



Planear actividades de construcción del software de acuerdo con el diseño establecido.

Producción y compilación del contenido digital Inst. Angélica M. Triana
Solo fines académicos



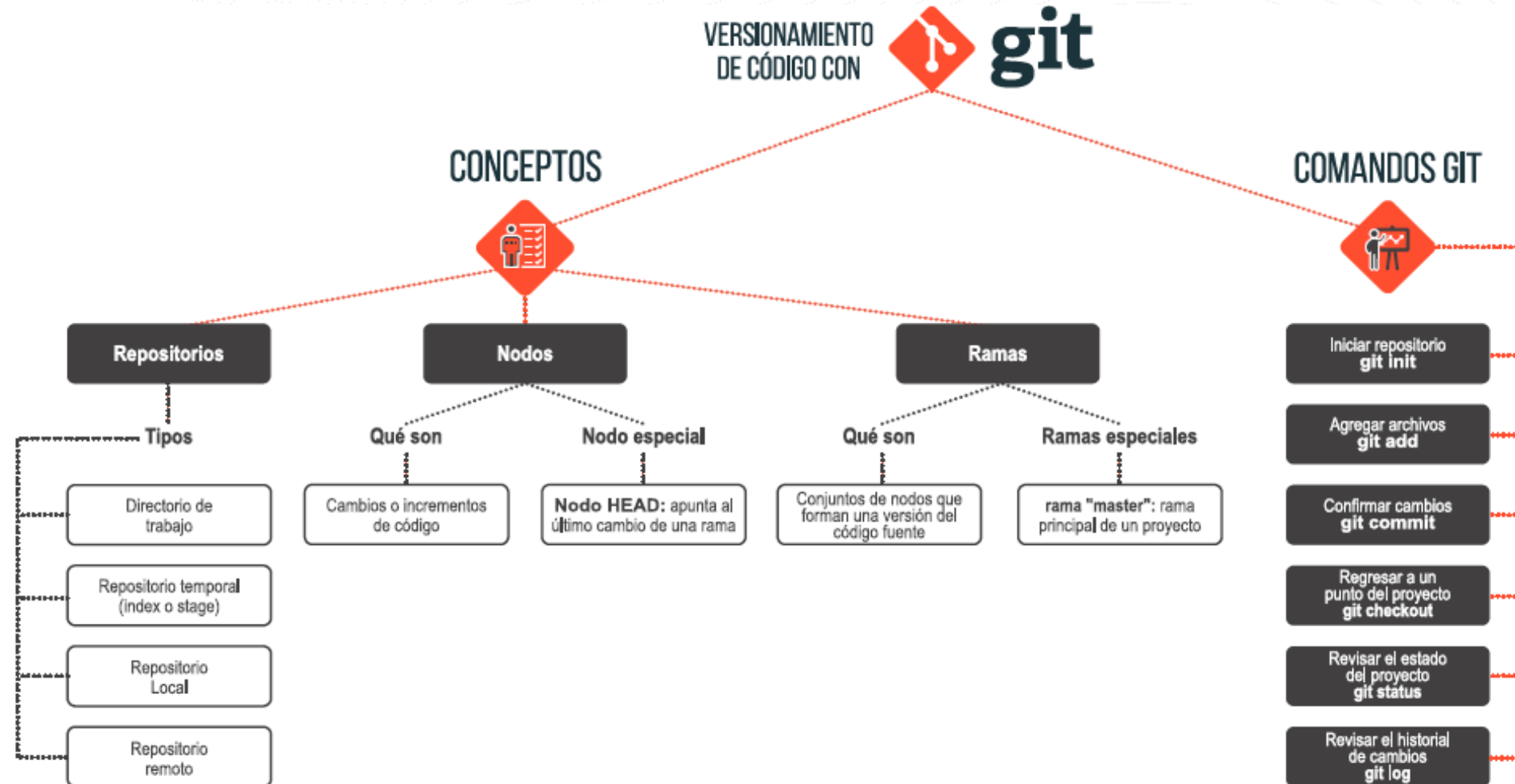
@SENAComunica

www.sena.edu.co

Contenido

Git-Github, conceptos.
Vincular git a un repositorio remoto
Actividades de la sesión

Sistema de control de versiones - Git



Laboratorio 1 (parte 1)



1. Instalar git: <https://git-scm.com/>
2. Si en la instalación lo pregunta, seleccionar: permitir usar Git desde la línea de comando y también se debe seleccionar la herramienta SSH (protocolo de seguridad).
3. Entre al cdm (símbolo del sistema de Windows)
4. Verificar la instalación: **git --versión**

```
C:\Users\Family>git --version
git version 2.35.1.windows.1
```

5. Crear la carpeta que contendrá el proyecto de sw. En este ejemplo se llamara proyecto_adso.

 > proyecto_adso

6. Retornando al cdm, seleccione la carpeta que contendrá el código al que se le hará el seguimiento:

Avanzar cd
Devolverse cd..

```
C:\Users\Family>cd desktop
C:\Users\Family\Desktop>cd proyecto_adso
C:\Users\Family\Desktop\proyecto_adso>
```

Paréntesis: Problema con el usuario



Si se presenta el inconveniente en donde no reconoce al usuario del equipo:

***** Please tell me who you are.**

Se debe configurar un usuario local para poder continuar y hacer el commit:

```
git config --global user.email "xyz@gmail.com"  
git config --local user.name "nombreUsuario"
```

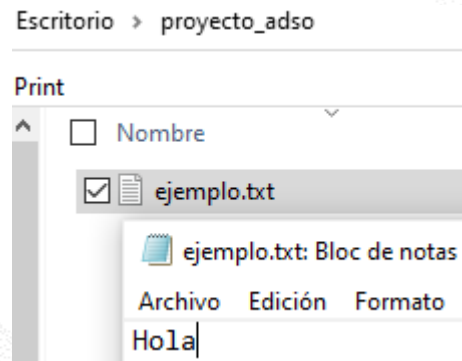
Verificar con **git config --list** que este el usuario.

Para volver a la línea de comando digitar la tecla: **Q**

Laboratorio 1 (parte 2)



7. En la carpeta del proyecto crea un documento llamado ejemplo.txt y escribe en el la palabra Hola y guarda el documento:



8. Luego se va a definir que archivo se va a controlar en el git. Con el ejemplo será: **git add** ejemplo.txt:

```
C:\Users\Family\Desktop\proyecto_adso>git init
Initialized empty Git repository in
C:/Users/Family/Desktop/proyecto_adso/.git/
C:\Users\Family\Desktop\proyecto_adso>git add ejemplo.txt
```

```
C:\Users\Family\Desktop\proyecto_adso>git init
Initialized empty Git repository in C:/Users/Family/Desktop/proyecto_adso/.git/
C:\Users\Family\Desktop\proyecto_adso>git add ejemplo.txt
```

Laboratorio 1 (parte 3)

9. Git no aplica automáticamente los cambios, sino que los almacena temporalmente en el repositorio o árbol “index”. Se puede revisar el estado de los cambios pendientes con el comando **git status**:

```
C:\Users\Family\Desktop\proyecto_adso>git status
On branch master

No commits yet

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

10. Para confirmar los cambios se usa el comando “git commit”. Este comando lo que hace es pasar los cambios de un repositorio “index” al repositorio definitivo o “head”. El comando “git commit” acepta comentarios usando la opción “-m” así: “git commit -m <mensaje>”. Para aplicar los cambios al ejercicio en curso se procede así: **git commit -m "version 1.0 original hola"**

```
C:\Users\Family\Desktop\proyecto_adso>git commit -m "version 1.0 original hola"
[master (root-commit) dc60fad] version 1.0 original hola
1 file changed, 1 insertion(+)
create mode 100644 ejemplo.txt
```


Laboratorio 1 (parte 4)



11. Revisar el historial de cambios con el comando “git log”:

```
C:\Users\Family\Desktop\proyecto_adso>git log
commit dc60fadf5ee79777a7f5bcc31ce3da3bf8156293 (HEAD -> master)
Author: amtriana62 <amtriana62@misena.edu.co>
Date: Sat Jul 27 15:17:28 2024 -0500
```

12. Realice una modificación en el documento. En este caso se agregará una línea de texto y guarde el documento:

 ejemplo.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Hola

Mi primera modificación seguimiento con git.

13. Guardada la modificación, ahora revise el status con **git status**:

```
C:\Users\Family\Desktop\proyecto_adso>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:   ejemplo.txt
```

¿Qué podemos concluir de lo que nos muestra el mensaje?

Laboratorio 1 (parte 5)



14. Se procederá primero a incluir los cambios en la cola de trabajo con el comando “**git add**” y revisar el estado del proyecto con el comando “**git status**”: **git add** ejemplo.txt y luego **git status**

```
C:\Users\Family\Desktop\proyecto_adso>git add ejemplo.txt  
  
C:\Users\Family\Desktop\proyecto_adso>git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    modified:   ejemplo.txt
```

15. Luego se confirman los cambios con el comando “**git commit**” agregándole un comentario sobre el cambio:

```
C:\Users\Family\Desktop\proyecto_adso>git commit -m "version 1.1 primer cambio"
```

```
C:\Users\Family\Desktop\proyecto_adso>git commit -m "version 1.1 primer cambio"  
[master 071e441] version 1.1 primer cambio  
1 file changed, 3 insertions(+), 1 deletion(-)
```

Laboratorio 1 (parte 6)



16. Por último, se revisa el historial de cambios con el comando “**git log**”: Los números de identificación son hash’s generados con el algoritmo SHA-1 (sello digital)

```
C:\Users\Family\Desktop\proyecto_adso>git log
commit 071e441156bbc3913ba1e17a7f3cec1c8e6c5f26 (HEAD -> master)
Author: amtriana62 <amtriana62@misena.edu.co>
Date: Sat Jul 27 15:48:14 2024 -0500

    version 1.1 primer cambio

commit dc60fadb5ee79777a7f5bcc31ce3da3bf8156293
Author: amtriana62 <amtriana62@misena.edu.co>
Date: Sat Jul 27 15:17:28 2024 -0500

    version 1.0 original hola
```

17. Otra forma de tener el id del commit de los cambios es la línea de comando **git log --oneline**:

```
C:\Users\Family\Desktop\proyecto_adso>git log --oneline
071e441 (HEAD -> master) version 1.1 primer cambio
dc60fad version 1.0 original hola
```

Tenga en cuenta los primeros 7 caracteres del commit, ya que le servirán para identificar la versión.

Laboratorio 1 (parte 7)



18. Para regresar o revisar un estado previo del proyecto se usa el comando “**git checkout id-commit**” (revise, debió volver al estado anterior donde solo estaba escrito “Hola”)

id-commit son los 7 primeros números del commit.

```
C:\Users\Family\Desktop\proyecto_adso>git checkout dc60fadf
Note: switching to 'dc60fadf'.

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:

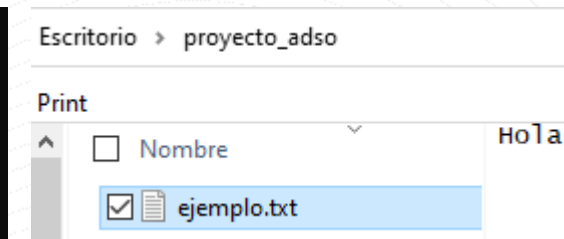
  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

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

HEAD is now at dc60fad version 1.0 original hola
```



Laboratorio 1 (parte 8)



19. Para retornar al último estado del proyecto se usa el comando “**git checkout master**”. (nuevamente, revise el documento para que observe en que estado se encuentra)

```
C:\Users\Family\Desktop\proyecto_adso>git checkout master
Previous HEAD position was dc60fad version 1.0 original hola
Switched to branch 'master'
```

Escritorio > proyecto_adso

Print

<input type="checkbox"/> Nombre	
<input checked="" type="checkbox"/> ejemplo.txt	Hola Mi primera modificación seguimiento con git.

20. Finalmente, realice un “**git log**” para revisar cual es la versión HEAD:

```
C:\Users\Family\Desktop\proyecto_adso>git log
commit 071e441156bbc3913ba1e17a7f3cec1c8e6c5f26 (HEAD -> master)
Author: amtriana62 <amtriana62@misena.edu.co>
Date: Sat Jul 27 15:48:14 2024 -0500

    version 1.1 primer cambio

commit dc60fADF5ee79777a7f5bcc31ce3da3bf8156293
Author: amtriana62 <amtriana62@misena.edu.co>
Date: Sat Jul 27 15:17:28 2024 -0500

    version 1.0 original hola
```

¿Qué se realizó?

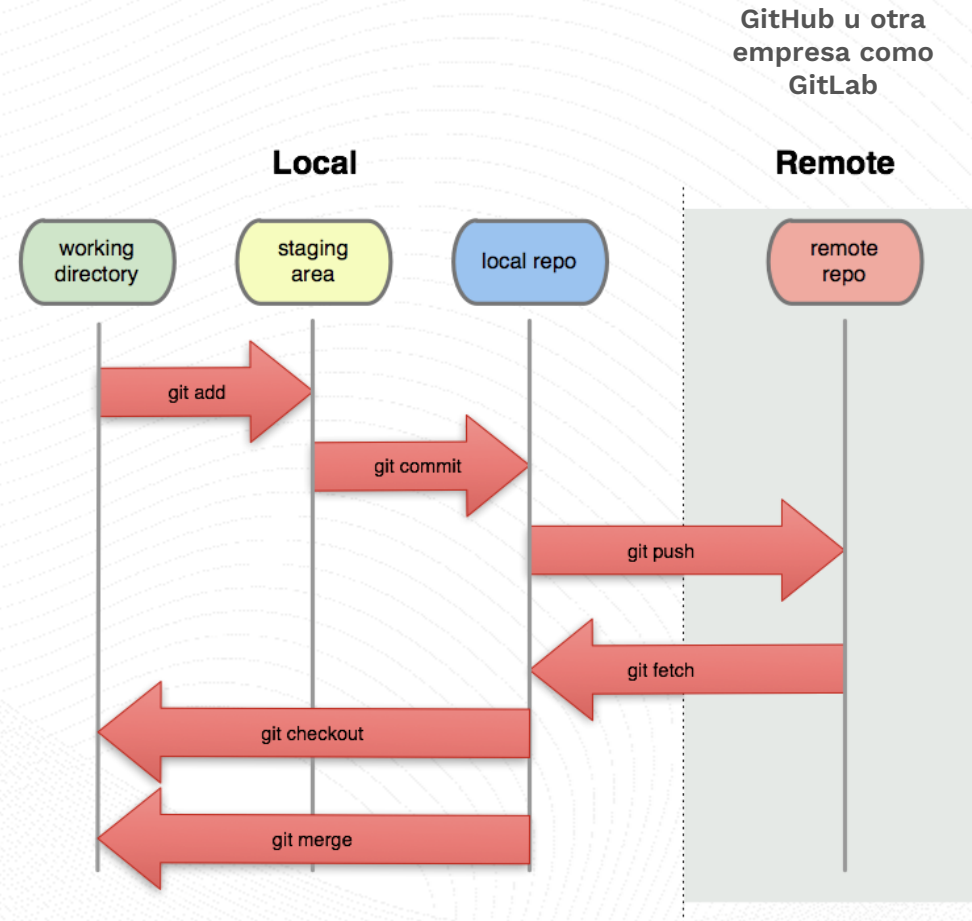
- ☐ Instalar git
- ☐ Crear un git
- ☐ Agregar los archivos al git
- ☐ Confirmar cambios con commit
- ☐ Navegar a través del historial de cambios (git checkout)

¿Qué podemos concluir del pequeño laboratorio?

Flujo de trabajo de Git

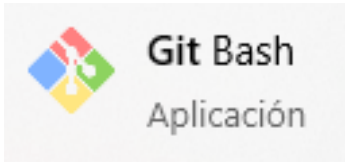


Tenemos nuestro directorio local (una carpeta en nuestro pc) con muchos archivos, Git nos irá registrando los cambios de archivos o códigos cuando nosotros le indiquemos, así podremos viajar en el tiempo retrocediendo cambios o restaurando versiones de código, ya sea en Local o de forma Remota (servidor externo).



"Trabajar en modo texto"

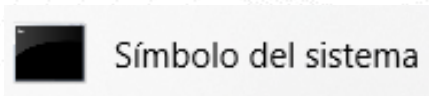
- Git Bash es una aplicación para entornos de Microsoft Windows que ofrece una capa de emulación para una experiencia de líneas de comandos de Git. (:q o :wq para regresar y guardar)



MINGW64:/c/Users/Family

```
Family@LAPTOP-QCE12K6U MINGW64 ~  
$ |
```

- Línea de comandos: Una terminal es un programa cuyo objetivo principal es leer comandos y ejecutar otros programas.



C:\> Símbolo del sistema

```
Microsoft Windows [Versión 10.0.19044.2604]  
(c) Microsoft Corporation. Todos los derechos reservados.  
C:\Users\Family>
```

Instalación previa (opcional)

- Visual Studio Code: <https://code.visualstudio.com/download>



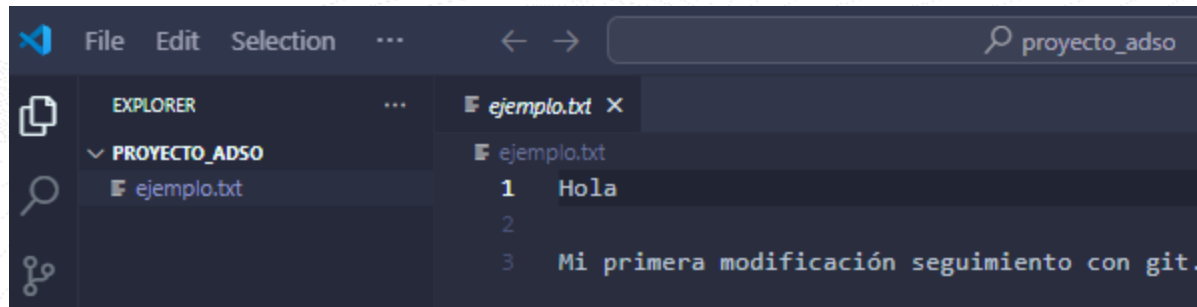
- Configurar idioma a Español: <https://youtu.be/b0E9tnIXo3E>

Uso del editor de código



En esta ocasión no se realizarán las líneas de comando por el cmd de Windows o el Git CMD o Bash. Se utilizará el editor de código fuente llamado Visual Studio Code.

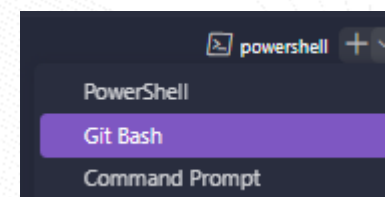
Para ello, abrimos Visual Studio Code, en File>Open folder buscamos la carpeta de proyecto_adso y seleccionamos el archivo ejemplo.txt:



Diríjase al menú Terminal>New terminal o seleccionamos el panel inferior:



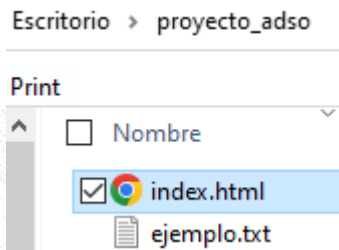
**Elegimos la terminal
de Git Bash.**



Laboratorio 2 (parte 1)



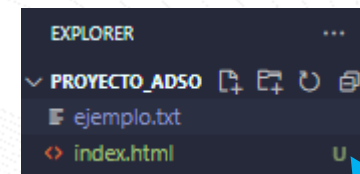
1. Revisar el estado del repositorio con **git status**
2. En la carpeta del proyecto_adso cree un bloc de notas y nombre al archivo index.html (fijese que la extensión sea .html al momento de guardarlo) le deberá preguntar por el cambio de la extensión, acepte. Se mostrará lo siguiente:



3. Ahora en la terminal de Visual, vuelva a ver el estado del repositorio:

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    index.html
```

Fijese en la parte izquierda del menú que el archivo indica un cambio.



Laboratorio 2 (parte 2)



4. Seleccione el archivo index.html en Visual Studio y edite el archivo agregado lo siguiente: html y seleccione html:5 y en medio de las etiquetas <body> escriba <h1>Prueba </h1> y guarde el archivo File>Save.

A screenshot of the Visual Studio code editor interface. On the left, the 'EXPLORER' sidebar shows a project named 'PROYECTO_AD50' with two files: 'ejemplo.txt' and 'index.html'. The 'index.html' file is selected and highlighted. The main editor area shows the content of 'index.html' with line numbers 1 through 11. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <h1>Prueba </h1>
10 </body>
11 </html>
```

5. Ahora prepare el archivo para adjuntarlo al repositorio con **git add index.html**, fíjese que el archivo en el menú izquierdo ahora tiene la letra A del add.

Nota. Si desea limpiar la pantalla de la terminal, escriba **clear**, con las flechas de arriba y abajo puede ver el los comandos que ha escrito.

Laboratorio 2 (parte 3)

6. // Crear commit (fotografía del proyecto en ese momento) del archivo index.html

`git commit -m "version original estructura html5"`

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (master)
$ git commit -m "version original estructura html5"
[master 0e45ef6] version original estructura html5
1 file changed, 11 insertions(+)
create mode 100644 index.html
```

Consejos: un archivo a la vez para la fotografía, un add y luego un commit. En cuanto a mensajes>

```
git commit -m "Add new search feature." # ✗
git commit -m "Fix a problem with the topbar..." # ✗
git commit -m "Change the default system color" # ✓
```

No colocar tildes ni caracteres especiales, ni puntos finales.

Observe los archivos del menú izquierdo, ya no muestran alguna indicación o letra en particular. Si necesitamos hacer un seguimiento de todos los archivos de la carpeta y no tengamos que agregarlos uno por uno se ingresa **git add** .

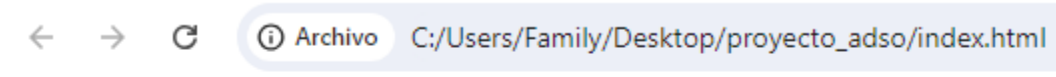
// Agregar todos los archivos para que esté pendiente de los cambios
`git add` .

Laboratorio 2 (parte 4)



Cambios y git log --online

- Abra el archivo index.html en el browser, verifique que se muestre lo siguiente:



Prueba

- Se procede a hacer un cambio en el archivo index.html, sin guardar, agregue `<p>Mi nombre</p>`:

```
8  <body>
9      <h1>Prueba </h1>
10 |  <p>Mi nombre</p>
11  </body>
12  </html>
```

- Posteriormente, ejecute en la terminal **git status -s** (observe las letra roja)

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (master)
$ git status -s
M index.html
```

Laboratorio 2 (parte 5)



Cambios y git log --online

10. Repetir el proceso: Guarde el archivo con los cambios realizados y envíelo al área temporal (staging area) con **git add** .

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (master)
$ git add .
```

11. Confirme los cambios al repositorio con un git commit.

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (master)
$ git commit -m "se agrega parrafo con mi nombre"
[master 3a04309] se agrega parrafo con mi nombre
1 file changed, 1 insertion(+)
```

Muestra cuantas inserciones o cambios se realizaron en las líneas.

// Muestra la lista de commit del mas reciente al más antiguo **git log**

12. Aplique el comando **git log --oneline**

// Muestra en una línea los commit realizados.

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (master)
$ git log --oneline
3a04309 (HEAD -> master) se agrega parrafo con mi nombre
0e45ef6 version original estructura html5
071e441 version 1.1 primer cambio
dc60fad version 1.0 original hola
```

Identificador –primeros 7- que nos sirve para viajar a través de los commit.

Laboratorio 2 (parte 6)



git reset --hard

13. Cree un archivo nuevo en la carpeta proyecto_adso llamado ejemplo2.txt
14. Agréguelo al repositorio y confírmelo con un mensaje “archivo vacio”.
15. Ejecute un **git log --oneline** y debe mostrarse algo similar a lo siguiente:

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (master)
• $ git log --oneline
b1f287a (HEAD -> master) archivo vacio
3a04309 se agrega parrafo con mi nombre
0e45ef6 version original estructura html5
071e441 version 1.1 primer cambio
dc60fad version 1.0 original hola
```

Identifique en que punto de la historia se encuentra con el HEAD.

16. Si viajemos a un commit en específico X#XxxX#, podremos eliminamos los cambios futuros con **git reset --hard**

En este caso nos devolveremos al commit antes de crear el archivo ejemplo2.txt

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (master)
• $ git reset --hard 3a04309
HEAD is now at 3a04309 se agrega parrafo con mi nombre
```

Si revisa la carpeta de proyecto_adso, el archivo ejemplo2.txt ya no debe mostrarse.

Laboratorio 2 (parte 7)



git reset --hard

17. Revise nuevamente el historial con `git log --oneline`

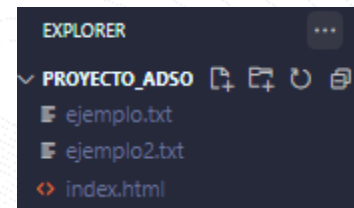
```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (master)
$ git log --oneline
3a04309 (HEAD -> master) se agrega parrafo con mi nombre
0e45ef6 version original estructura html5
071e441 version 1.1 primer cambio
dc60fad version 1.0 original hola
```

**Ya no esta el otro archivo creado.
Estamos posicionados en otro
momento de la historia.**

18. Para volver a la versión que tenia el archivo anterior utilice `git reset --hard` con el id-commit que se registro cuando existía ejemplo2.txt:

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (master)
$ git reset --hard b1f287a
HEAD is now at b1f287a archivo vacio
```

El archivo se muestra nuevamente en la carpeta:



Nota. Debe guardar los id-commit. Si quiere ver los cambios en los commit escriba `git log -p`, salga de ese modo con la letra q

Actividad de la sesión



Diferencias entre checkout, revert y reset



De manera individual,

Realizar la siguiente lectura: <https://www.aunitz.net/diferencias-checkout-revert-reset/>

Responda: ¿Cuales son sus diferencias?

GitHub



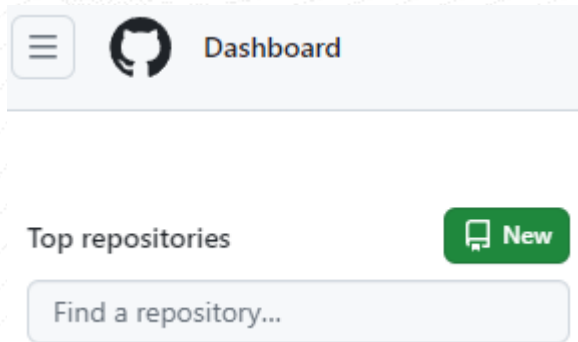
Respaldar los archivos en la nube: Cuenta en GitHub

Utilizaremos GitHub (empresa externa).

Cree una cuenta:

https://github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home

En home-dashboard, en el menú izquierdo seleccione New para crea un nuevo repositorio:



Crear el repositorio en la nube



Diligencie el formulario para crear un nuevo repositorio en la nube:

Create a new repository

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

[Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 amtriana62 ▾

Repository name *

/ proyecto_adso2024

✔ proyecto_adso2024 is available.

Un nombre sin espacio, no se puede repetir.

Great repository names are short and memorable. Need inspiration? How about [super-octo-disco](#) ?

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.

Quienes pueden ver el repositorio.

Create repository

Clic aquí para finalizar.

Vincular al repositorio remoto

Diríjase a la siguiente sección:

...or push an existing repository from the command line

```
git remote add origin https://github.com/amtriana62/proyecto_adso2024.git
git branch -M main
git push -u origin main
```

Copie cada línea en orden y una por una ejecútela en el Terminal de Visual Studio Code (clic derecho para pegar), fíjese que se encuentre en la carpeta del repositorio local que desea vincular al remoto:

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (master)
• $ git remote add origin https://github.com/amtriana62/proyecto_adso2024.git

Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (master)
• $ git branch -M main

Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (main)
• $ git push -u origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (15/15), 1.37 KiB | 200.00 KiB/s, done.
Total 15 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/amtriana62/proyecto_adso2024.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

El último comando puede tardar un poco mientras envía los archivos del repositorio local al repositorio remoto.

Explicación



Enviar el repositorio local a remoto:

`git remote add origin https://github.com/amtriana62/proyecto_adso2024.git` **Va a empezar a**

`git branch -M main` **Se crea la rama principal.**

`git push -u origin main` **Es posible que en esta línea les solicite iniciar sesión o usuario y contraseña, si no es así, ya lo ha identificado.**

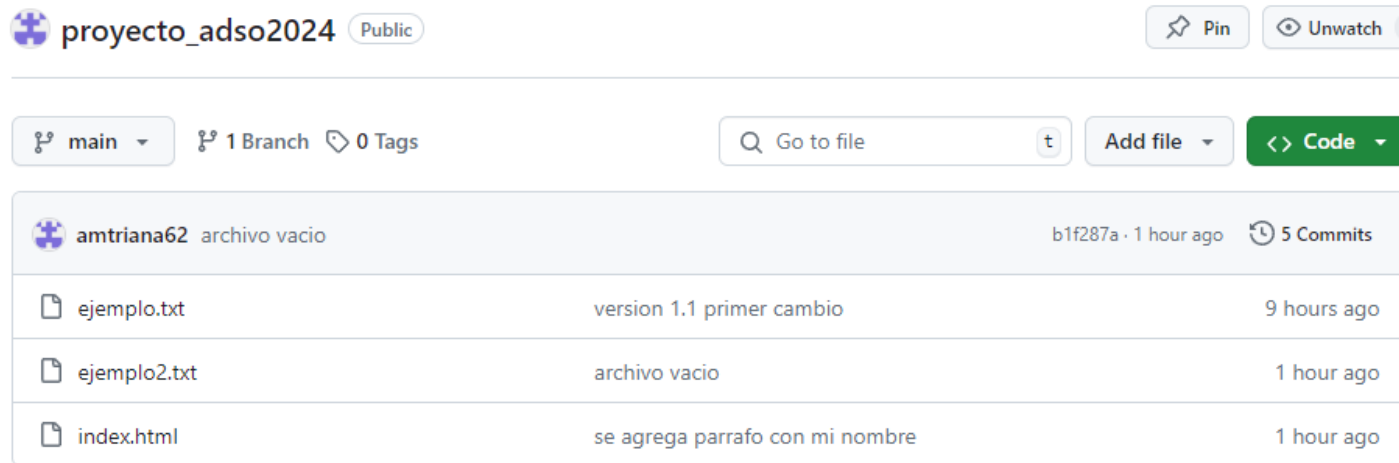
trabajar de forma remota. No se vuelve a colocar estos comandos porque queda asociada la url.

Nota: Si tiene problemas con el usuario desde la primera línea de comando y al parecer ya se encuentra uno activo, busque Administrado de credenciales>Credenciales de Windows>git:https>Seleccione Quitar y vuelva a intentar ejecutar.

Verificación de los commits

Regrese al repositorio en la nube y actualice la página F5.

Puede observar que se muestra el historial o logs y sus archivos correspondientes que tenía en el repositorio local:



The screenshot shows a GitHub repository interface. At the top, the repository name 'proyecto_adso2024' is displayed with a 'Public' badge. Below this, there are buttons for 'Pin' and 'Unwatch'. The main section shows the 'main' branch with '1 Branch' and '0 Tags'. A search bar 'Go to file' is present. Below the search bar, there is a table of files and their commit history.

File	Commit Message	Time Ago
ejemplo.txt	version 1.1 primer cambio	9 hours ago
ejemplo2.txt	archivo vacio	1 hour ago
index.html	se agrega parrafo con mi nombre	1 hour ago

Muestra cuantos commits se han realizado.

Para ver más detalles de clic allí y podrá ver los id-commit.

Laboratorio 3 (parte 1)



Verificar vínculo con nuevos cambios.

1. Ahora realice alguna modificación de un archivo de los que tiene en el **repositorio local**, en este caso en el archivo index.html:

```
8 <body>
9   <h1>Prueba </h1>
10  <p>Mi nombre</p>
11  <p>Este es un párrafo para ver a GitHub</p>
12 </body>
13 </html>
```

2. Agregue y confirme los cambios con los comandos antes señalados para estas acciones.
3. Verifique las modificaciones realizadas con un `git log --online`

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (main)
$ git log --oneline
5d26633 (HEAD -> main) se agrega parrafo para ver a github
b1f287a (origin/main) archivo vacio
3a04309 se agrega parrafo con mi nombre
0e45ef6 version original estructura html5
071e441 version 1.1 primer cambio
dc60fad version 1.0 original hola
```

(origin/main) es hasta donde el repositorio local con el remoto tiene su historia.

Laboratorio 3 (parte 2)



git push

4. Ejecute el comando **git push**

// Con el comando git push estaremos subiendo todos los cambios locales al servidor remoto de GitHub, tenga en cuenta que debe que estar enlazado con su repositorio.

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (main)
• $ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 351 bytes | 351.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/amtriana62/proyecto_adso2024.git
b1f287a..5d26633 main -> main
```

Recuerde que para subir los cambios al repositorio remoto debieron estar “comiteados” en el local.

Puede demorar un poco mientras sube los nuevos cambios.

5. Revise nuevamente el log, el (origin/main) y el HEAD se encuentra en la misma posición:

```
Family@LAPTOP-QCE12K6U MINGW64 ~/Desktop/proyecto_adso (main)
• $ git log --oneline
5d26633 (HEAD -> main, origin/main) se agrega parrafo para ver a github
b1f287a archivo vacío
```

Laboratorio 3 (parte 3)



6. Finalmente, diríjase al repositorio remoto, actualice la pagina y se deberá evidencia el aumento de commits:

A screenshot of a GitHub repository page for 'proyecto_adso2024'. The repository is public and has 1 branch and 0 tags. The commit history shows three commits. The most recent commit, by user 'amtriana62', has the message 'se agrega parrafo para ver a github' and is timestamped '11 minutes ago'. This commit includes three files: 'ejemplo.txt' (version 1.1 primer cambio, 10 hours ago), 'ejemplo2.txt' (archivo vacio, 1 hour ago), and 'index.html' (se agrega parrafo para ver a github, 11 minutes ago). A blue arrow points from the commit message of the most recent commit to the 'index.html' file entry. The commit hash '5d26633' is also visible next to the commit message.

proyecto_adso2024 Public

main 1 Branch 0 Tags

Go to file t Add file <> Code

amtriana62	se agrega parrafo para ver a github	5d26633 · 11 minutes ago	6 Commits
ejemplo.txt	version 1.1 primer cambio	10 hours ago	
ejemplo2.txt	archivo vacio	1 hour ago	
index.html	se agrega parrafo para ver a github	11 minutes ago	

Aumenta

Felicidades! Logro vincular su repositorio local con uno remoto.

¿Qué se realizó?

- ☐ Instalar Visual Studio Code
- ☐ Viajar en el tiempo de un repositorio con la opción `git reset --hard`, `git checkout`.
- ☐ Crear cuenta en GitHub
- ☐ Vincular el repositorio local o control de versiones con el respaldo en la nube.
- ☐ Enviar actualizaciones o cambios con `git push`.

¿Qué podemos concluir o que dudas se generaron de este pequeño laboratorio 3?

Actividad de la sesión



Actividad de la sesión



Asocie las definiciones con los comandos:

git add

Captura una instantánea de los cambios preparados en ese momento del proyecto. Las instantáneas confirmadas pueden considerarse como versiones "seguras" de un proyecto.

git commit

git remote

Se usa para cargar contenido del repositorio local a un repositorio remoto. El envío es la forma de transferir confirmaciones desde tu repositorio local a un repositorio remoto.

Indica a Git que quieres incluir actualizaciones en un archivo concreto en la próxima confirmación.

git push

Es una herramienta de gestión para el registro de repositorios remotos. Esto te permite guardar largas direcciones URL como cortos manejadores (handles), tales como 'origin', para que no tengas que escribir las URL todo el tiempo.


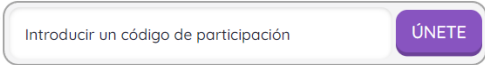
Actividades de la sesión



Quiz conceptos git



De manera individual,

- Ingrese a la plataforma de Quizizz 
- Escriba el código de participación: 
- Escriba primer nombre y primer apellido registrado en sofíaPlus.

En grupos de proyecto



Desarrollar tres (3) documentos con buena presentación, ortografía, estilo, glosario de términos y por supuesto bibliografía.

- El primer documento será sobre GitHub Desktop donde deberán registrar el paso a paso de la instalación y el tutorial que se da la primera vez cuando se ejecuta el programa, por favor describa lo que sucede en las imágenes y en las mismas señale las opciones utilizadas, finalmente de un ejemplo, del uso de esta herramienta.
- En el segundo documento será sobre GitHub en la nube, expliquen para que sirve o el propósito de cada una de las opciones allí mostradas –priorice por importancia.
- Instructivo paso a paso con capturas de pantalla de la creación del repositorio del proyecto formativo con los colaboradores correspondientes que conforman el grupo.

Finalmente entregar las evidencias en espacio indicado por el instructor.

Referencias



- Cantoral, C. (2017). ¿Que es una guía de código? Recuperado de: https://codigofacilito.com/articulos/guia_codigo
- Bose, S. (2021). Coding Standards and Best Practices To Follow. Recuperado de <https://www.browserstack.com/guide/coding-standards-best-practices#:~:text=Practices%20To%20Follow-,What%20are%20Coding%20Standards%3F,sophisticated%20and%20highly%20functional%20code.>
- Bluuweb. (2019) GIT / GITHUB [Tutorial en Español - Parte 1] ♥ Inicio Rápido para Principiantes ♥. Recuperado de: <https://youtu.be/hWglK8nWh60>
- EDteam (2017). ¿Cómo se deciden las versiones del software?. Recuperado de <https://ed.team/blog/como-se-deciden-las-versiones-del-software>
- Kumar, S. (2019). Software coding and testing. Recuperado de <https://www.slideshare.net/SandeepKumarNayak1/software-coding-and-testing>
- La rosa, A. (2021). Versionado de Software desde el punto de vista del usuario. Recuperado de <https://pandorafms.com/blog/es/versionado-de-software/>
- Santos, P. (2010) La historia de control de versiones. Recuperado de: <http://codicesoftware-es.blogspot.com/2010/11/la-historia-del-control-de-versiones.html>



GRACIAS

Línea de atención al ciudadano: 01 8000 910270
Línea de atención al empresario: 01 8000 910682



www.sena.edu.co