Ejercicios de comandos 7

Resolución de conflictos entre el repositorio local y el remoto

Objetivos:

- Resolución de conflictos.

Pre-requisitos:

- Disponer de un repositorio local con archivos, vinculado a uno remoto, y ambos sincronizados (sugerencia: partir de un `git clone` del repositorio remoto).
- El nombre que se usará para el repositorio es `repo_dummy`.

Comandos Git utilizados:

- `git status`
- `git pull`
- `git push`

Comandos Linux utilizados:

- `cd`
- `less`
- `vi`

1) Entrar en el directorio de trabajo del repositorio llamado repo_dummy.

```
cd /home/alumno/Documentos/gitRepos/repo_dummy
```

2) Comprobar el estado del repositorio.

```
git status
```

El resultado indica que el directorio de trabajo está limpio y que no hay archivos pendientes de ser confirmados:

```
On branch main
nothing to commit, working directory clean
```

3) Probar a hacer un `git push`.

```
git push origin main
```

Tras solicitar autenticación, el resultado indica que los repositorios local y remoto están sincronizados (`Everything up-to-date`):

```
Username for 'https://github.com': el.usuario
Password for 'https://el.usuario@github.com':
Everything up-to-date
```

4) Probar también a hacer un `git pull`.

```
git pull origin master
```

El resultado indica que los repositorios local y remoto están sincronizados (`Already up-to-date`):

```
From https://github.com/el.usuario/dummy_repo_remo
 * branch            main     -> FETCH_HEAD
Already up-to-date.
```

5) Modificar en el repositorio local un archivo `arch_01`, guardar las modificaciones, añadirlo al SA y realizar un *commit*.

```
nano arch_01
git add arch_01
commit –m "Modificación en local del archivo arch_01"
```

El resultado del *commit* es:

```
[main 52cc441] Modificado en local el archivo arch_01
 1 file changed, 1 insertion(+)
```

6) Acceder GitHub y a modificar en el repositorio remoto el mismo archivo `arch_01` y guardar las modificaciones (se hará un *commit*), al que se etiquetará como "`Modificación en remoto el archivo arch_01`"). Esto simula que el archivo hubiera sido modificado por otro usuario.

7) Probar de nuevo a hacer un `git push`.

```
git push origin main
```

Tras solicitar autenticación, el resultado indica que se ha rechazado la operación porque los repositorios local y remoto no están sincronizados, en concreto, el repositorio remoto contiene trabajo que no existe en el local. La recomendación es integrar primero en local los cambios remotos (con un `git pull`) y reintentar el `git push`.

```
Username for 'https://github.com': <usuario>
Password for 'https://el.usuario@github.com':
To https://github.com/el.usuario/dummy_repo_remo.git
 ! [rejected]        master -> main (fetch first)
error: failed to push some refs to
'https://github.com/<usuario>/dummy_repo_remo.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.
```

8) Se trata de seguir la recomendación y se hace un `git pull`.

```
git pull origin main
```

El resultado indica que se ha rechazado la operación porque los repositorios local y remoto no están sincronizados, en concreto, se intentó hacer un fusionado automático con el archivo arch_01 que falló debido a un conflicto. La recomendación es resolver manualmente los conflictos editando el archivo y luego hacer un `git commit`.

```
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-
reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/el.usuario/dummy_repo_remo
 * branch            main     -> FETCH_HEAD
```

```
   d393b18..e99dc8b  main     -> origin/master
Auto-merging arch_01
CONFLICT (content): Merge conflict in arch_01
Automatic merge failed; fix conflicts and then commit
the result.
```

9) Se edita el archivo arch_01.

```
nano arch_01
```

… y se encuentra el siguiente contenido:

```
<<<<<<< HEAD
Modificado en local el archivo arch_01
=======
Modificado en remoto el archivo arch_01
>>>>>>> e99dc8bab67b5a999b1dc9dc4c7be1cf01cd0b4d
```

Una primera sección (entre <<< y ===) muestra el contenido del archivo en su versión local. La segunda sección (entre === y >>> `<id_commit_remoto>`) muestra el contenido del archivo en su versión remota.
Hay que editar el archivo y dejarlo como se quiera en su versión final, eliminando todos los delimitadores (<<<, === y >>> `<id_commit_remoto>`). Después grabar.

```
nano arch_01
```

10) Añadir el archivo al SA y realizar un *commit*.

```
git add arch_01
commit –m "Resolución del conflicto en el archivo
arch_01"
```

El resultado del *commit* es:

```
[main 89a58fc] Resolución del conflicto en el archivo
arch_01
```

11) Probar de nuevo a hacer un `git push`.

```
git push origin master
```

Tras solicitar autenticación, el resultado indica que se realizó correctamente el `git push`.
Como consecuencia, los repositorios local y remoto vuelven a estar sincronizados.

```
Username for 'https://github.com': <usuario>
Password for 'https://el.usuario@github.com':
Counting objects: 10, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 750 bytes | 0 bytes/s,
done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/<usuario>/dummy_repo_remo.git
   e99dc8b..89a58fc  main -> main
```

12) Verificar en el repositorio remoto en GitHub que las modificaciones e actualizaron y que el identificador del último *commit* en local coincide con el identificador del último *commit* en remoto.

13) Volver a ejecutar un `git status`.

```
On branch main
nothing to commit, working directory clean
```