

Universidad Nacional del Centro de la Provincia de Buenos Aires
Facultad de Ciencias Exactas
Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas
Tecnología de la Información en las Organizaciones

Trabajo Práctico Especial Grupo 27

GIT APLICADO A QUENTIN TARANTINO

Grupo 27

Integrantes:

Bertoldi, Juan Ignacio.
Burlando, Melisa.
Casero, Lourdes Evangelina.
Dachille, Alejandro.
Iglesias Coqueti, Joaquín.
Milanesi, Pedro.
Olaechea, Miguel Ángel.
Ortiz Balbuena, Rosa Elizabeth.
Pironio, Mariano.

Profesores:

Alonso, Manuel.
González Cisaro, Sandra.
Montero, Norma.
Rodríguez, Florencia.

Link a tablero de Trello: <https://trello.com/b/5jSNucbE/tpe-grupo-27>

Link a repositorio GIT: https://github.com/exajoakinic/tpe_grupo_27

Contenido

I. Introducción	3
I.I. Objetivos	3
I.II. Motivación	3
I.III. Problema a abordar	4
II. Asignación de tareas en Trello:	5
III. Comenzando con GitHub	6
III.I. ¿Qué es GitHub?	6
III.II. Creación del repositorio remoto	6
III.III. Inicialización del proyecto	8
III.IV. Secuencias:	9
• add-commit	9
• Secuencia <i>pull-push</i>	9
• Conflictos al realizar un pull	10
• Eliminar y mover archivos del repositorio:	11
• Renombrar archivos del repositorio	12
• Deshacer confirmaciones:	12
• Añadir cambios y/o cambiar el mensaje a una confirmación:	13
• Retirar al menos una vez modificaciones del área de preparación.	13
• Crear una rama a nivel local y llevar cambios a la rama principal:	14
• Creación de rama remota, trabajo colaborativo y posterior fusión:	15
• Crear una carpeta e indicar a git que no le haga seguimiento:	16
IV. Conclusión	18
V. Bibliografía	19

I. Introducción

En el presente informe se podrá visualizar el proceso y los pasos seguidos para la creación de un repositorio remoto y la creación de un sitio web sobre las 10 mejores películas de Quentin Tarantino.

Desde la cátedra han propuesto la utilización de Git, que ha sido de utilidad para trabajar sobre el repositorio remoto y Trello como herramienta para la organización, asignación de roles y la distribución de las tareas para lograr un trabajo colaborativo.

A continuación se encuentran expuestos en orden los objetivos, las motivaciones, los pasos y las secuencias elaboradas para cumplir con las consignas propuestas acompañadas por las capturas de pantalla correspondientes a cada paso, de manera que se puedan visualizar de forma clara.

I.I. Objetivos

El objetivo del presente trabajo es poner en práctica los conocimientos que se adquirieron en la materia Tecnología de la Información en las Organizaciones la cual está a cargo de la Facultad de Ciencias Exactas de la UNICEN.

Para ello se utilizaron los siguientes recursos:

- Trabajo colaborativo en equipo.
- Utilización de herramientas tales como Trello, GitHub, Meet, Discord, Whatsapp, Google Docs, etc.

I.II. Motivación

La motivación de éste trabajo surge de poner en práctica de forma colaborativa los conceptos aprendidos durante la materia. También obtener la primera experiencia en el uso de las herramientas Git y Trello. Las cuales serán fundamentales a futuro para el desarrollo de cada miembro del equipo al momento de desenvolverse en un ámbito laboral.

Otro aspecto motivador fue el desafío de trabajar en equipo, poniendo a prueba la capacidad de comunicación, de interactuar con los compañeros y de resolver colaborativamente los problemas que se plantearon a medida que el proyecto se fue desarrollando.

I.III. Problema a abordar

El problema que se eligió para abordar el proyecto fue la creación de un sitio web que mostrara información sobre las 10 mejores películas de Quentin Tarantino.

Cada integrante tenía asignada una película sobre la cuál debía recopilar una cantidad determinada de información, para luego subirla al proyecto común en GitHub. La segunda parte del proyecto fue la creación de una página web con toda esa información que cada integrante del equipo fue subiendo al repositorio remoto.

Los integrantes del equipo, además de trabajar en su repositorio local, interactuaron sobre un mismo repositorio remoto trayendo -con el comando *git pull*- las actualizaciones hechas por los demás integrantes. Para luego trabajar sobre su local e incorporar esas modificaciones al proyecto común.

Cada miembro del grupo tuvo un *rol* asignado, el cuál abarcaba sus tareas individuales para cumplir con los *objetivos* propuestos por la cátedra; y aparte tareas grupales tales como: la elaboración del informe de manera colaborativa, creación de archivos *.jsons* para la posterior generación de página web, recopilación de información e imágenes sobre las películas, etc.

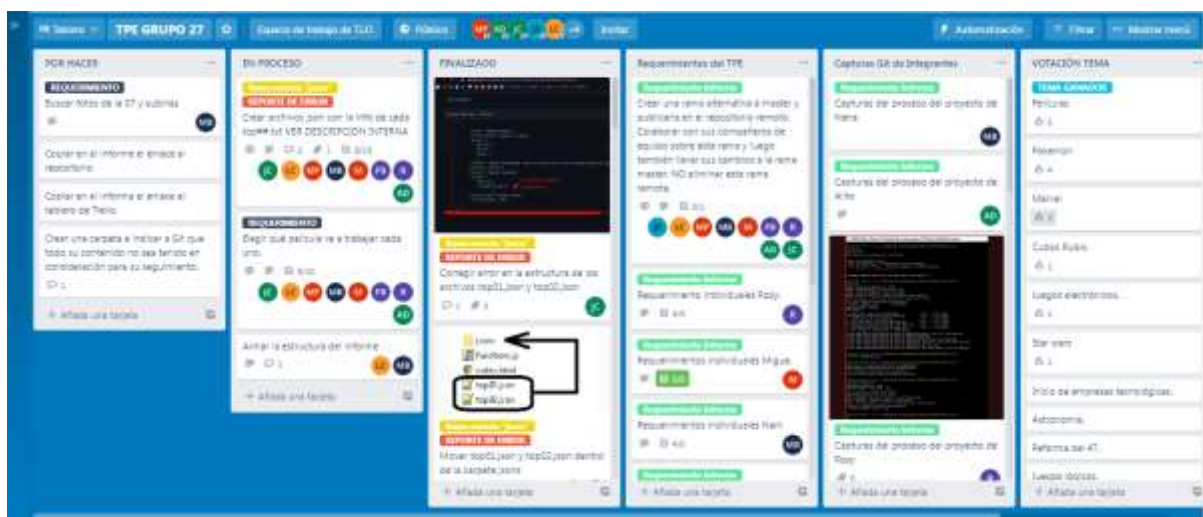
II. Asignación de tareas en Trello:

Trello es una herramienta virtual que permite *organizar, asignar y controlar las tareas* que se realizan dentro de un grupo de trabajo. Por tal motivo, se creó el tablero de Trello <https://trello.com/b/5jSNucbE/tpe-grupo-27> donde se agregaron la totalidad de las tareas a realizar.

Se realizó una “lista de tareas” en la que a cada integrante del grupo se le asignó una tarea, posicionada en la lista POR HACER. Cuando un integrante del equipo comenzaba a trabajar en una tarjeta, la deslizaba a la lista EN PROCESO y una vez resuelta la misma, hacia FINALIZADO.

A continuación nombraremos y explicaremos la utilidad

- 1) POR HACER: se utilizó para listar las tareas por realizar.
- 2) EN PROCESO: cada integrante movió a ésta lista la tarjeta correspondiente a la tarea iniciada.
- 3) FINALIZADO: como el nombre lo indica en esta lista cada colaborador movió la tarjeta referente al trabajo terminado.
- 4) Requerimientos del TPE: aquí se utilizó el recurso de checklist, permitiendo marcar cada tarea y llevar un orden de lo completado de una forma mucho más visible.
- 5) Capturas Git de integrantes: se creó con el fin de llevar un control de los requerimientos del trabajo.
- 6) VOTACIÓN DEL TEMA: este fue el primer apartado que se hizo ya que con esto se logró elegir el tema de nuestro proyecto el cual por votación fue películas.



III. Comenzando con GitHub

III.I. ¿Qué es GitHub?

GitHub es una plataforma la cual es propiedad de Microsoft, ésta sirve para darle alojamiento a repositorios remotos, tiene como función permitir crear repositorios de código y guardarlos de una forma segura. El sistema de control de versiones que se usa lleva el nombre de Git. Lo bueno de esta herramienta es que facilita muchas cosas a los desarrolladores, permitiendo un sistema de control de versiones distribuido.

Además cuenta con un sinfín de opciones, herramientas y funciones, ayudando a que cualquier persona con una pequeña base técnica pueda guardar el historial de su proyecto.

III.II. Creación del repositorio remoto

En primer lugar se creó el repositorio público en GitHub con el nombre “tpe_grupo_27”, generándose el siguiente link:

https://github.com/exajoakinic/tpe_grupo_27

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner *

 exajoakinic ▾

Repository name *

/ tpe_grupo_27 

Great repository names are short and memorable. Need inspiration? How about [curly-succotash?](#)

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

















You choose who can see and commit to this repository.

Esta captura corresponde al alumno Joaquín Iglesias Coqueti.

Para que todos los miembros del equipo puedan trabajar sobre el repositorio fue necesario agregarlos como colaboradores:

Manage access

[Add people](#)

<input type="checkbox"/> Select all	Type ▾
<input type="text" value="Find a collaborator..."/>	
<input type="checkbox"/>  NachoBer JIBOGP • Collaborator	
<input type="checkbox"/>  Melisa Burlando LadyShunney • Collaborator	
<input type="checkbox"/>  Itoakasha Collaborator	
<input type="checkbox"/>  Lourdes0 Collaborator	
<input type="checkbox"/>  Mpironio167 Collaborator	
<input type="checkbox"/>  NarBlit NarBlit • Collaborator	
<input type="checkbox"/>  PMilanesiB Collaborator	
<input type="checkbox"/>  Rosy Rosy333ro • Collaborator	

Esta captura corresponde al alumno Joaquín Iglesias Coqueti.

III.III. Inicialización del proyecto

Para inicializar el proyecto se ejecutaron los comandos:

git init para inicializar el proyecto

git remote add origin https://github.com/exajoakinic/tpe_grupo_27.git para agregar el proyecto remoto.

git branch -d main para cambiar el nombre de la rama local “master” a “main”

```
Usuario@Usuario-PC MINGW64 ~/Desktop/TPE T.I.O
$ git init
Initialized empty Git repository in C:/Users/Usuario/Desktop/TPE T.I.O/.git/
```

```
Usuario@Usuario-PC MINGW64 ~/Desktop/TPE T.I.O (master)
$ git remote add origin https://github.com/exajoakinic/tpe_grupo_27.git
```

Una vez iniciado el proyecto y habiendo hecho el primer commit. El resto de los miembros del equipo sólo tuvo que ejecutar el comando *git clone [url_remota]* para obtener su repositorio local.

```
JIBOGP@JIBOGP MINGW64 ~/Desktop/tp especial tio
$ git clone https://github.com/exajoakinic/tpe_grupo_27
Cloning into 'tpe_grupo_27'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 12 (delta 0), reused 12 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), 1.12 KiB | 2.00 KiB/s, done.
```

Esta captura corresponde al alumno Juan Ignacio Bertoldi.

III.IV. Secuencias:

En esta parte del informe se dará explicación de ciertos procesos que se fueron repitiendo muchas veces a lo largo del proyecto.

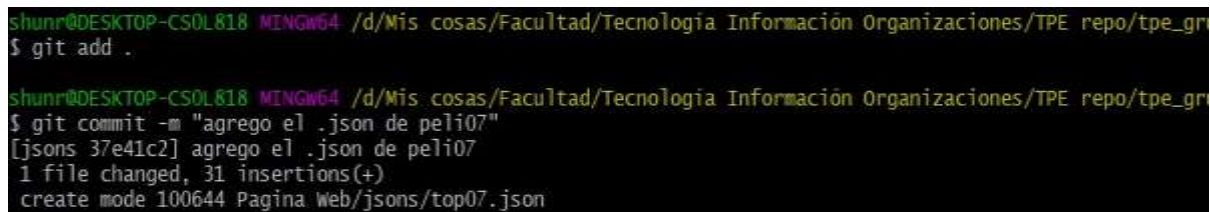
Para no extender innecesariamente el informe, además de explicar cada secuencia requerida, se ilustrará sólo con la captura de un integrante del equipo, entendiendo que todos hemos seguido los mismos pasos.

- **add-commit**

Se utilizó el comando *git add [archivo/carpeta]* para poner los archivos/carpetas en el área de preparación.

Luego se generó una nueva versión local con el comando *git commit -m "comentario representativo"*.

En el caso específico de este trabajo estas secuencias se utilizaron para agregar archivos de texto, imágenes, .json, .html, .css y .js.



```
shunr@DESKTOP-CS0L818 MINGW64 /d/Mis cosas/Facultad/Tecnología Información Organizaciones/TPE repo/tpe_gr
$ git add .
shunr@DESKTOP-CS0L818 MINGW64 /d/Mis cosas/Facultad/Tecnología Información Organizaciones/TPE repo/tpe_gr
$ git commit -m "agrego el .json de peli07"
[jsons 37e41c2] agrego el .json de peli07
1 file changed, 31 insertions(+)
create mode 100644 Pagina Web/jsons/top07.json
```

Esta captura corresponde a Melisa Burlando.

- **Secuencia *pull-push***

Cada vez que se necesitó hacer cambios en el proyecto, se decidió -para evitar inconvenientes- primero ejecutar el comando *git pull* o *git pull [remoto] [local]* que permite actualizar el repositorio local trayendo el último commit del remoto.

Luego se realizaron las correspondientes secuencias *add-commit* para finalmente subir los cambios al repositorio remoto mediante el comando *git push* o *git push [remote] [local]*. (Ver imágenes 2 y 3)

Cabe aclarar que para poder utilizar los comandos *git pull* o *git push* sin especificar remoto y local es necesario haber ejecutado anteriormente con éxito el comando *git push -u [remoto] [local]*. Éste le indica a git que debe recordar esa elección.

Si en el momento de ejecutar el comando push, el repositorio remoto está más actualizado que el local, git no permite hacer esta actualización ya que hay riesgo de que se pueda perder información (Imagen 1). En tal caso, se debe ejecutar nuevamente el comando `git pull [remoto] [local]` y luego reintentar el push.

Esta secuencia de comandos fue utilizada en el trabajo para actualizar tanto el repositorio remoto como el local.

```
70DESKTOP-OL02CNN MINGW64 ~/Desktop/TP T.I.O/tpe_grupo_27/Página Web (main)
$ git push origin main
To https://github.com/exajoakinic/tpe_grupo_27
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/exajoakinic/tpe_grupo_27'
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.
```

Imagen 1. Perteneciente a Miguel Ángel Olaechea.

```
70DESKTOP-OL02CNN MINGW64 ~/Desktop/TP T.I.O/tpe_grupo_27/Página Web (main)
$ git pull
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 15 (delta 7), reused 12 (delta 4), pack-reused 0
Unpacking objects: 100% (15/15), 233.80 KiB | 767.00 KiB/s, done.
From https://github.com/exajoakinic/tpe_grupo_27
 2fa2ae9..19268cf  main    -> origin/main
Merge made by the 'recursive' strategy.
...rama local capturas comiteo con error y corrijo.png | Bin 0 -> 69951 bytes
.../10 merge capturas a main.png | Bin 0 -> 51061 bytes
... por desactualizado, actualizo con pull y pusheo.png | Bin 0 -> 46657 bytes
... borro branch local capturas luego de mergearla.png | Bin 0 -> 37806 bytes
.../13 otro pull ye intento de push.png | Bin 0 -> 60952 bytes
5 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Media/Capturas de Pantalla Pironio/09 creo rama local capturas comiteo con error y corrijo.png
create mode 100644 Media/Capturas de Pantalla Pironio/10 merge capturas a main.png
create mode 100644 Media/Capturas de Pantalla Pironio/11 no podía push por desactualizado, actualizo con pull y pusheo.png
create mode 100644 Media/Capturas de Pantalla Pironio/12 borro branch local capturas luego de mergearla.png
create mode 100644 Media/Capturas de Pantalla Pironio/13 otro pull ye intento de push.png
```

Imagen 2. Perteneciente a Miguel Ángel Olaechea.

```
70DESKTOP-OL02CNN MINGW64 ~/Desktop/TP T.I.O/tpe_grupo_27/Página Web (main)
$ git push origin main
Enumerating objects: 15, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 695 bytes | 347.00 KiB/s, done.
Total 7 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 4 local objects.
To https://github.com/exajoakinic/tpe_grupo_27
 19268cf..ff0ed9d  main -> main
```

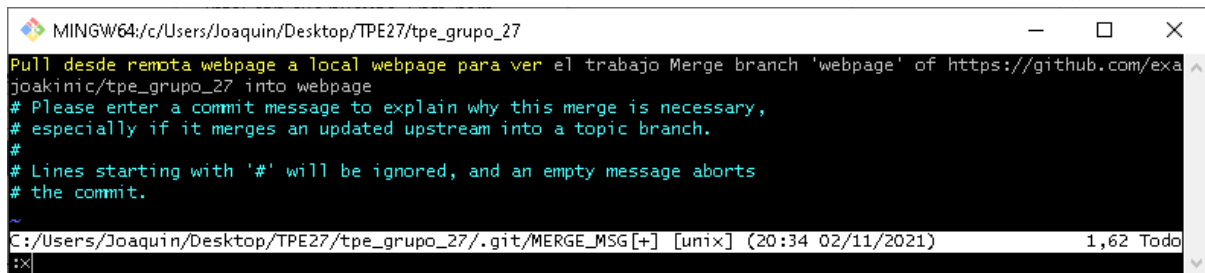
Imagen 3. Perteneciente a Miguel Ángel Olaechea.

- Conflictos al realizar un pull

Cuando se trabaja colaborativamente sobre una misma rama, es muy común que se produzca un conflicto al intentar realizar un pull. Esto se debe generalmente a que en el repositorio local se ha realizado un nuevo commit, y en paralelo otro integrante del equipo ha pusheado un commit al repositorio remoto.

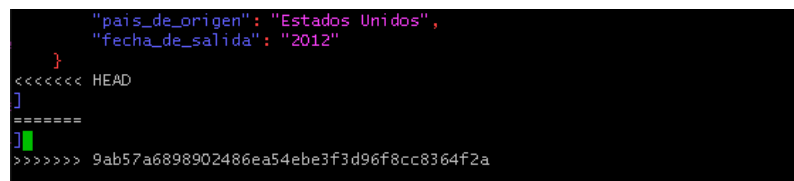
Muchas veces git lo soluciona automáticamente fusionando la rama remota con la rama local. En tal caso, es muy posible aparezca en la consola el editor vim, para que el

usuario local ingrese el mensaje del nuevo commit a generarse, como puede observarse a continuación:



```
MINGW64:/c/Users/Joaquin/Desktop/TPE27/tpe_grupo_27
Pull desde remota webpage a local webpage para ver el trabajo Merge branch 'webpage' of https://github.com/exa
joakinic/tpe_grupo_27 into webpage
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
~
C:/Users/Joaquin/Desktop/TPE27/tpe_grupo_27/.git/MERGE_MSG[+] [unix] (20:34 02/11/2021) 1,62 Todo
:~
```

Cuando GIT no pueda resolver el conflicto, la consola queda en un estado intermedio mostrando *MERGING* junto con el nombre de la rama. Ejecutando *git status* se puede saber cuáles son los archivos afectados que deberán editarse manualmente. GIT nos ayuda mostrándonos en el interior de cada archivo, las diferencias entre nuestro HEAD y el último commit remoto, como puede observarse a continuación:



```

    "pais_de_origen": "Estados Unidos",
    "fecha_de_salida": "2012"
  }
<<<<<< HEAD
]
=====
]
>>>>>> 9ab57a6898902486ea54ebe3f3d96f8cc8364f2a
```

Esta imagen corresponde al alumno Joaquín Iglesias Coqueti.

Finalmente, se procede con la correspondiente secuencia add-commit y pusheo.

- Eliminar y mover archivos del repositorio:

Para eliminar archivos haciendo que git reconozca los cambios se utilizó el comando: *git rm [nombre_del_archivo]*. Para mover archivos el comando fue *git mv [nombre_del_archivo] [destino]*. Luego de esto se continuó haciendo un commit con un mensaje representativo.

Esto puede observarse en la captura que sigue a continuación.



```
Joaquin@NBAtiril MINGW64 ~/Desktop/TPE27/tpe_grupo_27/Pagina Web (main)
$ git rm top01.json
rm 'Pagina Web/top01.json'

Joaquin@NBAtiril MINGW64 ~/Desktop/TPE27/tpe_grupo_27/Pagina Web (main)
$ git mv top02.json ./jsons/

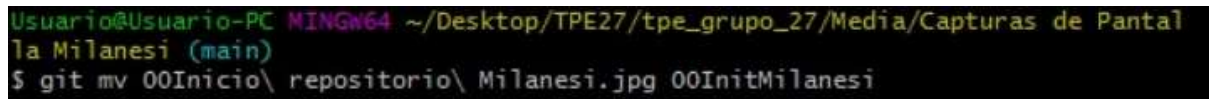
Joaquin@NBAtiril MINGW64 ~/Desktop/TPE27/tpe_grupo_27/Pagina Web (main)
$ git commit -m "Borrado duplicado top01.json y movido top02.json a /jsons/"
[main 9ac0575] Borrado duplicado top01.json y movido top02.json a /jsons/
2 files changed, 23 deletions(-)
rename Pagina Web/{ => jsons}/top02.json (100%)
delete mode 100644 Pagina Web/top01.json
```

Esta imagen corresponde al alumno Joaquín Iglesias Coqueti.

En el caso de esta imagen el integrante del equipo debió eliminar el archivo top01.json debido a que estaba duplicado y mover el archivo top02.json a la carpeta Pagina Web/jsns/.

- Renombrar archivos del repositorio

Para renombrar a un archivo del repositorio se utilizó el comando `git mv [archivo_original] [nuevo_archivo]`, como puede observarse en la siguiente imagen:



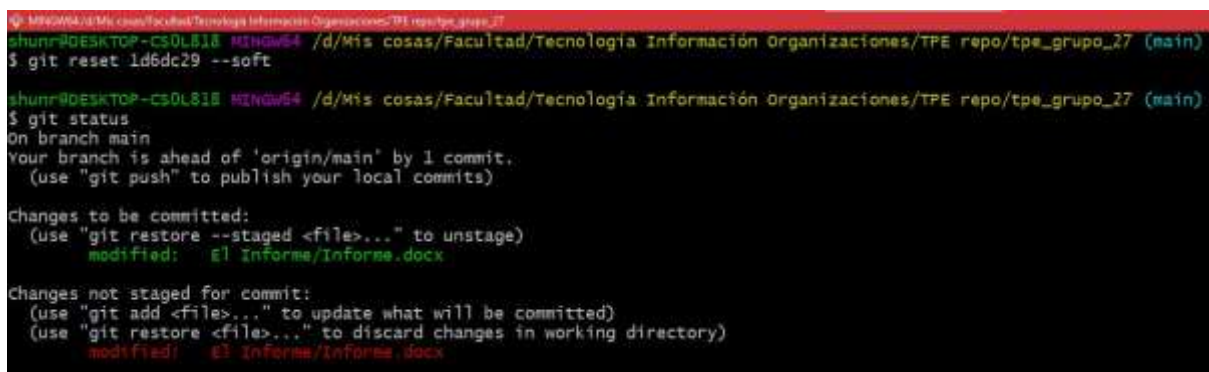
```
Usuario@Usuario-PC MINGW64 ~/Desktop/TPE27/tpe_grupo_27/Media/Capturas de Pantall
la Milanesi (main)
$ git mv 00Inicio\ repositorio\ Milanesi.jpg 00InitMilanesi
```

Imagen correspondiente al alumno Pedro Milanesi.

Luego ejecutamos `git status` para comprobar en que el archivo fue renombrado.

- Deshacer confirmaciones:

Ejecución del comando "`git reset <HASH> --soft`" para deshacer los commits siguientes a ese, manteniendo los cambios en staging área.



```
shunr@DESKTOP-C50L81E MINGW64 /d/Mis cosas/Facultad/Tecnología Información Organizaciones/TPE repo/tpe_grupo_27 (main)
$ git reset 1d6dc29 --soft

shunr@DESKTOP-C50L81E MINGW64 /d/Mis cosas/Facultad/Tecnología Información Organizaciones/TPE repo/tpe_grupo_27 (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   El Informe/Informe.docx

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   El Informe/Informe.docx
```

Imagen correspondiente a la alumna Melisa Burlando.

- Añadir cambios y/o cambiar el mensaje a una confirmación:

Para cambiar el mensaje del último commit se utilizó el comando `git commit --amend` <mensaje>, como se muestra en la siguiente captura de pantalla.

```
usuario@LAPTOP-IC3CAFMO MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Pagina web (main)
$ git add .

usuario@LAPTOP-IC3CAFMO MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Pagina web (main)
$ git co -m "cambio json a carpeta jsons"
[main 73fc585] cambio json a carpeta jsons
1 file changed, 0 insertions(+), 0 deletions(-)
rename Pagina web/ => jsons/top01.json (100%)

usuario@LAPTOP-IC3CAFMO MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Pagina web (main)
$ git commit --amend -m "cambio top01 a carpeta jsons"
[main 4690600] cambio top01 a carpeta jsons
Date: Thu Nov 4 21:40:00 2021 -0300
1 file changed, 0 insertions(+), 0 deletions(-)
rename Pagina web/ => jsons/top01.json (100%)
```

Correspondiente Lourdes Casero. "git co -m" es un alias referente al comando: `git commit -m`

- Retirar al menos una vez modificaciones del área de preparación.

En la siguiente imagen se muestra un caso donde hubo que retirar modificaciones del staging area ya que le faltaba contenido al archivo. Para ello se utilizó el comando `git reset HEAD <archivo>` aunque también se hubiera podido utilizar el comando `git restore --staged <archivo>` o simplemente `git reset`.

```
MINGW64 /d/Mis cosas/Facultad/Tecnología Información Organizaciones/TPE repo/tpe_grupo_27
(use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   El Informe/Informe.docx

no changes added to commit (use "git add" and/or "git commit -a")

shunrr@DESKTOP-CS0L818 MINGW64 /d/Mis cosas/Facultad/Tecnología Información Organizaciones/TPE repo/tpe_grupo_27 (main)
$ git add .

shunrr@DESKTOP-CS0L818 MINGW64 /d/Mis cosas/Facultad/Tecnología Información Organizaciones/TPE repo/tpe_grupo_27 (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   El Informe/Informe.docx

shunrr@DESKTOP-CS0L818 MINGW64 /d/Mis cosas/Facultad/Tecnología Información Organizaciones/TPE repo/tpe_grupo_27 (main)
$ git reset HEAD
FETCH_HEAD      ORIG_HEAD      origin/HEAD      origin/jsons      origin/webpage
HEAD            main            origin/comentarios  origin/main
shunrr@DESKTOP-CS0L818 MINGW64 /d/Mis cosas/Facultad/Tecnología Información Organizaciones/TPE repo/tpe_grupo_27 (main)
$ git reset HEAD El Informe/Informe.docx
Unstaged changes after reset:
M       El Informe/Informe.docx

shunrr@DESKTOP-CS0L818 MINGW64 /d/Mis cosas/Facultad/Tecnología Información Organizaciones/TPE repo/tpe_grupo_27 (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   El Informe/Informe.docx

no changes added to commit (use "git add" and/or "git commit -a")

shunrr@DESKTOP-CS0L818 MINGW64 /d/Mis cosas/Facultad/Tecnología Información Organizaciones/TPE repo/tpe_grupo_27 (main)
$ |
```

Captura de pantalla correspondiente a la alumna Melisa Burlando.

- Crear una rama a nivel local y llevar cambios a la rama principal:

Con el comando *git branch* se pueden observar las ramas locales ya creadas. Luego se ejecuta el comando *git branch [nombre_rama_nueva]*, y a continuación se comprueba que la nueva rama fue creada correctamente con el comando *git branch*. Para posicionar el HEAD en esta nueva branch se ejecuta el comando: *git checkout [nombre_rama_nueva]*:

```

Usuario@LAPTOP-IC3CAFMD MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (main)
$ git branch
* main

Usuario@LAPTOP-IC3CAFMD MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (main)
$ git branch lumos

Usuario@LAPTOP-IC3CAFMD MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (main)
$ git branch
  lumos
* main

```

Captura de pantalla perteneciente a Lourdes Casero

En la imagen de abajo, se observa que se utilizó *git add* para subir al área de preparación un archivo llamado "biografía tarantino", el cual posteriormente se commiteó.

Luego el usuario coloca el HEAD en la rama "main" utilizando *git checkout main*.

```

MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero
Usuario@LAPTOP-IC3CAFMD MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (lumos)
$ nano biografia\ tarantino

Usuario@LAPTOP-IC3CAFMD MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (lumos)
$ git status
On branch lumos
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    biografia tarantino

nothing added to commit but untracked files present (use "git add" to track)

Usuario@LAPTOP-IC3CAFMD MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (lumos)
$ git add "biografia tarantino"
warning: LF will be replaced by CRLF in Media/captura de pantalla casero/biografia tarantino.
The file will have its original line endings in your working directory

Usuario@LAPTOP-IC3CAFMD MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (lumos)
$ git commit -m "añadido biografia tarantino"
[lumos f402123] añadido biografia tarantino
1 file changed, 8 insertions(+)
 create mode 100644 Media/captura de pantalla casero/biografia tarantino

Usuario@LAPTOP-IC3CAFMD MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (lumos)
$ git status
On branch lumos
nothing to commit, working tree clean

Usuario@LAPTOP-IC3CAFMD MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (lumos)
$ git chec main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Usuario@LAPTOP-IC3CAFMD MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (main)
$ git status
On branch main

```

Captura de pantalla perteneciente a Lourdes Casero, *chec* es un alias configurado por el usuario para *checkout*

Por último, como se observa en la imagen posterior, se procedió a fusionar el contenido de la rama "lumos" con la "main" mediante el comando *git merge "lumos"*

Realizado el merge, se eliminó la rama "lumos" mediante *git branch -d lumos*

```

usuario@LAPTOP-IC3CAFMO MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (main)
$ git merge lunos
updating 0962e56..f402123
Fast-forward
 Media/captura de pantalla casero/biografia tarantino | 8 ++++++
 1 file changed, 8 insertions(+)
 create mode 100644 Media/captura de pantalla casero/biografia tarantino

usuario@LAPTOP-IC3CAFMO MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use 'git push' to publish your local commits)

nothing to commit, working tree clean

usuario@LAPTOP-IC3CAFMO MINGW64 ~/Desktop/grupo_27/tpe_grupo_27/Media/captura de pantalla casero (main)
$ git br -d lunos
Deleted branch lunos (was f402123).

```

Captura de pantalla correspondiente a la alumna Casero Lourdes Evangelina

ACLARACIÓN: en algunas imágenes se podrá observar el uso de ALIAS en los comandos. POR ESO ÉSTOS SE VEN DISTINTOS A las demás, A CONTINUACIÓN SE dejan asentados dichos ALIAS:

```

alias.chec=checkout
alias.br=branch
alias.co=commit
alias.st=status

```

para esto se utilizo el siguiente comando:

```
git config --global alias.(alias) (nombre por defecto)
```

Captura de pantalla correspondiente a la alumna Casero Lourdes. Listado de alias personalizados por ella.

- Creación de rama remota, trabajo colaborativo y posterior fusión:

El equipo decidió trabajar sobre la rama remota "jsons", con el objetivo de que cada uno de los integrantes suban al menos un archivo top##.json, necesarios para el funcionamiento de la página web. Para esto cada integrante del equipo siguió los siguientes pasos en GIT:

- 1) Creación de la rama local jsons:

```

Mariano@Mariano-PC MINGW64 ~/Desktop/Tio-tpe27/tpe_grupo_27 (main)
$ git branch jsons

Mariano@Mariano-PC MINGW64 ~/Desktop/Tio-tpe27/tpe_grupo_27 (main)
$ git branch
jsons
= main

```

- 2) Posicionamiento en la rama local jsons:

```

Mariano@Mariano-PC MINGW64 ~/Desktop/Tio-tpe27/tpe_grupo_27 (main)
$ git checkout jsons
Switched to branch 'jsons'

```

- 3) Creación de la rama remota jsons:

```

Mariano@Mariano-PC MINGW64 ~/Desktop/Tio-tpe27/tpe_grupo_27 (jsons)
$ git remote add jsons https://github.com/exajoakinic/tpe_grupo_27

```

- 4) Actualización de la rama local jsons:

```

Mariano@Mariano-PC MINGW64 ~/Desktop/Tio-tpe27/tpe_grupo_27 (jsons)
$ git pull jsons jsons
From https://github.com/exajoakinic/tpe_grupo_27
 = branch          jsons      -> FETCH_HEAD
 = [new branch]     jsons      -> jsons/jsons

```

5) Agregado del archivo .json en el área de intercambio y realizado del commit.

```
Mariano@Mariano-PC MINGW64 ~/Desktop/Tio-tpe27/tpe_grupo_27 (jsons)
$ git add Pagina\ Web\jsons\top03.json

Mariano@Mariano-PC MINGW64 ~/Desktop/Tio-tpe27/tpe_grupo_27 (jsons)
$ git status
On branch jsons
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   Pagina Web\jsons\top03.json

Mariano@Mariano-PC MINGW64 ~/Desktop/Tio-tpe27/tpe_grupo_27 (jsons)
$ git commit -m "Agrego top03.jsons"
[jsons 9f9ad86] Agrego top03.jsons
1 file changed, 25 insertions(+)
create mode 100644 Pagina Web\jsons\top03.json
```

6) Actualización del remoto jsons:

```
Mariano@Mariano-PC MINGW64 ~/Desktop/Tio-tpe27/tpe_grupo_27 (jsons)
$ git push jsons jsons
Username for 'https://github.com': Mpironio167
Counting objects: 44, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (30/30), done.
Writing objects: 100% (44/44), 987.30 KiB | 240.00 KiB/s, done.
Total 44 (delta 20), reused 31 (delta 12)
remote: Resolving deltas: 100% (20/20), completed with 6 local objects.
To https://github.com/exajoakinic/tpe_grupo_27
6ba7b13..9f9ad86 jsons -> jsons
```

Capturas de pantalla anteriores correspondiente al alumno Mariano Jesús Pironio

Una vez que todos los integrantes subieron los correspondientes archivos .json se procedió al fusinado de las ramas remotas, para esto, primero se las fusionó a nivel local con el comando *git merge jsons* estando situado en el branch main y luego se realizó un push, tal como se puede observar en las imágenes a continuación.

```
shunr@DESKTOP-C50H81B MINGW64 /d/Mis cosas/Facultad/Tecnología Información Organizaciones/TPC repo/tpe_grupo_27/Pagina web/jsons (main)
$ git merge jsons
Updating 116c213..9f9ad86
Fast-forward
 Pagina web/jsons/top03.json | 25 +++++
 Pagina web/jsons/top07.json | 31 +++++
 Pagina web/jsons/top09.json | 23 +++++
 Peliculas/top 09/Portada de Jackie brown.jpg | Bin 0 -> 20054 bytes
 Peliculas/top 09/jackie brown afiche.jfif | Bin 0 -> 11473 bytes
 ../jackie-brown-1997-001-jackie-max-romance.jpg | Bin 0 -> 62157 bytes
 6 files changed, 79 insertions(+)
 create mode 100644 Pagina web/jsons/top01.json
 create mode 100644 Pagina web/jsons/top07.json
 create mode 100644 Pagina web/jsons/top09.json
 create mode 100644 Peliculas/top 09/Portada de Jackie brown.jpg
 create mode 100644 Peliculas/top 09/jackie brown afiche.jfif
 create mode 100644 Peliculas/top 09/jackie-brown-1997-001-jackie-max-romance.jpg
```

Captura de pantalla de Melisa Burlando.

- Crear una carpeta e indicar a git que no le haga seguimiento:

Se creó el archivo *.gitignore*, y dentro de él se indicó a GIT el nombre de la carpeta de la cual se pretendía que GIT no realizara seguimiento.

Inicialmente se consideró que sería buena idea indicar a git que ignorara la carpeta MEDIA (donde se alojaron las capturas de pantalla de git de los miembros) para que, de esta forma, no se mezclara en cada commit el material para el sitio web con las capturas de

pantalla para el informe. Lamentablemente eso generó algunos conflictos, dado que al crear el .gitignore la carpeta MEDIA ya existía en el proyecto y contenía archivos con seguimiento dentro de git, produciendo que algunos integrantes puedan seguir subiendo sus capturas sin entender por qué y otros no puedan subirla e incluso llegando a perdiéndose algunas capturas.

Finalmente, se modificó .gitignore para que no realizada seguimiento la carpeta *Privado* -carpeta para la que git aún no había realizado ningún seguimiento-, de esta forma tuvimos la posibilidad de agregar archivos dentro de esta carpeta sin que se suban al remoto, esto último puede observarse en la siguiente captura:

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Privado/

nothing added to commit but untracked files present (use "git add" to track)
Joaquin@NBAttril MINGW64 ~/Desktop/TPE27/tpe_grupo_27 (main)
$ nano .gitignore

Joaquin@NBAttril MINGW64 ~/Desktop/TPE27/tpe_grupo_27 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
  modified:   .gitignore

no changes added to commit (use "git add" and/or "git commit -a")

Joaquin@NBAttril MINGW64 ~/Desktop/TPE27/tpe_grupo_27 (main)
$ git add .gitignore
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory

Joaquin@NBAttril MINGW64 ~/Desktop/TPE27/tpe_grupo_27 (main)
$ git commit -m "Agrego .gitignore para no realizar seguimiento sobre la carpeta Privado"
[main b919660] Agrego .gitignore para no realizar seguimiento sobre la carpeta Privado
 1 file changed, 1 insertion(+)

Joaquin@NBAttril MINGW64 ~/Desktop/TPE27/tpe_grupo_27 (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 348 bytes | 348.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/exajoakinic/tpe_grupo_27
 136c215..b919660 main -> main
```

Captura de pantalla de Joaquín Iglesias Coqueti.

IV. Conclusión

Para muchos de los miembros del grupo éste fue uno de los primeros acercamientos a *GIT*, por este motivo se presentaron grandes desafíos, tanto en la utilización de los comandos, como en el trabajo colaborativo.

Uno de los principales problemas al comienzo del trabajo, fue la elección de un tema sobre el cuál poder desarrollar el proyecto, ya que TIO es la primera materia de la mayoría de los integrantes del equipo. Todo esto para poner en práctica los comandos de *GIT*, y luego poder elaborar el informe usando una metodología colaborativa de “equipo de trabajo”.

Otro problema que se presentó fue la coordinación entre los miembros del grupo, debido a la cantidad de integrantes y la inexperiencia en trabajo *colaborativo, asincrónico y remoto*. La herramienta *Trello* fue de gran utilidad para solucionarlo por sus funcionalidades para asignar tareas y ver las etapas de desarrollo del proyecto. De esa manera cada integrante del grupo sabía cuáles eran sus responsabilidades, evitando la superposición de tareas.

A lo largo del proyecto surgieron diversos inconvenientes con la actualización del repositorio local ya que al querer hacer un *git push*, la consola devolvía un mensaje de error, el cuál era solucionado con el comando *git pull*. Para la solución de éste y otros errores se utilizaron las filminas brindadas por la cátedra.

Se cumplieron con los *objetivos* y las *metas* propuestas, logrando una buena interacción entre todos los miembros del equipo a través de diferentes plataformas como grupo de *WhatsApp*, *Discord* y *Meet*, adquiriendo practicidad y agilidad en la comunicación grupal.

El hecho de tener una serie de pautas individuales permitió a cada uno practicar los comandos aprendidos en la cátedra y buscar soluciones conjuntas para cada caso en particular en el que algún compañero tenía un inconveniente.

El presente trabajo ayudó a consolidar los conceptos aprendidos sobre el funcionamiento de Git, aclarando cuestiones de comprensión sobre la utilidad que cumplen los comandos, lo cual nos acerca al objetivo de saber usar la herramienta de forma correcta.

V. Bibliografía

Autor desconocido. (Fecha desconocida). *git --local-branching-on-the-cheap*. Git-SCM.

<https://git-scm.com/docs/gitignore>

Camacho, D. (Agosto de 2021). *Qué es GitHub y cómo usarlo para aprovechar sus beneficios*. Platzi.

https://platzi.com/blog/que-es-github-como-funciona/?gclid=Cj0KCQiA-K2MBhC-ARIsAMtLKRsACCDKfRIBFebIEi75qeI_TXF30W1tlfRG_Eh7-kCL6tXWGRY1Qd4aApkHEALw_wcB&gclsrc=aw.ds

Chacon, S., Straub, B. (14 de Agosto de 2021). *Pro Git: todo lo que necesitas saber acerca de GIT*. Apress. Git-SCM.

<https://git-scm.com/book/es/v2>

Tecnología de Información de las Organizaciones 2021, TUDAI, EXACTAS, UNICEN
Filminas provistas por la cátedra.