

Conflictos

Suele suceder cuando trabajamos en equipo o desde más de una computadora sobre el mismo proyecto, que se sobreescriba o modifique una misma porción de código dentro de un archivo.

En estos casos se puede abordar el problema de dos maneras, una más sencilla y otra un poco más trabajosa.

Conflictos

Primera opción:

`git pull` antes de realizar algún cambio

Segunda opción:

Se da principalmente cuando se trabaja simultáneamente o cuando no se realizó un `git pull` antes de modificar el archivo

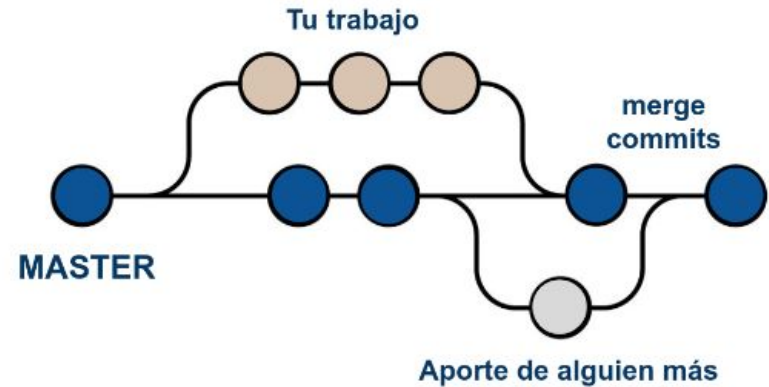
Conflictos

git pull
las partes.

```
<<<<<< HEAD
Modificaciones
hechas
por
una
persona
=====
Modificaciones
hechas
por
otra
persona
>>>>>>
```

— archivo con todos los cambios realizados por todas

Nos vamos por las ramas



El sistema de “branches” o ramas nos permite desarrollar en paralelo diferentes funcionalidades (cada una en una rama distinta), sin mezclar el código, hasta tener una versión final y estable. Luego esos cambios se pueden integrar y obtener el producto completo.

Ramas

Los comandos más generales que se usan para trabajar con ramas son:

- `git branch`
- `git branch nombre`
- `git branch -d nombre`
- `git branch -D nombre`
- `git branch -m nombre2`

Ramas

La manera de poder moverse entre ramas es con el comando `checkout` el cual también puede ser usado para crear nuevas ramas. Los comandos serían:

- `git checkout nombre`
- `git checkout -b nombre`

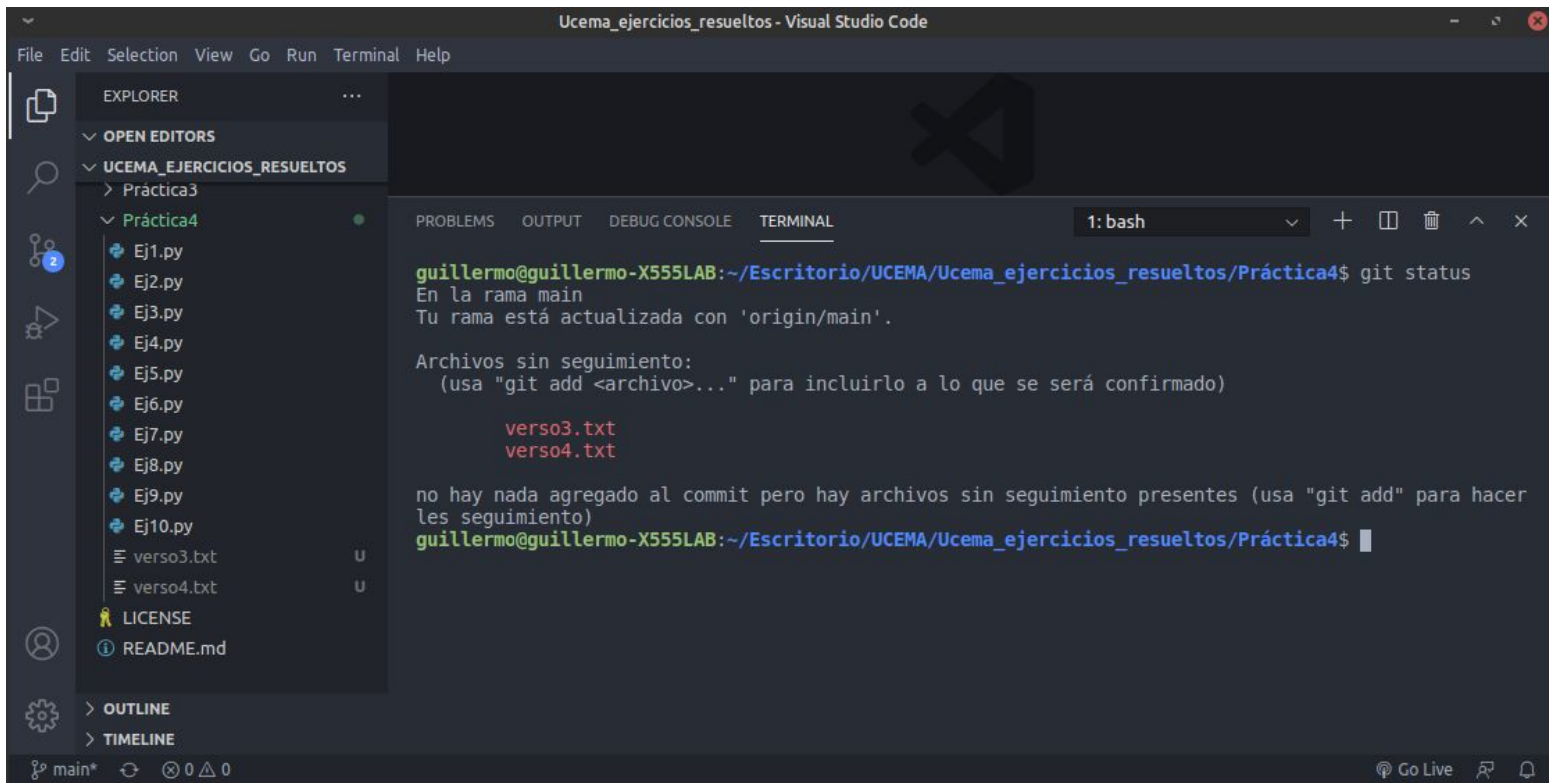
Finalmente cuando queremos juntar las ramas con la principal hay que usar el comando `merge` (desde la rama principal):

- `git merge nombre`

Ignorando

Cuando trabajamos con control de versiones la idea es siempre hacer un seguimiento al código que estamos escribiendo, sin embargo puede ser que dentro de la carpeta tengamos archivos que no pertenecen al código y que no quisiéramos (o necesitamos) hacerle un seguimiento (por ejemplo, archivos de texto que se generen al ejecutar el programa, o que usamos como prueba). Para esto git tiene una forma de evitar que se le preste atención a estos archivos: `.gitignore`

Ignorando



Ucema_ejercicios_resueltos - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS

UCEMA_EJERCICIOS_RESUELTOS

Práctica3

Práctica4

- Ej1.py
- Ej2.py
- Ej3.py
- Ej4.py
- Ej5.py
- Ej6.py
- Ej7.py
- Ej8.py
- Ej9.py
- Ej10.py
- verso3.txt
- verso4.txt

LICENSE

README.md

OUTLINE

TIMELINE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: bash

```
guillermo@guillermo-X555LAB:~/Escritorio/UCEMA/Ucema_ejercicios_resueltos/Práctica4$ git status
En la rama main
Tu rama está actualizada con 'origin/main'.

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)

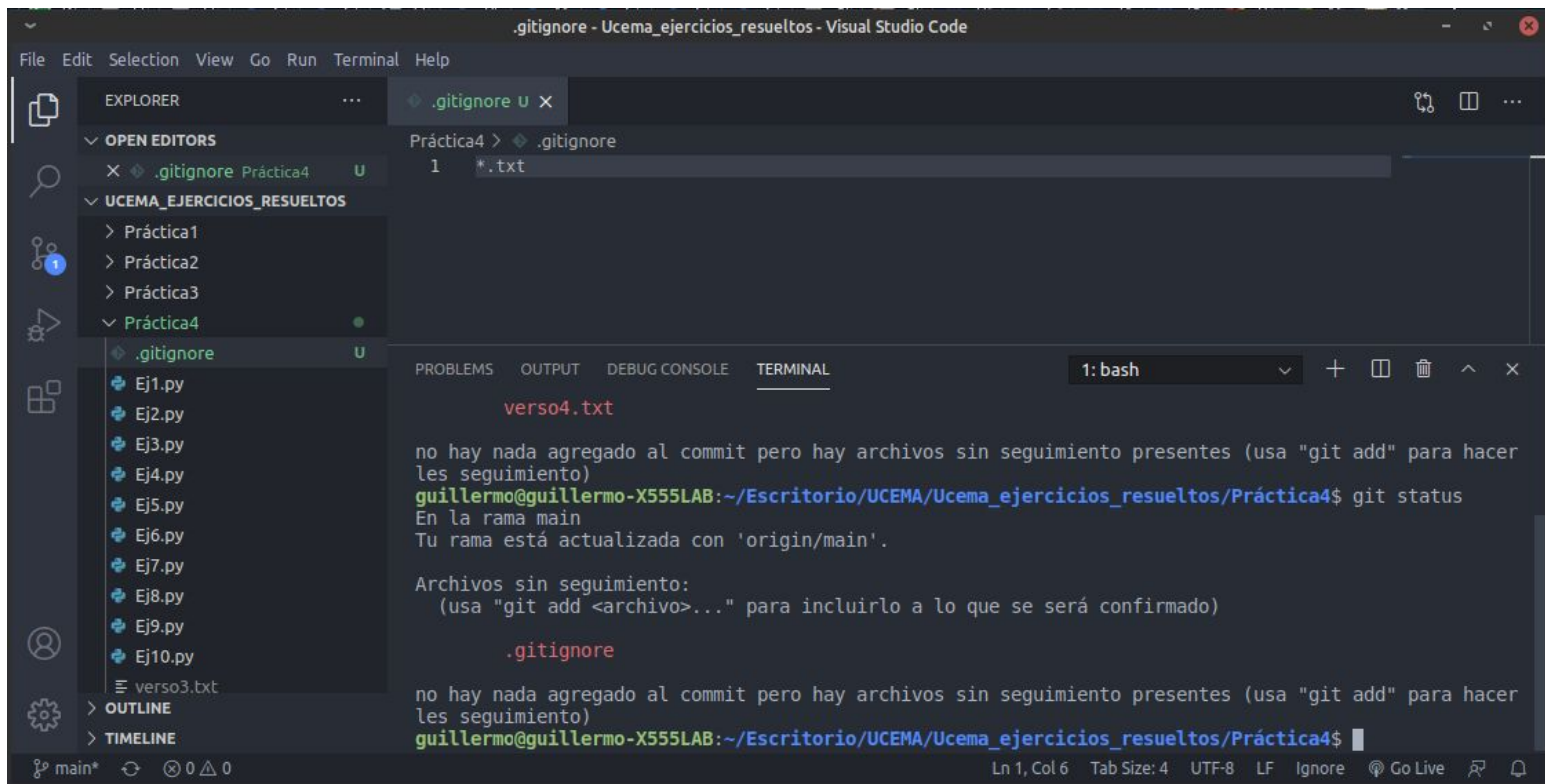
    verso3.txt
    verso4.txt

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacer
les seguimiento)
guillermo@guillermo-X555LAB:~/Escritorio/UCEMA/Ucema_ejercicios_resueltos/Práctica4$
```

main* 0 0 0

Go Live

Ignorando



The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left displays the file structure of a project named 'Ucema_ejercicios_resueltos'. Under the 'Práctica4' folder, a '.gitignore' file is highlighted. The main editor area shows the content of this file, which contains a single line: `*.txt`. The bottom panel contains a terminal window with a bash prompt. The terminal output shows the user running `git status`, which reports that the repository is in the 'main' branch and that there are files not staged for commit. The user then runs `git add *.txt`, and the terminal shows the file being staged. The status bar at the bottom indicates the current file is `.gitignore` and the working directory is `Ln 1, Col 6`.

```
.gitignore - Ucema_ejercicios_resueltos - Visual Studio Code
File Edit Selection View Go Run Terminal Help

EXPLORER
└─ OPEN EDITORS
  └─ .gitignore Práctica4 U
  └─ UCEMA_EJERCICIOS_RESUELTOS
    └─ Práctica1
    └─ Práctica2
    └─ Práctica3
    └─ Práctica4
      └─ .gitignore U
      └─ Ej1.py
      └─ Ej2.py
      └─ Ej3.py
      └─ Ej4.py
      └─ Ej5.py
      └─ Ej6.py
      └─ Ej7.py
      └─ Ej8.py
      └─ Ej9.py
      └─ Ej10.py
      └─ versio3.txt
    └─ OUTLINE
    └─ TIMELINE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: bash

verso4.txt

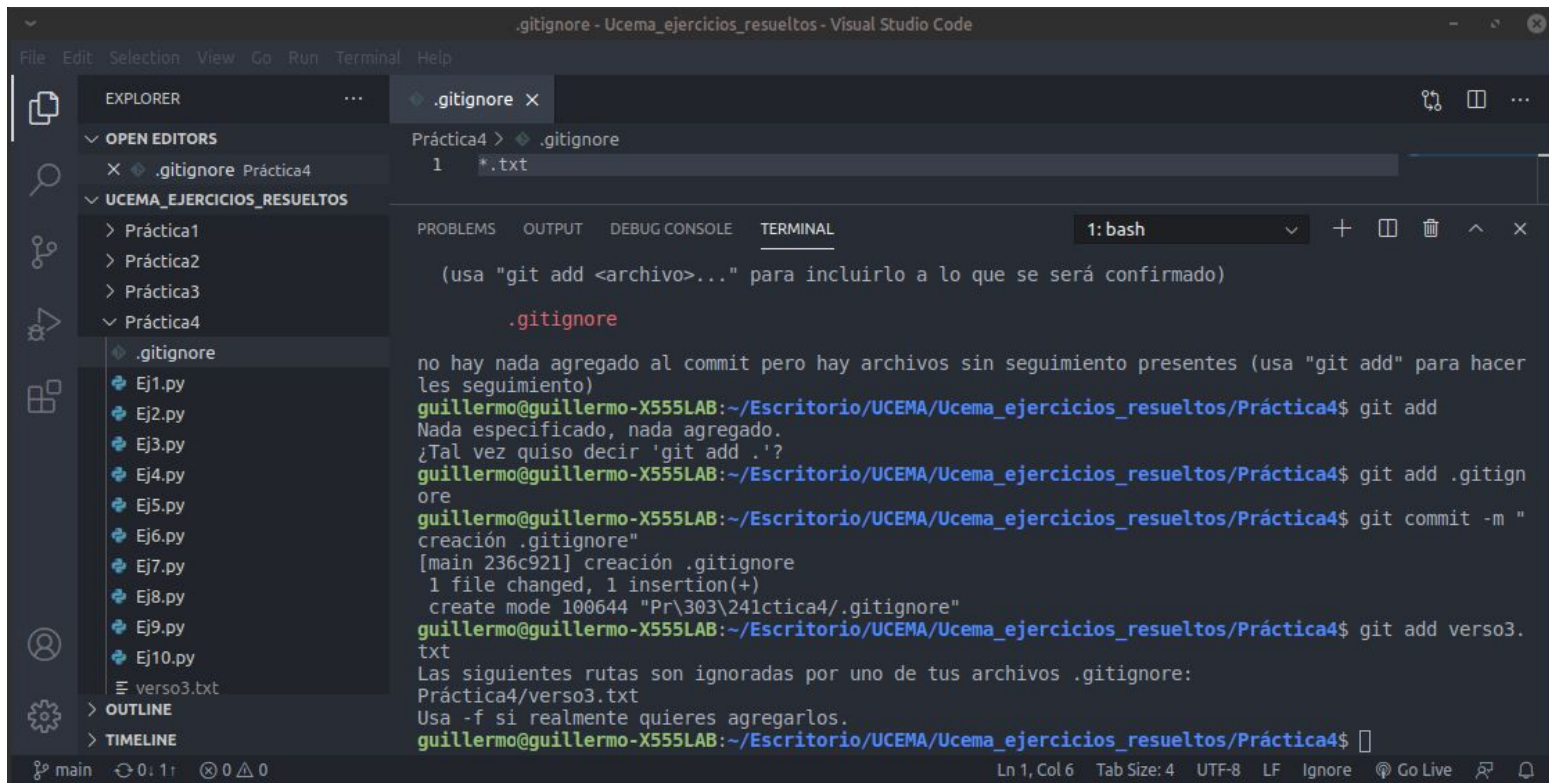
no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacer
les seguimiento)
guillermo@guillermo-X555LAB:~/Escritorio/UCEMA/Ucema_ejercicios_resueltos/Práctica4$ git status
En la rama main
Tu rama está actualizada con 'origin/main'.

Archivos sin seguimiento:
(usa "git add <archivo>..." para incluirlo a lo que se será confirmado)

.gitignore

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacer
les seguimiento)
guillermo@guillermo-X555LAB:~/Escritorio/UCEMA/Ucema_ejercicios_resueltos/Práctica4$
```

Ignorando



The screenshot shows the Visual Studio Code interface with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure with a directory named 'Práctica4' containing a '.gitignore' file and several Python files (Ej1.py to Ej10.py). The terminal shows the following commands and output:

```
Práctica4 > .gitignore
1 *.txt

(usa "git add <archivo>..." para incluirlo a lo que se será confirmado)

.gitignore

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacer
les seguimiento)
guillermo@guillermo-X555LAB:~/Escritorio/UCEMA/Ucema_ejercicios_resueltos/Práctica4$ git add
Nada especificado, nada agregado.
¿Tal vez quiso decir 'git add .' ?
guillermo@guillermo-X555LAB:~/Escritorio/UCEMA/Ucema_ejercicios_resueltos/Práctica4$ git add .gitign
ore
guillermo@guillermo-X555LAB:~/Escritorio/UCEMA/Ucema_ejercicios_resueltos/Práctica4$ git commit -m "
creación .gitignore"
[main 236c921] creación .gitignore
1 file changed, 1 insertion(+)
create mode 100644 "Pr\303\241ctica4/.gitignore"
guillermo@guillermo-X555LAB:~/Escritorio/UCEMA/Ucema_ejercicios_resueltos/Práctica4$ git add verso3.
txt
Las siguientes rutas son ignoradas por uno de tus archivos .gitignore:
Práctica4/verso3.txt
Usa -f si realmente quieres agregarlos.
guillermo@guillermo-X555LAB:~/Escritorio/UCEMA/Ucema_ejercicios_resueltos/Práctica4$
```