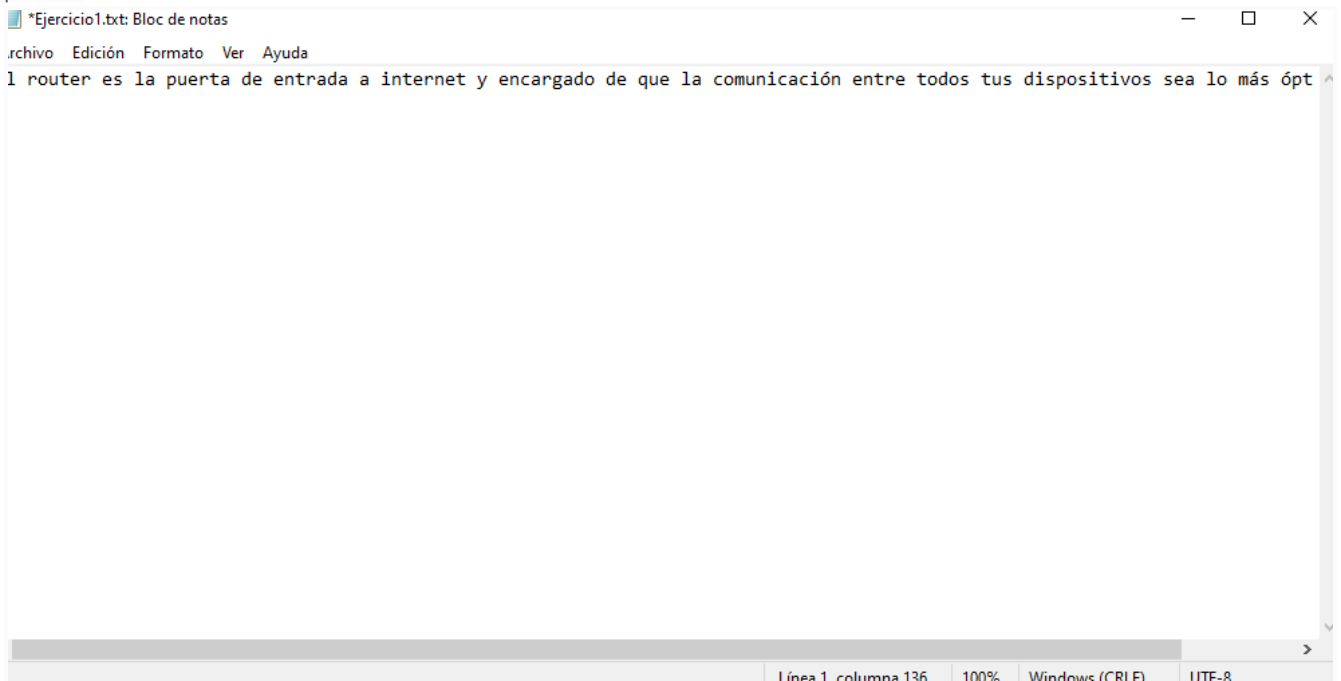


# EJERCICIOS INICIO con GIT

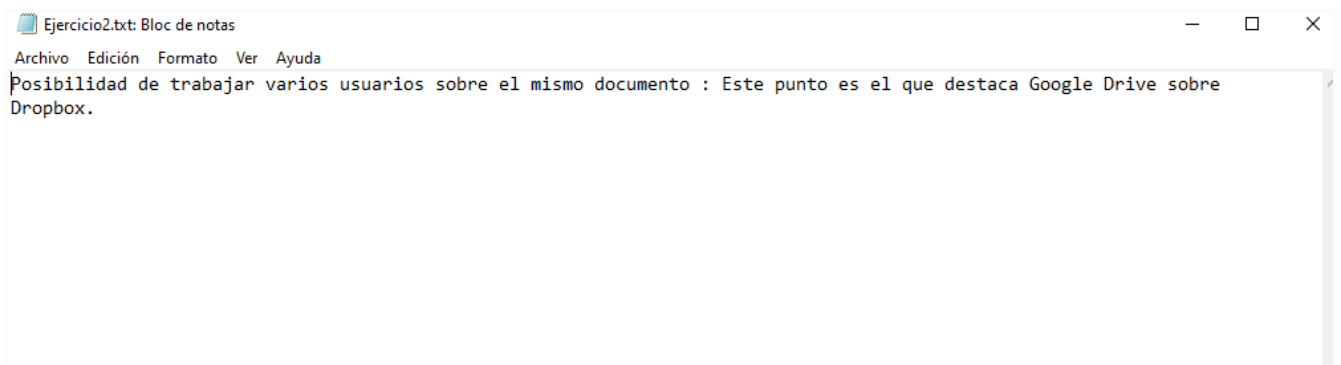
a) Instalación: <https://youtu.be/ES2xtLyl-B8>

b) Explicación en video : <https://youtu.be/imFd4zI7224>

**1) Crear nueva carpeta "Ejercicios".** Crear en ella el documento : Ejercicio1.txt , que incluye la siguiente frase:  
El router es la puerta de entrada a internet y encargado de que la comunicación entre todos tus dispositivos sea lo más óptima posible.



**2) Crear nuevo documento, Ejercicio2.txt, en la carpeta "Ejercicios".** Escribir en él la frase:  
Posibilidad de trabajar varios usuarios sobre el mismo documento : Este punto es el que destaca Google Drive sobre Dropbox.



**3) Sobre la carpeta "Ejercicios" .... Boton derecho.... Git Bash Here** (<https://git-for-windows.github.io/>)



*Si en Git elegimos: Git Bash Here , ¿qué hace?*

"Al abrir una terminal de Git Bash en un directorio específico (dentro de una carpeta), podemos trabajar con Git en ese directorio y realizar diversas operaciones, como confirmar cambios, hacer seguimientos, crear nuevas ramas. Esta se hará utilizando comandos de Git en la línea de comandos."

**4) En la ventana de comandos que aparece, escribir el comando:** `$ git init`

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git init
```

##### 5) Escribir el siguiente comando (con tu nombre de usuario)

```
$ git config user.name matias
```

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git config user.name JoseRuizJurado
```



¿Puedes explicarme el siguiente comando? `git config user.name matias`

Al ejecutar este comando, se establece el nombre de usuario "matias" en el repositorio de Git local (la carpeta donde estas trabajando). Si trabajas con otros colaboradores, permite identificar fácilmente quién realizó cada modificación. Es importante tener en cuenta que este comando solo configura el nombre del usuario en el repositorio de Git local (una carpeta). Para configurar el nombre de usuario en todos los repositorios de Git en el equipo, se puede utilizar el modificador **--global** junto con el comando `git config`. Por ejemplo:

```
git config --global user.name matias
```

##### 6) Escribir el siguiente comando (con tu email):

```
$ git config user.email matias@ibserveis.com
```

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git config user.email jruizj70@alumnes.santjosepobrer.es
```

##### 7) Explica con tus palabras que pueden haber hecho el comando anterior,

Hemos creado un usuario llamado JoseRuizJurado, este hereda el valor del nombre de usuario, luego hemos establecido un email para dicho usuario

##### 8) Escribir el comando: `$git status`

¿Cuántos archivos hay "untracked"? ¿Qué significa?

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Ejercicio1.txt
  Ejercicio2.txt

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

Hay dos archivos que no estan siendo supervisados actualmente



"Untracked" ¿Qué significa en git?

Un archivo "untracked" es aquel que Git no está supervisando actualmente. Esto puede deberse a que es un archivo nuevo que nunca ha sido añadido al control de versiones, o puede ser un archivo que anteriormente estaba "tracked" pero fue eliminado del seguimiento.

9) Escribir el comando: `$git add Ejercicio1.txt`

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git add Ejercicio1.txt
```

10) Escribir el comando: `$git status`

¿Qué archivo no está "controlado"?

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Ejercicio1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Ejercicio2.txt
```

El archivo que no esta siendo controlado es el archivo cuyo nombre es ejercicio2.txt



¿Qué significa lo siguiente en git?

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

Este resultado de Git indica que el archivo **prueba.docx** se ha añadido al área de staging y está listo para ser confirmado (cometido) en el repositorio de Git.

La sección "Changes to be committed" muestra los archivos que se han añadido al área de staging y que están listos para ser confirmados. En este caso, hay un archivo nuevo llamado **prueba.docx** que se ha añadido al área de staging, lo que significa que Git está preparado para incluir este archivo en la próxima confirmación. Si se ejecuta el comando **git commit**, el archivo **prueba.docx** se confirmará y se guardará en el repositorio de Git.

11) Escribir el comando: `$git commit -m "Inicio texto, version 1"`

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git commit -m "Inicio texto, version 1"
[master (root-commit) 4833e0c] Inicio texto, version 1
1 file changed, 2 insertions(+)
 create mode 100644 Ejercicio1.txt
```

12) Escribir el comando: `$git status`

¿Qué significa lo que contesta Git ?

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Ejercicio2.txt

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

Git nos esta diciendo que no se ha agregado nada que nos permita confirmar el estado actual

### Explicación en video:

a) Configuraciones iniciales: [https://youtu.be/ISpTok3\\_Jqw](https://youtu.be/ISpTok3_Jqw)

b) Primer repositorio: <https://youtu.be/m4wh8GhzcYg>

### 13 ) Abrir documento Ejercicio1.txt ,escribir la siguiente frase;

Si buscas nuevo router puede que el Asus RT AC88U te interese.

Guardar.

### 14) Escribir el comando: `$git status`

¿Se ha dado cuenta que archivo se ha modificado?

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (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:   Ejercicio1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Ejercicio2.txt

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

### 15) Escribir los siguientes comandos al sistema Git: `$git add Ejercicio1.txt` `$git status`

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Ejercicio1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Ejercicio2.txt
```

### 16) Añadimos "Ejercicio2.txt" al sistema:

`$git add Ejercicio2.txt`

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git add Ejercicio2.txt
```

**17) Realizamos "commit" conjunto los archivos :**

`$git commit -m "Actualización Ejercicio1.txt y de Ejercicio2.txt"`

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git commit -m "Actualización Ejercicio1.txt y de Ejercicio2.txt"
[master 87668c3] Actualización Ejercicio1.txt y de Ejercicio2.txt
 2 files changed, 3 insertions(+), 2 deletions(-)
 create mode 100644 Ejercicio2.txt
```

**18) Escribir el comando:** `$git status`

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git status
On branch master
nothing to commit, working tree clean
```

**19) Escribir el comando:** `$git log`

Ahí esta nuestra "línea del tiempo" donde podemos retroceder a cada versión

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git log
commit 87668c3ba015de0e8449b3ecc379a76e6990075a (HEAD -> master)
Author: JoseRuizJurado <jruizj70@alumnos.santjosepobrer.es>
Date: Tue Mar 14 16:20:33 2023 +0100

    Actualización Ejercicio1.txt y de Ejercicio2.txt

commit 4833e0ce0b82d7514beb1b4d5e57102c094cab33
Author: JoseRuizJurado <jruizj70@alumnos.santjosepobrer.es>
Date: Tue Mar 14 16:13:00 2023 +0100

    Inicio texto, version 1
```

**17) Escribir el siguiente comando,**

substituyendo <aquí .... hash> por los 6 primeros caracteres del primer commit 'Inicio texto':

`$git checkout <aquí se escriben los 6 primeros n°del hash>`

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git checkout 4833e0
Note: switching to '4833e0'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 4833e0c Inicio texto, version 1
```

18) Ir a la carpeta de "Ejercicios" y comprobar que texto hay en los archivos- ¿Es el texto inicial?

Si, se trata del éxito inicial

19) Volver a colocar a línea del tiempo al último commit : `$ git checkout master`

En la carpeta Ejercicios., comprobar que texto tienen los archivos, en que estado se encuentran

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios ((4833e0c...))
$ git checkout master
Previous HEAD position was 4833e0c Inicio texto, version 1
Switched to branch 'master'
```

20) Utilizar esta otra forma git log : `$git log --pretty=oneline`

```
AzureAD+JoseRuizJurado@TM214-06 MINGW64 ~/Downloads/Ejercicios (master)
$ git log --pretty=oneline
87668c3ba015de0e8449b3ecc379a76e6990075a (HEAD -> master) Actualización Ejercici
o1.txt y de Ejercicio2.txt
4833e0ce0b82d7514beb1b4d5e57102c094cab33 Inicio texto, version 1
```