


Taller Unidad 1 GIT

1. Crear un repositorio local en su sistema de archivos con el proyecto de ejemplo Taller0003Git proporcionado en clase, el cual contiene el código de una “Calculadora” desarrollada en JavaScript, una vez creado el repositorio crear un Commit inicial.
 - a. Primero debemos inicializar el repositorio local con el comando “git init”

 MINGW64:/d/diplomado/modulo 1/modulo git/Taller0003Git

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git
$ git init
Initialized empty Git repository in D:/diplomado/modulo 1/modulo git/Taller0003Git/.git/

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
```

- b. En el segundo paso debemos verificar los archivos que tenemos listos para agregarlos, esto lo hacemos con el comando “git status”

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git status
On branch main

No commits yet

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

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

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$
```

- c. Proseguimos agregando los archivos al área de intercambio, esto lo hacemos con el comando “git add .” o bien podemos agregar cada archivo individualmente con el comando “git add css/”, “git add js/”, “git add index.html”

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git add .

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$
```

- d. Verificamos los archivos que fueron agregados y están listos para hacer el commit, lo hacemos con el comando “git status”

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   css/style.css
        new file:   index.html
        new file:   js/script.js

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ |
```

- e. Ahora debemos hacer el commit, con el siguiente comando “git commit -m “descripción””, colocando una descripción del commit que vamos a realizar.

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git commit -m "primer commit"
[main (root-commit) 7bd1b94] primer commit
3 files changed, 238 insertions(+)
create mode 100644 css/style.css
create mode 100644 index.html
create mode 100644 js/script.js

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ |
```

2. Crear un repositorio remoto en GitHub con el mismo nombre del repositorio local es decir “Taller0003Git”, enviar el avance del repositorio local al remoto.
- a. Primero creamos un nuevo repositorio en nuestra cuenta personal de Github, con el nombre Taller003git y lo dejamos como proyecto público.

Crear un nuevo repositorio

Un repositorio contiene todos los archivos del proyecto, incluido el historial de revisiones. ¿Ya tienes un repositorio de proyectos en otro lugar? [Importar un repositorio.](#)

Los campos obligatorios están marcados con un asterisco (*).

Dueño *

 Miller98

Nombre del repositorio *

Taller003git

 **Taller003git está disponible.**

Los grandes nombres de repositorios son breves y fáciles de recordar. ¿Necesitas inspiración? Qué tal si [gofre-didáctico](#) ?

Descripción (opcional)

proyecto calculadora en javascript

☒  **Público**
Cualquiera en Internet puede ver este repositorio. Tú eliges quién puede comprometerse.

☐  **Privado**
Tú eliges quién puede ver y comprometerse con este repositorio.

- b. Ahora conectamos el repositorio local al remoto con el siguiente comando “git remote add origin https://github.com/Miler98/Taller003git.git”

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git remote add origin https://github.com/Miler98/Taller003git.git

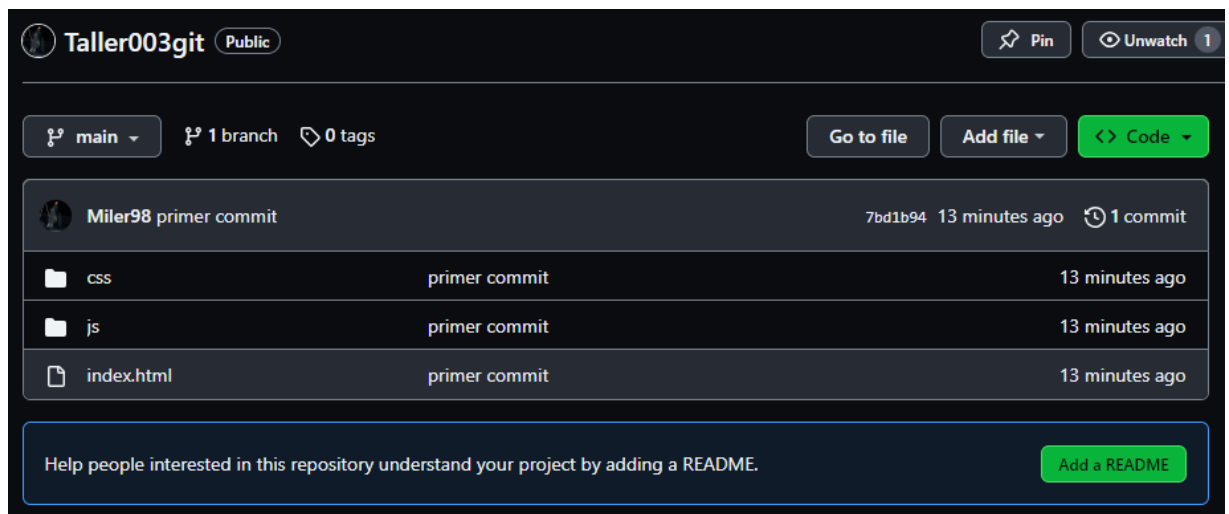
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ |
```

- c. Finalmente subimos el archivo local al remoto con el siguiente comando “git push -u origin main”

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 2.51 KiB | 1.26 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Miler98/Taller003git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$
```

- d. Verificamos en Github que los cambios se hayan subido correctamente



The screenshot shows the GitHub interface for a repository named 'Taller003git', which is marked as 'Public'. At the top, there are buttons for 'Pin' and 'Unwatch'. Below this, the repository's branch structure is shown: 'main' (selected), '1 branch', and '0 tags'. There are buttons for 'Go to file', 'Add file', and 'Code'. The main content area displays a commit by 'Miler98' titled 'primer commit' with hash '7bd1b94', made '13 minutes ago'. Below the commit, a list of files is shown: 'css', 'js', and 'index.html', all from the 'primer commit' and added '13 minutes ago'. At the bottom, there is a prompt to 'Add a README' to help people understand the project.

3. En el repositorio local crear una rama llamada “diseno”, en ella actualizar las referencias a Bootstrap en su versión más actual y modificar el diseño visual del proyecto de tal forma que tenga una interfaz gráfica más agradable, realizar los commit que sean necesarios.

- a. primero debemos crear la nueva rama llamada diseno con el siguiente comando “git Branch diseno” y cambiarnos a la rama diseño recién creada con el comando “git checkout diseno”

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git branch diseno

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git checkout diseno
Switched to branch 'diseno'
```

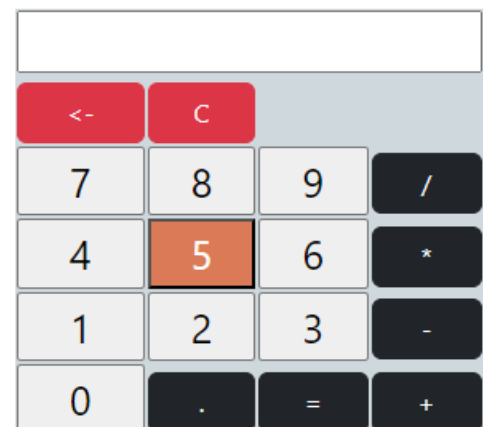
- b. verificamos que nos cambiamos a la rama diseno recién creada con el comando “git Branch”, la cual nos indicara con un asterisco la rama en la cual estamos ubicados.

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (diseno)
$ git branch
* diseno
main

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (diseno)
$
```

- c. realizamos algunos cambios en el diseño de la calculadora usando Bootstrap y Css y se vera de la siguiente manera

Calculadora JavaScript



Editado por Miler España.

- d. Ahora haremos el commit con la nueva interfaz de la calculadora con los comandos ya vistos anteriormente

```
miller@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (diseno)
$ git status
On branch diseno
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:   css/style.css
        modified:   index.html

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

miller@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (diseno)
$ git add .

miller@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (diseno)
$ git status
On branch diseno
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   css/style.css
        modified:   index.html

miller@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (diseno)
$ git commit -m "cambio de diseño desde la rama diseno"
[diseno ff4c90a] cambio de diseño desde la rama diseno
 2 files changed, 12 insertions(+), 7 deletions(-)

miller@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (diseno)
$
```

4. Una vez terminado el diseño realizar un merge (fusión) de los cambios realizados en la rama "diseno" a la rama "main".
- a. Primero debemos volver a la rama principal (main) para hacer el merge con el comando "git checkout main"

```
miller@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (diseno)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

miller@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ |
```

- b. Ahora, verificamos que realmente estamos en la rama main para hacer el merge con el siguiente comando "git Branch"

```
miller@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git branch
  diseno
* main ←
miller@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ |
```

- c. Ahora procedemos a hacer el merge con el siguiente comando “git merge diseno”

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git merge diseno
Updating 7bd1b94..ff4c90a
Fast-forward
 css/style.css | 7 +++++-
 index.html    | 12 ++++++-----
 2 files changed, 12 insertions(+), 7 deletions(-)

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ |
```

5. Crear una rama llamada “funcionalidad” en la cual agregaremos un botón al final de la GUI del proyecto que me permita convertir grados Centígrados a Fahrenheit, realizar en esta rama los commit que sean necesarios.

- a. primero debemos crear la nueva rama llamada funcionalidad con el siguiente comando “git Branch funcionalidad” y cambiarnos a la rama diseño recién creada con el comando “git checkout funcionalidad”

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git branch funcionalidad

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git checkout funcionalidad
Switched to branch 'funcionalidad'

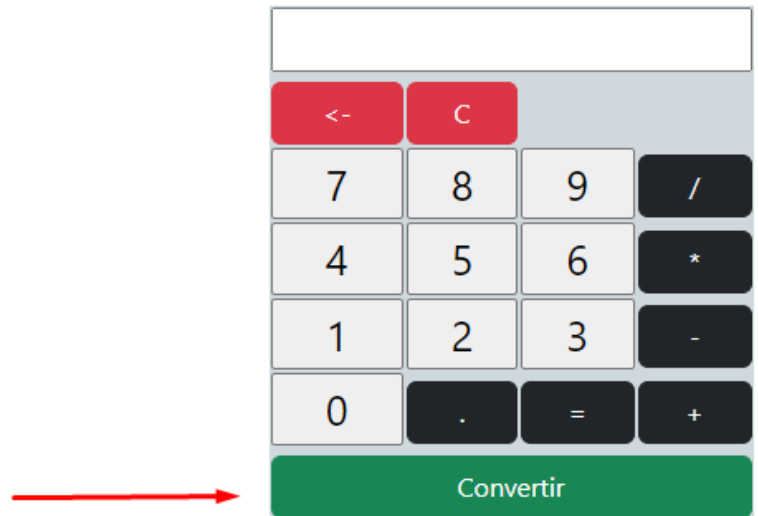
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (funcionalidad)
$ |
```

- b. verificamos que nos cambiamos a la rama funcionalidad recién creada con el comando “git Branch”, la cual nos indicara con un asterisco la rama en la cual estamos ubicados.

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (funcionalidad)
$ git branch
diseno
* funcionalidad
main

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (funcionalidad)
$ |
```

- c. agregamos un nuevo botón en el diseño de la calculadora y su respectiva funcionalidad en el archivo script.js, la cual se vera de la siguiente manera



Editado por Miler España.

- d. Ahora haremos el commit con la nueva funcionalidad de la calculadora con los comandos ya vistos anteriormente.

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (funcionalidad)
$ git status
On branch funcionalidad
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:   index.html
    modified:   js/script.js

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

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (funcionalidad)
$ git add .

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (funcionalidad)
$ git status
On branch funcionalidad
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   index.html
    modified:   js/script.js

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (funcionalidad)
$ git commit -m "tercer commit, funcionalidad de calculadora con javascript"
[funcionalidad c07448b] tercer commit, funcionalidad de calculadora con javascript
 2 files changed, 11 insertions(+), 11 deletions(-)

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (funcionalidad)
$ |
```

6. Una vez terminada la funcionalidad realizar un merge (fusión) de los cambios realizados en la rama "funcionalidad" a la rama "main".

- a. Primero debemos volver a la rama principal (main) para hacer el merge con el comando "git checkout main"

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (funcionalidad)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$
```

- b. Ahora, verificamos que realmente estamos en la rama main para hacer el merge con el siguiente comando "git Branch"

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git branch
diseno
funcionalidad
* main ←
```

- c. Nos disponemos a hacer el merge con el siguiente comando "git merge funcionalidad"

```
miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git merge funcionalidad
Updating ff4c90a..c07448b
Fast-forward
 index.html | 2 --
 js/script.js | 20 ++++++++-----
 2 files changed, 11 insertions(+), 11 deletions(-)

miler@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$
```

7. Actualizar el repositorio remoto y realizar un pequeño informe presentado de manera individual en el cual se evidencie por medio de capturas de pantalla el trabajo realizado, el informe debe ser almacenado como un archivo PDF dentro de la carpeta principal del proyecto con el nombre "informe.pdf"

- a. Finalmente actualizaremos el repositorio remoto con las actualizaciones que hemos realizado con el siguiente comando "git push -u origin main"


```
miller@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ git push -u origin main
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 1.07 KiB | 1.07 MiB/s, done.
Total 10 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To https://github.com/Miler98/Taller003git.git
   7bd1b94..c07448b  main -> main
branch 'main' set up to track 'origin/main'.

miller@DESKTOP-I96PSVH MINGW64 /d/diplomado/modulo 1/modulo git/Taller0003Git (main)
$ |
```

- b. Evidenciamos que los cambios se hayan guardado en Github.