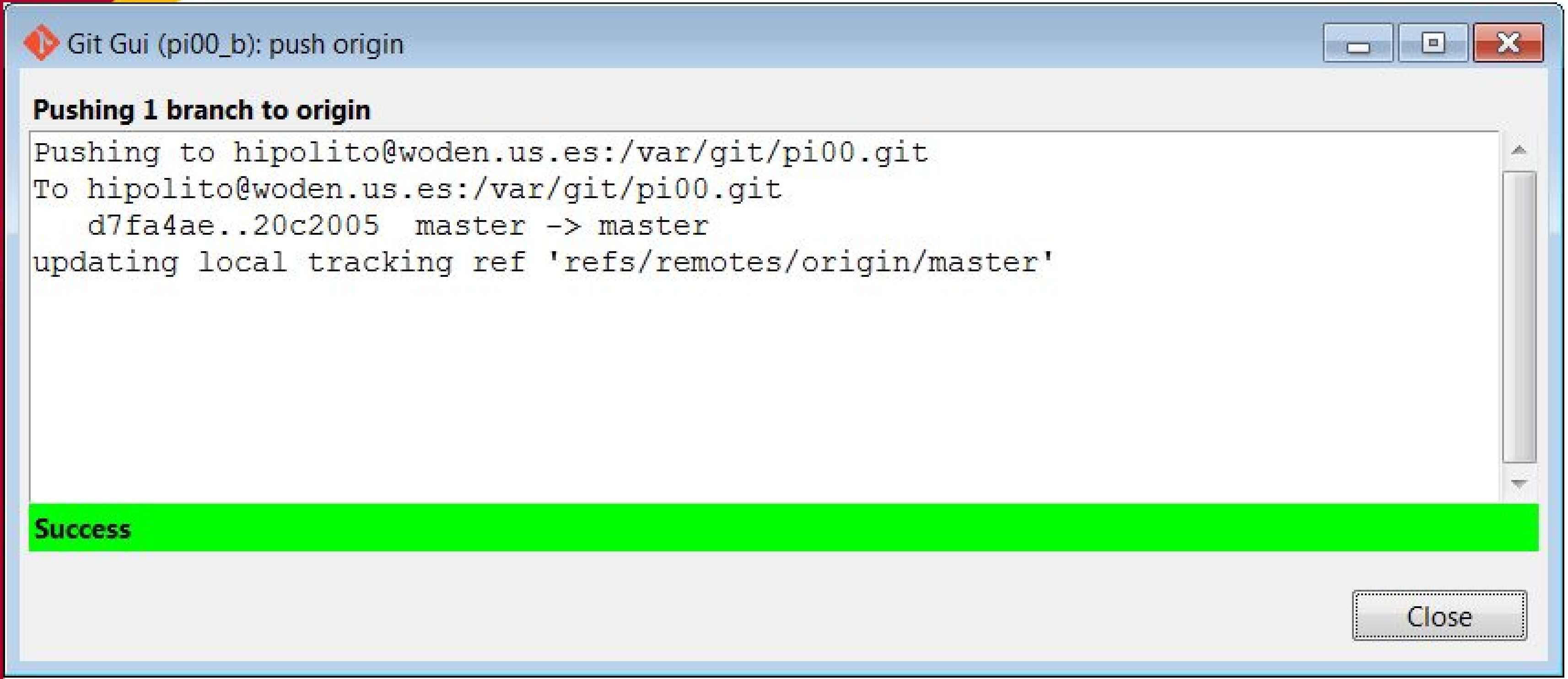
Push (2)



Git Gui (pi00_b) C:/Users/hipolito/pi00_b

Tras el merge, ya podemos ‘pushear’ nuestros cambios

Push (2)



Pushing 1 branch to origin

```
Pushing to hipolito@woden.us.es:/var/git/pi00.git
To hipolito@woden.us.es:/var/git/pi00.git
   d7fa4ae..20c2005  master -> master
updating local tracking ref 'refs/remotes/origin/master'
```

Success

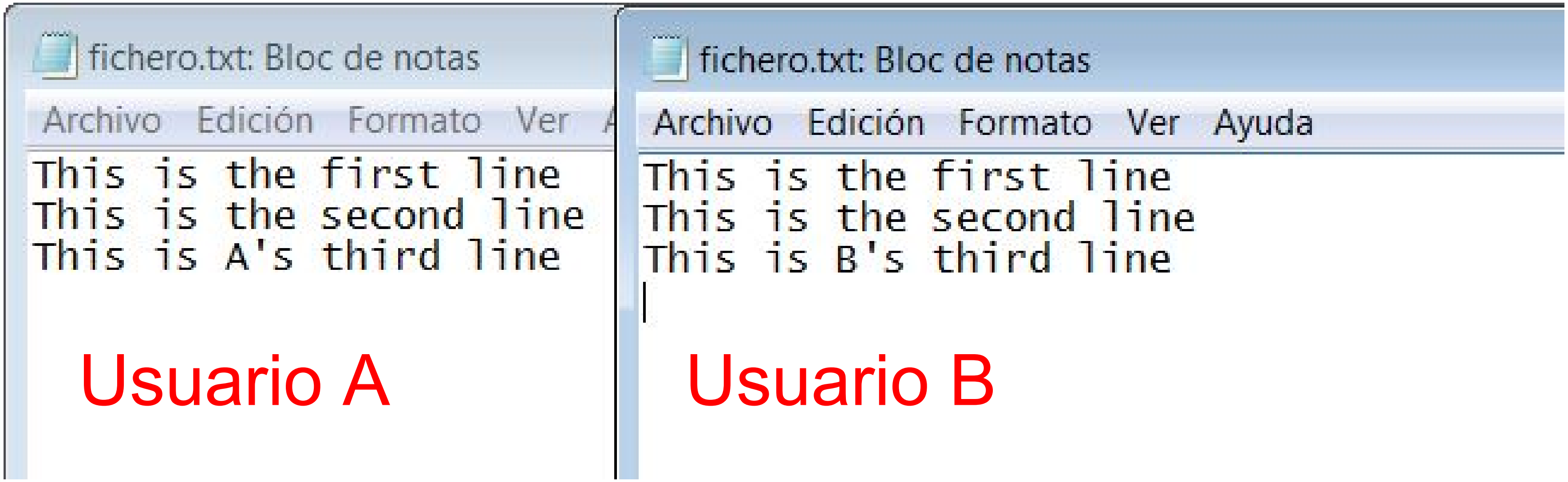
Y esta vez, el push termina correctamente

Contenido

- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout

Resolución de Conflictos

¿Qué ocurre si dos personas modifican la misma zona del mismo fichero?


fichero.txt: Bloc de notas
Archivo Edición Formato Ver A
fichero.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda

```
This is the first line
This is the second line
This is A's third line
```

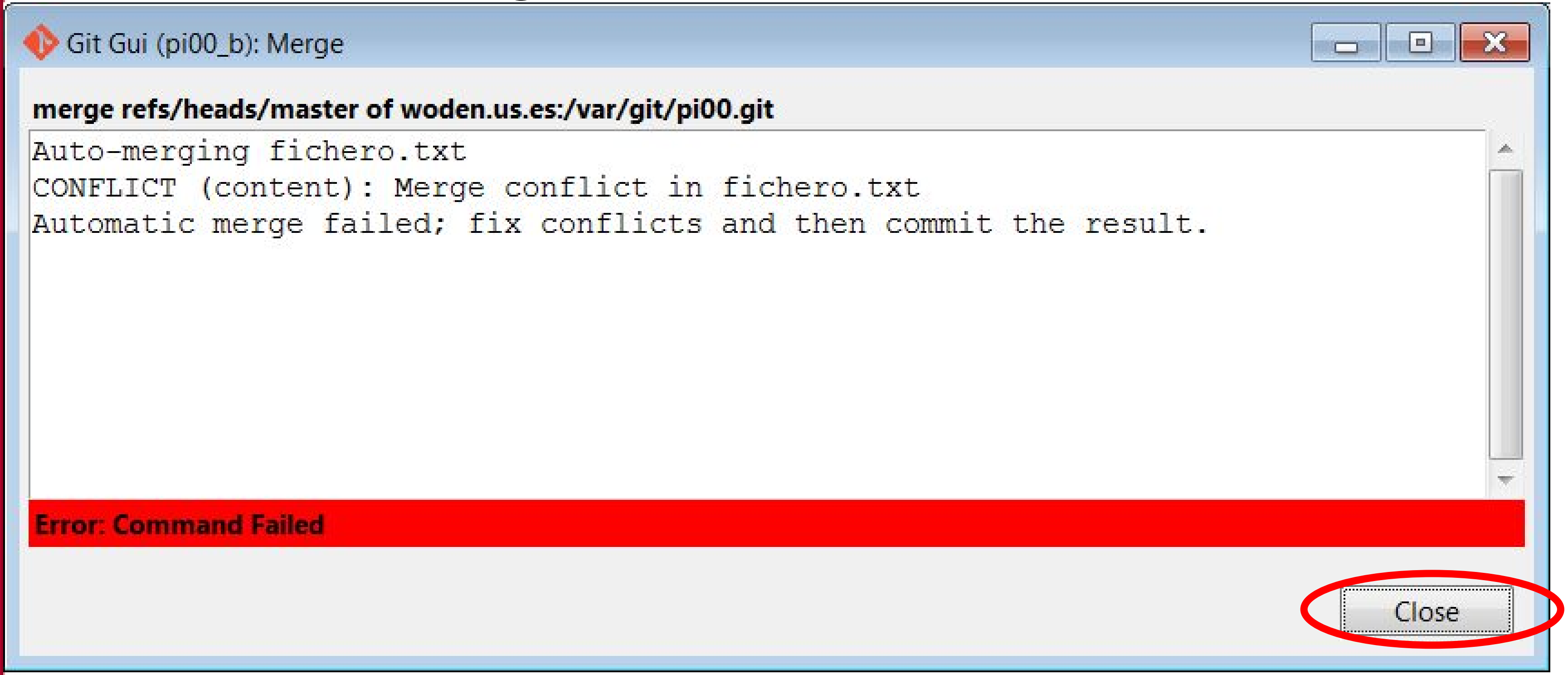
```
This is the first line
This is the second line
This is B's third line
```

Usuario A

Usuario B

A ‘pushea’ primero y no ve nada raro

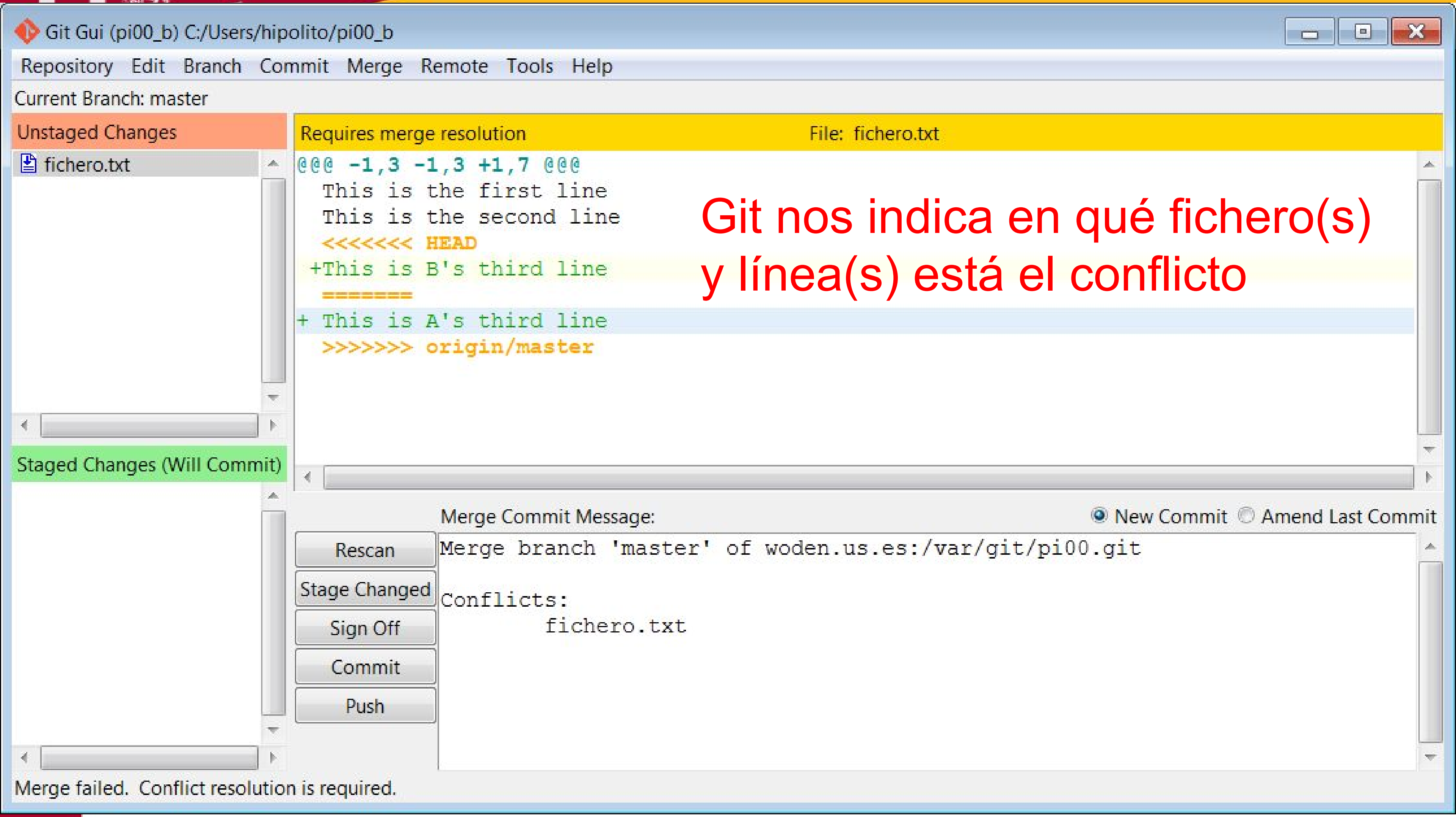
B tiene que hacer fetch + merge, pero al hacer merge:



merge refs/heads/master of woden.us.es:/var/git/pi00.git

```
Auto-merging fichero.txt
CONFLICT (content): Merge conflict in fichero.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Error: Command Failed



Current Branch: master

Unstaged Changes: fichero.txt

Requires merge resolution File: fichero.txt

```
@@@ -1,3 -1,3 +1,7 @@@
  This is the first line
  This is the second line
  <<<<<<< HEAD
 +This is B's third line
  =======
+ This is A's third line
  >>>>>>> origin/master
```

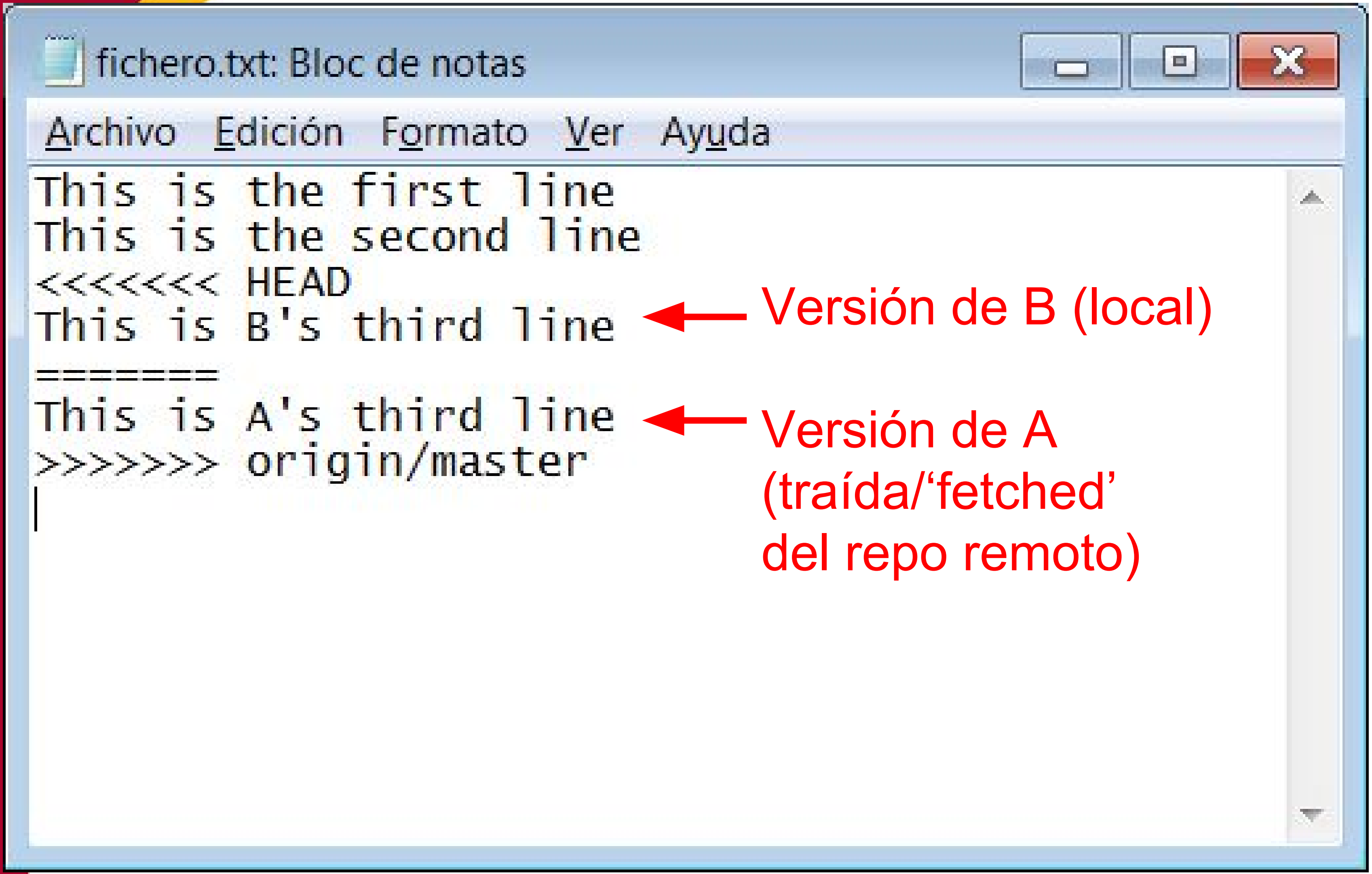
Merge Commit Message: ● New Commit ○ Amend Last Commit

```
Merge branch 'master' of woden.us.es:/var/git/pi00.git

Conflicts:
	fichero.txt
```

Merge failed. Conflict resolution is required.

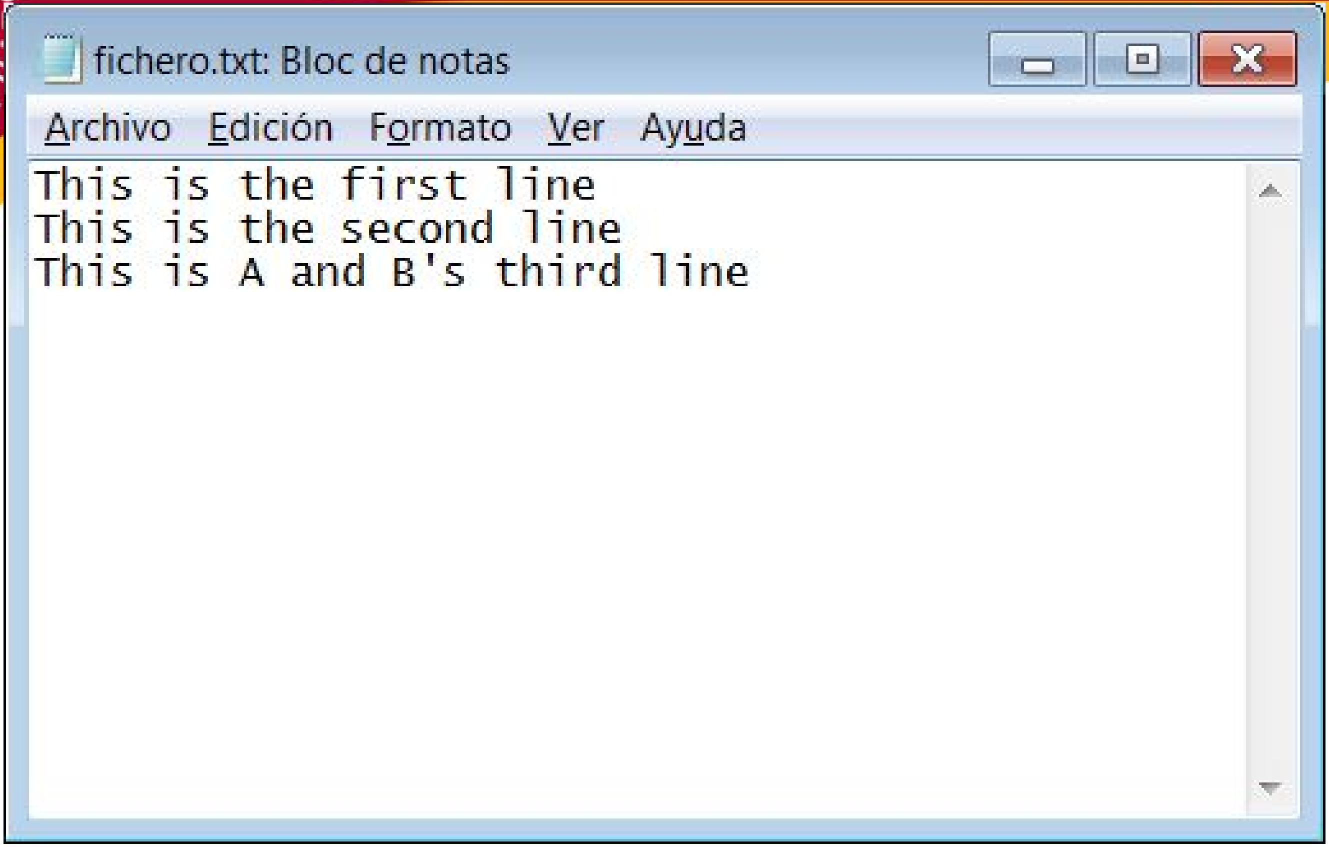
Git nos indica en qué fichero(s) y línea(s) está el conflicto



```
This is the first line
This is the second line
<<<<<<< HEAD
This is B's third line
=======
This is A's third line
>>>>>>> origin/master
```

Versión de B (local)

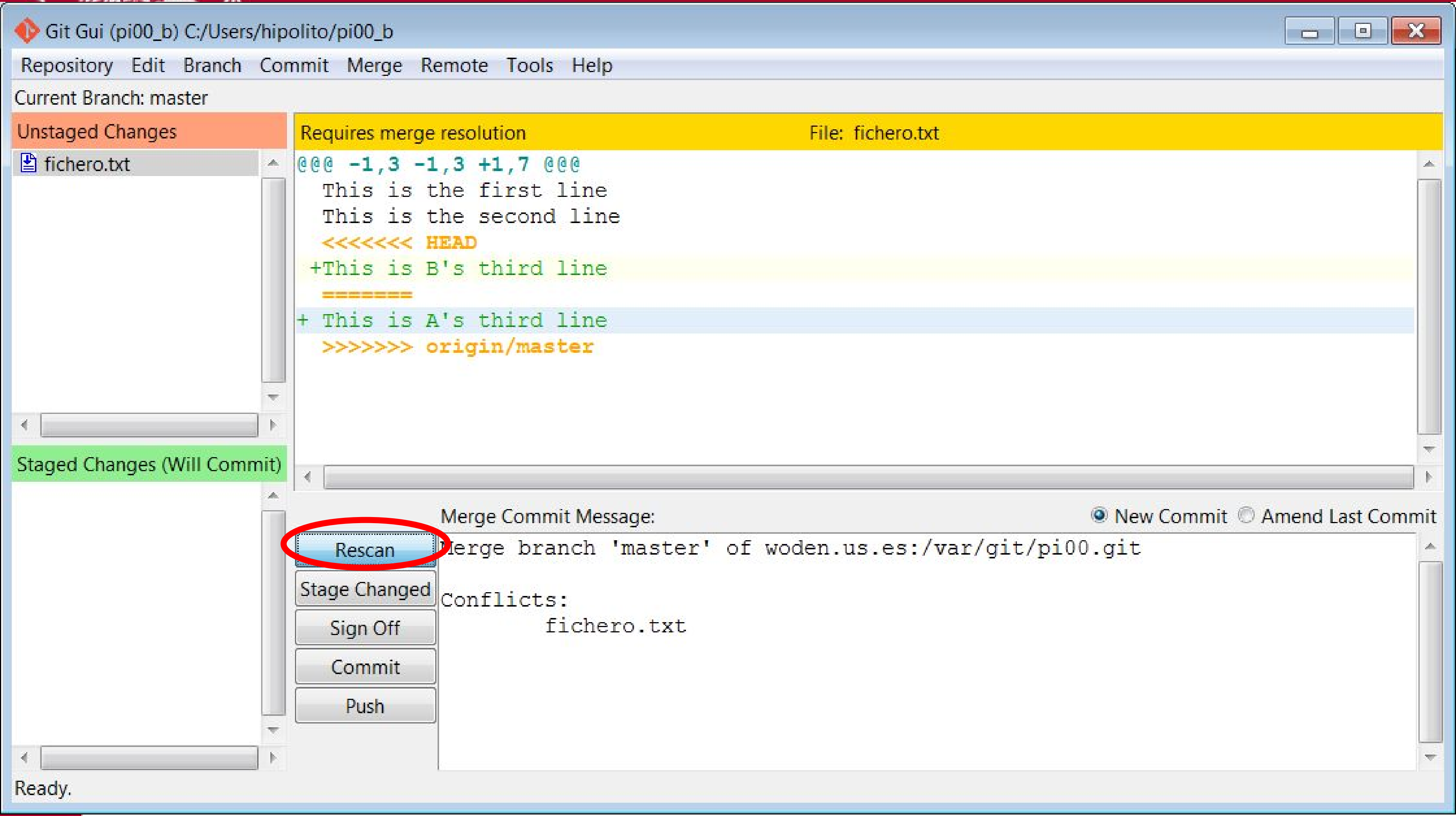
Versión de A (traída/‘fetched’ del repo remoto)



```
This is the first line
This is the second line
This is A and B's third line
```

Solucionamos el conflicto:

- Decidiendo qué versión es la correcta
- Eliminando los caracteres de control (<<, >>, ==)



Current Branch: master

Requires merge resolution File: fichero.txt

Unstaged Changes: fichero.txt

```
@@@ -1,3 -1,3 +1,7 @@@
  This is the first line
  This is the second line
  <<<<<<< HEAD
 +This is B's third line
  =======
+ This is A's third line
  >>>>>>> origin/master
```

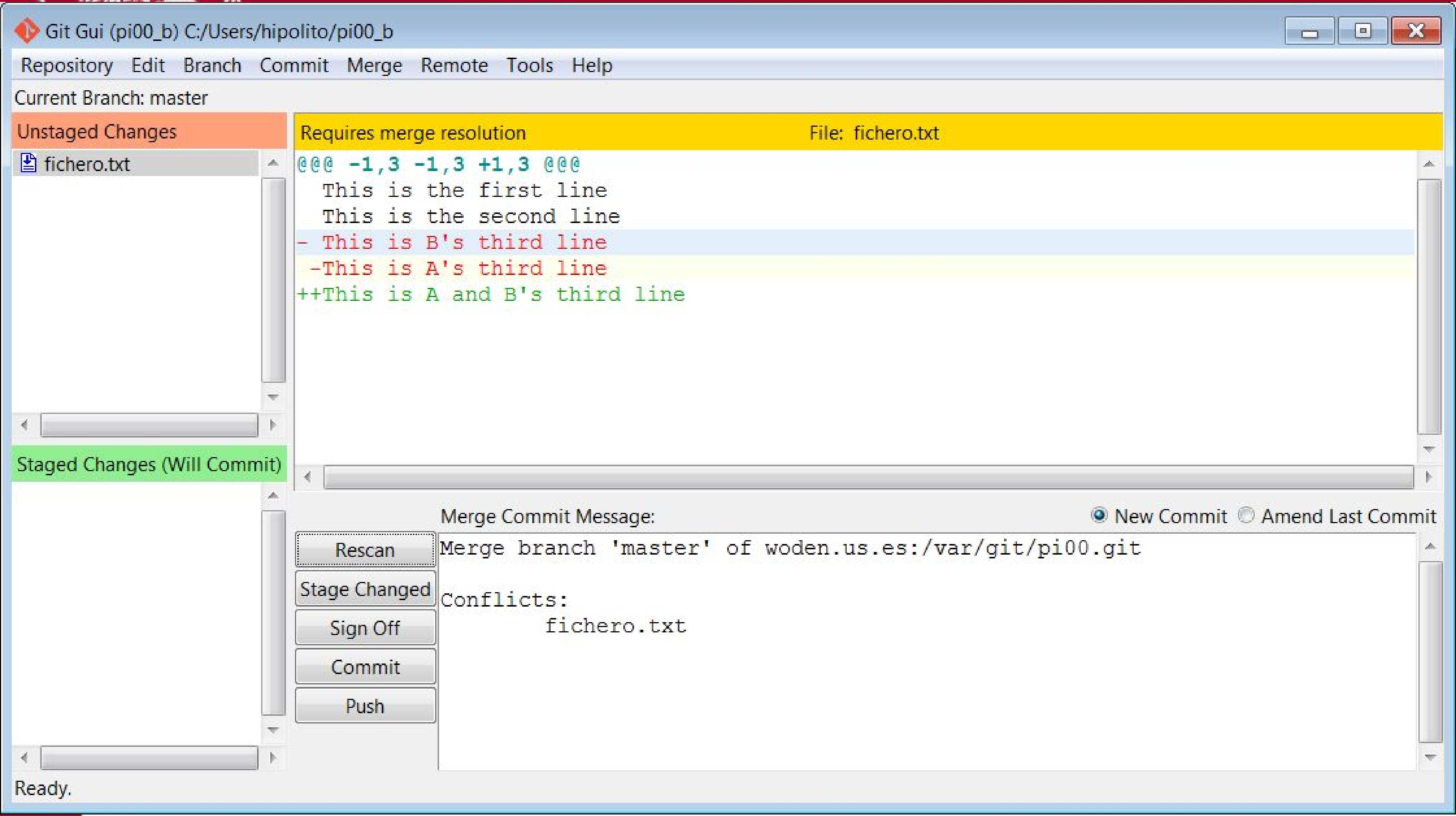
Merge Commit Message: ● New Commit ○ Amend Last Commit

```
Merge branch 'master' of woden.us.es:/var/git/pi00.git

Conflicts:
	fichero.txt
```

Ready.

Pulsamos ‘Rescan’ para buscar cambios en el workspace



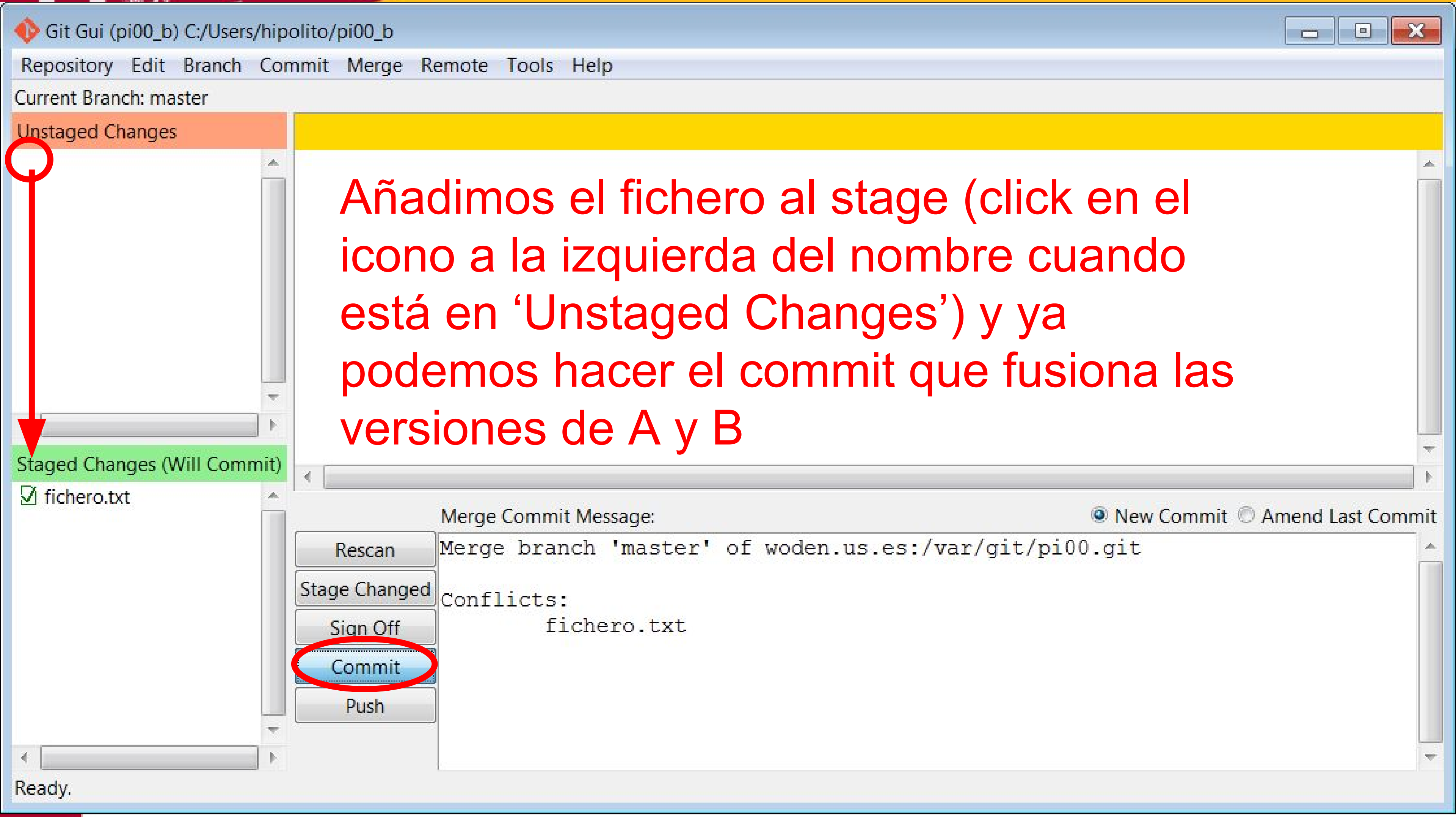
Git Gui (pi00_b) C:/Users/hipolito/pi00_b
Repository Edit Branch Commit Merge Remote Tools Help
Current Branch: master
Unstaged Changes
fichero.txt
Requires merge resolution
File: fichero.txt
Staged Changes (Will Commit)
Merge Commit Message:
● New Commit ○ Amend Last Commit
Rescan
Stage Changed
Sign Off
Commit
Push
Ready.

```
@@@ -1,3 -1,3 +1,3 @@@
  This is the first line
  This is the second line
- This is B's third line
 -This is A's third line
++This is A and B's third line
```

```
Merge branch 'master' of woden.us.es:/var/git/pi00.git

Conflicts:
        fichero.txt
```

Vemos la versión corregida del fichero

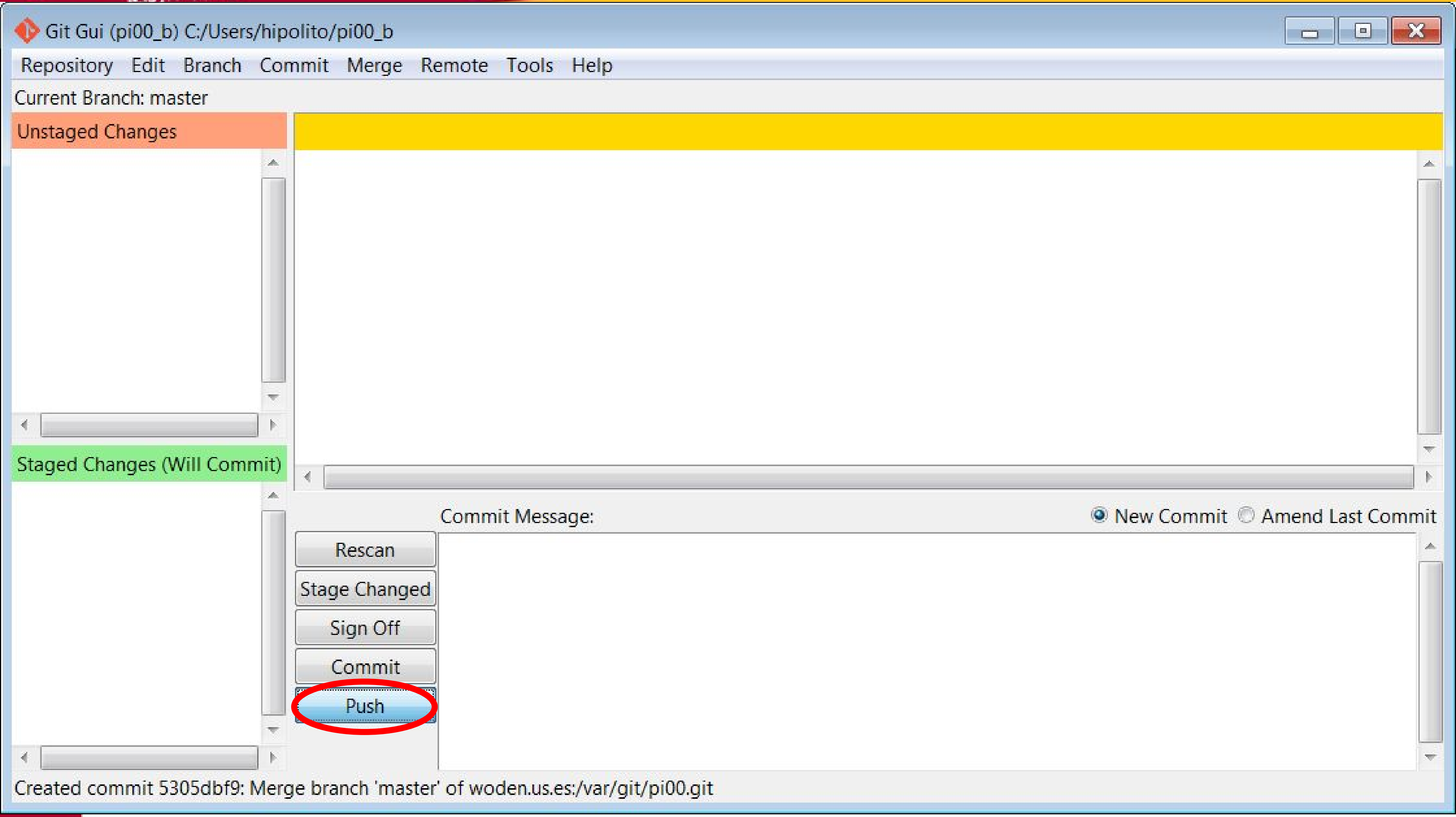


Git Gui (pi00_b) C:/Users/hipolito/pi00_b
Repository Edit Branch Commit Merge Remote Tools Help
Current Branch: master
Unstaged Changes
Staged Changes (Will Commit)
☑ fichero.txt
Rescan
Stage Changed
Sign Off
Commit
Push
Merge Commit Message:
● New Commit ○ Amend Last Commit
Ready.

Añadimos el fichero al stage (click en el icono a la izquierda del nombre cuando está en ‘Unstaged Changes’) y ya podemos hacer el commit que fusiona las versiones de A y B

```
Merge branch 'master' of woden.us.es:/var/git/pi00.git

Conflicts:
        fichero.txt
```



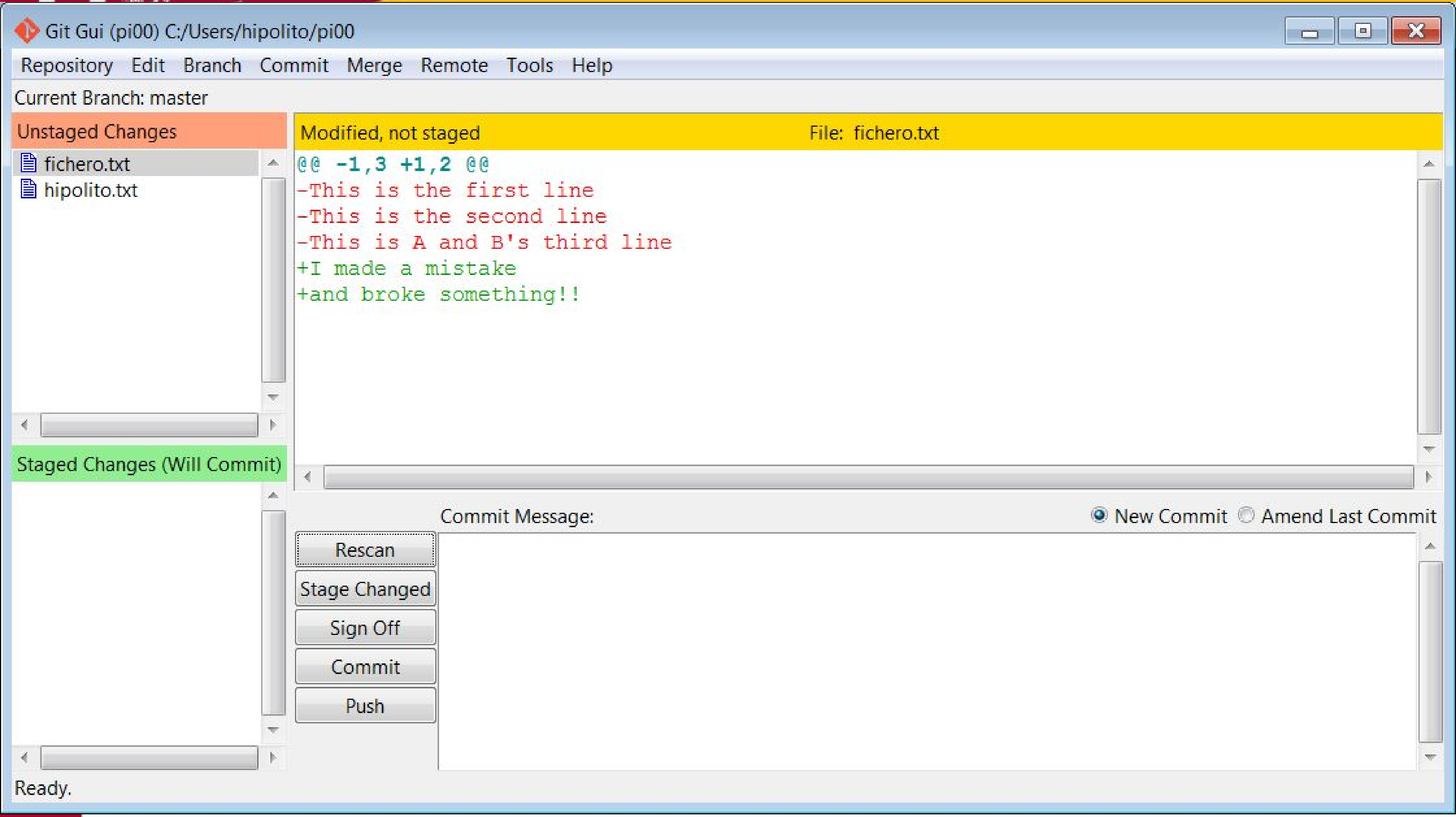
Created commit 5305dbf9: Merge branch 'master' of woden.us.es:/var/git/pi00.git

Y ya podemos empujar este nuevo commit al repositorio remoto

Contenido

- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout

Revertir errores usando checkout



Git Gui (pi00) C:/Users/hipolito/pi00

Current Branch: master

Unstaged Changes

- fichero.txt
- hipolito.txt

Modified, not staged File: fichero.txt

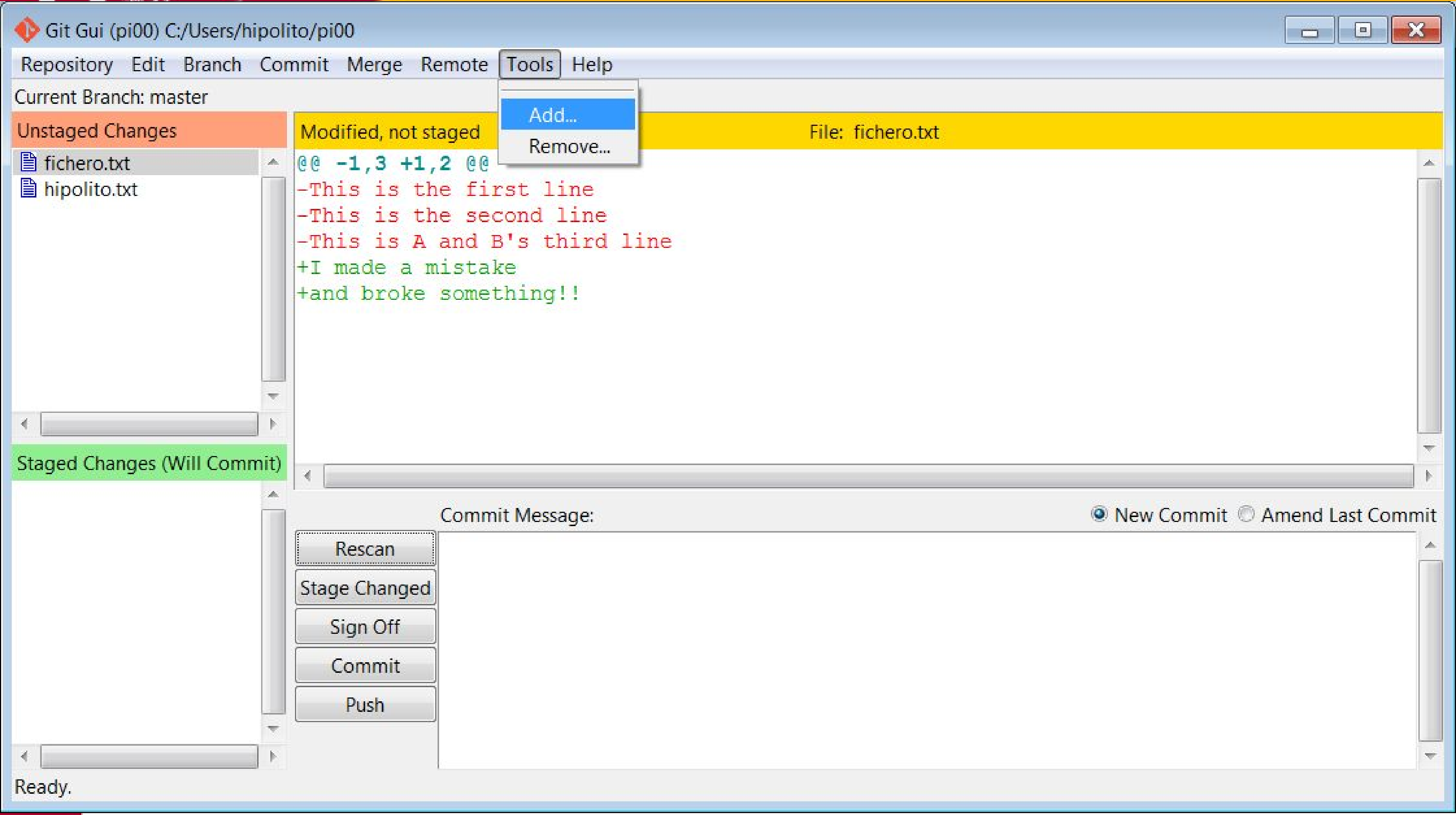
```
@@ -1,3 +1,2 @@
-This is the first line
-This is the second line
-This is A and B's third line
+I made a mistake
+and broke something!!
```

Staged Changes (Will Commit)

Commit Message: ● New Commit ○ Amend Last Commit

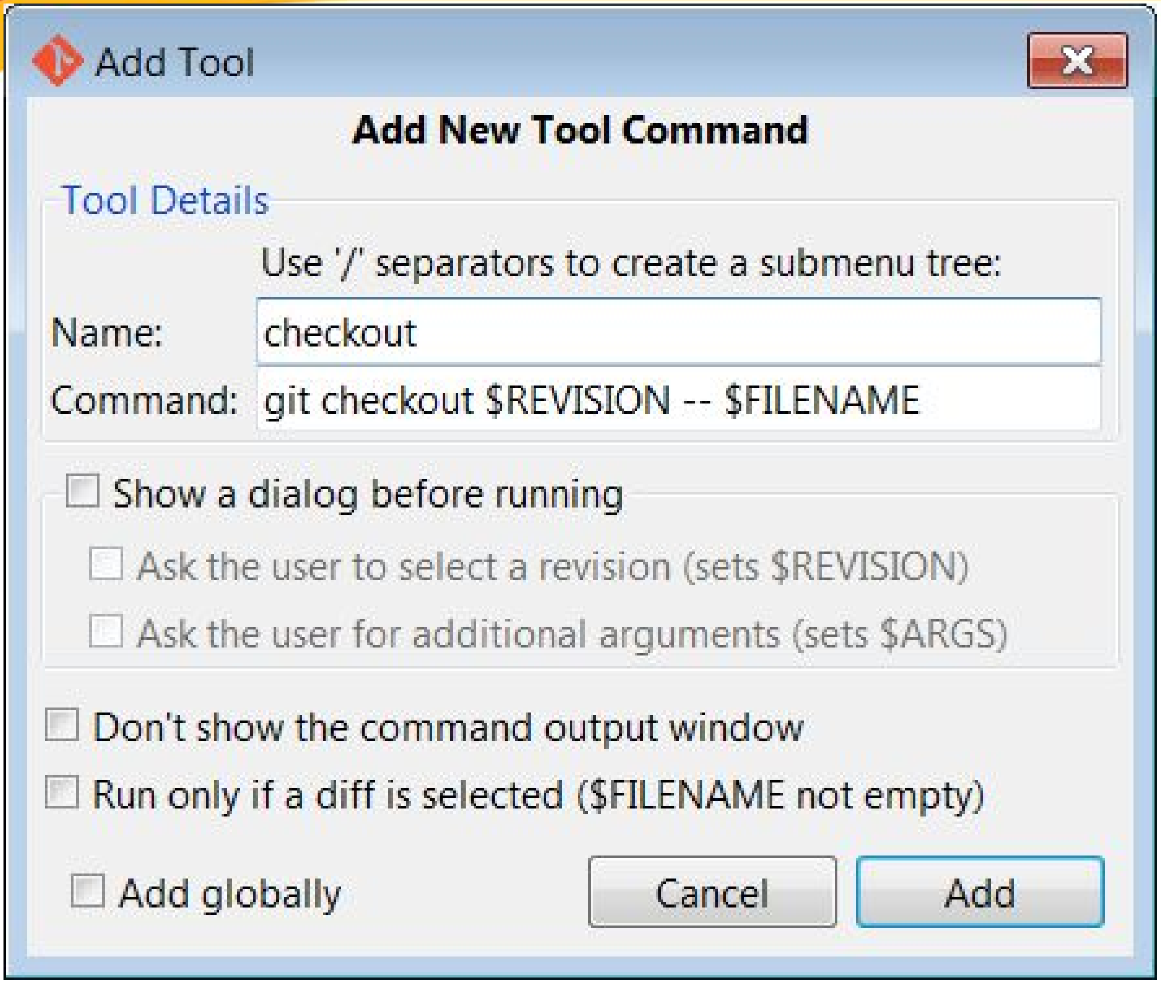
Ready.

Revertir errores usando checkout



```
@@ -1,3 +1,2 @@
-This is the first line
-This is the second line
-This is A and B's third line
+I made a mistake
+and broke something!!
```

Revertir errores usando checkout



Add New Tool Command

Tool Details

Use '/' separators to create a submenu tree:

Name: checkout

Command: git checkout $REVISION -- $FILENAME

□ Show a dialog before running

□ Ask the user to select a revision (sets $REVISION)

□ Ask the user for additional arguments (sets $ARGS)

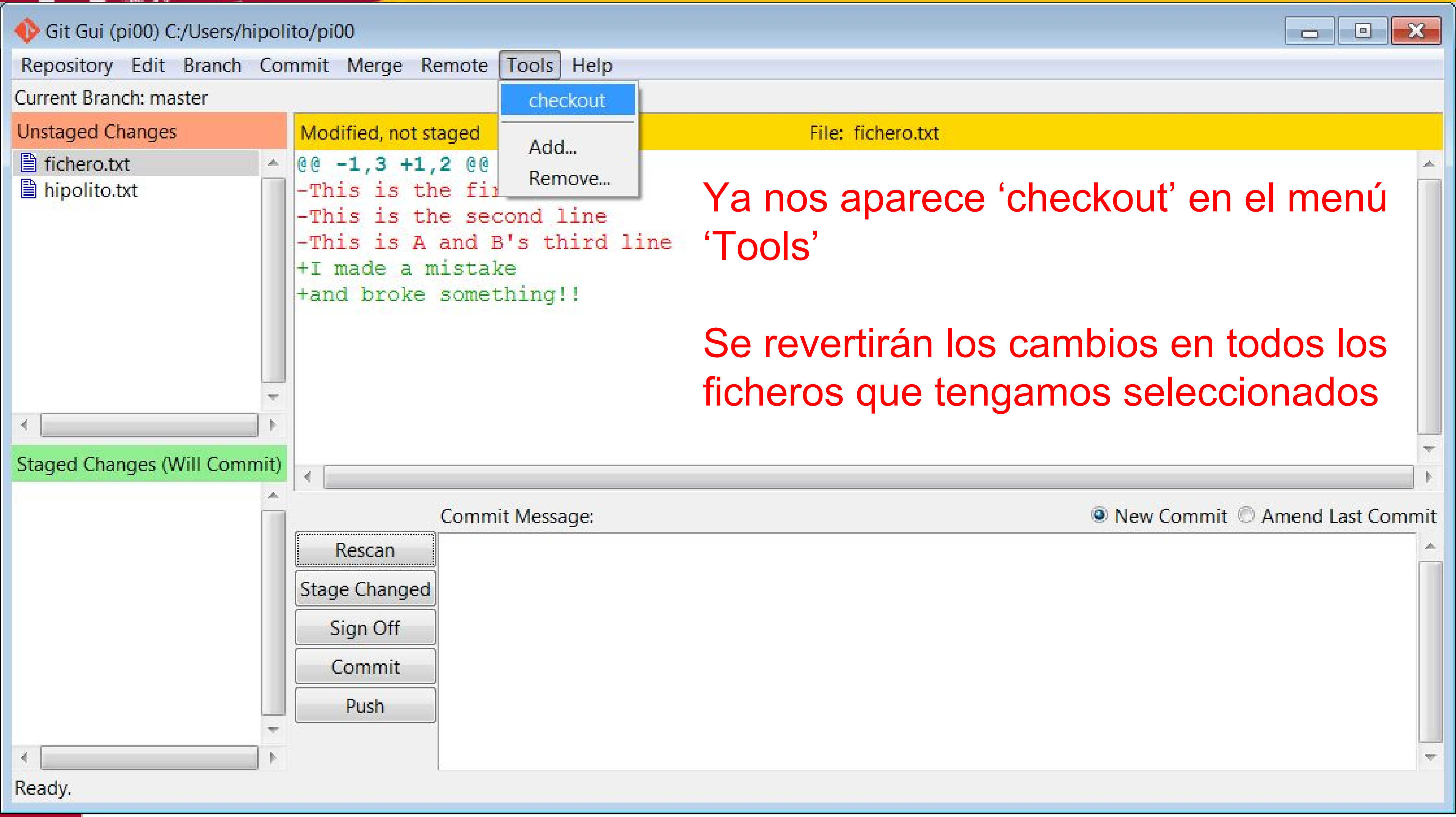
□ Don't show the command output window

□ Run only if a diff is selected ($FILENAME not empty)

□ Add globally

(Sólo tenemos que añadir el comando si no lo tenemos de antes)

Revertir errores usando checkout

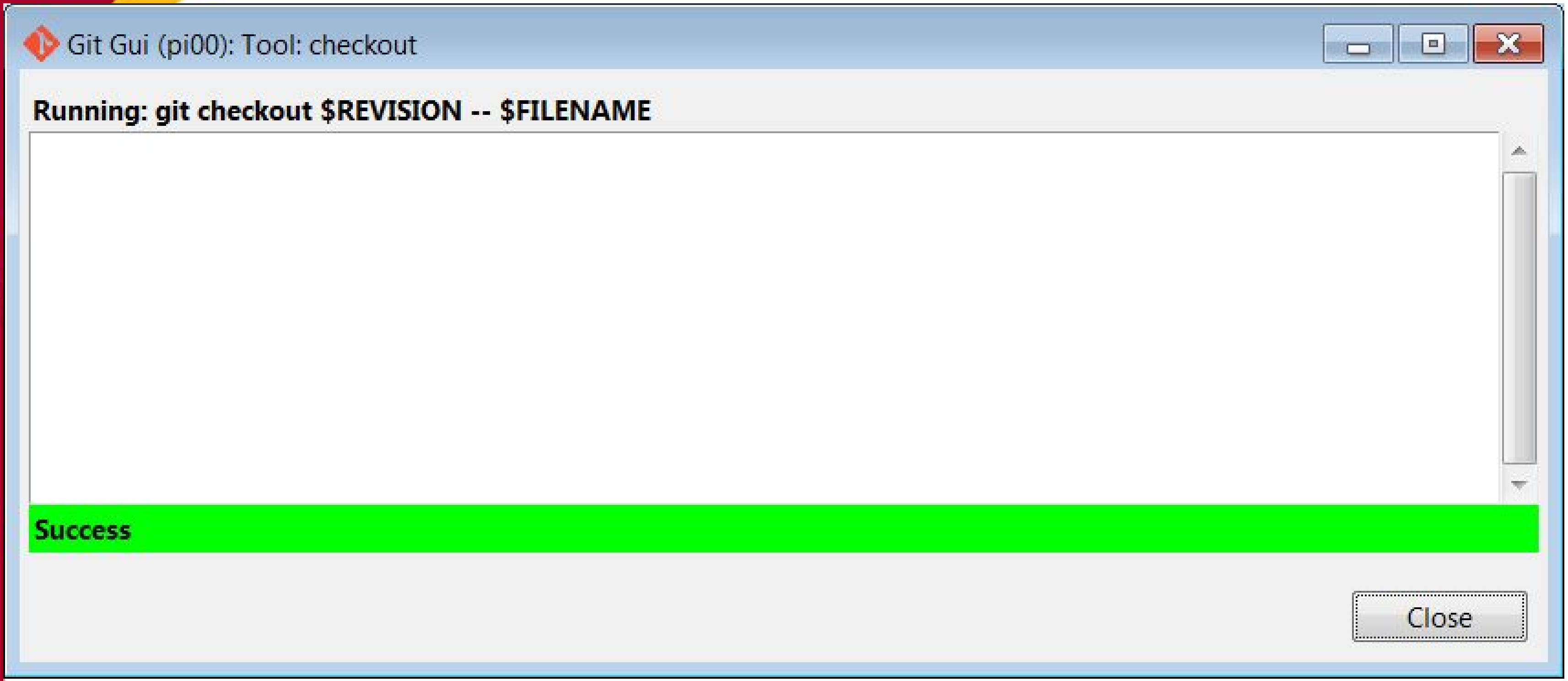


```
@@ -1,3 +1,2 @@
-This is the fi
-This is the second line
-This is A and B's third line
+I made a mistake
+and broke something!!
```

Ya nos aparece ‘checkout’ en el menú ‘Tools’

Se revertirán los cambios en todos los ficheros que tengamos seleccionados

Revertir errores usando checkout



Git Gui (pi00): Tool: checkout

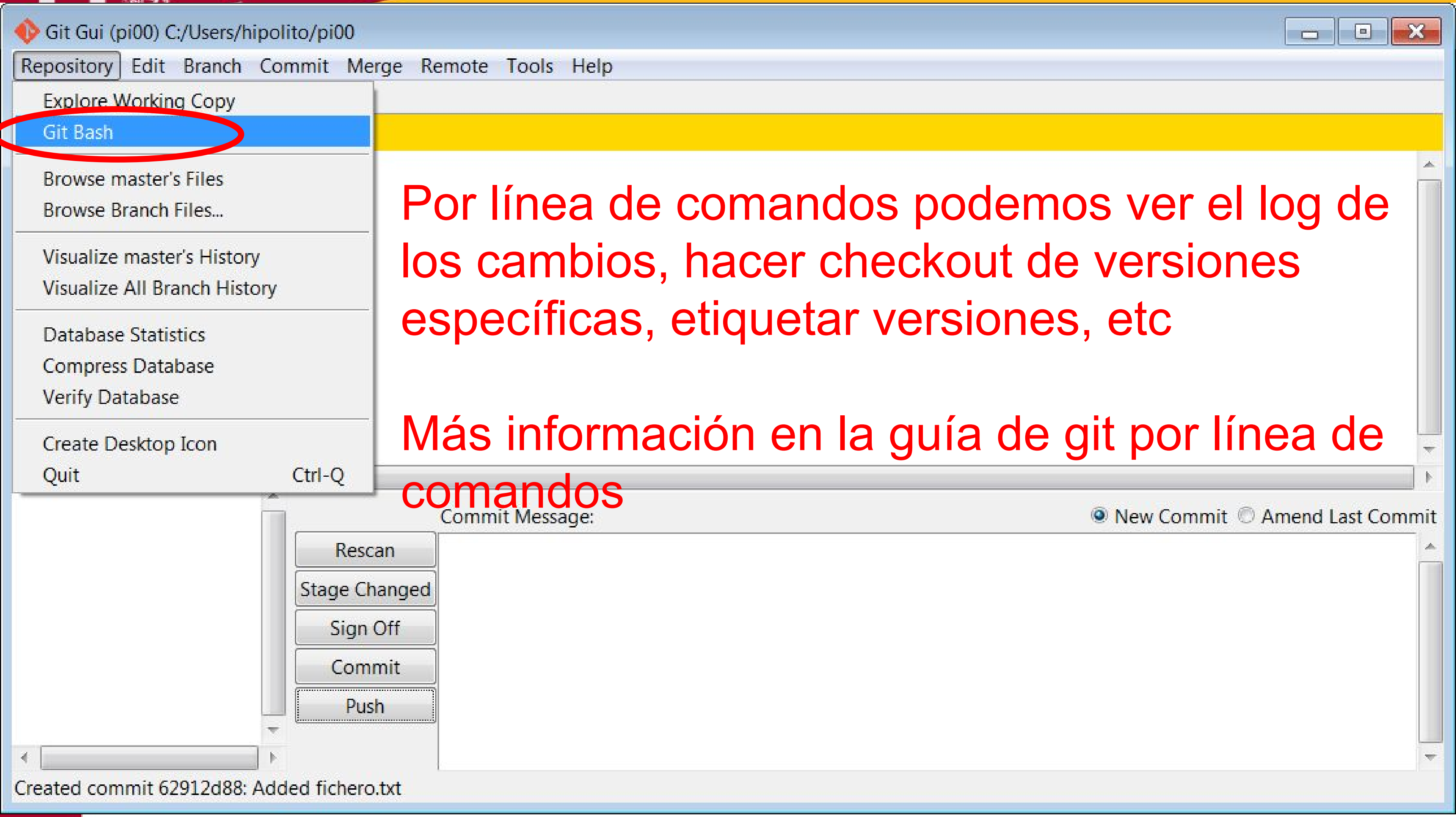
Running: git checkout $REVISION -- $FILENAME

Success

Tras esto, los ficheros que hayamos ‘checked out’ estarán igual que la versión del repo

Para profundizar

Para profundizar



Commit Message:

● New Commit ○ Amend Last Commit

Created commit 62912d88: Added fichero.txt

Por línea de comandos podemos ver el log de los cambios, hacer checkout de versiones específicas, etiquetar versiones, etc

Más información en la guía de git por línea de comandos

Valor… y al repo!