

Universidad de Nariño.

Ingeniería de Sistemas.

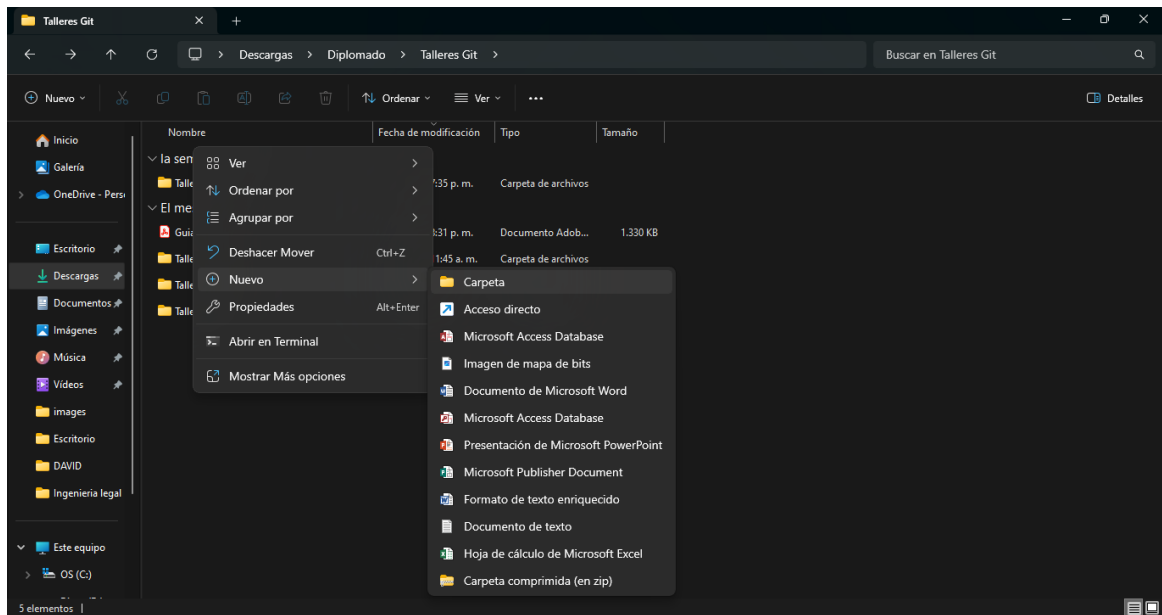
Diplomado de actualización en nuevas tecnologías para el desarrollo de Software.

Estudiante: David Alejandro Rodríguez Acosta

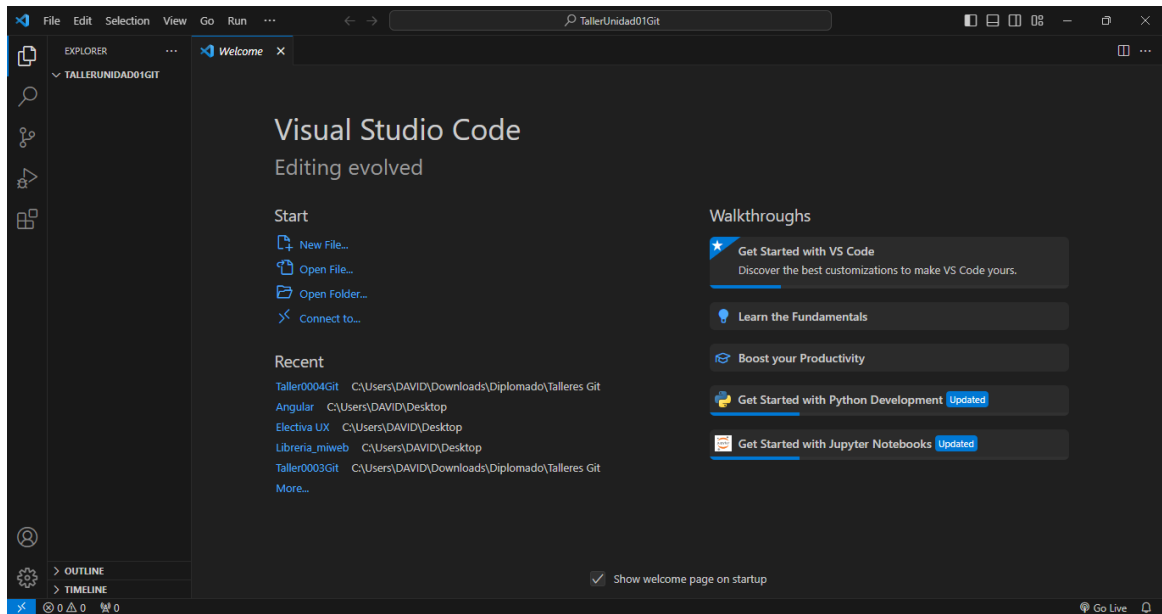
Taller Unidad 1 GIT.

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.

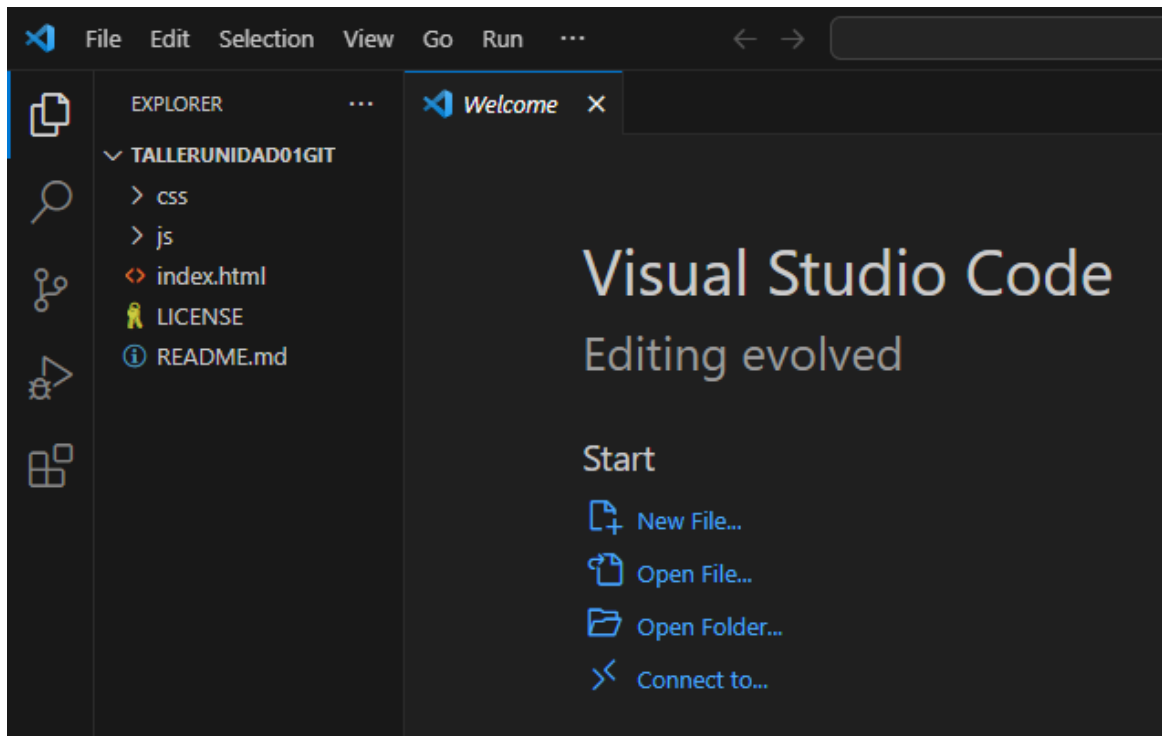
Para crear el repositorio local primero creamos una carpeta la cual vaya a contener nuestro proyecto.



Después abrimos Visual Studio Code y abrimos la carpeta que acabamos de crear.



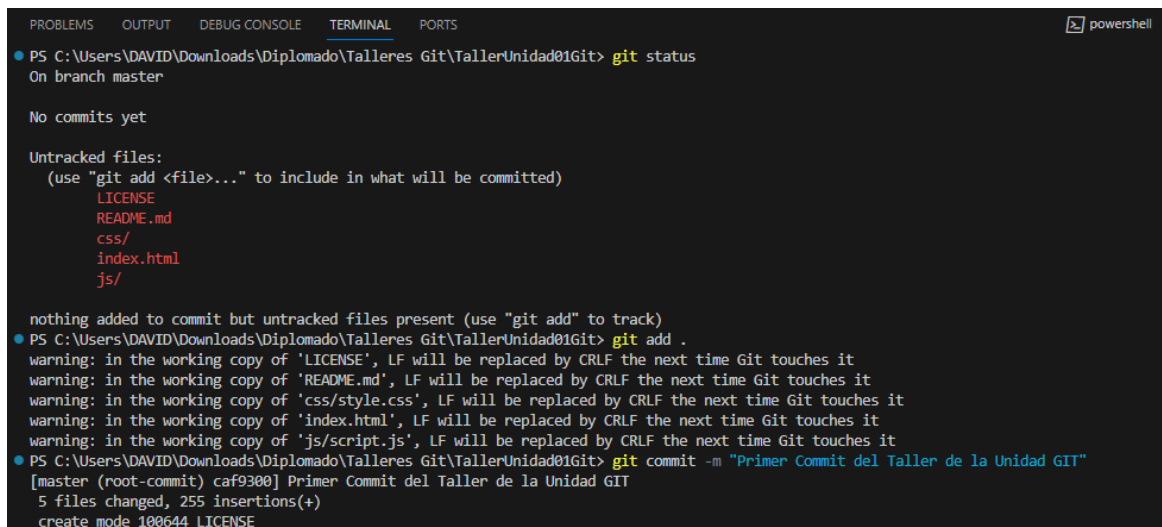
Posteriormente copiamos a la carpeta los archivos con el código proporcionado en clase de la calculadora.



Después inicializamos una terminal dentro de Visual Studio Code y dentro de la terminal escribimos el comando “**git init**” con el fin de crear el repositorio de forma local

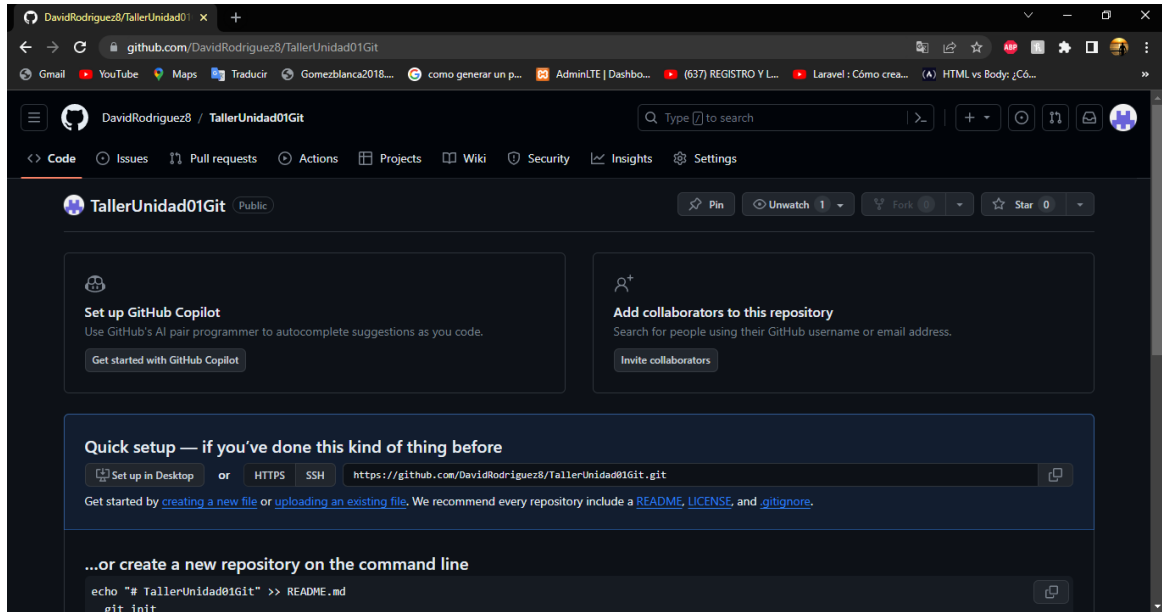


Posterior a esto realizaremos el commit inicial, para ello vemos el estado de los archivos con “**git status**” y luego si los archivos aparecen de color rojo es porque necesitamos añadirlos al repositorio local, para eso escribiremos “**git add .**” a continuación haremos el commit inicial con el comando “**git commit -m <“mensaje opcional”>**”



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.

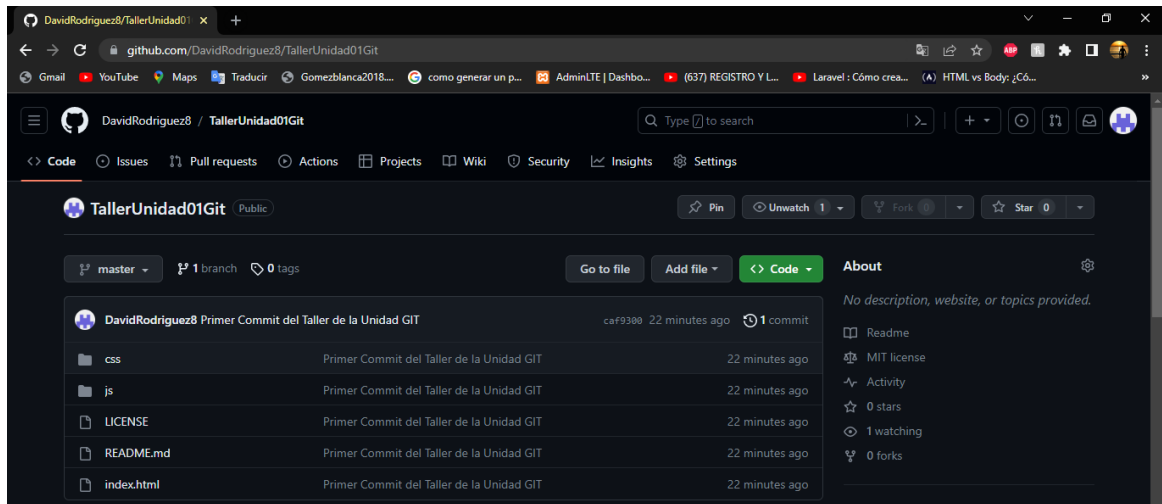
En este caso el repositorio remoto va a llamarse igual que la carpeta creada, con el fin de diferenciar el taller realizado en clase al taller de la unidad es decir se llamará “TallerUnidad01Git”



Ahora conectamos el repositorio local con el remoto y enviamos el avance de la siguiente manera, escribimos en la terminal **“git remote add origin https://github.com/...”** y a continuación escribimos **“git push origin master”**

```
PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git remote add origin https://github.com/DavidRodriguez8/TallerUnidad01Git.git
PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git push origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 3.45 KiB | 442.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DavidRodriguez8/TallerUnidad01Git.git
 * [new branch]      master -> master
PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git>
```

Y comprobamos en nuestro repositorio remoto si los cambios fueron realizados.



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.

Para la realización de este punto vamos a crear la rama “diseño” con el siguiente comando **“git branch diseño”** luego para pasarnos a esa rama escribimos el comando **“git checkout diseño”**

```
PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git branch diseño
PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git checkout diseño
Switched to branch 'diseño'
PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git>
```

Editamos el diseño con Bootstrap y realizamos los commit

```
PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git add .
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git commit -m "Segundo Commit desde la Rama Diseño"
[diseño 806447a] Segundo Commit desde la Rama Diseño
1 file changed, 22 insertions(+), 22 deletions(-)
PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git>
```

4. Una vez terminado el diseño realizar un merge (fusión) de los cambios realizados en la rama “diseno” a la rama “main”.

Realizamos la fusión de la rama escribiendo en la terminal el siguiente comando **“git merge diseño”** pero para hacer esto primero debemos ir a la rama master o main y desde ahí escribir el comando.

```
● PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git checkout master
Switched to branch 'master'
● PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git merge diseño
Updating caf9300..806447a
Fast-forward
 index.html | 44 ++++++-----
 1 file changed, 22 insertions(+), 22 deletions(-)
○ PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> █
```

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.

Creamos la rama “funcionalidad” igual que en punto 3 con el comando **“git branch funcionalidad”** creamos la rama, luego para pasarnos a esa rama escribimos el comando **“git checkout funcionalidad”**

```
● PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git branch funcionalidad
● PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git checkout funcionalidad
Switched to branch 'funcionalidad'
○ PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> █
```

Ahora agregamos el nuevo botón que permita convertir Centígrados a Fahrenheit y realizamos los commit

```
● PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git add .
warning: in the working copy of 'js/script.js', LF will be replaced by CRLF the next time Git touches it
● PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git commit -m "Tercer Commit desde la Rama Funcionalidad"
[funcionalidad 2278740] Tercer Commit desde la Rama Funcionalidad
 2 files changed, 26 insertions(+)
○ PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> █
```

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

Una vez terminada la funcionalidad nos vamos a la rama principal así **“git checkout master”** y una vez en esa rama realizamos la fusión así **“git merge funcionalidad”**

```
● PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
● PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git merge funcionalidad
Updating 806447a..2278740
Fast-forward
 index.html | 5 ++++
 js/script.js | 21 ++++++
 2 files changed, 26 insertions(+)
○ PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> █
```

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”

Por último, actualizamos el repositorio remoto escribiendo en la terminal de Visual Studio Code el comando **“git push origin master”**

```
● PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DavidRodriguez8/TallerUnidad01Git.git
 806447a..2278740 master -> master
○ PS C:\Users\DAVID\Downloads\Diplomado\Talleres Git\TallerUnidad01Git> █
```

Adicionalmente en el repositorio debe aparecer el informe realizado entonces debemos agregar este archivo al repositorio local, por lo tanto haremos un **“git add .”** y posteriormente podríamos hacer un commit y un push para que los cambios se suban al repositorio remoto.