

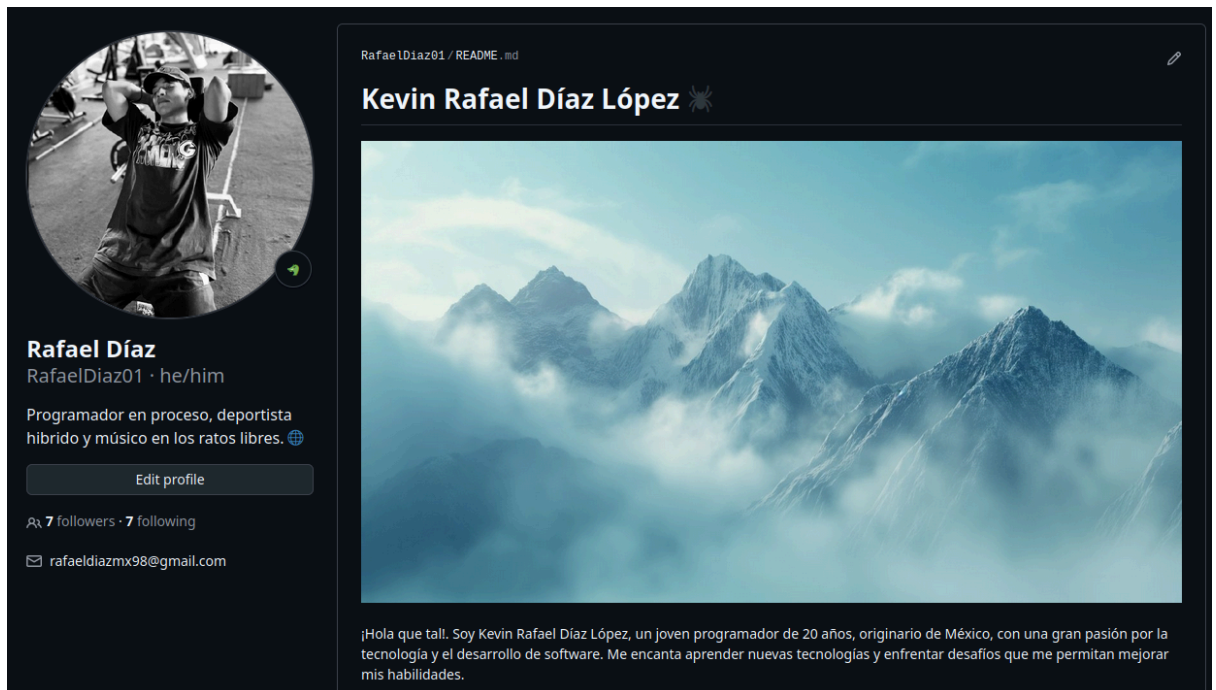
# SISTEMAS OPERATIVOS I

## UNIDAD 1

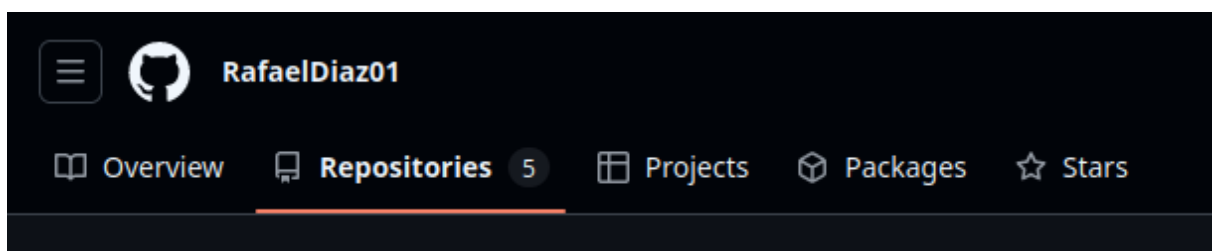
### ENTORNO DE TRABAJO

#### 1. Crear el Repositorio en GitHub

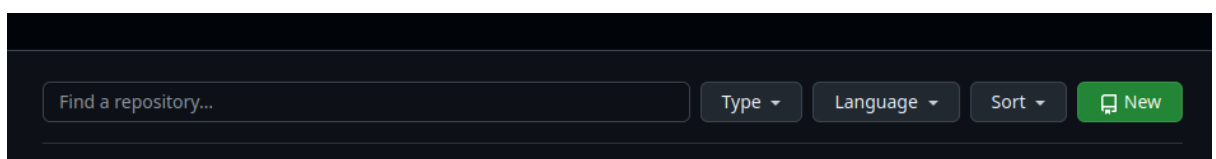
Primero debemos crear una cuenta en GitHub. En mi caso, ya tengo creada mi cuenta con el nombre de RafaelDiaz01. Mi cuenta: <https://github.com/RafaelDiaz01>



Ahora vamos a ir a la sección “Repositorios”.



Una vez en la sección “Repositorios”, vamos a crear un nuevo repositorio en el botón “New”.



Ahora le pondremos un nombre a nuestro repositorio, para este caso mi repositorio se va a llamar “*Sistemas Operativos I*”. Luego agregamos una descripción, dejamos el repositorio en público para que pueda ser visible por cualquier persona y agregamos el archivo README. Al final solo damos click en “Create Repository”.

## 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 (\*).*

Repository owner and name  
RafaelDiaz01

/

Repository name \*  
Sistemas Operativos


✔ Your new repository will be created as **Sistemas-Operativos**.  
The repository name can only contain ASCII letters, digits, and the characters ., -, and \_.


Great repository names are short and memorable. Need inspiration? How about **refactored-broccoli** ?

**Description** (optional)

Repositorio para subir los trabajos de la materia Sistemas Operativos.

---

☒  **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.

---

**Initialize this repository with:**

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**


.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)


**Choose a license**

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

---

 You are creating a public repository in your personal account.

---

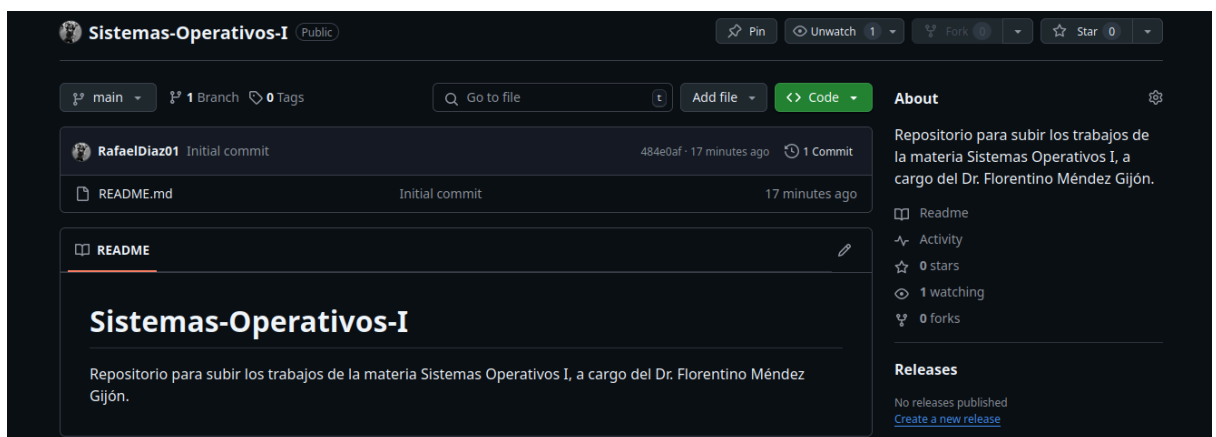
Create repository

Y aquí podemos observar que ya se creó nuestro repositorio.



## 2. Clonar el Repositorio en tu Computadora

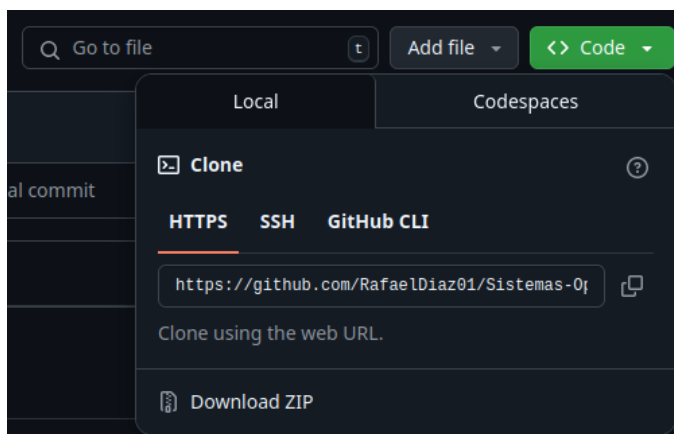
En GitHub, vamos a ir hasta la página principal del repositorio.



Hay que buscar el botón “<Code>” y hacer click en él.



Copiamos la dirección URL del repositorio.



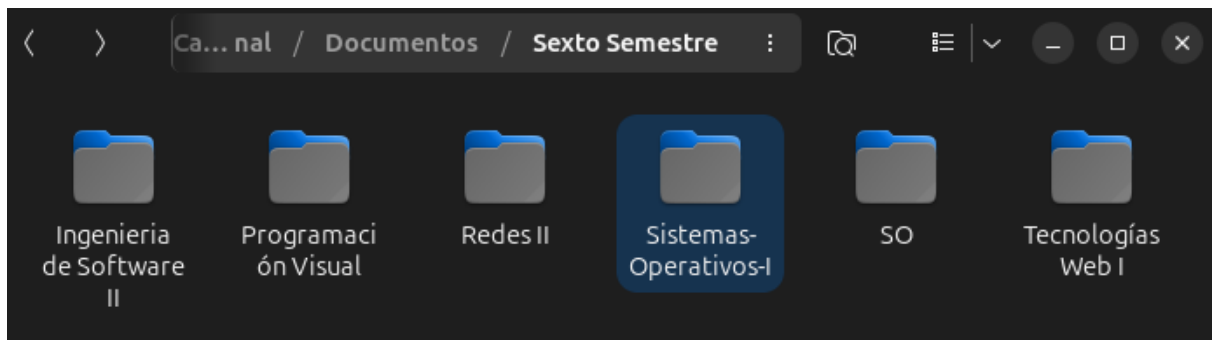
Ahora abrimos una terminal y navegamos hasta la ubicación en donde queremos clonar el directorio.

```
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~$ cd Documentos/Sexto\ Semestre/  
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~/Documentos/Sexto Semestre$
```

Una vez en esta dirección, escribimos “*get clone*”, pegamos la dirección URL que copiamos antes y damos “*Enter*”.

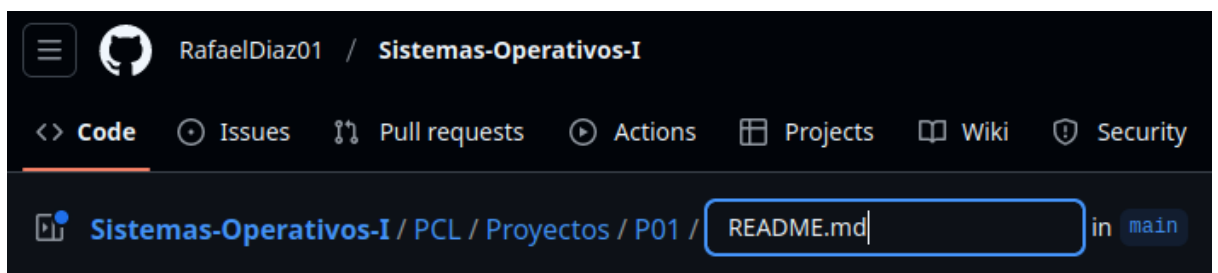
```
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~/Documentos/Sexto Semestre$ git clone https://github.com/RafaelDiaz01/Sistemas-Operativos-I.git  
Clonando en 'Sistemas-Operativos-I'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
Recibiendo objetos: 100% (3/3), listo.
```

Listo, ya tenemos clonado nuestro repositorio en nuestra computadora.



3. Crear la Estructura de Directorios, [descrito en esta liga](#)

Para crear esta estructura, nos vamos a GitHub y primero creamos la carpeta “*PCL*” (hay que agregar / después de escribir el nombre de la carpeta para crearla). Luego creamos la carpeta “*Proyectos*”, seguido de la carpeta “*P01*” y al final el archivo *README.md*.



Listo, ya tenemos la estructura creada, ahora solo falta agregar actividades a cada una de nuestras carpetas. Las actividades vendrían siendo “*A01*”, “*A02*”, etc.

#### 4. Agregar y Subir los Cambios a GitHub

Para agregar y subir los cambios a GitHub vamos a navegar hasta nuestra carpeta: **cd Documentos/Sexto\ Semestre/Sistemas-Operativos-I/** y una vez en esta dirección escribiremos el comando **git status**.

```
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~$ cd Documentos/Sexto\ Semestre/Sistemas-Operativos-I/
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~/Documentos/Sexto Semestre/Sistemas-Operativos-I$ git status
En la rama main
Tu rama está actualizada con 'origin/main'.
```

Ahora hay que agregar las nuevas carpetas al área de preparación con el siguiente comando: **git add .**

El **.** agrega todos los cambios en la carpeta actual.

```
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~/Documentos/Sexto Semestre/Sistemas-Operativos-I$ git add .
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~/Documentos/Sexto Semestre/Sistemas-Operativos-I$
```

Debemos confirmar los cambios con un mensaje descriptivo para tener un mejor control en nuestro GitHub.

```
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~/Documentos/Sexto Semestre/Sistemas-Operativos-I$ git commit -m "Se añadieron 3
nuevas carpetas correspondientes a los parciales, esto con el fin de tener un mejor control de archivos"
En la rama main
Tu rama está actualizada con 'origin/main'.

nada para hacer commit, el árbol de trabajo está limpio
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~/Documentos/Sexto Semestre/Sistemas-Operativos-I$
```

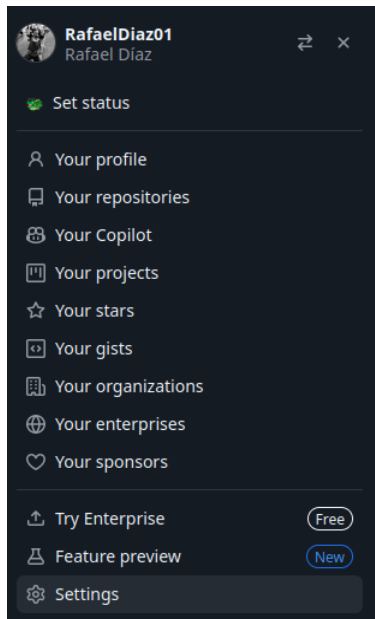
Finalmente, hay que subir los cambios al repositorio en GitHub con el comando: **git push origin main**.

```
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~/Documentos/Sexto Semestre/Sistemas-Operativos-I$ git push origin main
Username for 'https://github.com': RafaelDiaz01
Password for 'https://RafaelDiaz01@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls
for information on currently recommended modes of authentication.
fatal: Autenticación falló para 'https://github.com/RafaelDiaz01/Sistemas-Operativos-I.git/'
```

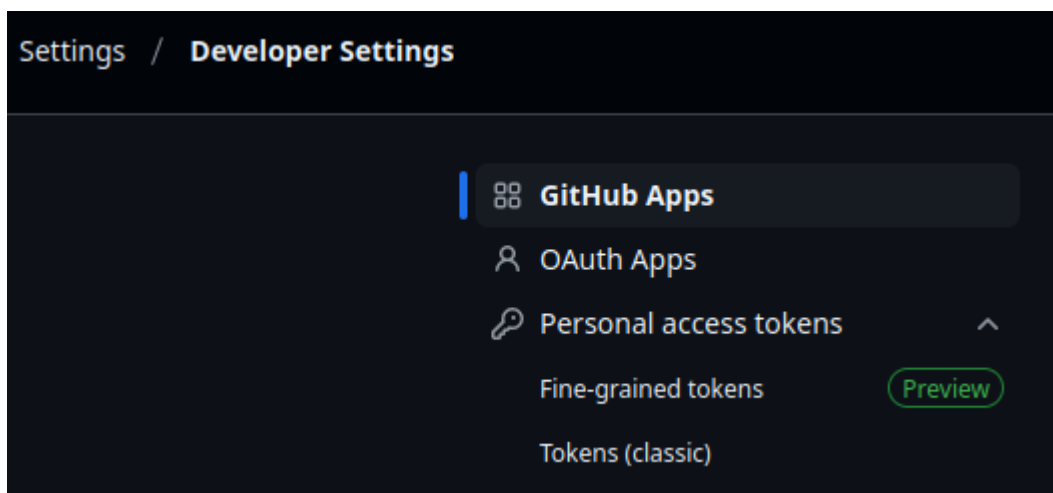
Si marca el error de la imagen , se debe a que GitHub ya no permite la autenticación mediante nombre de usuario y contraseña para operaciones como **git push** cuando usas HTTPS.

Para solucionarlo debemos crear un Token en nuestro perfil.

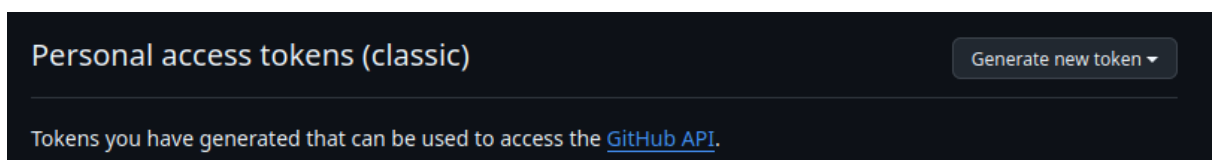
Primero vamos a “Settings”.



Después vamos a “*Developer setting*”, luego en “*Personal access tokens*” y hacemos clic en “*Token (classic)*”.



Click en “*Generate new token*”.



De nuevo hay que subir los cambios al repositorio en GitHub con el comando: **git push origin main**, pero ahora en lugar de poner nuestra contraseña pondremos el token que generamos.

```
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~/Documentos/Sexto_Semestre/Sistemas-Operativos-1$ git push origin main
Username for 'https://github.com': RafaelDiaz01
Password for 'https://RafaelDiaz01@github.com':
Everything up-to-date
```

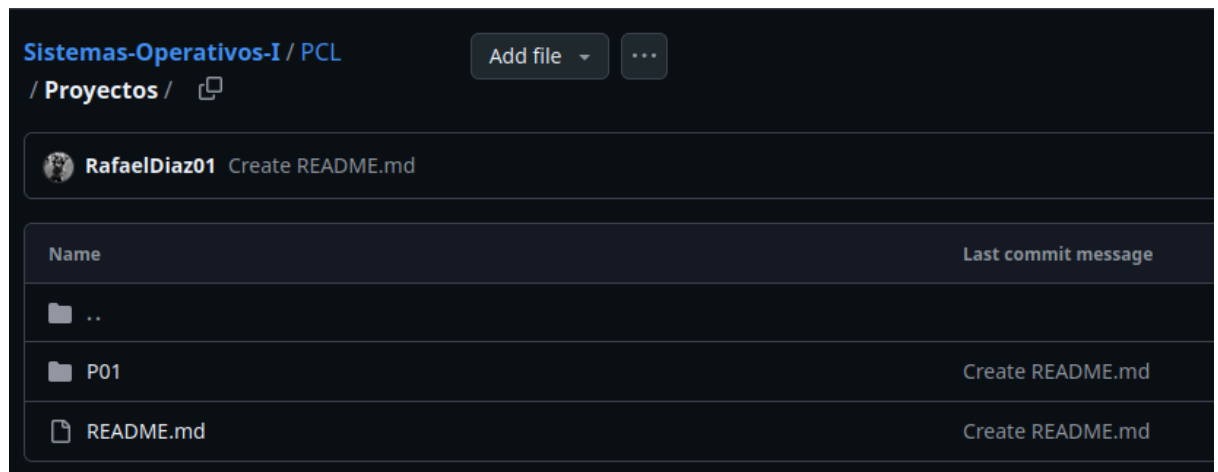
Ya quedó, no nos marca error.

5. Verificar en GitHub

Para verificar que nuestros cambios se hayan realizado en GitHub vamos a ir a nuestro repositorio en el navegador de nuestra preferencia.



Como podemos ver tenemos creada la estructura de trabajo creada.

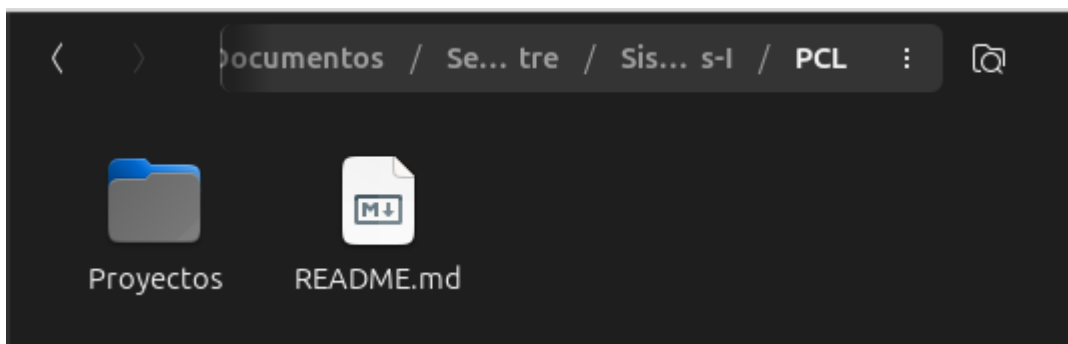
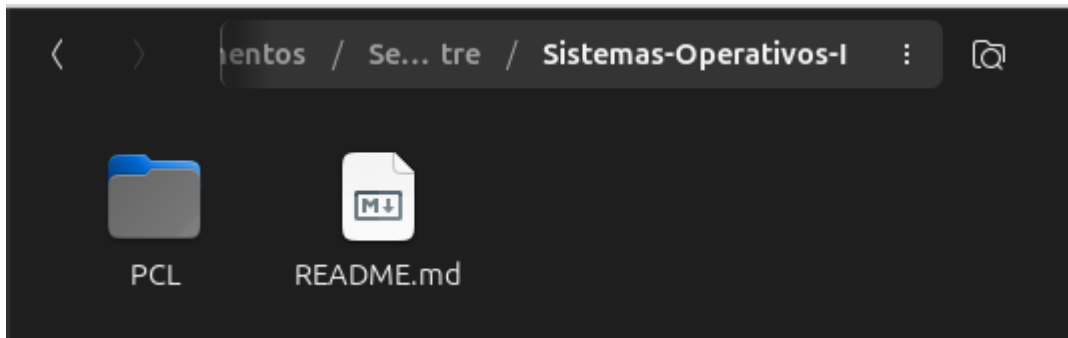


Para ver los cambios en nuestra computadora, haremos lo siguiente: Navegamos hasta donde tenemos el repositorio clonado y escribiremos el comando ***“git pull origin main”*** ya que este comando descarga los cambios

remotos y los fusiona con el repositorio local.

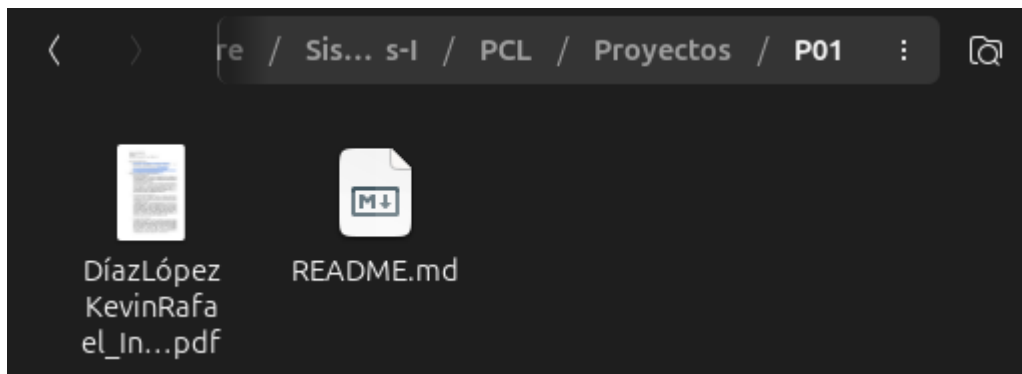
```
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~/Documentos/Sexto Semestre/Sistemas-Operativos-I$ git pull origin main
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 21 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Desempaquetando objetos: 100% (21/21), 8.02 KiB | 821.00 KiB/s, listo.
Desde https://github.com/RafaelDiaz01/Sistemas-Operativos-I
* branch          main      -> FETCH_HEAD
484e0af..3ba678f  main      -> origin/main
Actualizando 484e0af..3ba678f
Fast-forward
 PCL/Proyectos/P01/README.md | 1 +
 PCL/Proyectos/README.md     | 1 +
 PCL/README.md               | 1 +
3 files changed, 3 insertions(+)
create mode 100644 PCL/Proyectos/P01/README.md
create mode 100644 PCL/Proyectos/README.md
create mode 100644 PCL/README.md
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~/Documentos/Sexto Semestre/Sistemas-Operativos-I$
```

Ya tenemos los cambios que hicimos en GitHub en nuestra computadora.



Ahora agregaré un archivo PDF en la carpeta PO1, simulando que es un trabajo.





Y haremos los pasos correspondientes para subir los cambios a GitHub.

```
rafaeldiaz@labred04-OptiPlex-SFF-Plus-7010:~/Documentos/Sexto Sem  
vos-I$ git push origin main  
Username for 'https://github.com': RafaelDiaz01  
Password for 'https://RafaelDiaz01@github.com':  
Enumerando objetos: 10, listo.  
Contando objetos: 100% (10/10), listo.  
Compresión delta usando hasta 20 hilos  
Comprimiendo objetos: 100% (6/6), listo.  
Escribiendo objetos: 100% (6/6), 1.05 MiB | 923.00 KiB/s, listo.  
Total 6 (delta 0), reusados 0 (delta 0), pack-reusados 0  
To https://github.com/RafaelDiaz01/Sistemas-Operativos-I.git  
3ba678f..564cea8 main -> main
```

Como podemos observar, el pdf se cargó correctamente en GitHub.

