



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de Programación

Grupo: #3

No de Práctica(s): #1

Integrante(s): Torres Mendoza Alexa Erandy

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 49

Semestre: 2020-1

Fecha de entrega: 26- agosto- 2019

Observaciones: Excelente

CALIFICACIÓN: 10



Se utiliza AND para la búsqueda que contengan las dos palabras a su lado.



Se utilizan las comillas para la búsqueda exacta de las palabras en ellas



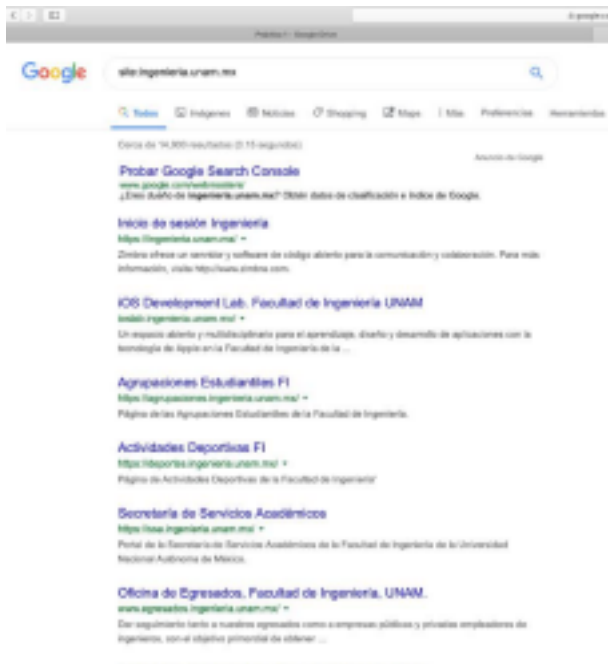
Se utiliza para que te de resultados con algunas de las dos palabras de OR pero no las dos



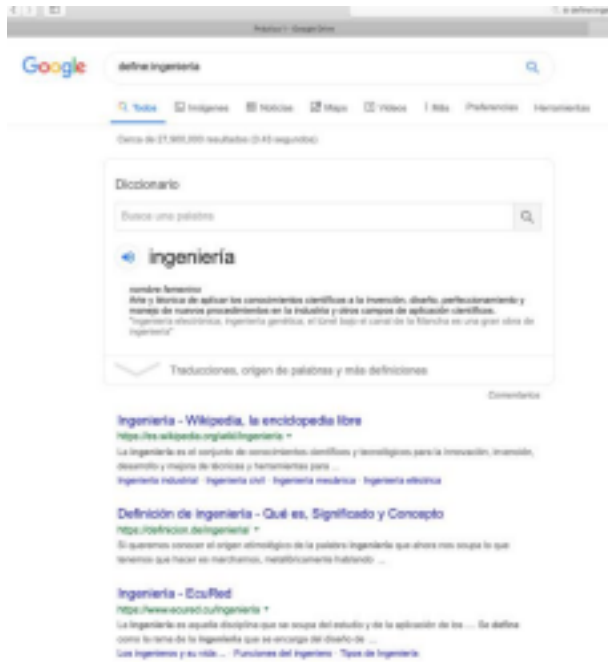
Se utiliza el menos para que te de resultados de la búsqueda excepto esa palabra



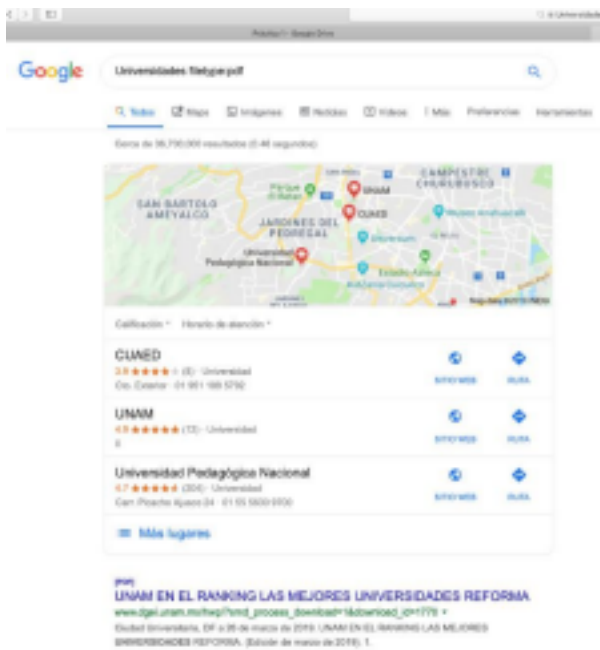
Se utiliza el asterisco para tomar cualquier palabra antes del asterisco pero con el formato de la búsqueda



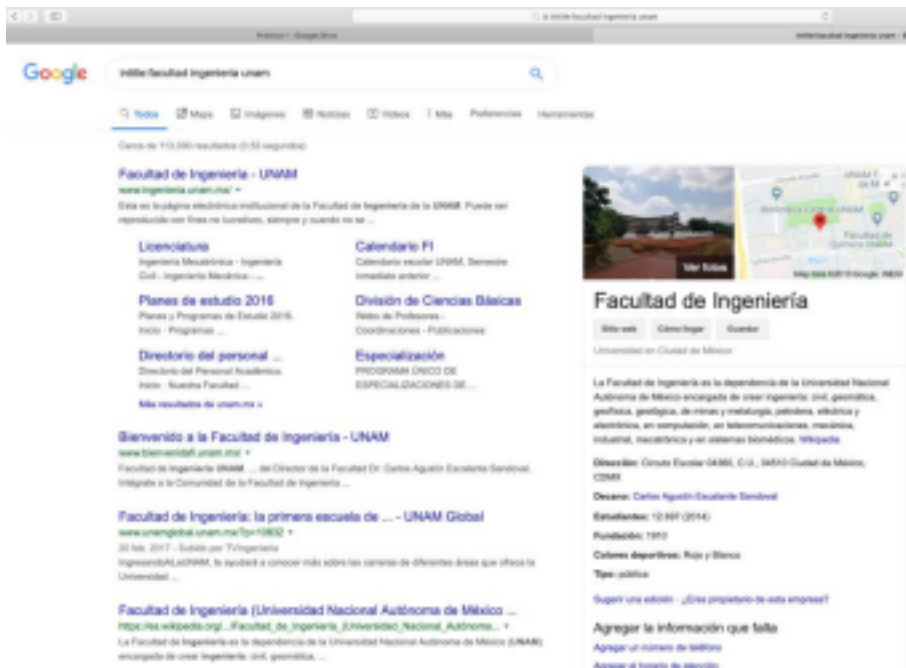
Se ocupa site para que las palabras de busqueda sean las misma del sitio we



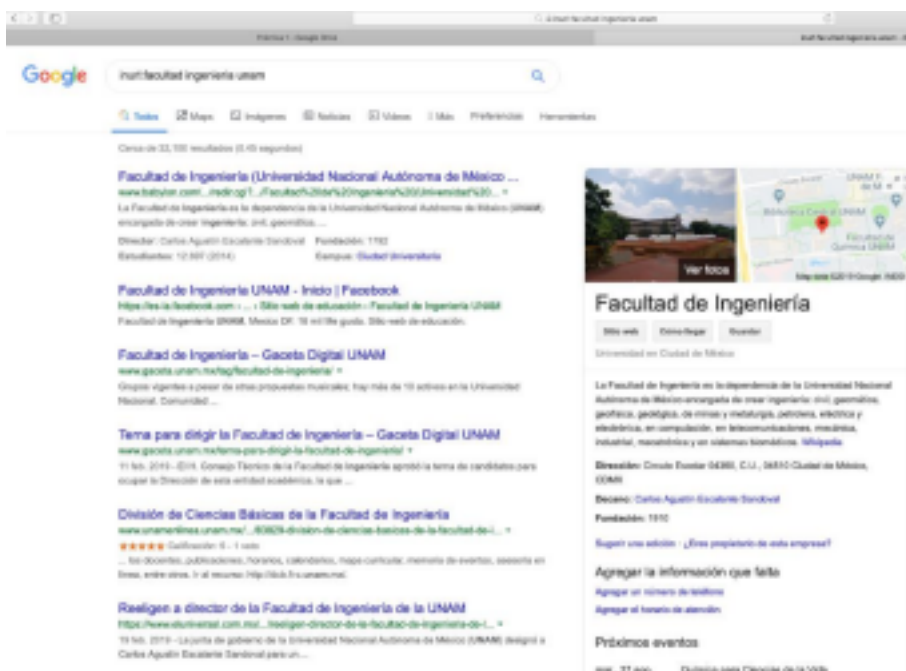
Te da resultados de la búsqueda con sus definiciones



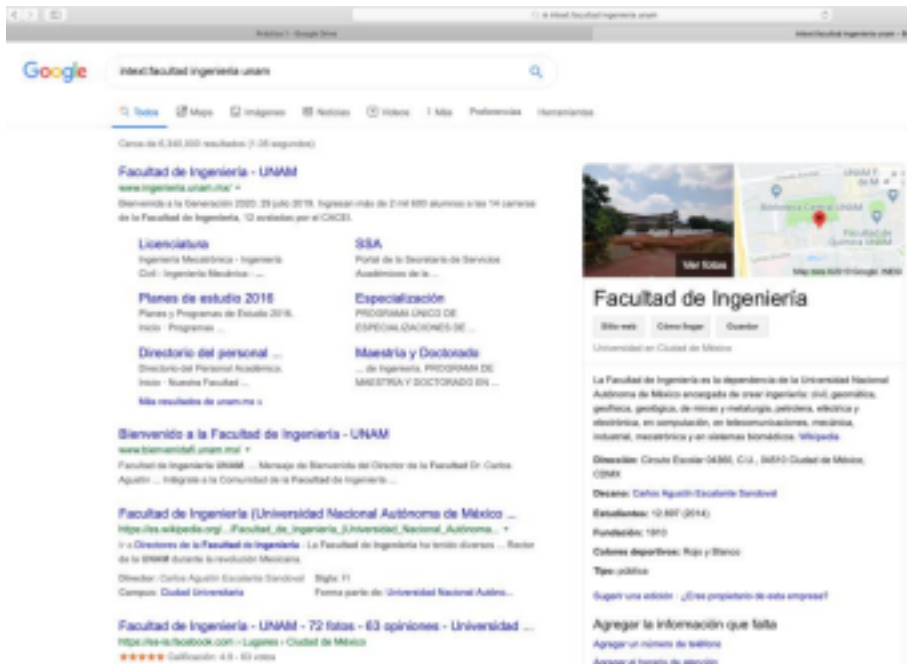
El filetype es para especificar el tipo de documento requerido



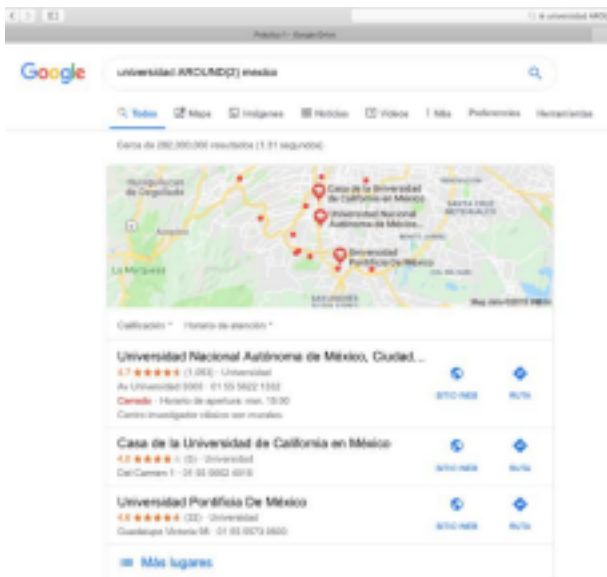
Intitle se utiliza para que en tus resultados estén las palabras de la búsqueda dentro del título



Se ocupa inurl para que las palabras de búsqueda aparezcan en el URL



Intext se utiliza para que dentro de tu búsqueda aparezcan las palabras requeridas



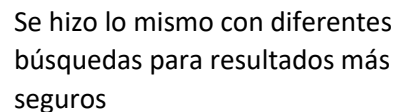
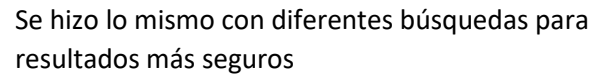
Se ocupa AROUND para que varias palabras esten en el intervalo de la búsqueda



Combine diferentes comandos para una sola búsqueda mas específica de un tema libre



Combine otros comandos diferentes que son intext, AND y OR para obtener los resultados esperados

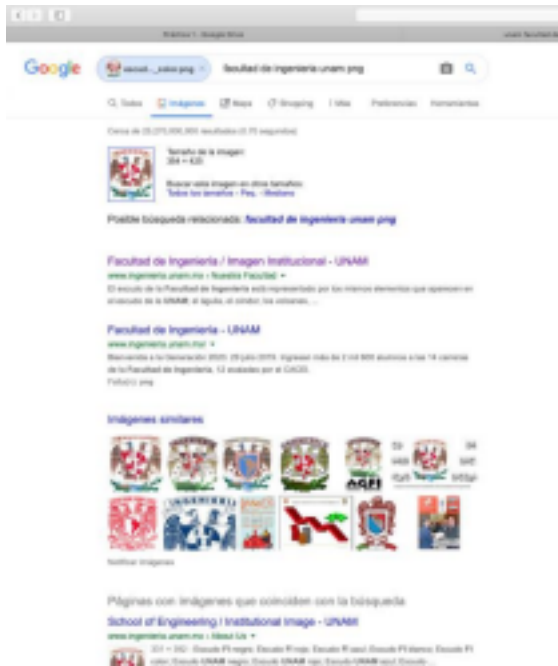




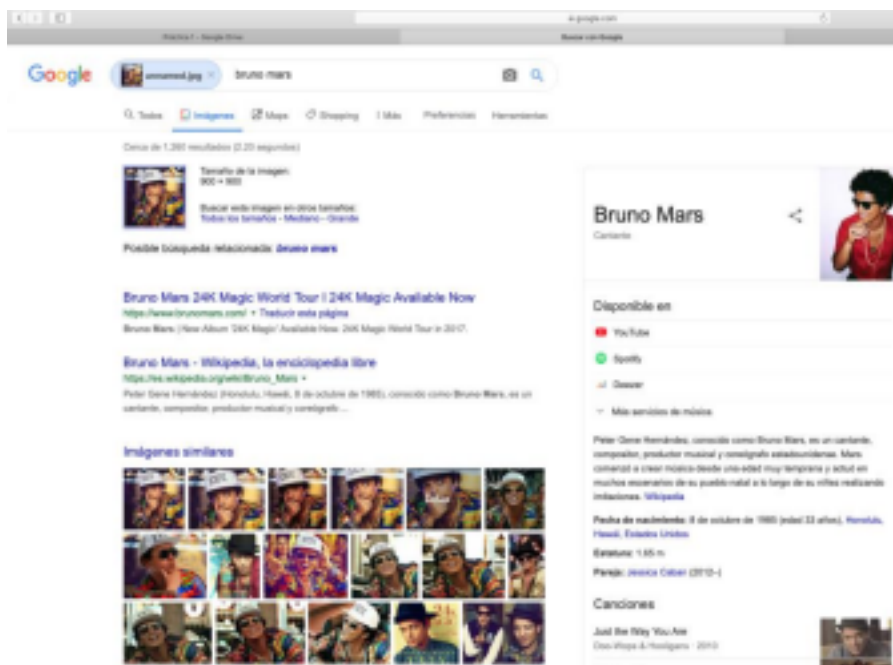
Se hizo lo mismo con diferentes búsquedas para resultados más seguros



Se hizo lo mismo con diferentes búsquedas para resultados más seguros



Usé google imágenes para buscar una imagen en específico y poderla subir



Hice lo mismo con una imagen opcional para poder subirla

```
MINGW64/c/Users/alexarandy/FP_2020-1_1641
alexarandy@Alexa MINGW64 ~
$ pwd
/c:/Users/alexarandy/
alexarandy@Alexa MINGW64 ~
$ git config --global user.name "Alexa Torres"
alexarandy@Alexa MINGW64 ~
$ git config --global user.email alexatorres2909@gmail.com
alexarandy@Alexa MINGW64 ~
$ git config --global core.editor nano
alexarandy@Alexa MINGW64 ~
$ mkdir ~/PROYECTO
alexarandy@Alexa MINGW64 ~
$ cd ~/PROYECTO
alexarandy@Alexa MINGW64 ~/PROYECTO
$ git init
Initialized empty Git repository in C:/Users/alexarandy/PROYECTO/.git/
alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ echo "Mi primer proyecto" > README.txt
alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ git status
On branch master

No commits yet
```

En esta primera captura configure el git para poner mis datos personales.

Después cree una carpeta llamada proyecto para comenzar a usar el git ahí.

Observe los cambios con git status hasta añadir archivo README.

Añadí este archivo para ver seguimiento.

```
MINGW64/c/Users/alexarandy/FP_2020-1_1641
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  README.txt

nothing added to commit but untracked files present (use "git add" to track)
alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ git add README.txt
warning: LF will be replaced by CRLF in README.txt.
The file will have its original line endings in your working directory
alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   README.txt

alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   README.txt

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:   README.txt

alexarandy@Alexa MINGW64 ~/PROYECTO (master)
```

No se notó el seguimiento así que use git add para README tuviera seguimiento por parte de git

```
MINOWS4/c/Users/alexander/PP_2020-1_7641
alexander@Alexa KINGS4 ~/PROYECTO (master)
$ git add README.txt
warning: LF will be replaced by CRLF in README.txt.
The file will have its original line endings in your working directory
alexander@Alexa KINGS4 ~/PROYECTO (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.txt

alexander@Alexa KINGS4 ~/PROYECTO (master)
$ touch archivo1

alexander@Alexa KINGS4 ~/PROYECTO (master)
$ touch archivo2

alexander@Alexa KINGS4 ~/PROYECTO (master)
$ touch archivo3

alexander@Alexa KINGS4 ~/PROYECTO (master)
$ touch archivo4

alexander@Alexa KINGS4 ~/PROYECTO (master)
$ toucharchivo5
bash: toucharchivo5: command not found

alexander@Alexa KINGS4 ~/PROYECTO (master)
$ touch archivo5

alexander@Alexa KINGS4 ~/PROYECTO (master)
$ git add .

alexander@Alexa KINGS4 ~/PROYECTO (master)
$ git status
```

Cree 5 archivos y todos los añadí al mismo con git add . y observe los cambios

```
MINOWS4/c/Users/alexander/PP_2020-1_7641
alexander@Alexa KINGS4 ~/PROYECTO (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.txt
    new file:   archivo1
    new file:   archivo2
    new file:   archivo3
    new file:   archivo4
    new file:   archivo5

alexander@Alexa KINGS4 ~/PROYECTO (master)
$ touch pruebas.temp

alexander@Alexa KINGS4 ~/PROYECTO (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.txt
    new file:   archivo1
    new file:   archivo2
    new file:   archivo3
    new file:   archivo4
    new file:   archivo5

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

alexander@Alexa KINGS4 ~/PROYECTO (master)
```

Hize un archivo llamado pruebas.temp donde git lo reconocia despues con gitignore hicimos que ignorara todos los archivos con terminación .temp

```
MINGW64/c:/Users/alexander/FP_2020-1_1641
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  pruebas.tmp

alexander@ALEXA MINGW64 ~/PROYECTO (master)
$ echo "*.tmp" > .gitignore

alexander@ALEXA MINGW64 ~/PROYECTO (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.txt
    new file:   archivo1
    new file:   archivo2
    new file:   archivo3
    new file:   archivo4
    new file:   archivo5

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

alexander@ALEXA MINGW64 ~/PROYECTO (master)
$ open .gitignore
back: open: command not found

alexander@ALEXA MINGW64 ~/PROYECTO (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
```

Dentro del documento también se indicó que ignorara a gitignore

```
MINGW64/c:/Users/alexander/FP_2020-1_1641
  (use "git rm --cached <file>..." to unstage)
    new file:   README.txt
    new file:   archivo1
    new file:   archivo2
    new file:   archivo3
    new file:   archivo4
    new file:   archivo5

alexander@ALEXA MINGW64 ~/PROYECTO (master)
$ git commit -m "Aqui narro los cambios que se hicieron"
[master (root-commit) 45a0123] Aqui narro los cambios que se hicieron
 3 files changed, 3 insertions(+)
 create mode 100644 README.txt
 create mode 100644 archivo1
 create mode 100644 archivo2
 create mode 100644 archivo3
 create mode 100644 archivo4
 create mode 100644 archivo5

alexander@ALEXA MINGW64 ~/PROYECTO (master)
$ git status
On branch master
nothing to commit, working tree clean

alexander@ALEXA MINGW64 ~/PROYECTO (master)
$ git log
commit 45a012380182c75cb01261061a3523e07bbf51e (HEAD -> master)
Author: Alexa Torres <alexatorres2909@gmail.com>
Date:   Sat Aug 24 17:58:12 2019 -0500

    Aqui narro los cambios que se hicieron

alexander@ALEXA MINGW64 ~/PROYECTO (master)
$ rm archivo5

alexander@ALEXA MINGW64 ~/PROYECTO (master)
$ git status
On branch master
Changes not staged for commit:
```

Hice mi primer commit para guardar los cambios que hice y después con git log para visualizar mi commit. Utilice rm para borrar el archivo 5 con git status lo comprobé.

```
MINGW64/c/Users/Alexander/TP_2020-1_1641
alexandrady@Alexa MINGW64 ~/PROJECT0 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    archivod

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

alexandrady@Alexa MINGW64 ~/PROJECT0 (master)
$ git rm archivod
rm 'archivod'

alexandrady@Alexa MINGW64 ~/PROJECT0 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    archivod

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

alexandrady@Alexa MINGW64 ~/PROJECT0 (master)
$ mv archivod archivoe

alexandrady@Alexa MINGW64 ~/PROJECT0 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    archivod

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

Utilice git rm para el archivo 4 para le diera seguimiento mientras que el archivo 5 sigue sin seguimiento. Después con mv hicimos el cambio del archivo 3 al archivo 6.

```
(use "git restore <file>..." to discard changes in working directory)
    deleted:    archivod
    deleted:    archivoe

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

alexandrady@Alexa MINGW64 ~/PROJECT0 (master)
$ git mv archivod archivoe

alexandrady@Alexa MINGW64 ~/PROJECT0 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    archivod
    renamed:   archivod => archivoe

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

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

alexandrady@Alexa MINGW64 ~/PROJECT0 (master)
$ git commit -m "Borrando y moviendo archivos 1"
[master d754381] Borrando y moviendo archivos 1
 2 files changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 archivod
 rename archivod => archivoe (100%)

alexandrady@Alexa MINGW64 ~/PROJECT0 (master)
$ git status
```

Se hizo lo mismo para cambiar del archivo 2 al archivo 7. Después guardamos los cambios en un nuevo commit con el nombre borrando y moviendo archivos 1.


```
MINGW64/c/Users/alexander/FP_2020-1_1641
alexander@alexa: MINGW64 ~/PROJECT0 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    archiv01
    deleted:    archiv06

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    archiv06

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

alexander@alexa: MINGW64 ~/PROJECT0 (master)
$ git rm archiv01
rm 'archiv01'

alexander@alexa: MINGW64 ~/PROJECT0 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    archiv01

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    archiv06

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    archiv06

alexander@alexa: MINGW64 ~/PROJECT0 (master)
$ echo "Archivo temporal" > temp
alexander@alexa: MINGW64 ~/PROJECT0 (master)
```

Después de borrar y mover archivos creamos el archivo temporal.

```
MINGW64/c/Users/alexander/FP_2020-1_1641
alexander@alexa: MINGW64 ~/PROJECT0 (master)
$ git add .
warning: LF will be replaced by CRLF in temp.
The file will have its original line endings in your working directory

alexander@alexa: MINGW64 ~/PROJECT0 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    archiv01
    renamed:    archiv01 -> archiv06
    new file:   temp

alexander@alexa: MINGW64 ~/PROJECT0 (master)
$ git rm --cached temp
rm 'temp'

alexander@alexa: MINGW64 ~/PROJECT0 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    archiv01
    renamed:    archiv01 -> archiv06

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

alexander@alexa: MINGW64 ~/PROJECT0 (master)
$ git commit -m "Borrando y moviendo archivos 2"
[master add6d6f] Borrando y moviendo archivos 2
2 files changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 archiv01
rename archiv01 -> archiv06 (100%)
```

Usamos rm cached para que el archivo ya no tuviera seguimiento por parte de git pero seguía estando dentro de la carpeta.

Guardamos los cambios en commit con el título borrando y moviendo archivos 2

```
MINIGW64/c/Users/alexandry/FP_2020-1_1641

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ mv temp no_temp

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ git add .
warning: LF will be replaced by CRLF in no_temp.
The file will have its original line endings in your working directory

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ git commit -m "Siempre si quiero agregar el archivo temporal"
[master ae178dd] Siempre si quiero agregar el archivo temporal
1 file changed, 1 insertion(+)
create mode 100644 no_temp

alexandrady@Alexa KING64 ~/PROYECTO (master)
$

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ echo "Olvide editar mi nuevo archivo!" >> no_temp

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ git add .
warning: LF will be replaced by CRLF in no_temp.
The file will have its original line endings in your working directory

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ git commit --amend
[master e4b8dc3] Siempre si quiero agregar el archivo temporal
Date: Sat Aug 24 18:12:43 2019 -0500
1 file changed, 2 insertions(+)
create mode 100644 no_temp

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ echo "Cambios al archivol" >> archivol

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ echo "Cambios al archivod" >> archivod
```

Movimos el archivo temp para hacer un commit con el titulo siempre si quiero agregar el archivo temporal. Ocupamos echo para añadir --olvide editar mi nuevo archivo- en no_temp.

Se puso amend para añadir el cambio en el commit anterior. Se hizo una modificación en los archivos 1,6 y 7 y se guardaron en un commit mas cambios.

```
MINIGW64/c/Users/alexandry/FP_2020-1_1641

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ echo "Cambios al archivol" >> archivol

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ echo "Cambios al archivod" >> archivod

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ echo "Cambios al archivod" >> archivod

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ git add .
warning: LF will be replaced by CRLF in archivol.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in archivod.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in archivod.
The file will have its original line endings in your working directory

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ git commit -m "Cambios para hacer pruebas"
[master eb2be7e] Cambios para hacer pruebas
3 files changed, 3 insertions(+)

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ echo "Mas cambios al archivol" >> archivol

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ echo "Mas cambios al archivod" >> archivod

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ echo "Mas cambios al archivod" >> archivod

alexandrady@Alexa KING64 ~/PROYECTO (master)
$ git add .
warning: LF will be replaced by CRLF in archivol.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in archivod.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in archivod.
```

Se hizo una modificación en los archivos 1,6 y 7 y se guardaron en un commit mas cambios. Y despues se vuelven a modificar.

```
MINGW64/Users/AlexaTorres/FP_2020-1_1641
alexatorres@Alexa MINGW64 ~/PROJECT0 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   archiv6
        modified:   archiv8
        modified:   archiv7

alexatorres@Alexa MINGW64 ~/PROJECT0 (master)
$ git reset HEAD archiv6
Unstaged changes after reset:
M   archiv6
M   archiv7

alexatorres@Alexa MINGW64 ~/PROJECT0 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   archiv6
        modified:   archiv7

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:   archiv6

alexatorres@Alexa MINGW64 ~/PROJECT0 (master)
$ git checkout -- archiv6

alexatorres@Alexa MINGW64 ~/PROJECT0 (master)
$ git reset HEAD
Unstaged changes after reset:
M   archiv6
M   archiv7

alexatorres@Alexa MINGW64 ~/PROJECT0 (master)
$ git status
```

Se saca el archivo 1 del seguimiento de git y se devolvio el archivo uno a su estado anterior con checkout.

```
MINGW64/Users/AlexaTorres/FP_2020-1_1641
alexatorres@Alexa MINGW64 ~/PROJECT0 (master)
$ git status
On branch master
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:   archiv6
        modified:   archiv7

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

alexatorres@Alexa MINGW64 ~/PROJECT0 (master)
$ git log
commit 6b2bc7685c8649f4b37bc76b4c98305f6e77dc2e (HEAD -> master)
Author: Alexa Torres <alexatorres2909@gmail.com>
Date: Sat Aug 24 18:12:09 2019 -0500

    Cambios para hacer pruebas

commit cdb0cc4831a1cb778f8bb21e65119c378ce7dad
Author: Alexa Torres <alexatorres2909@gmail.com>
Date: Sat Aug 24 18:12:43 2019 -0500

    Siempre si quiero agregar el archivo temporal

commit ad6d46f77a1b00b8f69fada6f711fc9156aa627c
Author: Alexa Torres <alexatorres2909@gmail.com>
Date: Sat Aug 24 18:19:21 2019 -0500

    Borrando y moviendo archivos 2

commit d794382bb75d043685143cfedaa9325c7b409c0b
Author: Alexa Torres <alexatorres2909@gmail.com>
Date: Sat Aug 24 17:58:12 2019 -0500

    Borrando y moviendo archivos 1

commit 61a0129182182c75cb82761241a611e67b6f1e
Author: Alexa Torres <alexatorres2909@gmail.com>
Date: Sat Aug 24 17:58:12 2019 -0500
```

Con reset head se elimino a los archivos 6 y 7 del seguimiento de git.

Usamos git log para ver nuestras versiones de los commits guardados anteriormente.

```
commit d754381bb75d043681143cfdaa9921c7b409c0b
Author: Alexa Torres <alexatorres299@gmail.com>
Date:   Sat Aug 24 17:55:48 2023 -0500

    Borrando y moviendo archivos 1

commit 41a0329108182c7fcb42a1061a1128d07b0f5e
Author: Alexa Torres <alexatorres299@gmail.com>
Date:   Sat Aug 24 17:50:12 2023 -0500

    Aquí narra los cambios que se hicieron

alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ git checkout d75438
error: Your local changes to the following files would be overwritten by checkout:
    archivo6
    archivo7
Please commit your changes or stash them before you switch branches.
Aborting

alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ git checkout d75438 --archivo7
Updated 1 path from 9f5f0de

alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   archivo7

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:   archivo6

alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ git checkout d75438 .
```

Se regreso el archivo 7 al commit de borrando y moviendo archivos 1. El archivo 7 se devolvio al commit y el 6 no esta recibiendo seguimiento.

```
alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ git checkout d75438 .
Updated 3 paths from 9f5f0de

alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ pwd
/c:/Users/alexarandy/PROYECTO

alexarandy@Alexa MINGW64 ~/PROYECTO (master)
$ cd ..

alexarandy@Alexa MINGW64 ~
$ pwd
/c:/Users/alexarandy

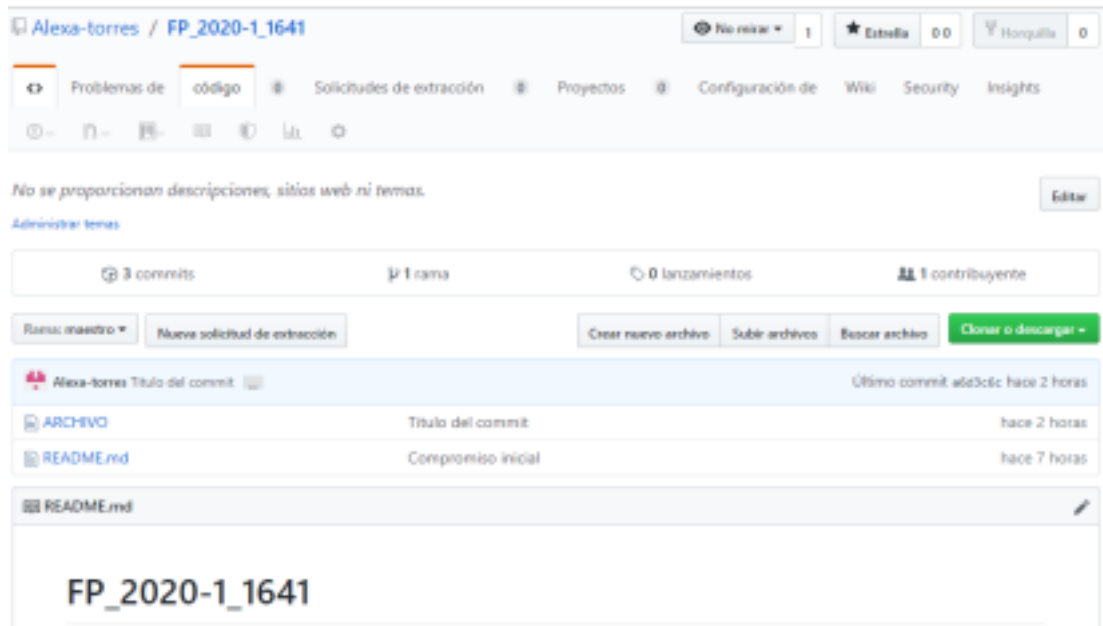
alexarandy@Alexa MINGW64 ~
$ git clone ^C

alexarandy@Alexa MINGW64 ~
$ git clone ^C

alexarandy@Alexa MINGW64 ~
$ git clone https://github.com/Alexa-Torres/FP_2020-1_1641.git
Cloning into 'FP_2020-1_1641'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

alexarandy@Alexa MINGW64 ~
$ cd
.gitconfig
ID Objects/
AppData/
Configuración local/
Contacts/
Cookies/
Datos de programa/
Desktop/
Documents/
```

Con el checkout se regresaron todos los archivos a la carpeta borrando y moviendo archivos 1. Despues con cd sales de la carpeta proyecto y te regresas a la carpeta con tu nombre.



Creamos un repositorio en github para tenerlo en linea en la pagina web y copiamos en url en la terminal para descargar el repositorio en linea en la computadora

```
MINGW64 ~/Users/alexarand/FP_2020-1_1641
Desktop/
Documents/
Downloads/
Entorno de red/
Favoritos/
FP_2020-1_1641/
ICloudDrive/
Impresoras/
IntelGraphicsProfiles/
Links/
Menu Inicio/
MicrosoftEdgeBackups/
Mis documentos/
Music/
NTUSER.DAT
ntuser.dat.LOG1
ntuser.dat.LOG2
NTUSER.DAT[1459e887-6ac5-11e9-bbe1-d8b7ba45cab].TM.b1P
NTUSER.DAT[1459e887-6ac5-11e9-bbe1-d8b7ba45cab].TMContainer00000000000000000001.nagtrans-ms
NTUSER.DAT[1459e887-6ac5-11e9-bbe1-d8b7ba45cab].TMContainer0000000000000000000002.nagtrans-ms
ntuser.ini
OneDrive/
Pictures/
Plantillas/
PROXCTO/
Recargas/
Send Games/
SendTo/
Videos/

alexarand@alexa MINGW64 ~
$ cd FP_2020-1_1641/

alexarand@alexa MINGW64 ~/FP_2020-1_1641 (master)
$ pwd
~/Users/alexarand/FP_2020-1_1641

alexarand@alexa MINGW64 ~/FP_2020-1_1641 (master)
$ echo "Creando un archivo de prueba" > ARCHIVO
```

Se escribe cd y el nombre del primer repositorio para que lo identifique y entrar a la carpeta y se confirma con pwd, se crea un archivo para el repositorio y se guardan los cambios en el commit de github.

```
MINGW64 ~/Documents/PP_2020-1_1641
$ cd PP_2020-1_1641/

alexander@alex: ~/PP_2020-1_1641 (master)
$ cd
~/Documents/PP_2020-1_1641
alexander@alex: ~/PP_2020-1_1641 (master)
$ echo "Creando un archivo de prueba" > ARCHIVO
alexander@alex: ~/PP_2020-1_1641 (master)
$ git add
warning: LF will be replaced by CRLF in ARCHIVO.
The file will have the original line endings in your working directory
alexander@alex: ~/PP_2020-1_1641 (master)
$ git commit -a "Primer commit para github"
[master: 8bda30c] Primer commit para github
1 file changed, 1 insertion(+),
create mode 100644 ARCHIVO
alexander@alex: ~/PP_2020-1_1641 (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (1/1), 332 bytes | 187.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com:Alex-heras/PP_2020-1_1641.git
 8bda30c..8bda30c master -> master
alexander@alex: ~/PP_2020-1_1641 (master)
$ git commit -a "Trifurca" -m "Mensaje largo con detalles"
On branch master
Your branch is up to date with 'origin/master'.
nothing to commit, working tree clean
alexander@alex: ~/PP_2020-1_1641 (master)
$ git pull
```

En el primer commit de github enviamos los cambios que hicimos al archivo de github. Se hacen cambios directos en github para que despues se pueda descargar con git pull

```
MINGW64 ~/Documents/PP_2020-1_1641
alexander@alex: ~/PP_2020-1_1641 (master)
$ git commit -a "Primer commit para github"
[master: 8bda30c] Primer commit para github
1 file changed, 1 insertion(+),
create mode 100644 ARCHIVO
alexander@alex: ~/PP_2020-1_1641 (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (1/1), 332 bytes | 207.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com:Alex-heras/PP_2020-1_1641.git
 8bda30c..8bda30c master -> master
alexander@alex: ~/PP_2020-1_1641 (master)
$ git commit -a "Trifurca" -m "Mensaje largo con detalles"
On branch master
Your branch is up to date with 'origin/master'.
nothing to commit, working tree clean
alexander@alex: ~/PP_2020-1_1641 (master)
$ git pull
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), 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:Alex-heras/PP_2020-1_1641
 8bda30c..8bda30c master -> origin/master
Updating 8bda30c..8bda30c
Fast-forward
  ARCHIVO | 1 +
  1 file changed, 1 insertion(+)
```

git pull se actualizaron los cambios en el repositorio local.