

Guía rápida de Git en modo gráfico

Hipólito Guzmán Miranda
Universidad de Sevilla

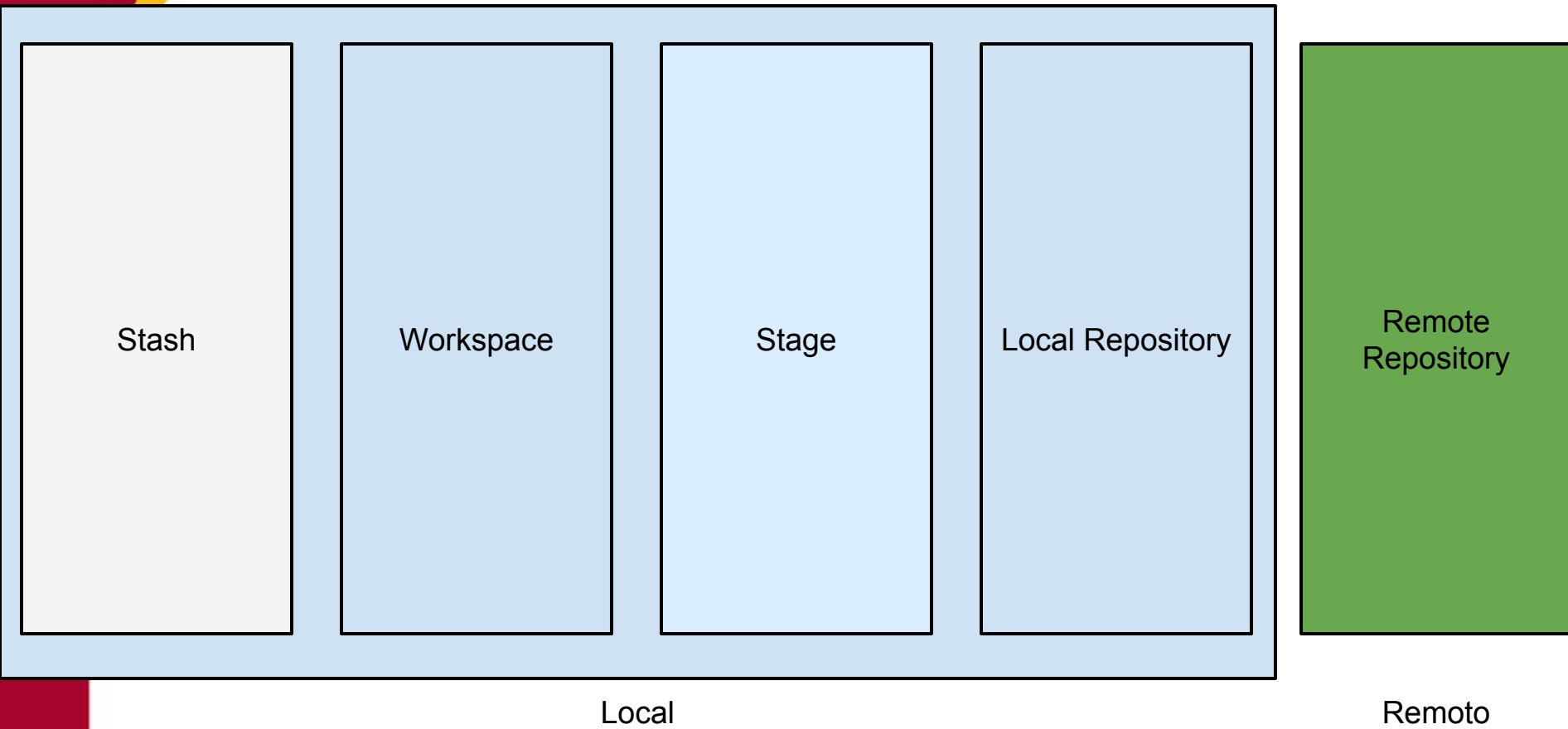
Contenido

- 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



Descripción

Espacio para esconder cambios temporalmente

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

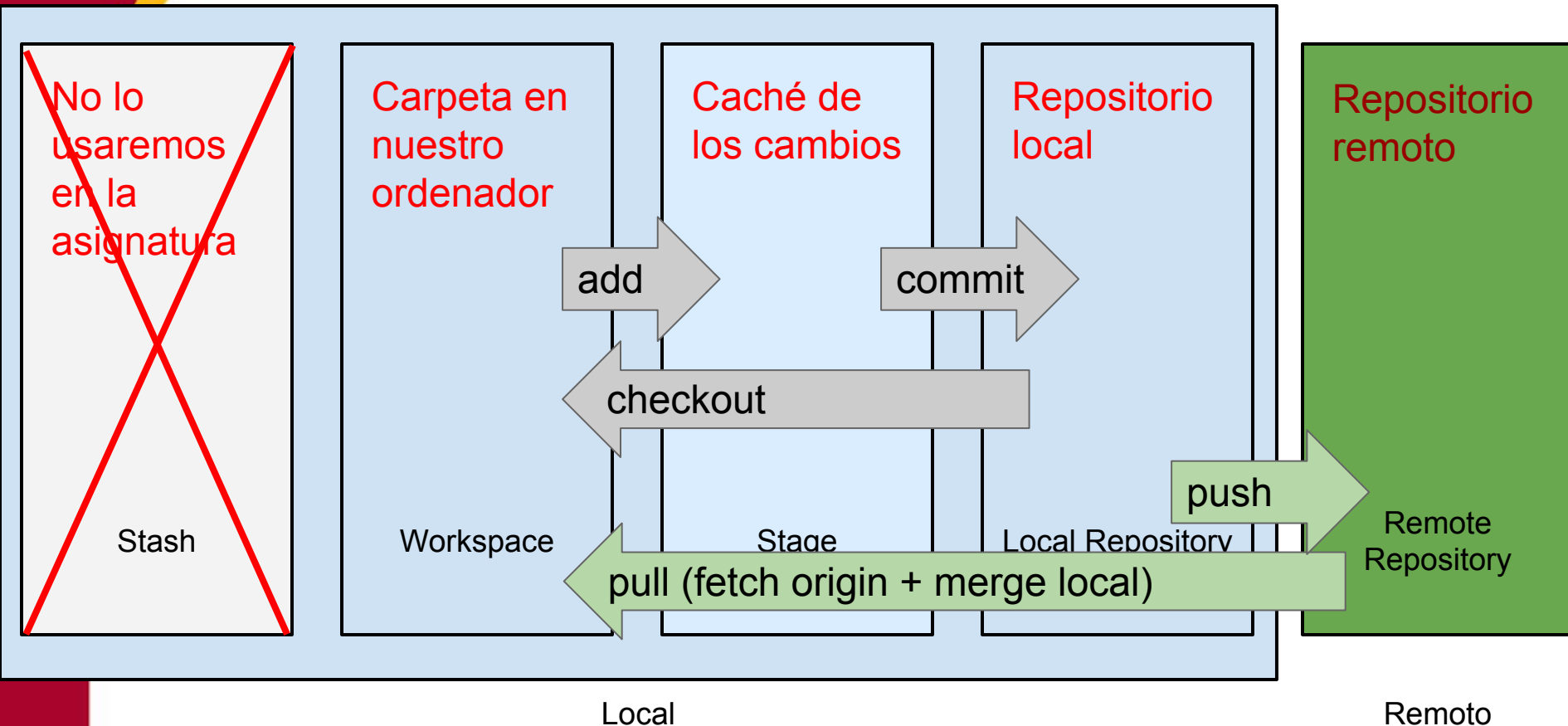
Repositorio externo, accesible por todos los miembros del grupo

Remote Repository

Local

Remoto

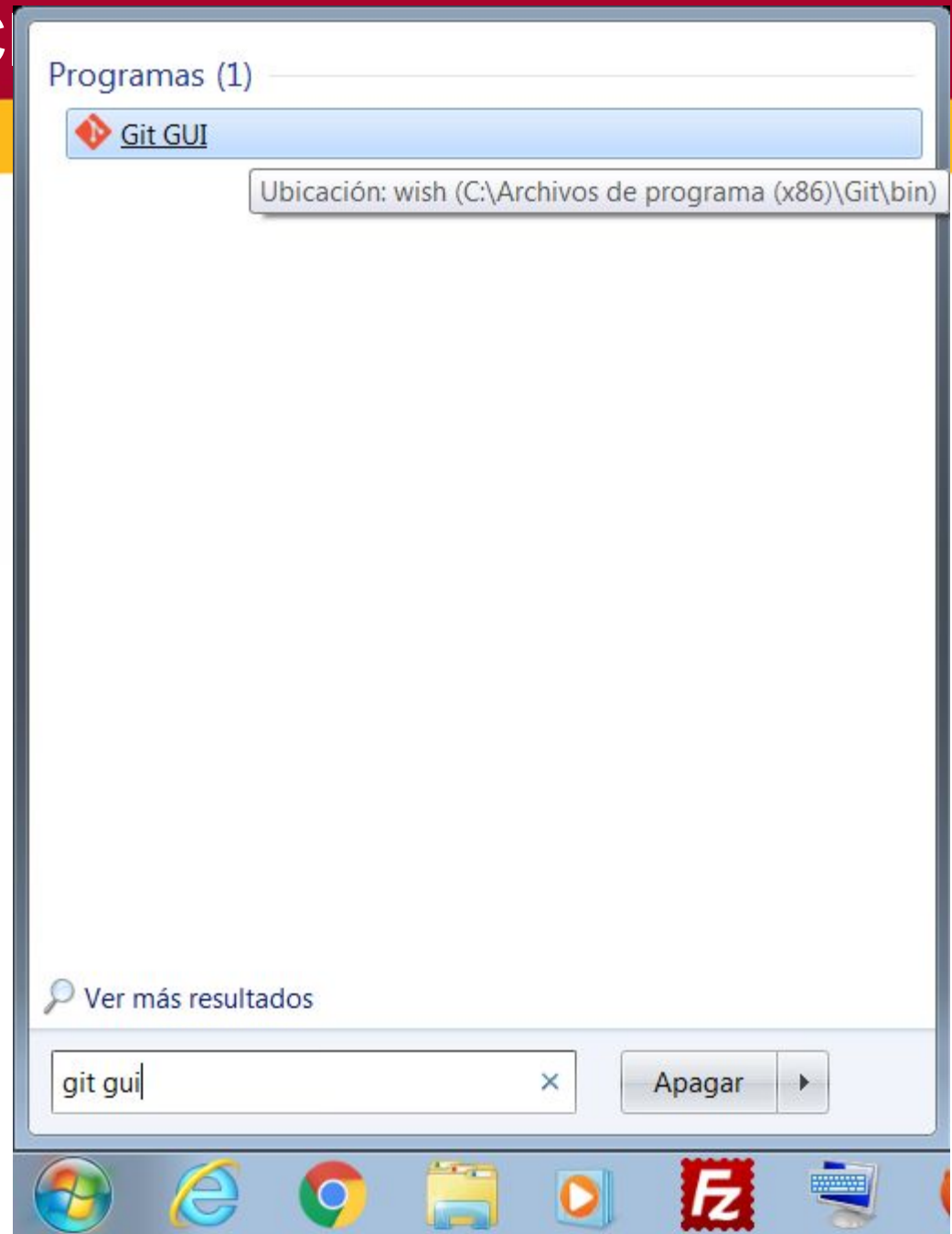
Comandos



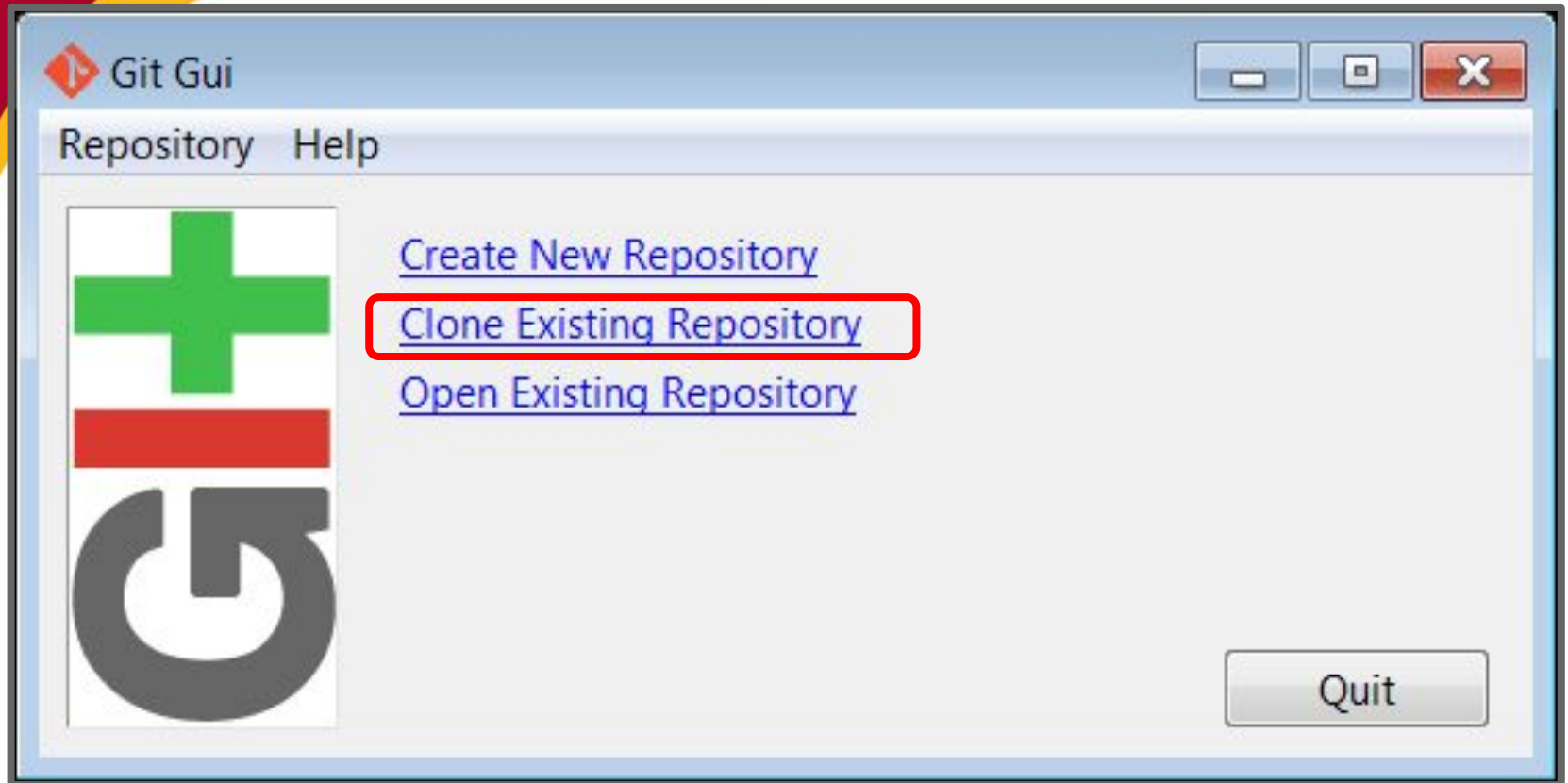
Contenido

- 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

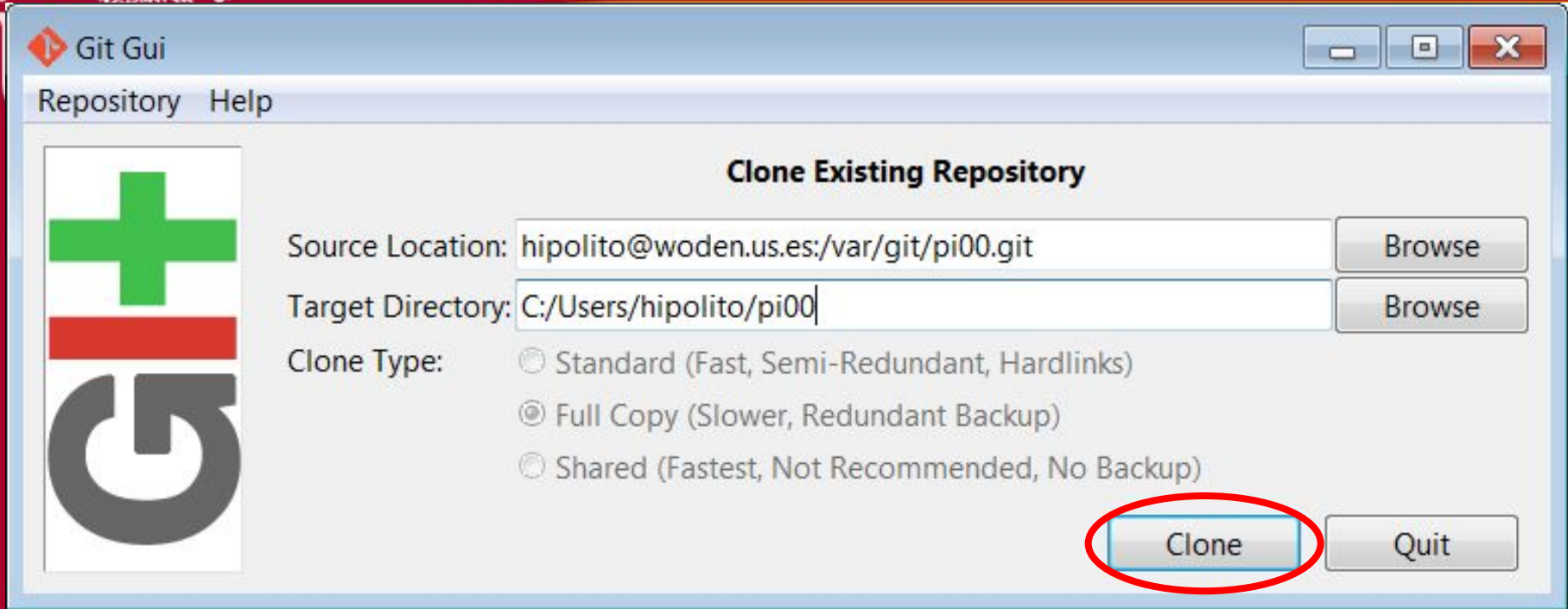
Abrimos Git GUI



Clonar el repo

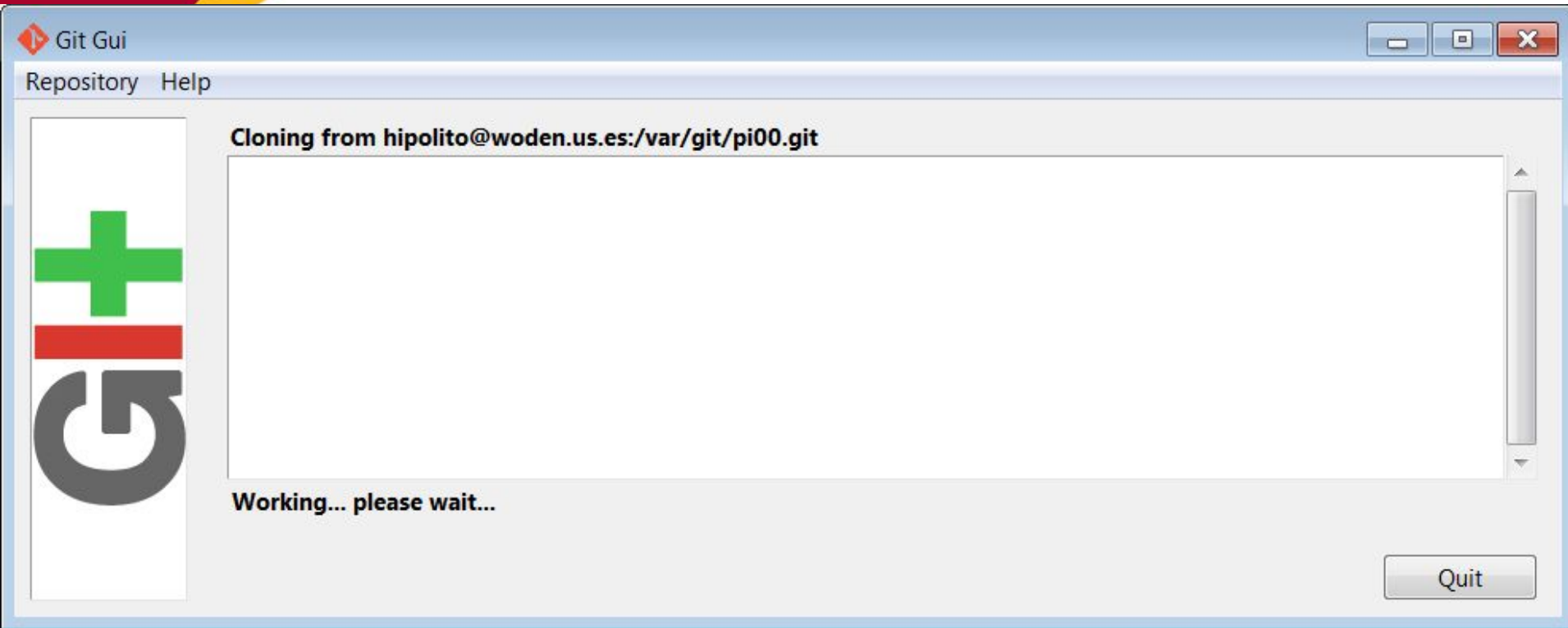


Clonar el repo

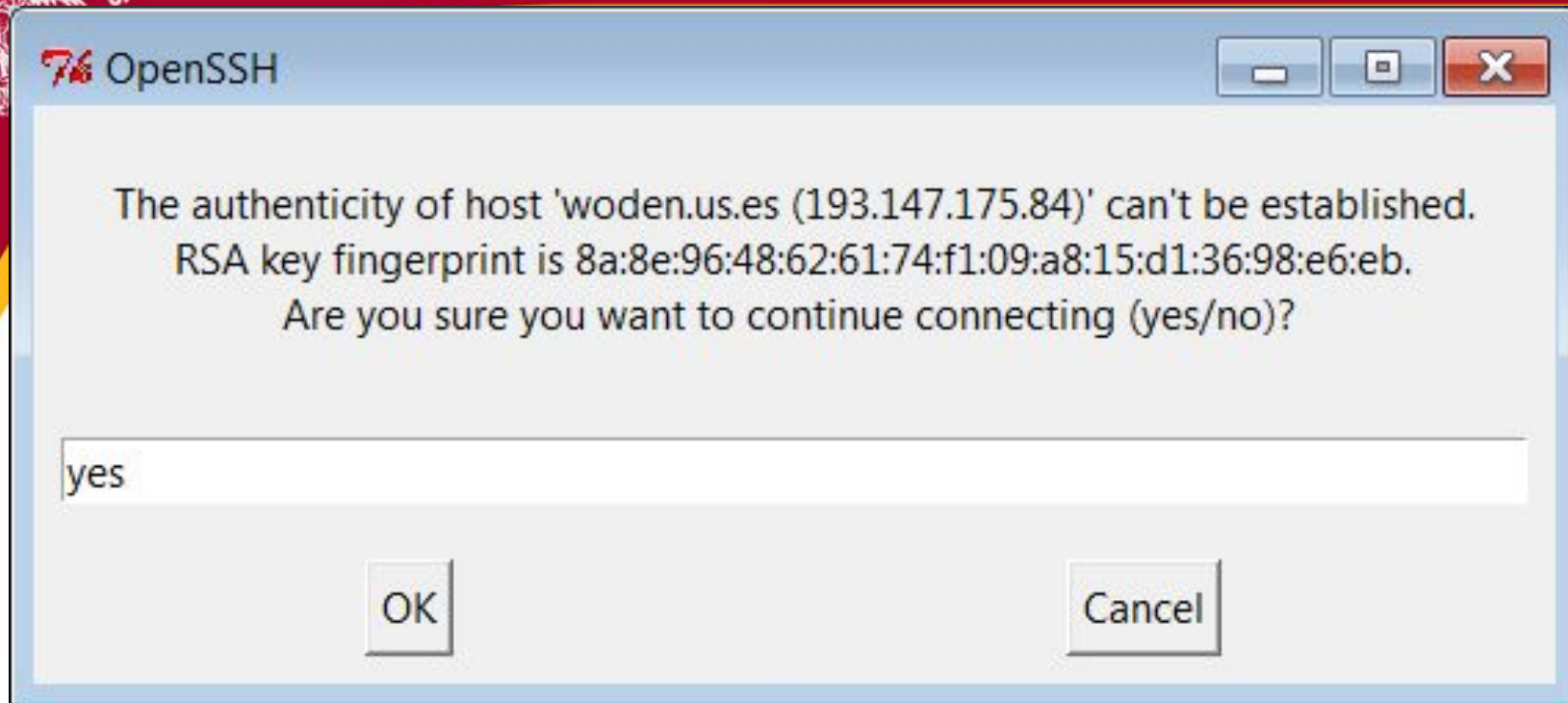


- Source es `usuario@servidor:ruta_al_repo`
- Cambia `hipolito` por tu usuario y `pi00.git` por el repositorio de tu grupo
- El target directory será creado al clonar el repositorio

Clonar el repo

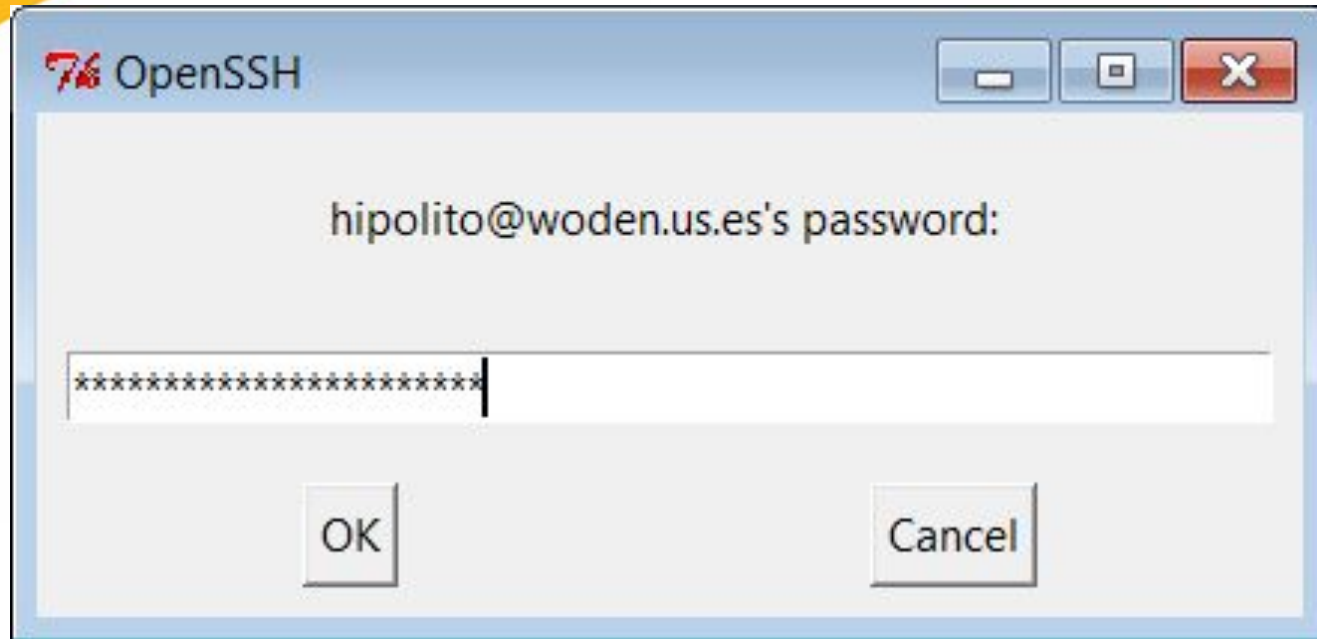


Clonar el repo

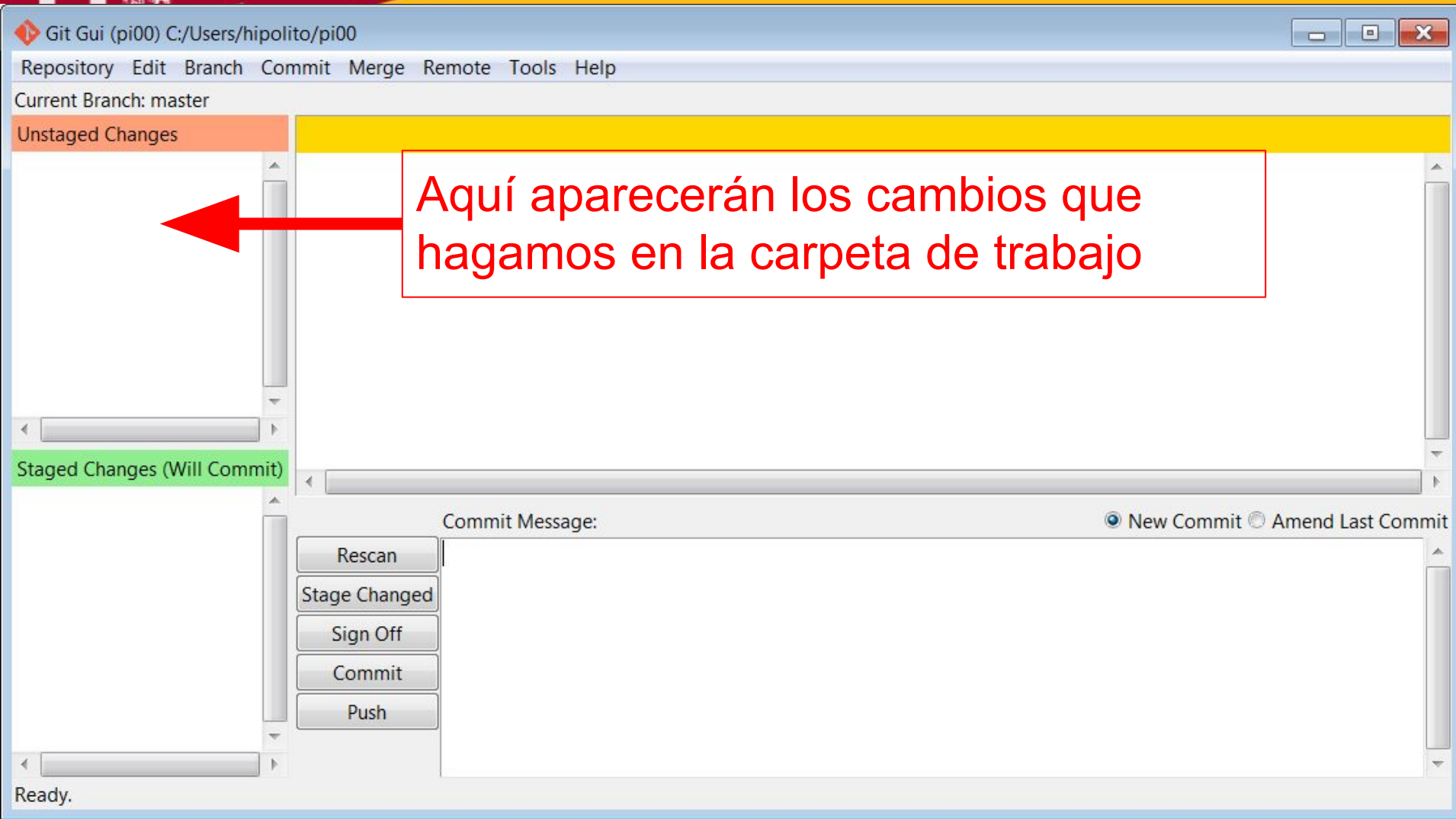


Este mensaje sólo lo veremos la primera vez que conectemos con un servidor (en nuestro caso, el servidor es woden.us.es)

Clonar el repo

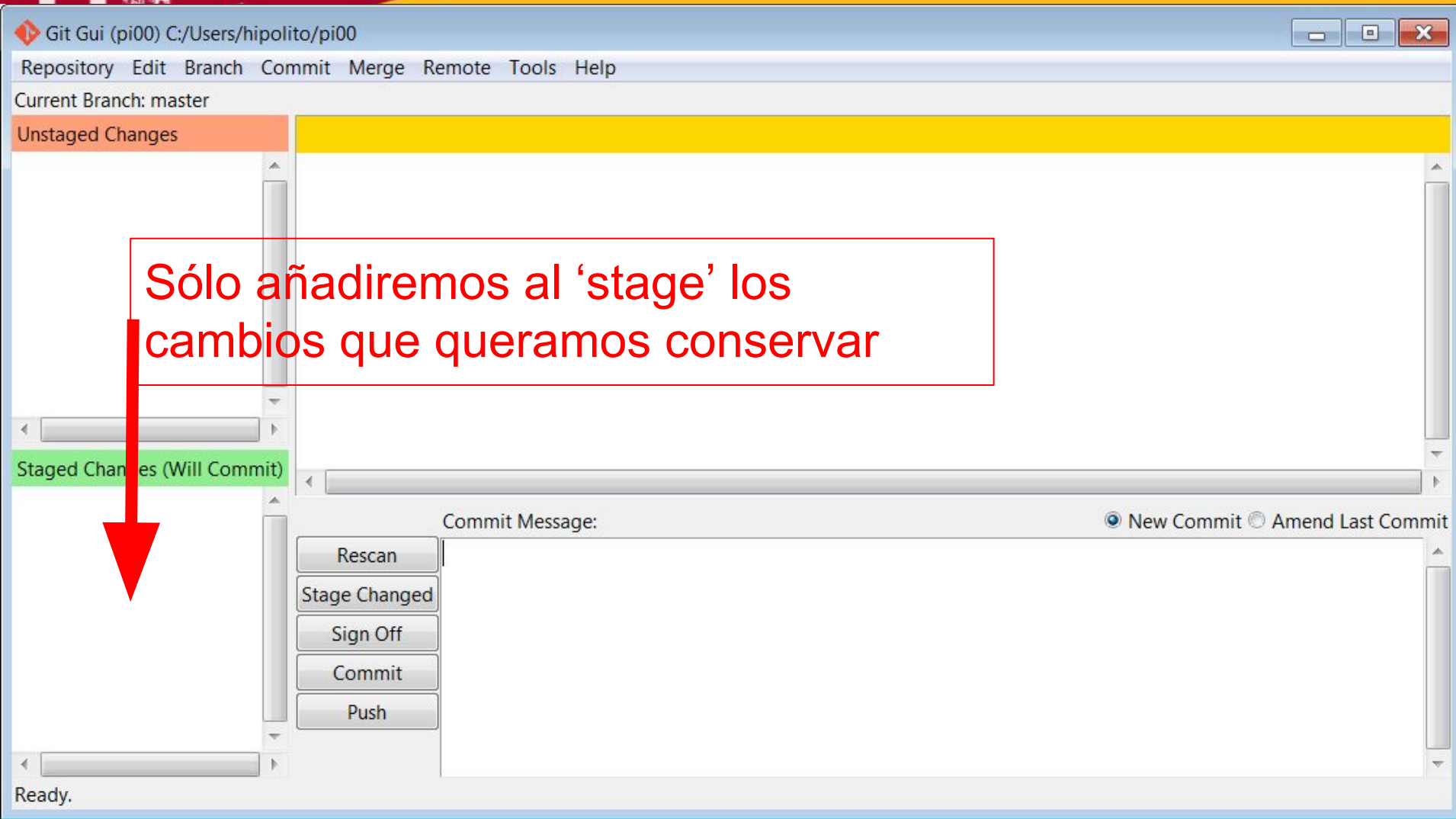


Siempre que hagamos una operación con el repositorio remoto nos pedirá el password



The screenshot shows the Git GUI application window. The title bar reads "Git Gui (pi00) C:/Users/hipolito/pi00". The menu bar includes "Repository", "Edit", "Branch", "Commit", "Merge", "Remote", "Tools", and "Help". Below the menu bar, it says "Current Branch: master". The main area is divided into two sections: "Unstaged Changes" (highlighted in orange) and "Staged Changes (Will Commit)" (highlighted in green). A red arrow points from a text box to the "Unstaged Changes" section. The "Commit Message:" field is visible, with radio buttons for "New Commit" (selected) and "Amend Last Commit". A vertical toolbar on the left contains buttons for "Rescan", "Stage Changed", "Sign Off", "Commit", and "Push". The status bar at the bottom left says "Ready."

Aquí aparecerán los cambios que hagamos en la carpeta de trabajo

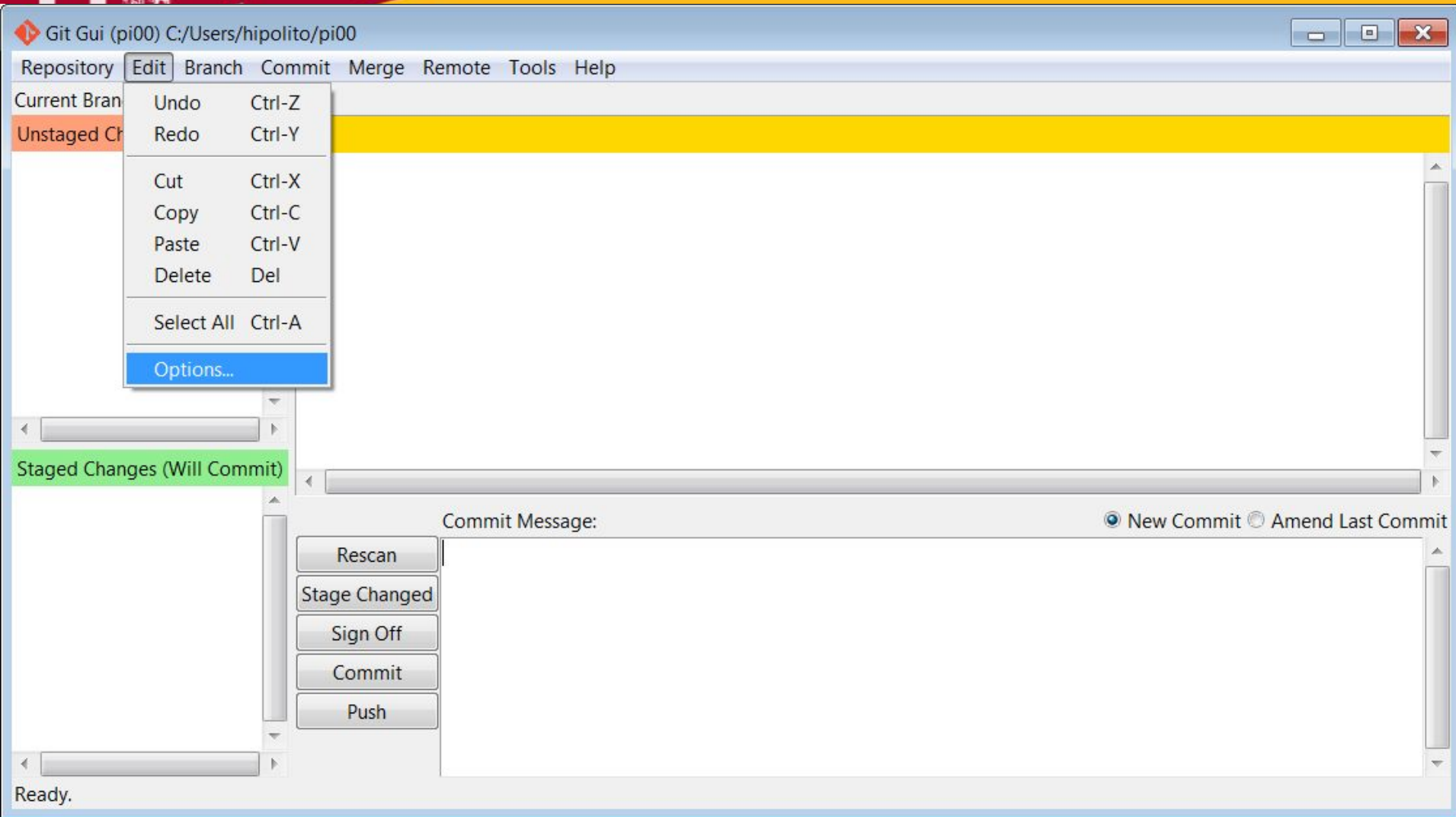


The screenshot shows the Git GUI application window. The title bar reads "Git Gui (pi00) C:/Users/hipolito/pi00". The menu bar includes "Repository", "Edit", "Branch", "Commit", "Merge", "Remote", "Tools", and "Help". Below the menu bar, it says "Current Branch: master". The main area is divided into two sections: "Unstaged Changes" (highlighted in yellow) and "Staged Changes (Will Commit)" (highlighted in green). A red box with the text "Sólo añadiremos al 'stage' los cambios que queremos conservar" is overlaid on the "Unstaged Changes" section. A red arrow points from this box down to the "Staged Changes" section. At the bottom, there is a "Commit Message:" field with radio buttons for "New Commit" (selected) and "Amend Last Commit". Below the message field are buttons for "Rescan", "Stage Changed", "Sign Off", "Commit", and "Push". The status bar at the bottom left says "Ready."

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



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

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

Warn before committing to a detached head

Staging of untracked files: ask

Show untracked files

Spelling Dictionary:

Main Font: Segoe UI 9 pt.

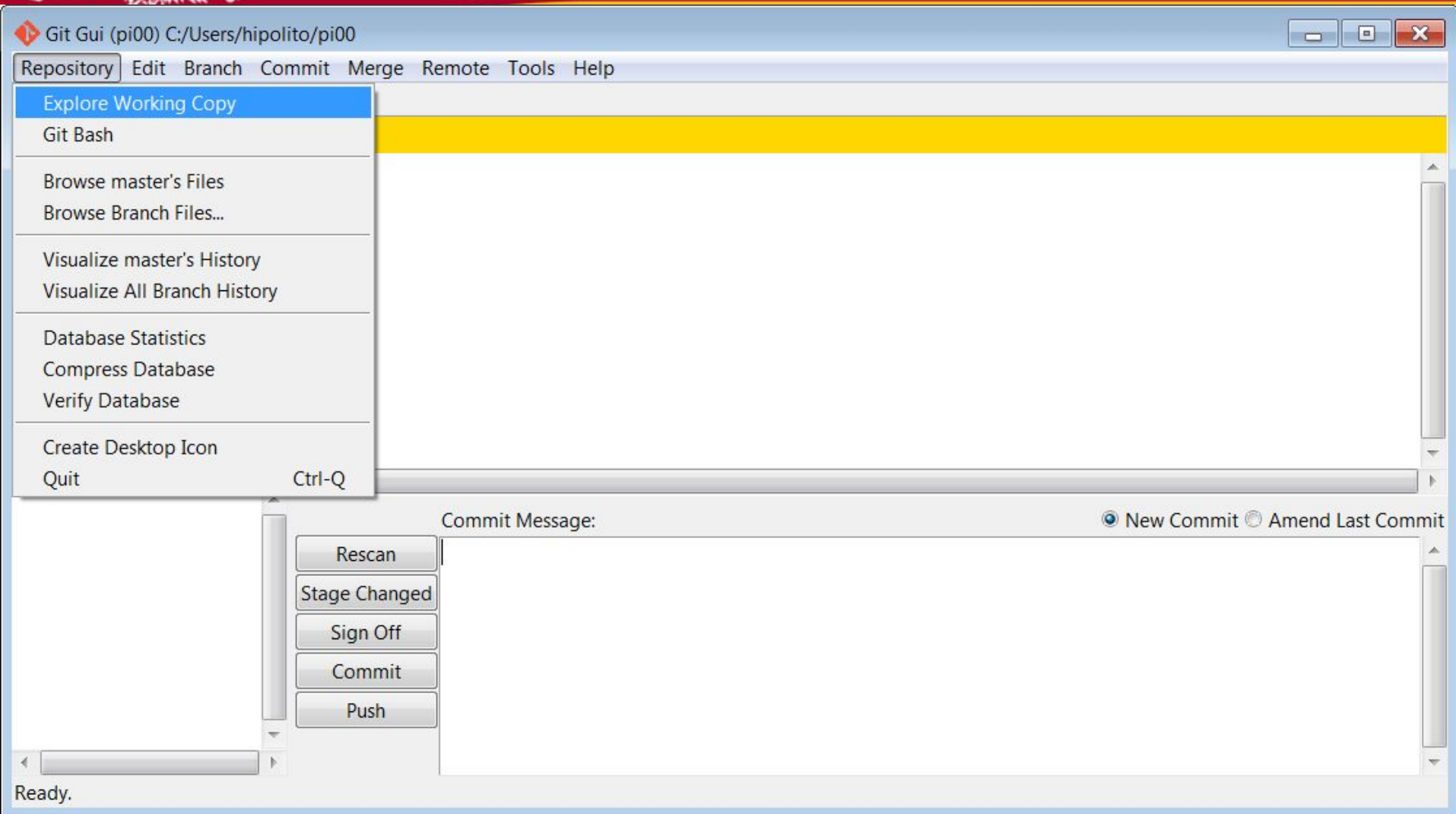
Diff/Console Font: Courier New 10 pt.

Poned vuestro
nombre y
vuestro email

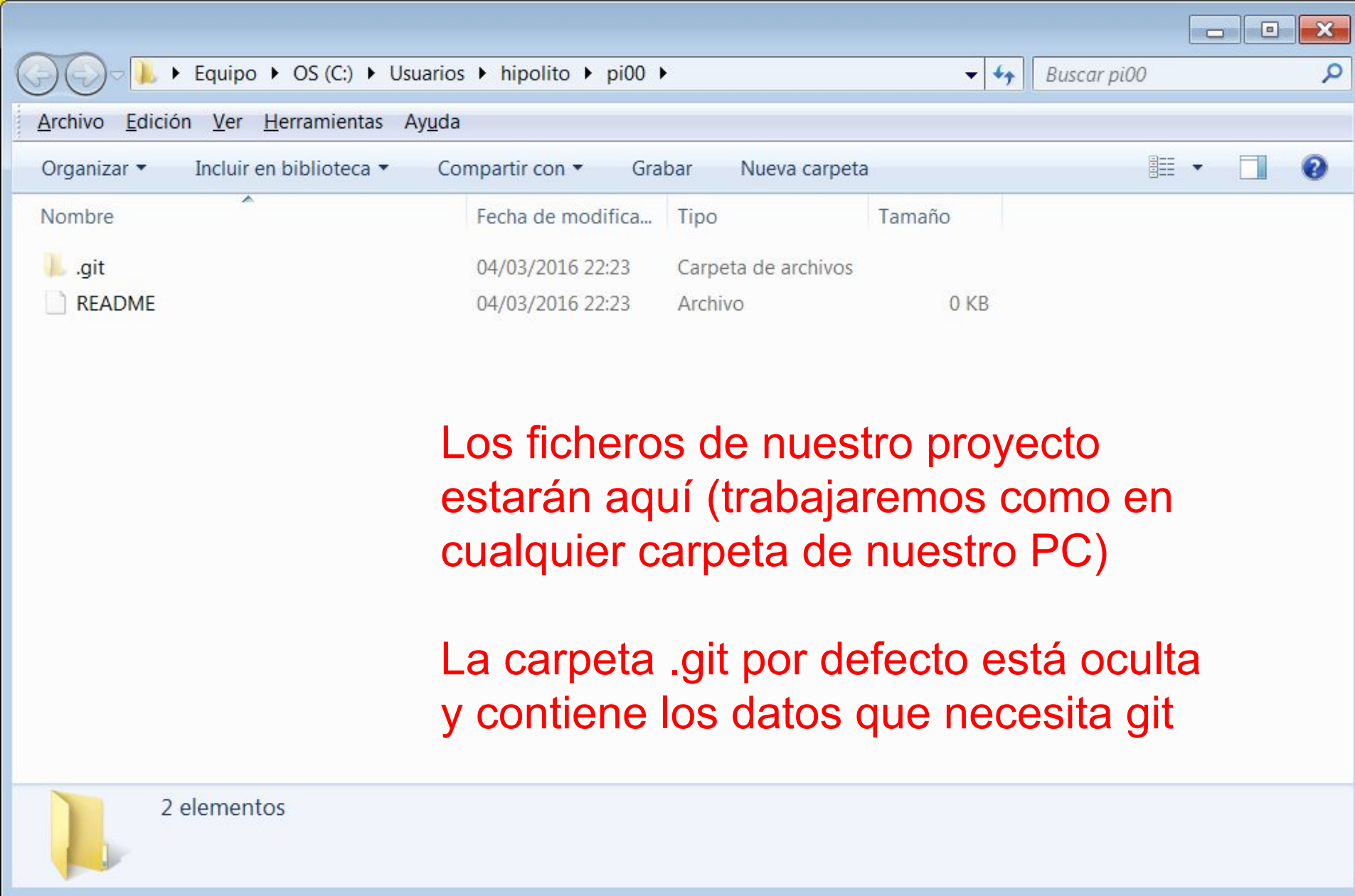
Contenido

- 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



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



Equipo > OS (C:) > Usuarios > hipolito > pi00

Buscar pi00

Archivo Edición Ver Herramientas Ayuda

Organizar Abrir Compartir con Imprimir Grabar Nueva carpeta

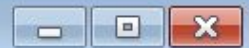
Nombre	Fecha de modifica...	Tipo	Tamaño
.git	04/03/2016 22:29	Carpeta de archivos	
hipolito.txt	04/03/2016 22:29	Documento de tex...	0 KB
README	04/03/2016 22:23	Archivo	0 KB



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

Creamos un fichero con nuestro nombre

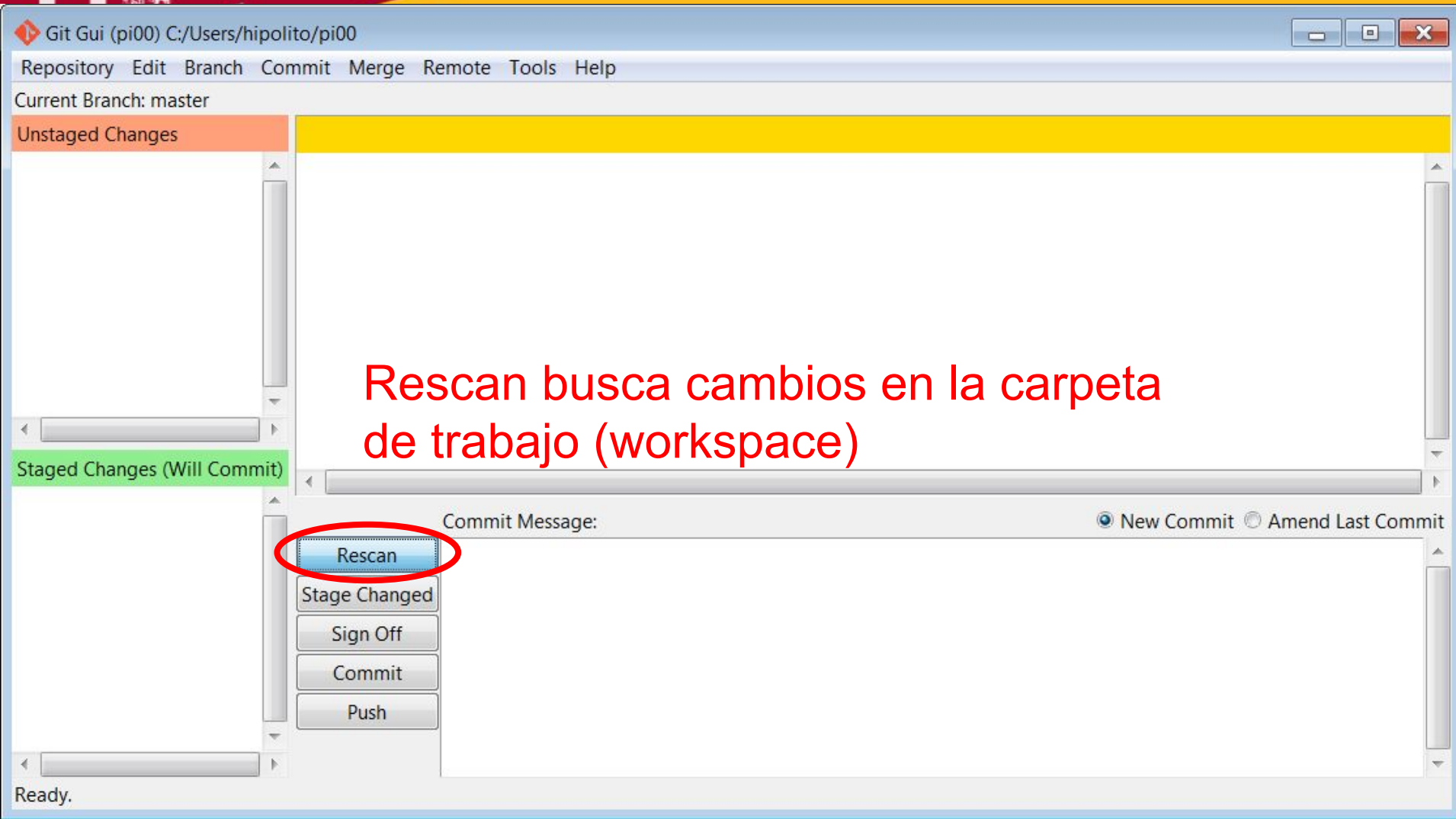
hipolito.txt: Bloc de notas



Archivo Edición Formato Ver Ayuda

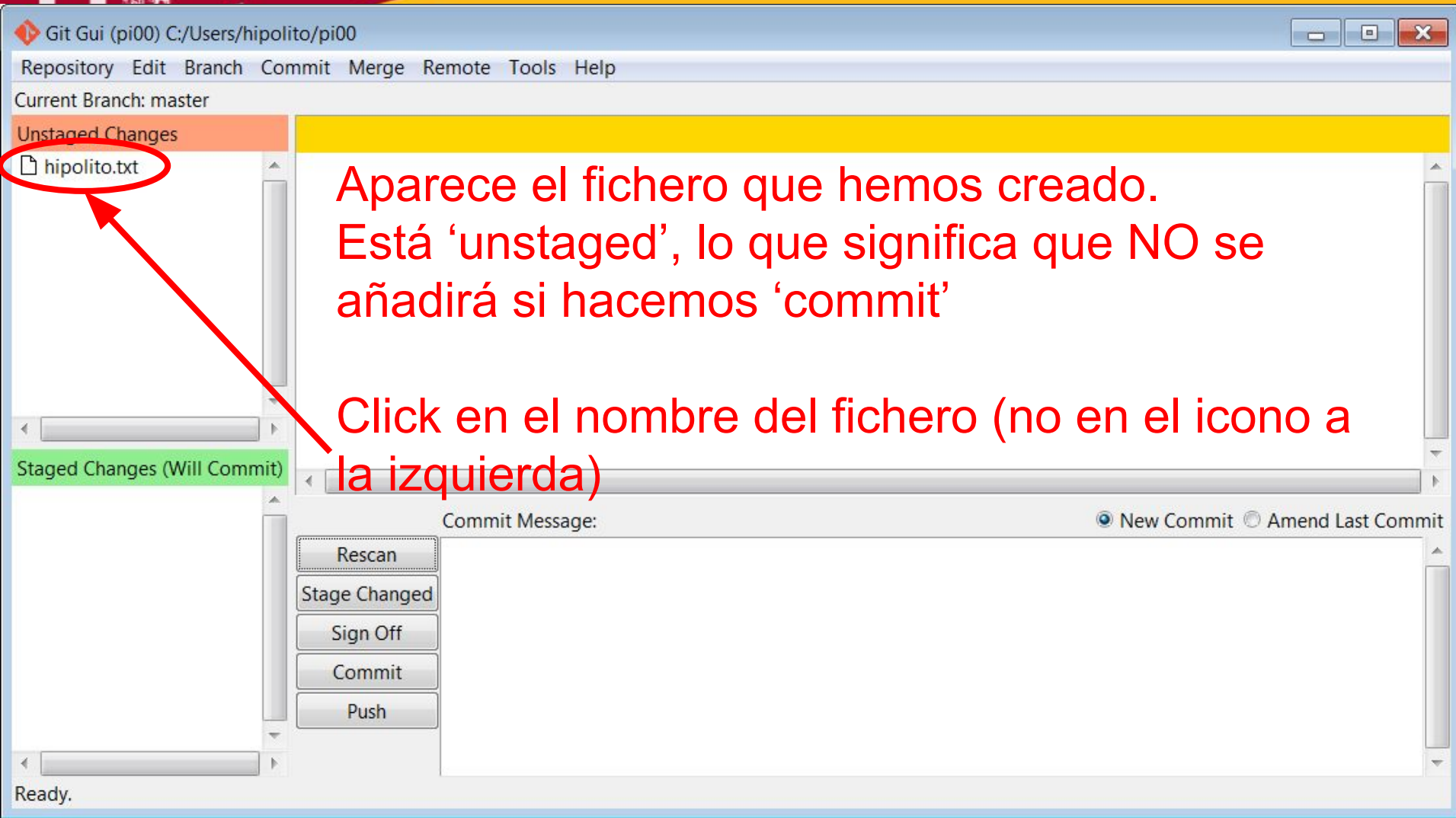
Contenido de 'hipolito.txt'

Añadir ficheros al stage



The screenshot shows the Git GUI application window. The title bar reads "Git Gui (pi00) C:/Users/hipolito/pi00". The menu bar includes "Repository", "Edit", "Branch", "Commit", "Merge", "Remote", "Tools", and "Help". Below the menu bar, it says "Current Branch: master". The main area is divided into "Unstaged Changes" (highlighted in orange) and "Staged Changes (Will Commit)" (highlighted in green). A large red text overlay in the center reads "Rescan busca cambios en la carpeta de trabajo (workspace)". At the bottom left, a vertical toolbar contains buttons for "Rescan", "Stage Changed", "Sign Off", "Commit", and "Push". The "Rescan" button is circled in red. To the right of the toolbar is a "Commit Message:" field with radio buttons for "New Commit" (selected) and "Amend Last Commit". The status bar at the bottom left says "Ready."

Añadir ficheros al stage



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

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

Repository Edit Branch Commit Merge Remote Tools Help

Current Branch: master

Unstaged Changes **Untracked not staged** File: hipolito.txt

hipolito.txt Contenido de 'hipolito.txt'

Staged Changes (Will Commit)

Commit Message: New Commit Amend Last Commit

Rescan
Stage Changed
Sign Off
Commit
Push

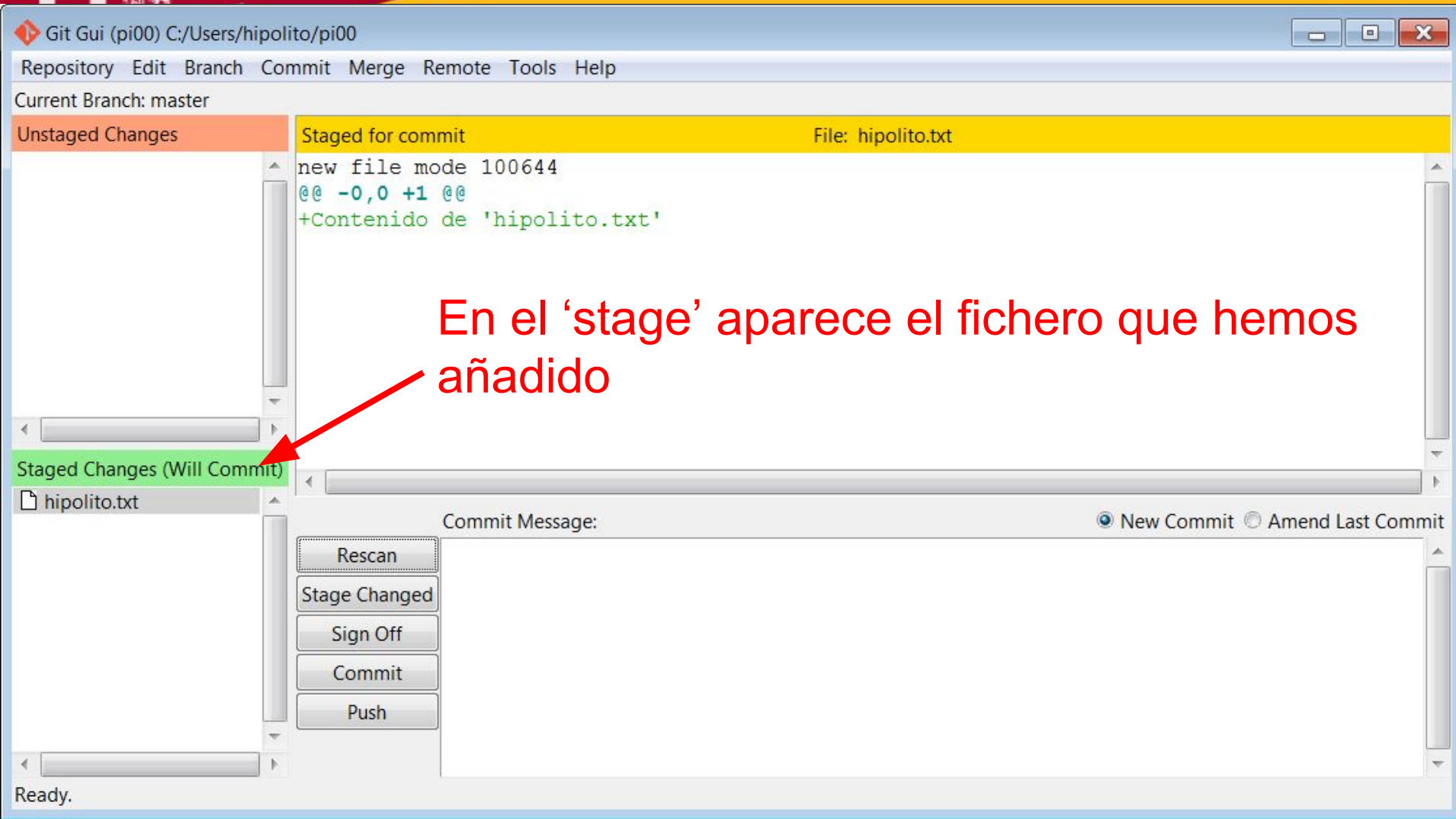
Ready.

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



The screenshot shows the Git GUI interface. The window title is "Git Gui (pi00) C:/Users/hipolito/pi00". The menu bar includes "Repository", "Edit", "Branch", "Commit", "Merge", "Remote", "Tools", and "Help". The current branch is "master".

The interface is divided into several sections:

- Unstaged Changes:** This section is currently empty.
- Staged for commit:** This section is highlighted in yellow and shows the details of the file "hipolito.txt" that has been staged. The content is:

```
new file mode 100644
@@ -0,0 +1 @@
+Contenido de 'hipolito.txt'
```
- Staged Changes (Will Commit):** This section is highlighted in green and shows the file "hipolito.txt" as a single item.
- Commit Message:** This section contains a text area for entering a commit message. It also has radio buttons for "New Commit" (selected) and "Amend Last Commit".
- Buttons:** A vertical stack of buttons is located below the commit message area: "Rescan", "Stage Changed", "Sign Off", "Commit", and "Push".

A red arrow points from the text "En el 'stage' aparece el fichero que hemos añadido" to the "Staged Changes (Will Commit)" section.

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

Current Branch: master

Unstaged Changes	Staged for commit	File: hipolito.txt
	<pre>new file mode 100644 @@ -0,0 +1 @@ +Contenido de 'hipolito.txt'</pre>	

Staged Changes (Will Commit)

-  hipolito.txt

Commit Message: New Commit Amend Last Commit

Buttons: Rescan, Stage Changed, Sign Off, Commit, Push

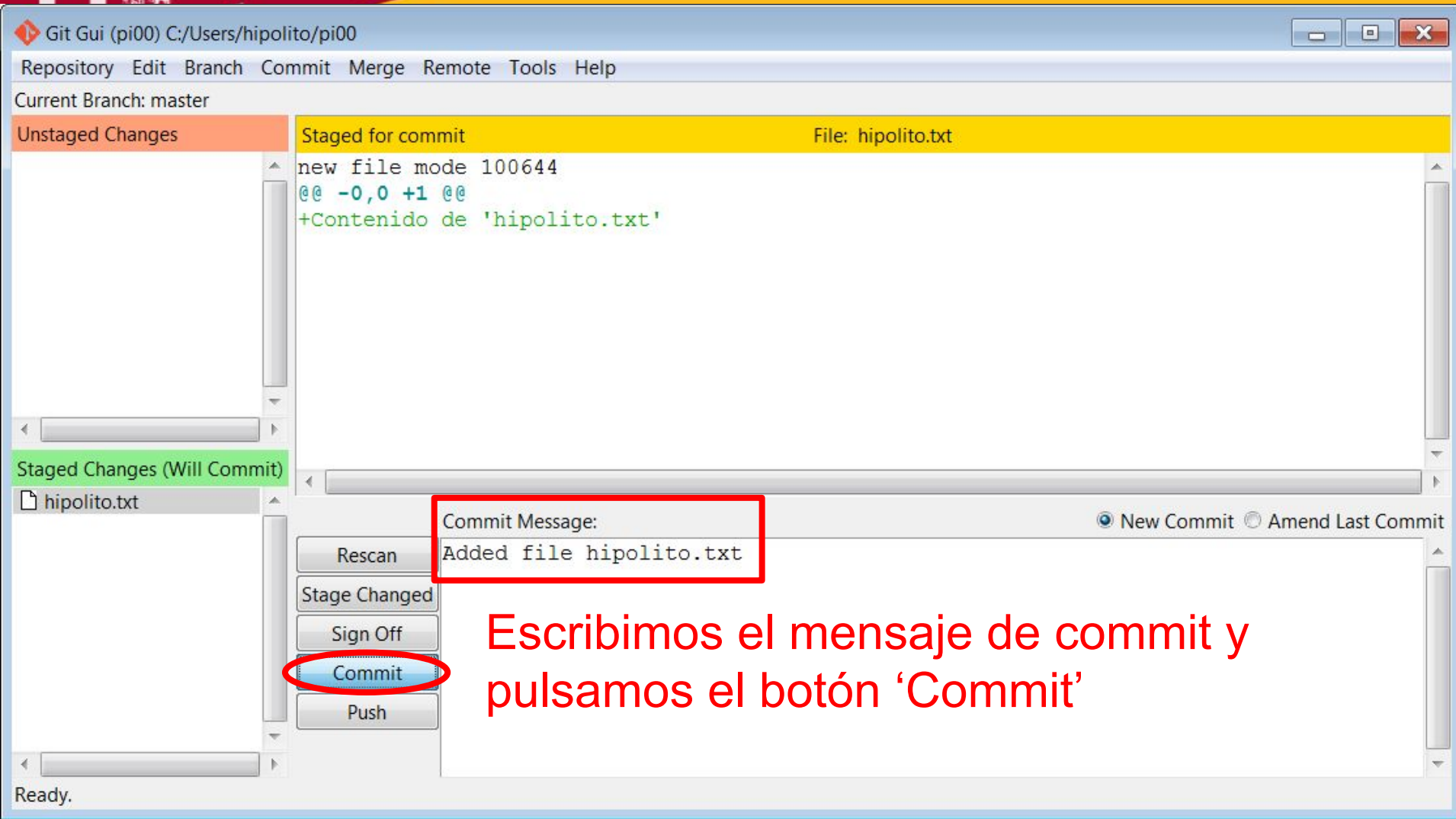
Ready.

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



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

Repository Edit Branch Commit Merge Remote Tools Help

Current Branch: master

Unstaged Changes

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: Added file hipolito.txt

New Commit Amend Last Commit

Rescan

Stage Changed

Sign Off

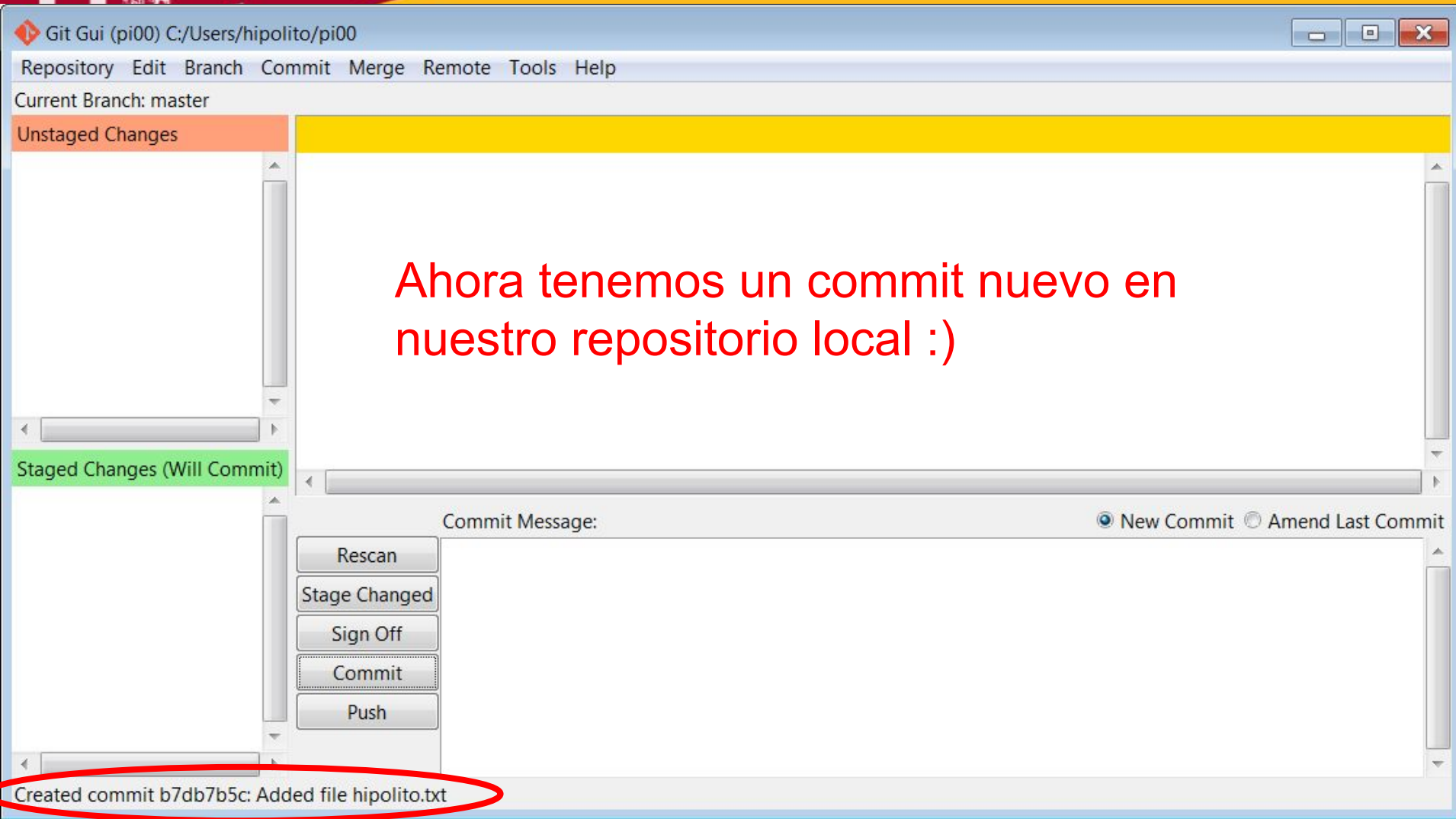
Commit

Push

Ready.

Escribimos el mensaje de commit y pulsamos el botón 'Commit'

Creando commits



The screenshot shows the Git GUI application window. The title bar reads "Git Gui (pi00) C:/Users/hipolito/pi00". The menu bar includes "Repository", "Edit", "Branch", "Commit", "Merge", "Remote", "Tools", and "Help". Below the menu bar, it says "Current Branch: master".

The main area is divided into two sections: "Unstaged Changes" (highlighted in orange) and "Staged Changes (Will Commit)" (highlighted in green). The "Staged Changes" section is currently empty.

Below the staged changes, there is a "Commit Message:" field with a text area. To the right of this field are two radio buttons: "New Commit" (selected) and "Amend Last Commit".

On the left side of the commit message area, there is a vertical stack of buttons: "Rescan", "Stage Changed", "Sign Off", "Commit" (highlighted with a dotted border), and "Push".

At the bottom of the window, a status bar displays the message "Created commit b7db7b5c: Added file hipolito.txt", which is circled in red.

Ahora tenemos un commit nuevo en nuestro repositorio local :)

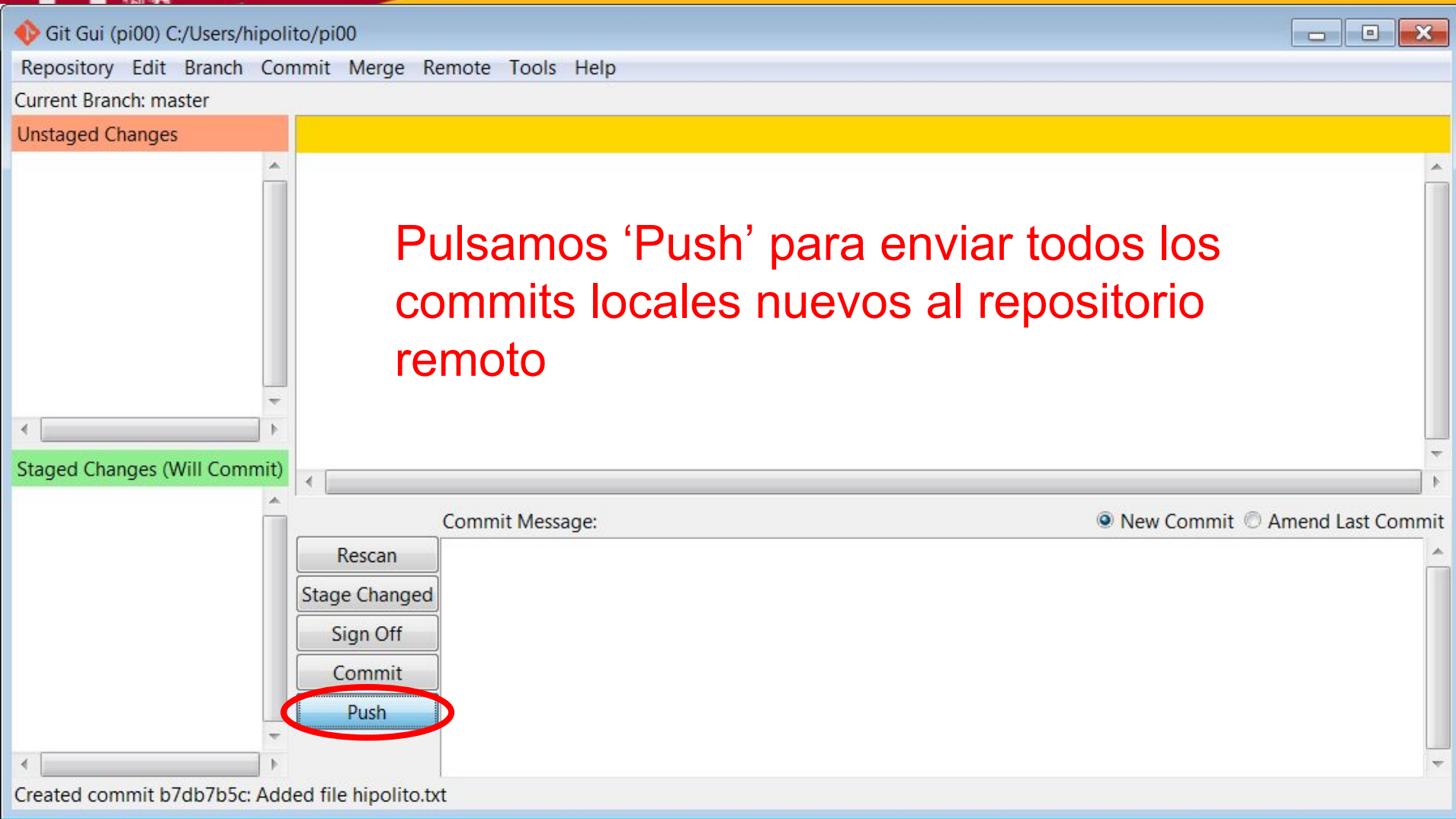
Formato correcto de un mensaje de commit

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

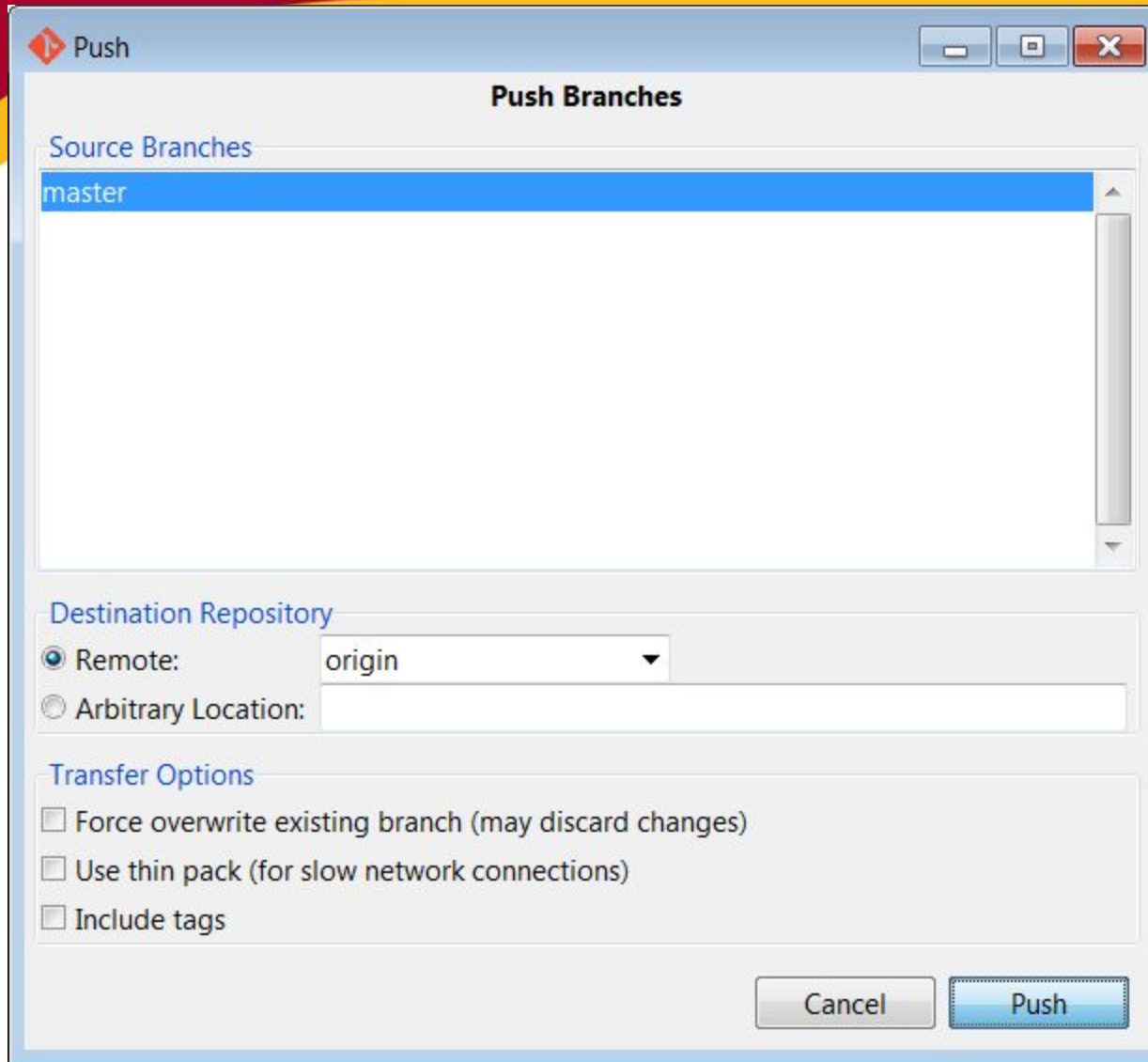
- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout



The screenshot shows the Git GUI application window. The title bar reads "Git Gui (pi00) C:/Users/hipolito/pi00". The menu bar includes "Repository", "Edit", "Branch", "Commit", "Merge", "Remote", "Tools", and "Help". The status bar at the top indicates "Current Branch: master". The interface is divided into several sections: "Unstaged Changes" (highlighted in orange), "Staged Changes (Will Commit)" (highlighted in green), and a "Commit Message" field. A vertical toolbar on the left contains buttons for "Rescan", "Stage Changed", "Sign Off", "Commit", and "Push". The "Push" button is circled in red. The main workspace area contains the text: "Pulsamos 'Push' para enviar todos los commits locales nuevos al repositorio remoto". At the bottom, a status bar shows "Created commit b7db7b5c: Added file hipolito.txt".

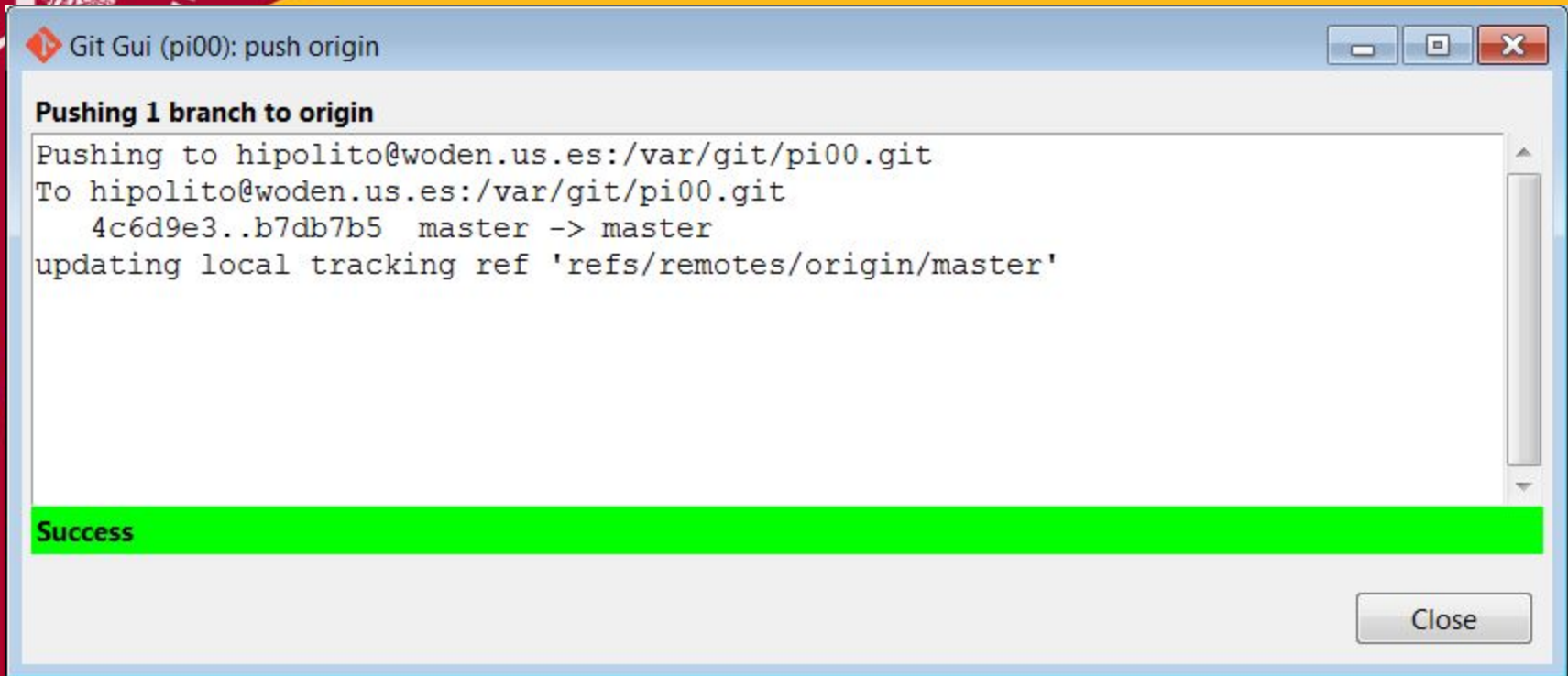
Pulsamos 'Push' para enviar todos los commits locales nuevos al repositorio remoto

Push



Por defecto 'pusheamos' a la rama master

Push



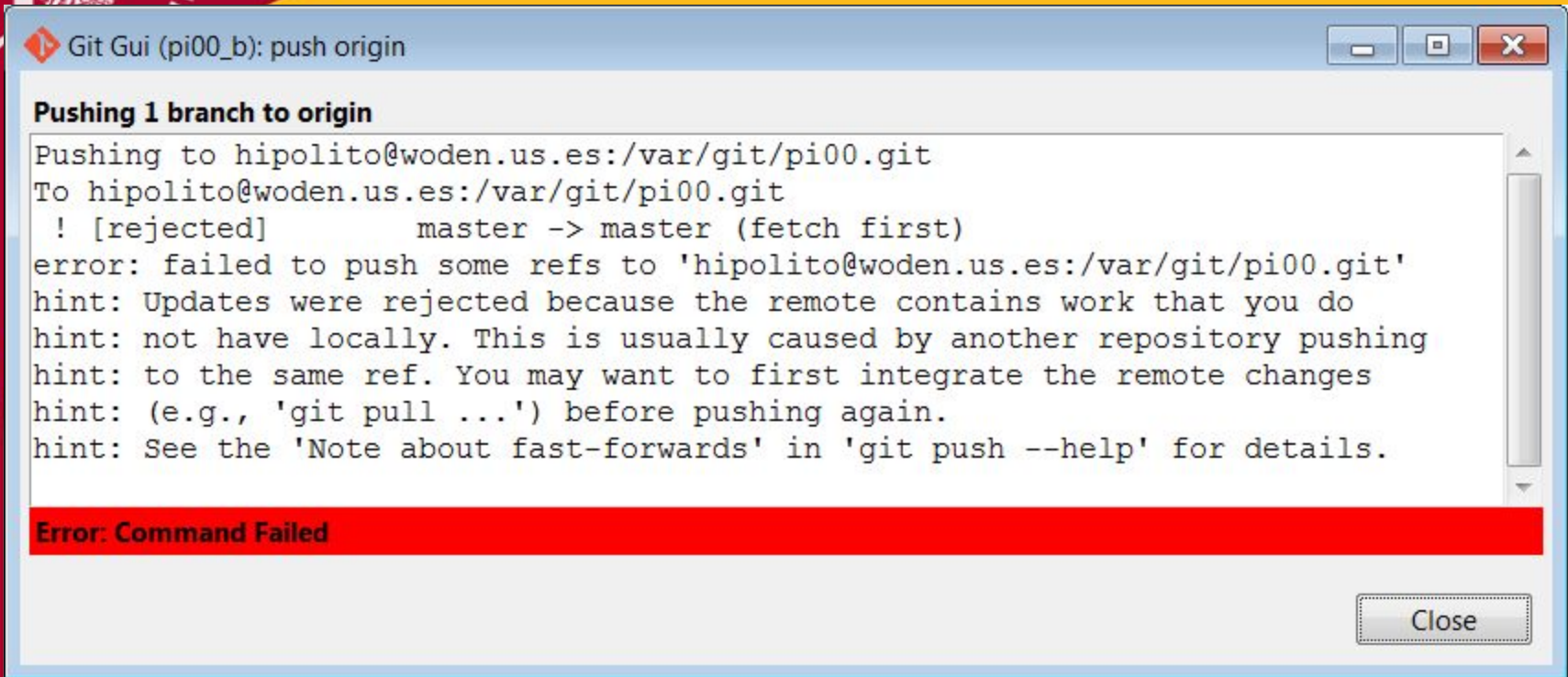
```
Git Gui (pi00): push origin

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
```

Close

Normalmente todo va bien...



```
Git Gui (pi00_b): push origin

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

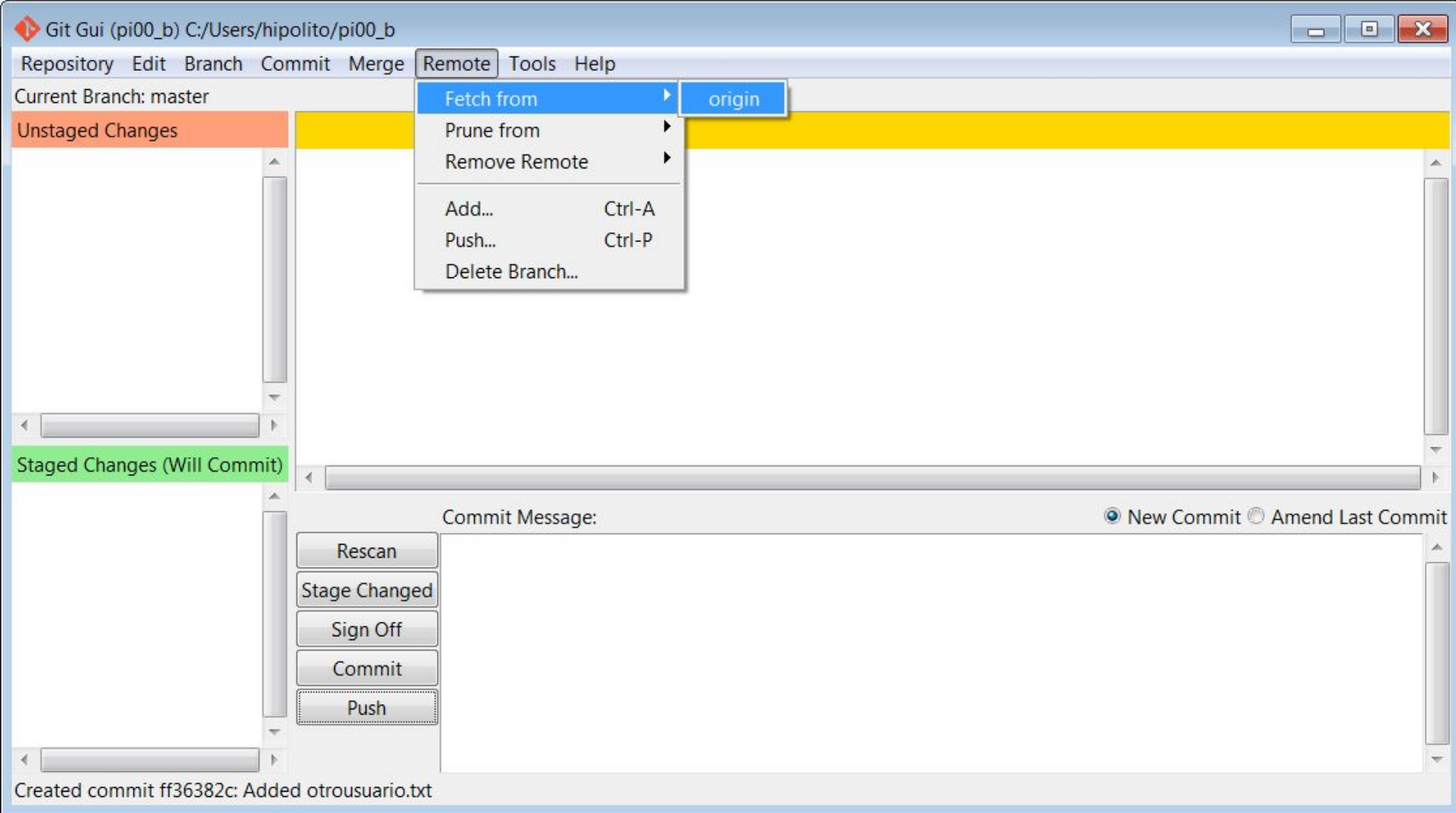
Close
```

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

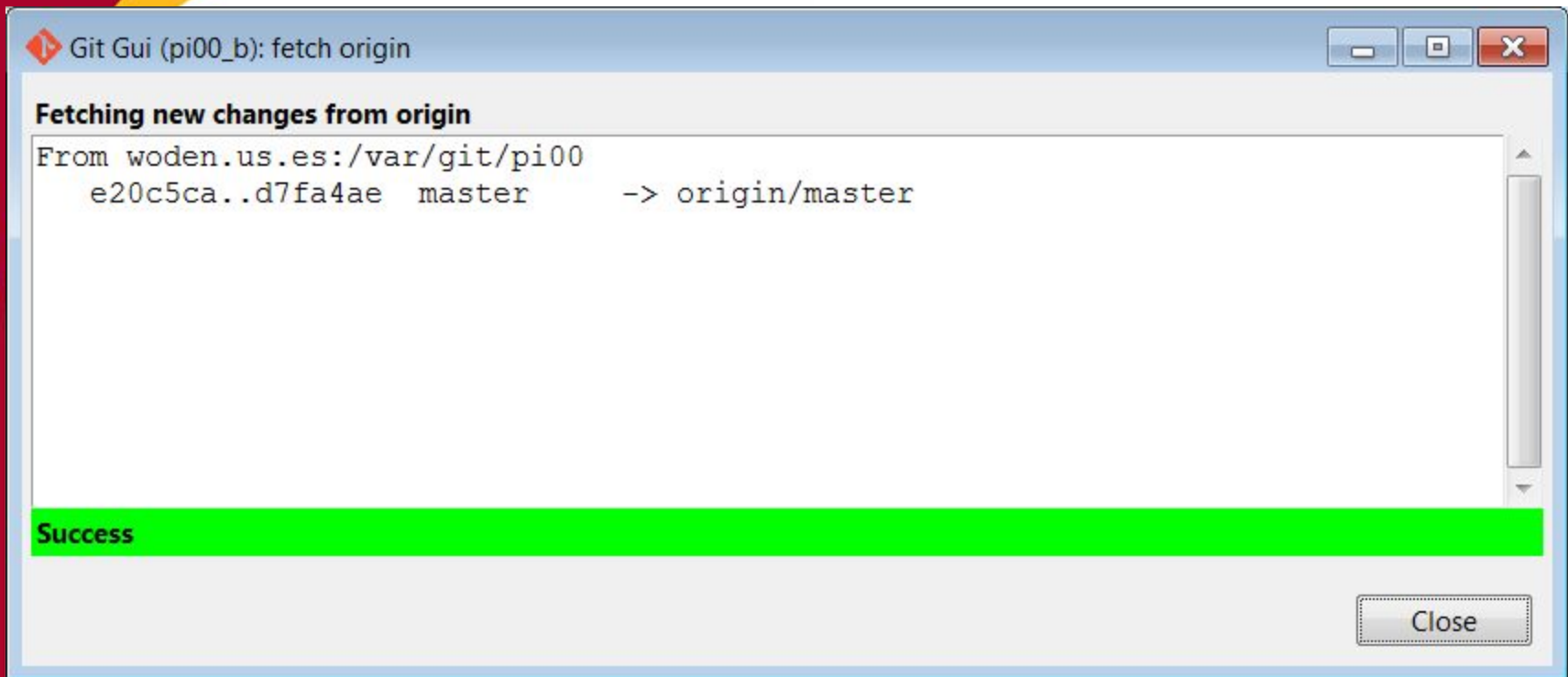
- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout

Pull: fetch + merge



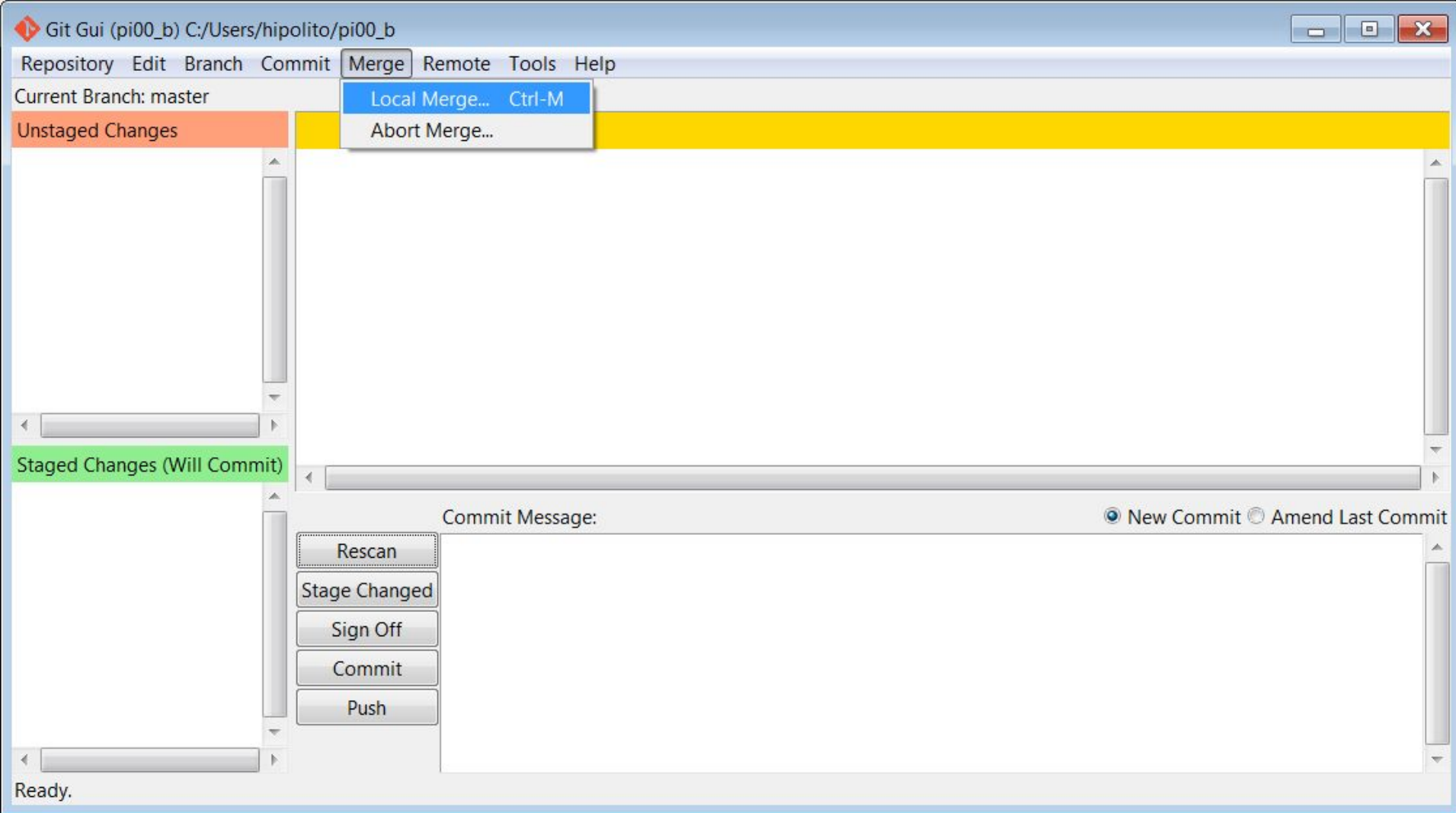
Fetch from origin descarga los cambios nuevos que haya en el repo remoto

Pull: fetch + merge



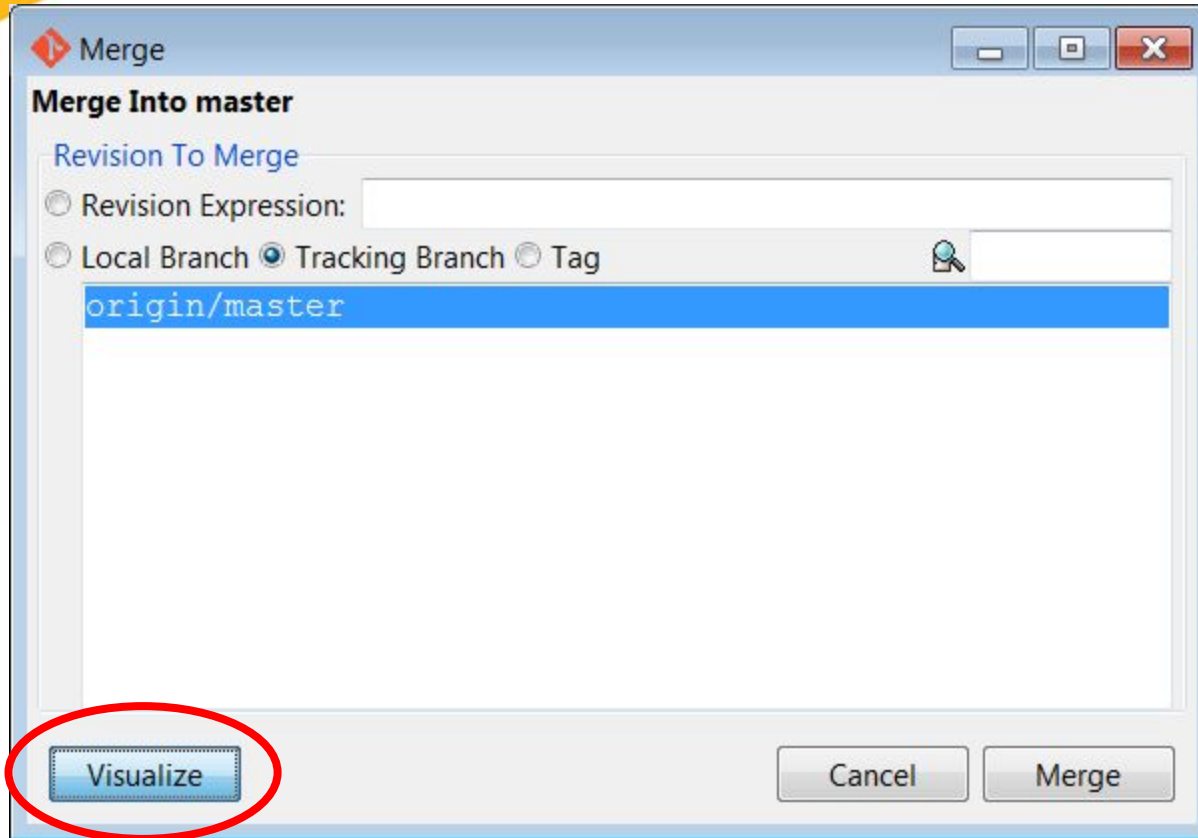
(Nota: si al hacer fetch este cuadro de diálogo no contiene texto, es que no hay cambios nuevos, por lo que no será necesario un merge)

Pull: fetch + merge



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

Pull: fetch + merge



Podemos ver los cambios antes de fusionar

Pull: fetch + merge

gitk pi00_b

Archivo Editar Vista Ayuda

remotes/origin/master Added hipolito.txt
Initial commit

Hipolito Guzman <hipolito@challenger> 2016-03-04 21:56:00
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

Diferencia Versión antigua Versión nueva Líneas de contexto: 3 Ignora c

Parche Árbol

Comentarios
hipolito.txt

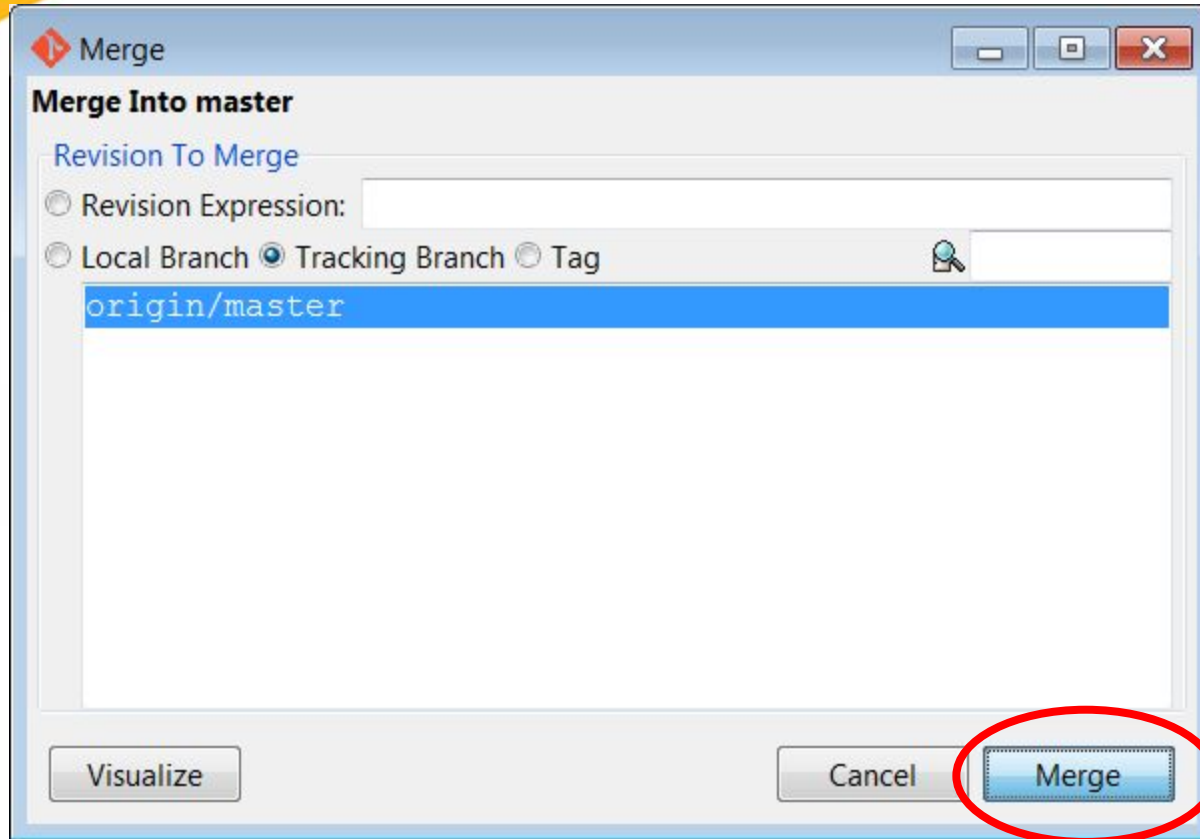
```

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'
    
```

Pull: fetch + merge



Pulsamos 'Merge'

Pull: fetch + merge



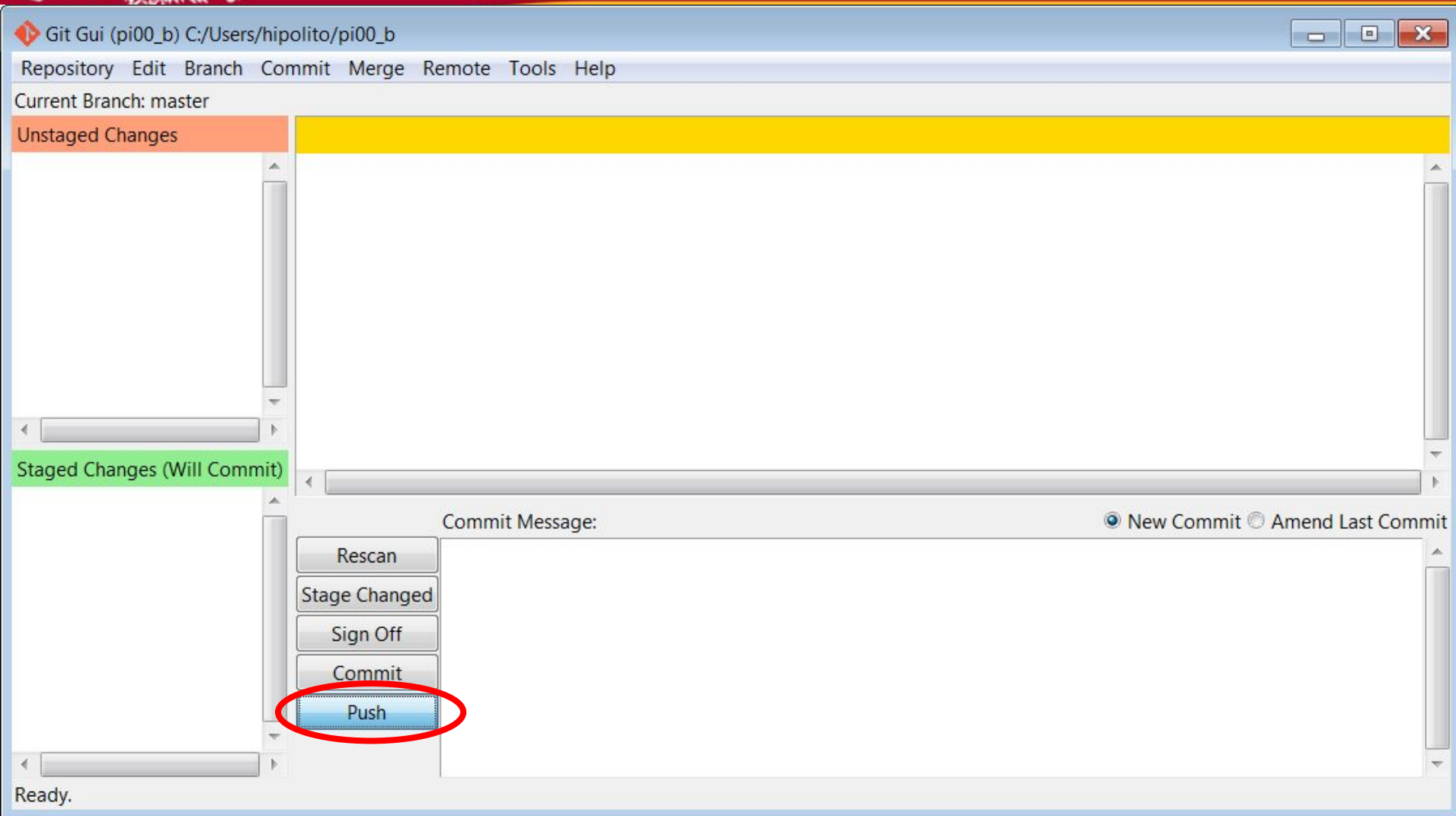
The screenshot shows a window titled "Git Gui (pi00_b): Merge". The main content area displays the following text:

```
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
```

A green bar at the bottom of the window contains the word "Success". A "Close" button is located in the bottom right corner of the window.

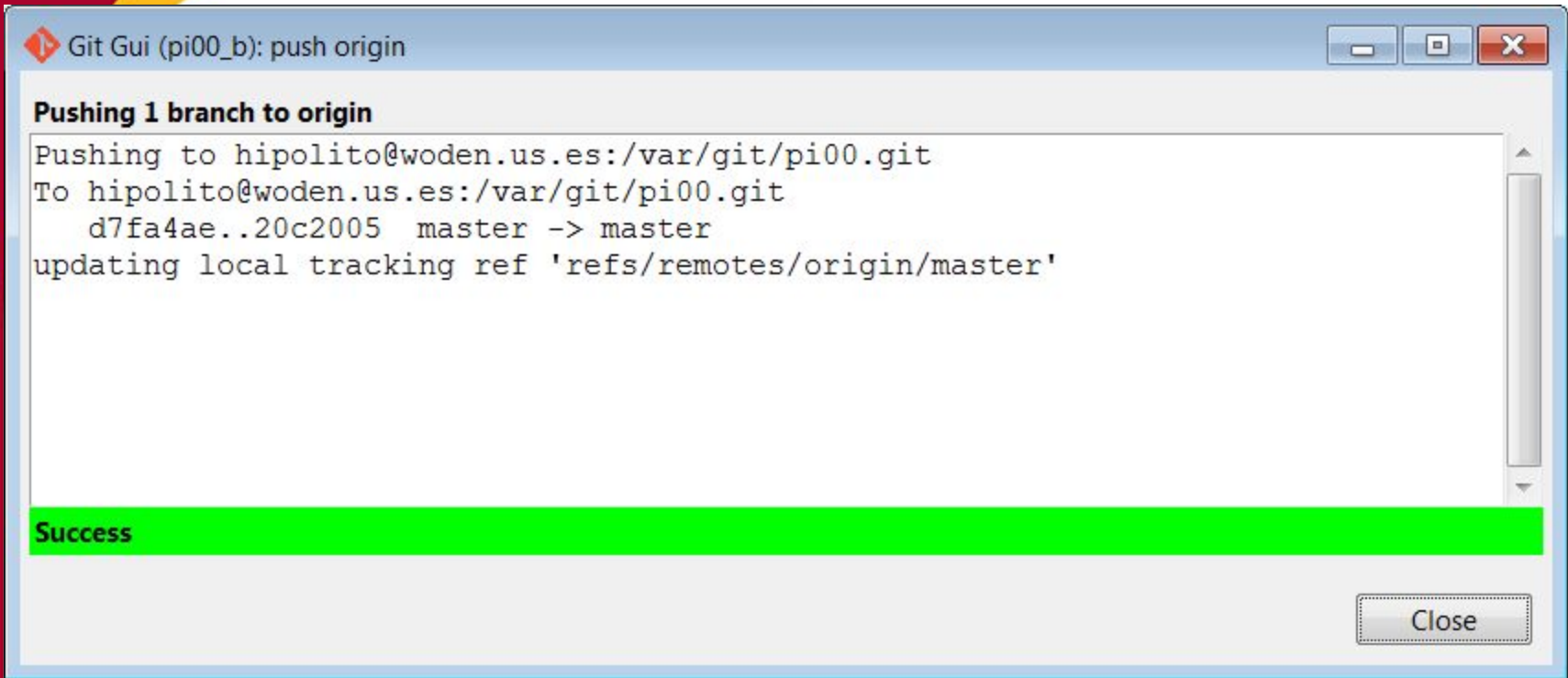
Merge completado con éxito

Push (2)



Tras el merge, ya podemos 'pushear' nuestros cambios

Push (2)



Git Gui (pi00_b): push origin

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

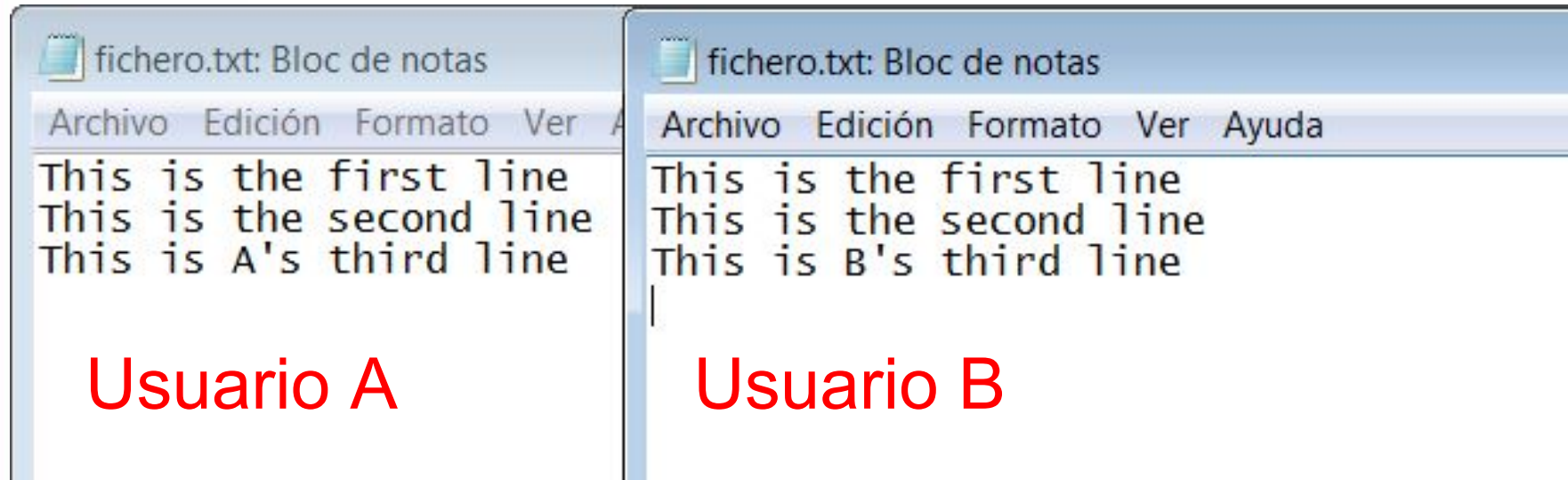
Close

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

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



The image shows two side-by-side Notepad windows, both titled 'fichero.txt: Bloc de notas'. The left window, labeled 'Usuario A', has a menu bar with 'Archivo', 'Edición', 'Formato', and 'Ver'. Its text contains three lines: 'This is the first line', 'This is the second line', and 'This is A's third line'. The right window, labeled 'Usuario B', has a menu bar with 'Archivo', 'Edición', 'Formato', 'Ver', and 'Ayuda'. Its text contains three lines: 'This is the first line', 'This is the second line', and 'This is B's third line'. The third line in the right window is highlighted in yellow, indicating it is the current selection.

Usuario A

Usuario B

A 'pushea' primero y no ve nada raro
B tiene que hacer fetch + merge, pero al hacer merge:

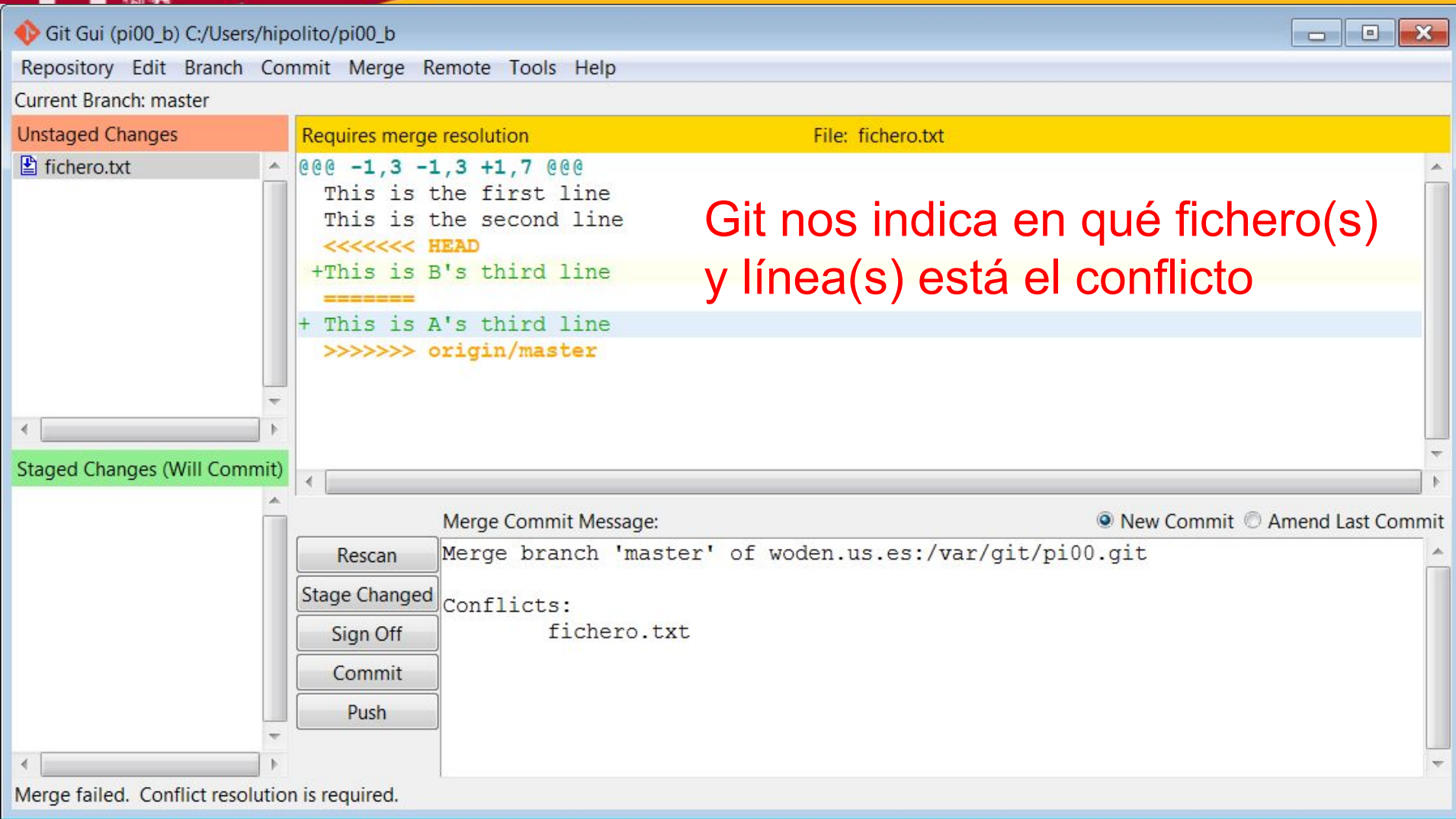


Git Gui (pi00_b): Merge

```
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

Close



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

```
@@@ -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
```

Staged Changes (Will Commit)

Merge Commit Message: New Commit Amend Last Commit

Rescan

Stage Changed

Sign Off

Commit

Push

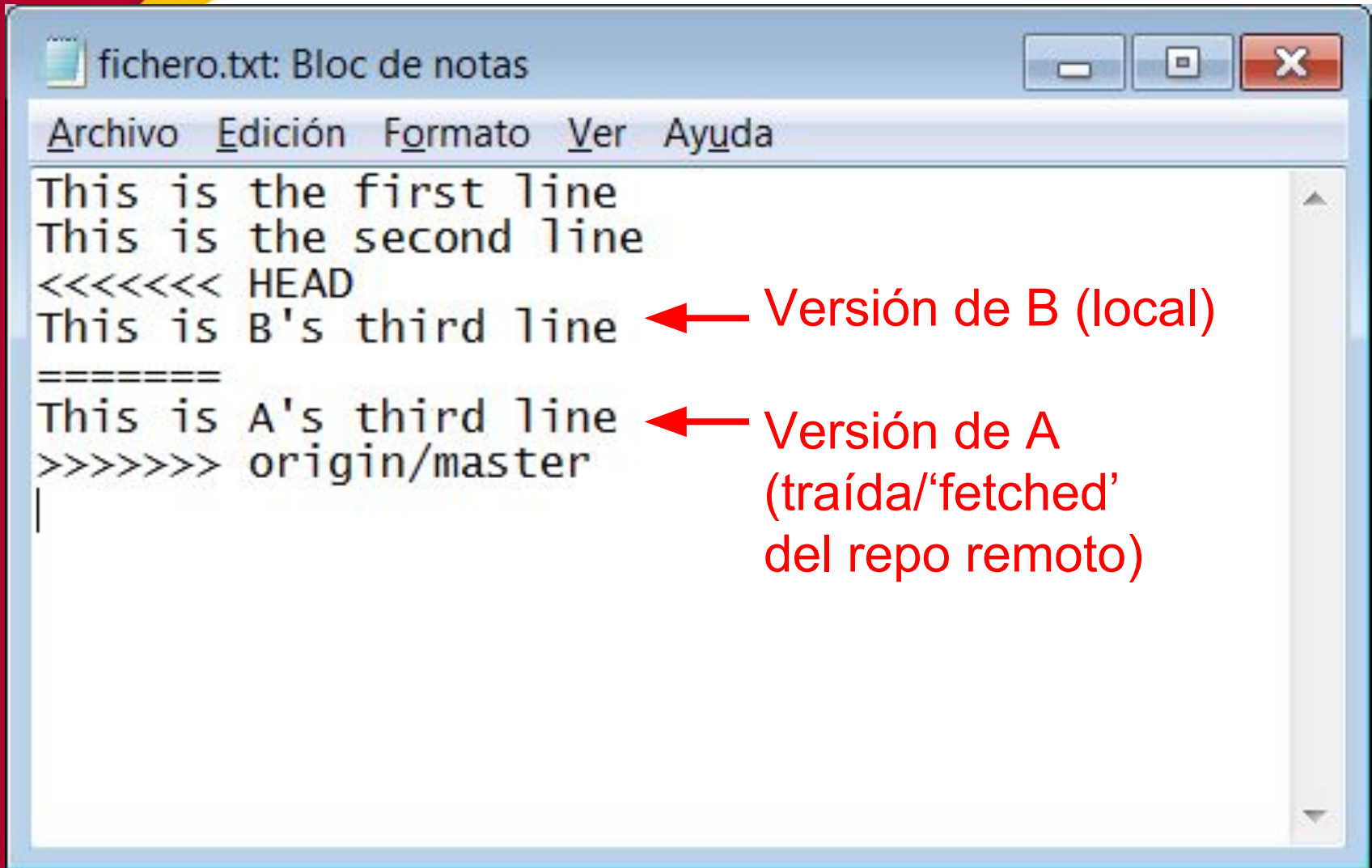
Merge branch 'master' of woden.us.es:/var/git/pi00.git

Conflicts:

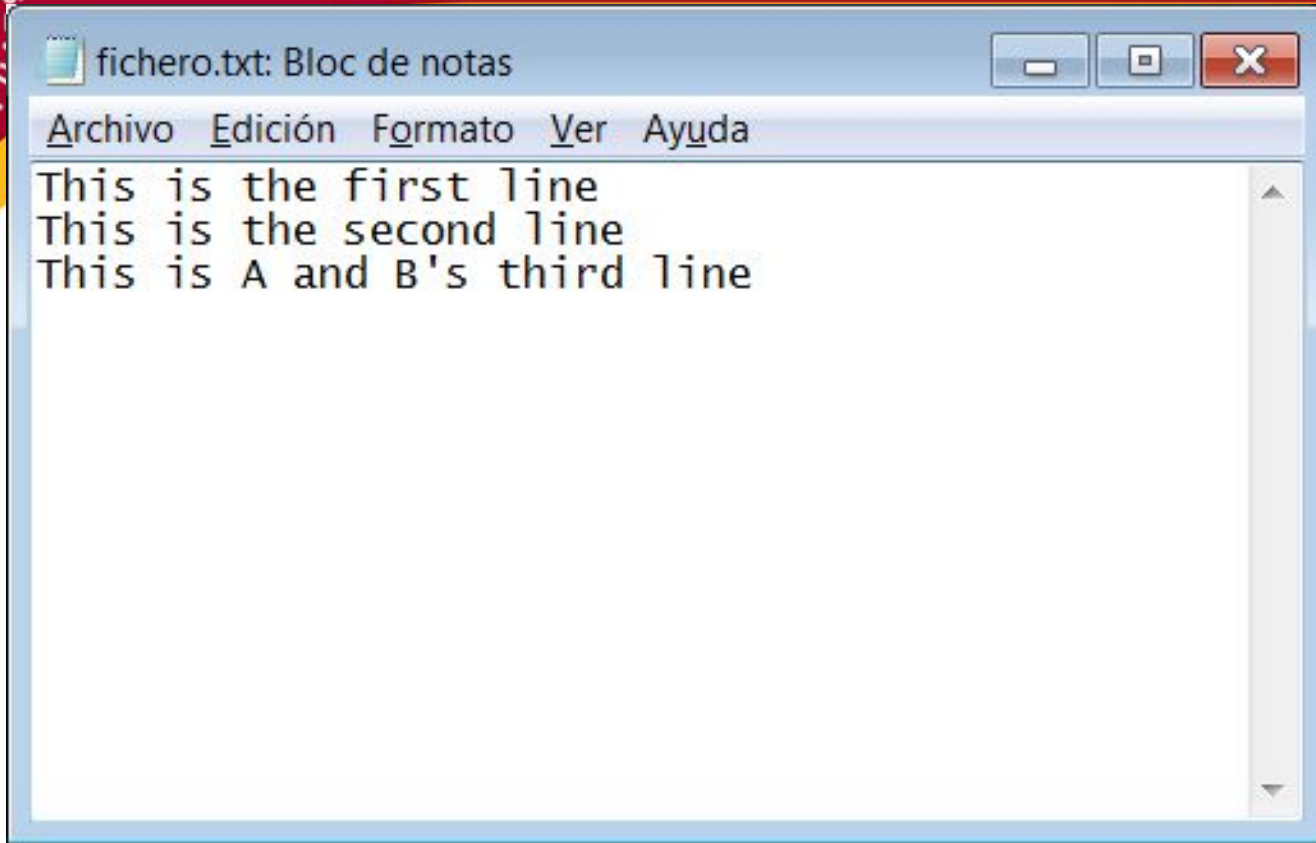
fichero.txt

Merge failed. Conflict resolution is required.

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

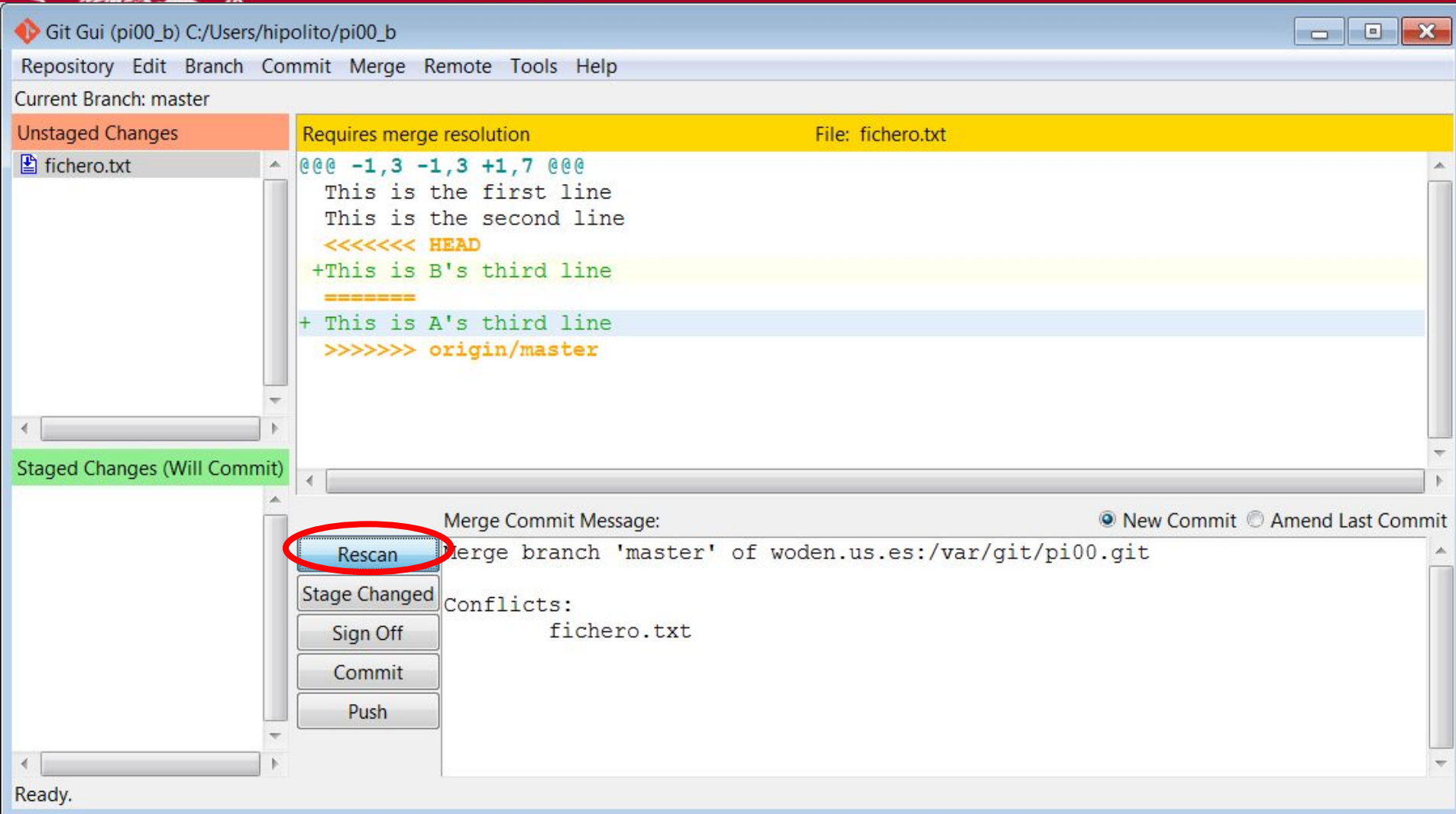


```
fichero.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
This is the first line
This is the second line
<<<<<<< HEAD
This is B's third line
======'
This is A's third line
>>>>>>> origin/master
|
```



Solucionamos el conflicto:

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



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

Repository Edit Branch Commit Merge Remote Tools Help

Current Branch: master

Unstaged Changes

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
  
```

Staged Changes (Will Commit)

Merge Commit Message: New Commit Amend Last Commit

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

Conflicts:

fichero.txt

Rescan

Stage Changed

Sign Off

Commit

Push

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

Requires merge resolution File: fichero.txt

```
@@@ -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
```

Staged Changes (Will Commit)

Merge Commit Message: New Commit Amend Last Commit

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

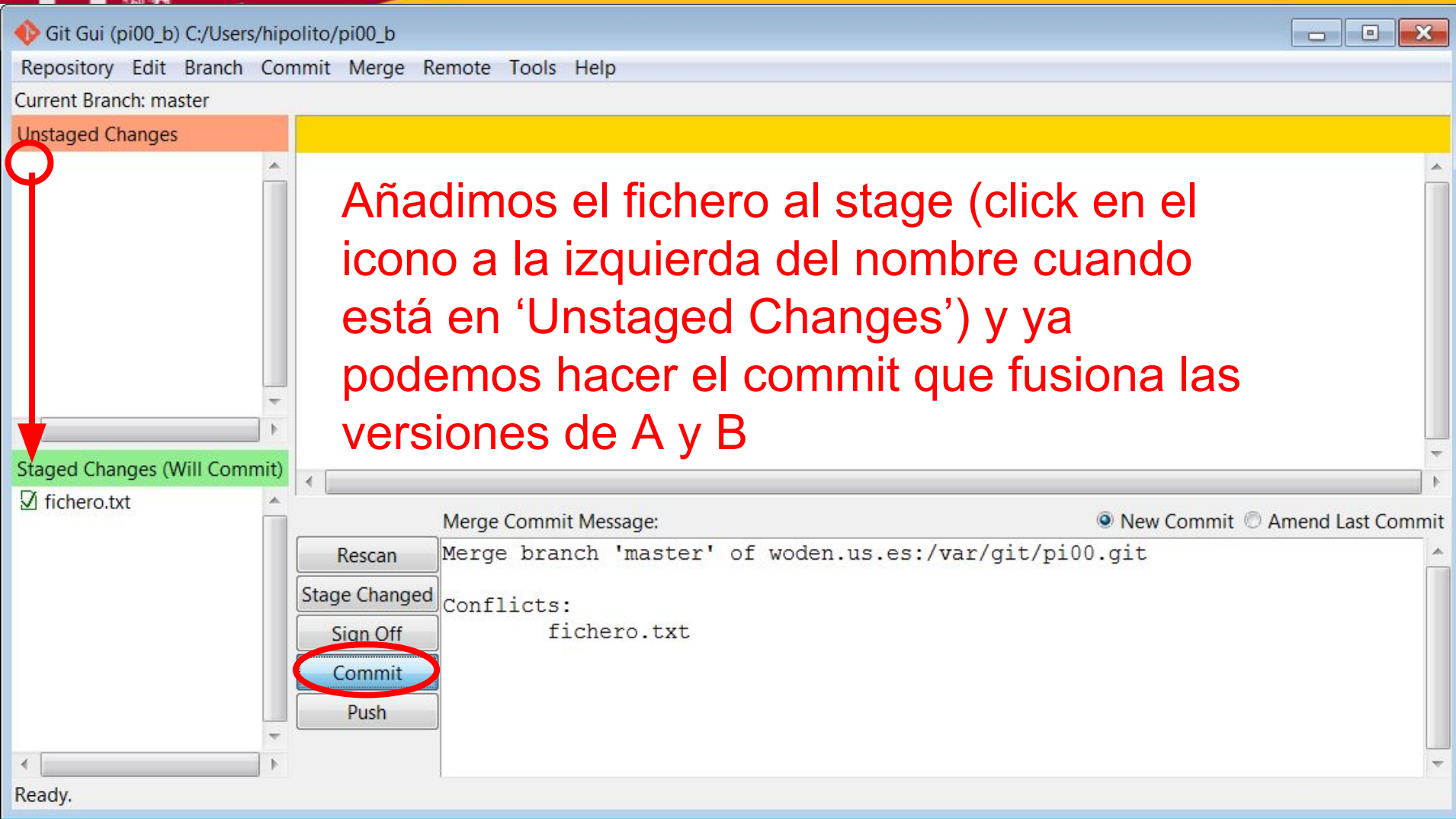
Conflicts:

 fichero.txt

Rescan
Stage Changed
Sign Off
Commit
Push

Ready.

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

Merge Commit Message: New Commit Amend Last Commit

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

Conflicts:

- fichero.txt

Rescan

Stage Changed

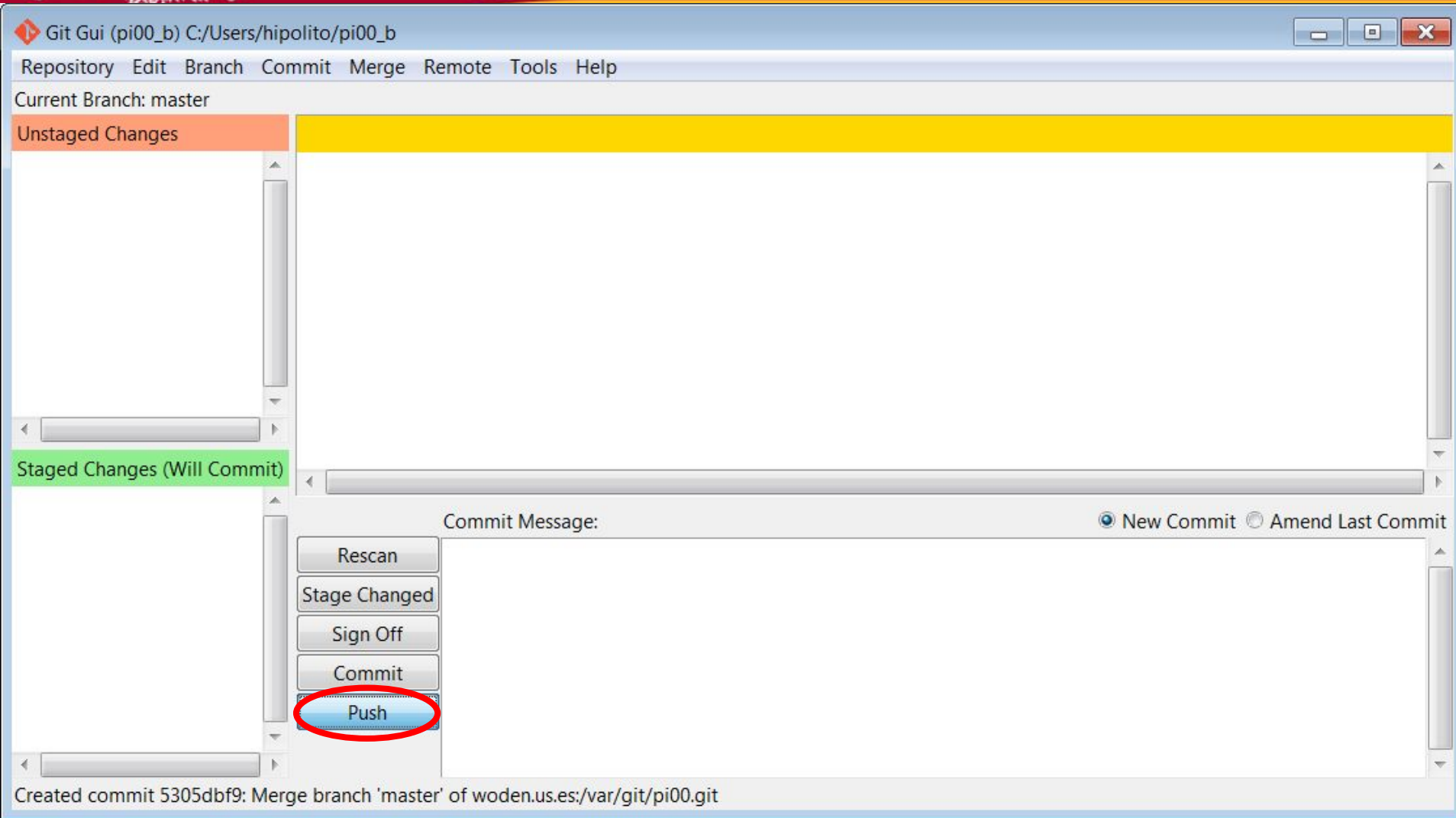
Sign Off

Commit

Push

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



Y ya podemos empujar este nuevo commit al repositorio remoto

Contenido

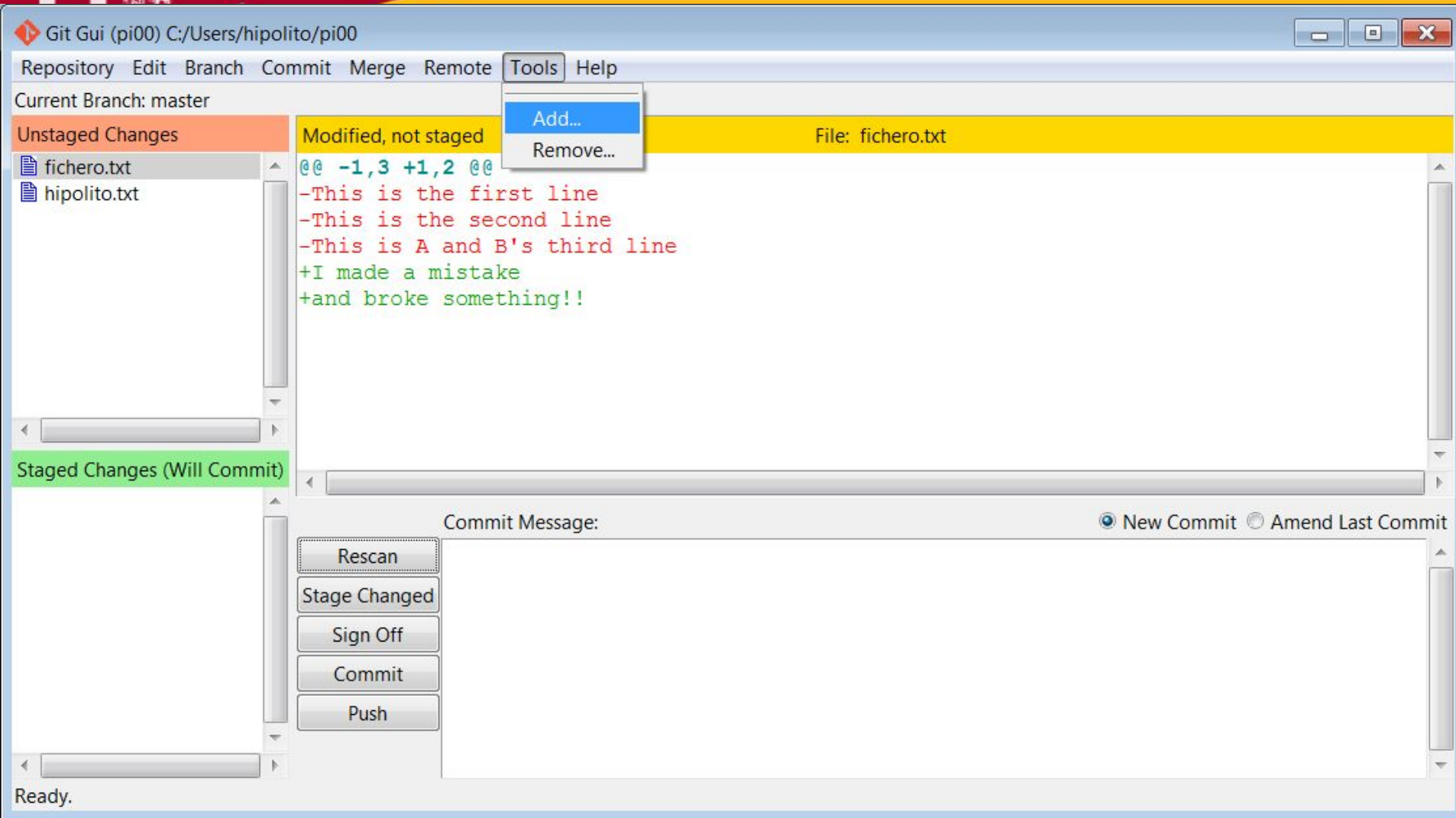
- Áreas de git
- Clonar el repositorio
- Configurar git
- Añadir ficheros al stage
- Creando commits
- Push
- Pull = Fetch from origin + Merge local
- Resolución de conflictos
- Revertir errores usando checkout

Revertir errores usando checkout



The screenshot shows the Git GUI application window. The title bar reads 'Git Gui (pi00) C:/Users/hipolito/pi00'. The menu bar includes 'Repository', 'Edit', 'Branch', 'Commit', 'Merge', 'Remote', 'Tools', and 'Help'. Below the menu bar, it says 'Current Branch: master'. The main area is divided into two sections: 'Unstaged Changes' and 'Staged Changes (Will Commit)'. The 'Unstaged Changes' section is highlighted in orange and shows a list of files: 'fichero.txt' and 'hipolito.txt'. The 'Staged Changes' section is highlighted in green and is currently empty. The 'fichero.txt' file is selected, and its content is displayed in a text area. The text shows three lines that were removed (indicated by red minus signs) and two lines that were added (indicated by green plus signs). The commit message field is empty, and the 'New Commit' radio button is selected. A vertical toolbar on the left side of the commit message field contains buttons for 'Rescan', 'Stage Changed', 'Sign Off', 'Commit', and 'Push'. The status bar at the bottom left says 'Ready.'.

Revertir errores usando checkout



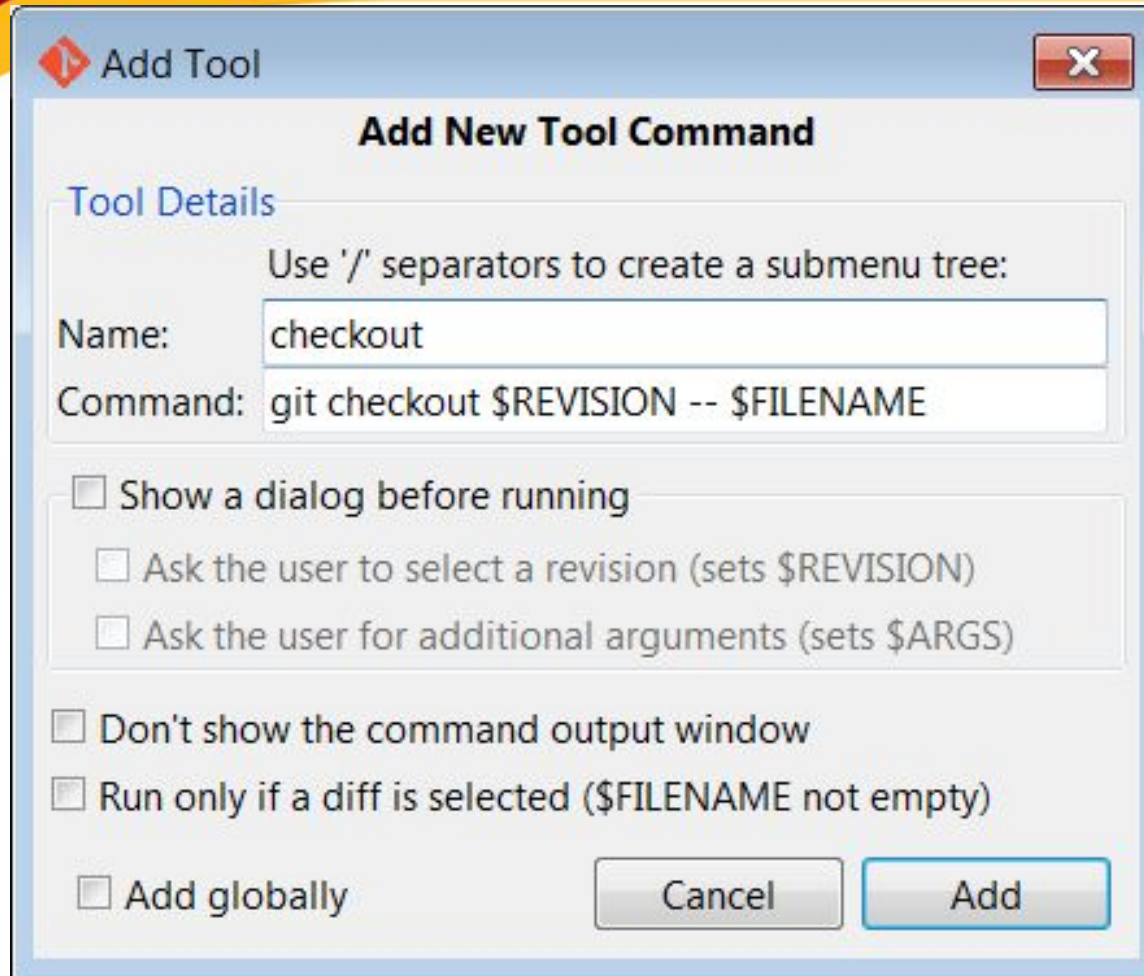
The screenshot shows the Git GUI interface for a repository located at C:/Users/hipolito/pi00. The current branch is master. The interface is divided into several sections:

- Unstaged Changes:** A list on the left shows 'fichero.txt' and 'hipolito.txt'. The main area displays a diff for 'fichero.txt' with the following content:

```
@@ -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):** This section is currently empty.
- Commit Message:** A text area for entering a commit message, with radio buttons for 'New Commit' (selected) and 'Amend Last Commit'.
- Buttons:** A vertical stack of buttons on the left includes 'Rescan', 'Stage Changed', 'Sign Off', 'Commit', and 'Push'.

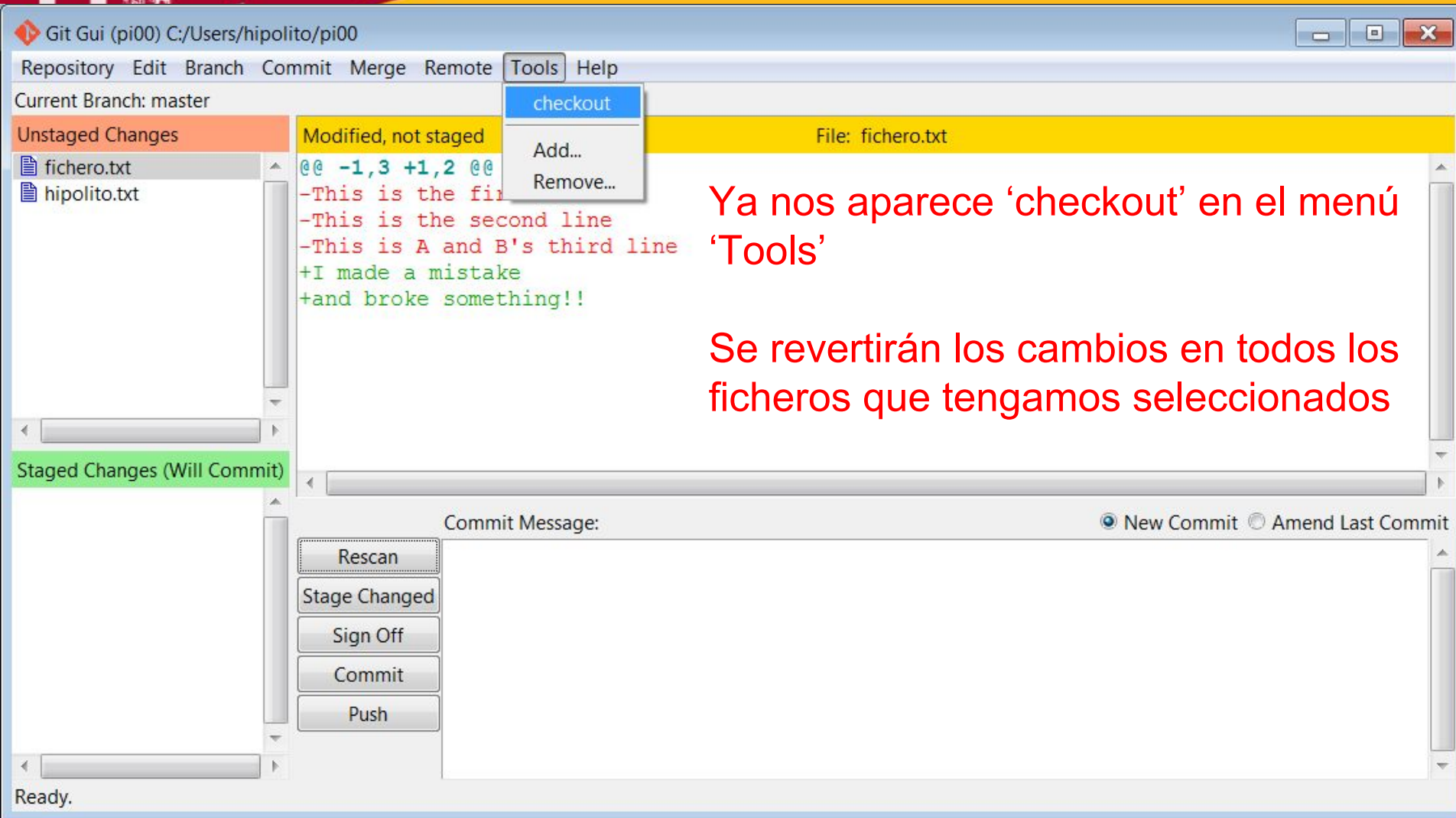
The status bar at the bottom left indicates 'Ready.'.

Revertir errores usando checkout



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

Revertir errores usando checkout



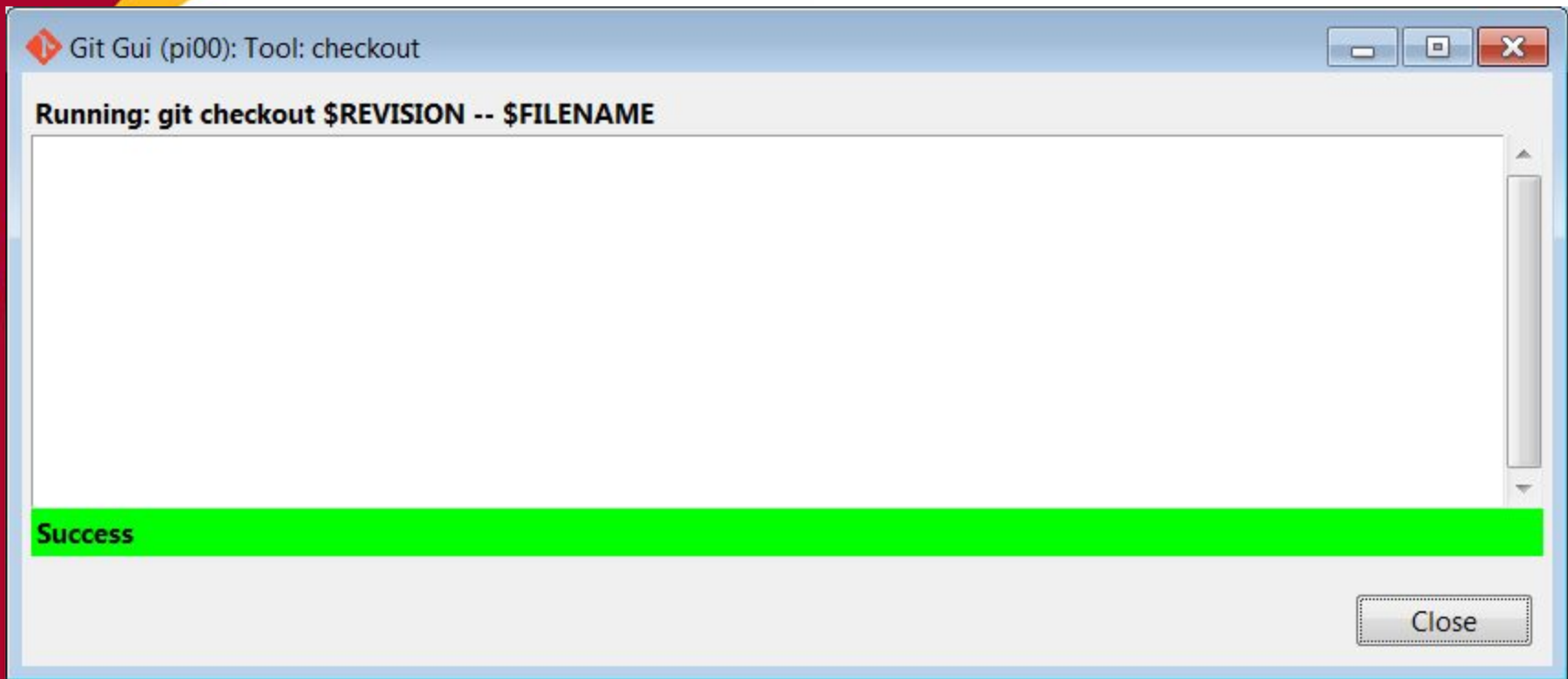
The screenshot shows the Git GUI interface. The 'Tools' menu is open, and the 'checkout' option is highlighted. The main window displays the file 'fichero.txt' with the following content:

```
@@ -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!!
```

Below the file content, there is a 'Commit Message' field with radio buttons for 'New Commit' (selected) and 'Amend Last Commit'. Buttons for 'Rescan', 'Stage Changed', 'Sign Off', 'Commit', and 'Push' are visible on the left side of the commit message area.

Ya nos aparece 'checkout' en el menú 'Tools'

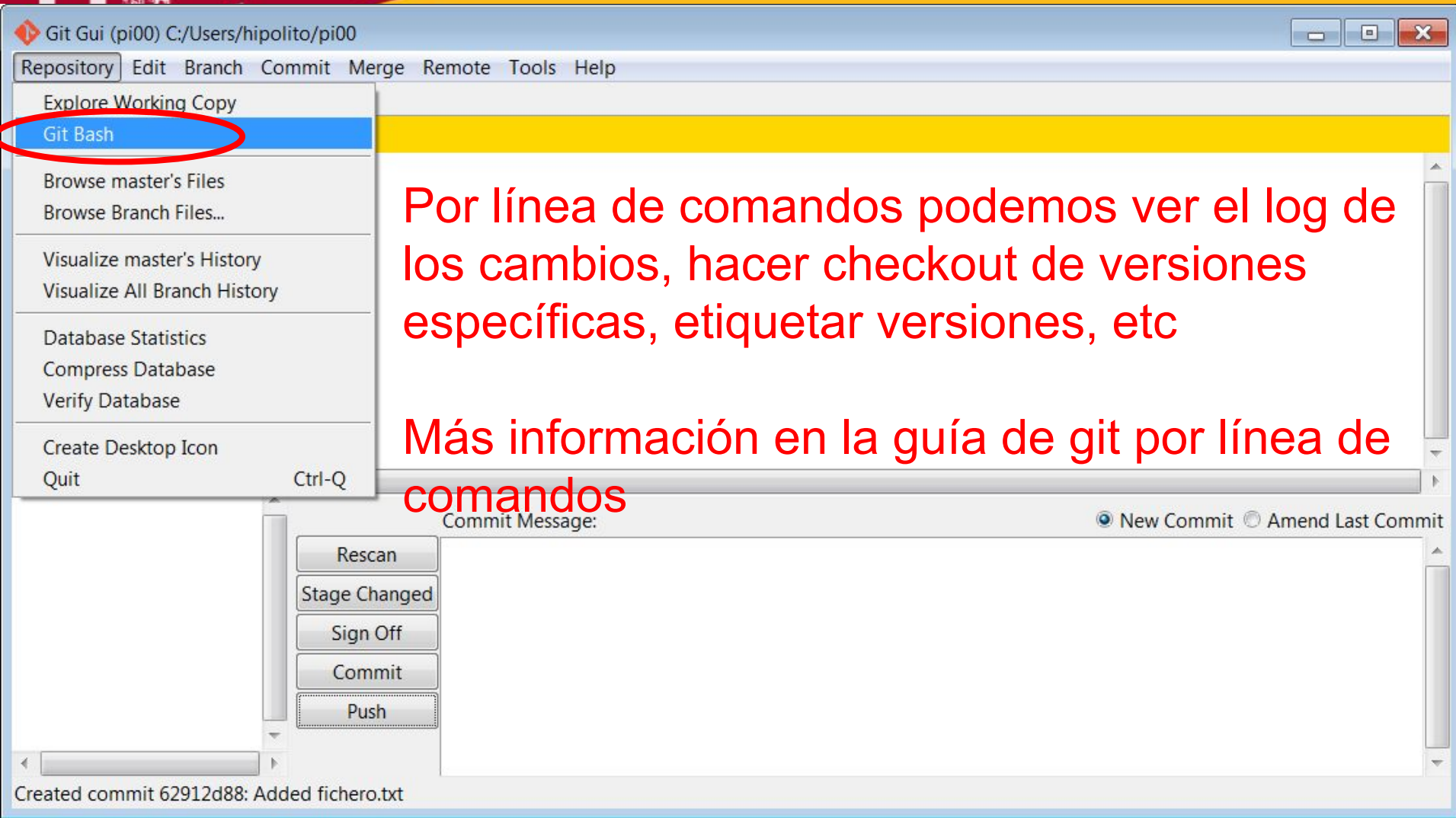
Se revertirán los cambios en todos los ficheros que tengamos seleccionados



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

Para profundizar

Para profundizar



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!

