

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
Ingeniería en Software y Tecnologías Emergentes



Práctica 0: Gestión de Repositorios con Git y GitHub

Materia: Paradigmas de programación

ALUMNO: Barrios Retana Lizeth

MATRÍCULA: 372813

GRUPO: 941

PROFESOR: Jose Carlos Gallegos Mariscal

Ensenada, Baja California a 29 de marzo de 2024.

Práctica 0: Gestión de Repositorios con Git y GitHub.

Introducción:

En este reporte de práctica exploraremos la gestión de repositorios utilizando Git y GitHub. Comenzaremos creando y verificando una cuenta en GitHub, instalando y verificando Git, y estableciendo el entorno de trabajo necesario. Luego, inicializaremos un repositorio local, crearemos una llave pública/privada para conexión segura con GitHub, y configuraremos un repositorio en la plataforma. Abordaremos el clonado del repositorio, la gestión de ramas para cambios eficientes, la restauración de cambios y la fusión de ramas. A través de este proceso, obtendremos una comprensión integral de cómo colaborar efectivamente en proyectos de desarrollo de software utilizando estas herramientas de control de versiones.

Desarrollo:

Crear / Verificar cuenta de Github

- Abre tu navegador web y ve a [Github.com](https://github.com).
- Haz clic en "Sign up" si aún no tienes una cuenta. Si ya tienes una cuenta, inicia sesión.

Instalar / Verificar Git

- Verifica la instalación ejecutando `git --version` en tu terminal.
- Si aún no tienes Git instalado, descárgalo e instálalo desde git-scm.com.

Crear folder del proyecto

- Decide dónde deseas almacenar tu proyecto en tu sistema de archivos.
- Crea una nueva carpeta para tu proyecto si aún no tienes una.
- En la terminal, en este caso bash se pueden ejecutar los siguientes comandos (ejemplo):

```
$mkdir Pparadigmas/practica0      // Crear carpetas
$cd Pparadigmas/practica0         // Cambiar directorio
$touch prueba.md                  // Crear archivo
$nano prueba.md                   // Abrir archivo en editor de texto
$cat prueba.md                    // Leer archivo en terminal
$rm pureba.md                     // Eliminar archivo
$code README.md                   // Abrir archivo en VSCode
```

Práctica 0: Gestión de Repositorios con Git y GitHub.

```
MINGW64:/c:/Users/fer ojeda/OneDrive/Escritorio/4 semestre/PProgramacion/temp/PProg...
fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion
$ mkdir practica0

fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion
$ cd PProgramacion/practica0
bash: cd: PProgramacion/practica0: No such file or directory

fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion
$ cd practica0

fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion
/practica0
$ touch prueba.md

fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion
/practica0
$ nano prueba.md

fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion
/practica0
$ cat prueba.md
titulo

fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion
/practica0
$ rm prueba.md

fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion
/practica0
$ code README.md
```

Inicializar repositorio

- Abre tu terminal o símbolo del sistema.
- Navega hasta la carpeta de tu proyecto usando el comando `cd`.
- Ejecuta `git init` para inicializar un nuevo repositorio Git en la carpeta.

```
$cd ..          // Volver al directorio anterior (atras)
$git init       // Inicializar repositorio local de git
$git status     // Checar el estado de los archivos en el repositorio
$git add README.md // Añadir archivo a los archivos rastreados
$git commit -m " mi primer commit" // Crear una captura de los cambios
```

```
MINGW64:/c:/Users/fer ojeda/OneDrive/Escritorio/4 semestre/PProgramacion/temp/PProg...
/practica0
$ cd ..

fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion
$ git init
Initialized empty Git repository in C:/Users/fer ojeda/OneDrive/Escritorio/4 sem
estre/PProgramacion/.git/

fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion
(master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

Práctica 0: Gestión de Repositorios con Git y GitHub.

```
MINGW64:/c:/Users/fer ojeda/OneDrive/Escritorio/4 semestre/PProgramacion/temp/PProg...
fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion/practica0 (master)
$ git add README.md

fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion/practica0 (master)
$ git commit -m "primer commit"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'fer ojeda@DESKTOP-5A9PR68.(none)')
```

Crear llave pública / privada

- Si aún no tienes una llave SSH, genera una usando el comando ssh-keygen.
- Localizar el archivo creado y leerlo usando el comando cat.

```
$ssh-keygen
```

```
$ls ~/.ssh
```

```
$cat ~/.ssh/id_ed25519.pub
```

```
MINGW64:/c:/Users/fer ojeda/OneDrive/Escritorio/4 semestre/PProgramacion/temp/PProg...
fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion/practica0 (master)
$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c:/Users/fer ojeda/.ssh/id_ed25519):
Created directory '/c:/Users/fer ojeda/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c:/Users/fer ojeda/.ssh/id_ed25519
Your public key has been saved in /c:/Users/fer ojeda/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:F9w6SxAhlydkJjuPmATFTPvffTyoyB1cqmrIFaskWwM fer ojeda@DESKTOP-5A9PR68
The key's randomart image is:
+--[ED25519 256]--+
|  .o .++0+  . |
|  .o .E++ooo... |
|  . o .+++++...o |
|  . o =+*+=o  .. |
|  o . S+0o    |
|    . + o     |
|  .           |
|  +-----+
+-----[SHA256]-----+

fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion/practica0 (master)
$ ls ~/.ssh
id_ed25519  id_ed25519.pub
```

Práctica 0: Gestión de Repositorios con Git y GitHub.

```
fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion
/practica0 (master)
$ cat ~/.ssh/id_rsa.pub
cat: '/c/Users/fer ojeda/.ssh/id_rsa.pub': No such file or directory

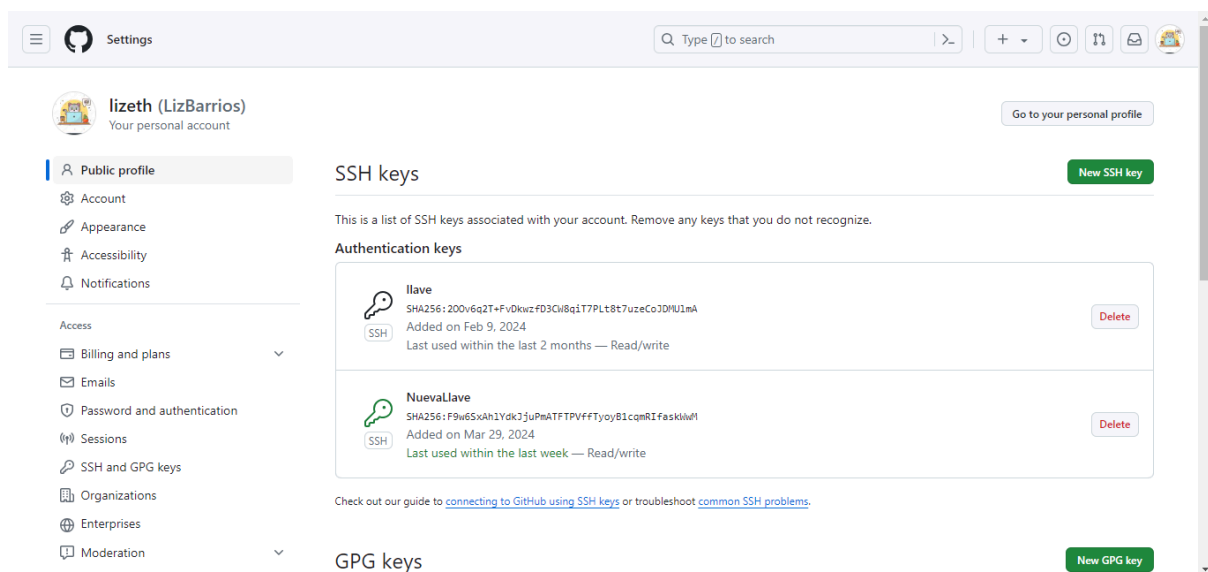
fer ojeda@DESKTOP-5A9PR68 MINGW64 ~/OneDrive/Escritorio/4 semestre/PProgramacion
/practica0 (master)
$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIjdAG6qANxVu5kOpP61kJOXBfHKK8dp33D6/DfZeDwWx
fer ojeda@DESKTOP-5A9PR68
```

Crear repositorio en Github

- Inicia sesión en tu cuenta de Github.
- Haz clic en el signo más en la esquina superior derecha y selecciona "New repository".
- Dale un nombre a tu repositorio, establece la visibilidad y otras configuraciones según tu preferencia.
- Haz clic en "Create repository".

Agregar llave a Github

- Copia el contenido de tu llave pública creada anteriormente.
- Ve a la configuración de tu cuenta de Github haciendo clic en tu avatar en la esquina superior derecha y seleccionando "Settings".
- En el menú de la izquierda, haz clic en "SSH and GPG keys".
- Haz clic en "New SSH key".
- Pega el contenido de tu llave pública en el campo "Key" y dale un nombre descriptivo.
- Haz clic en "Add SSH key" para guardar.



Práctica 0: Gestión de Repositorios con Git y GitHub.

Clonar repositorio

- Ve a Github y haz clic en "Code", de ahí copia el código de SSH.
- En tu terminal, navega hasta la carpeta donde deseas clonar el repositorio.
- Ejecuta git clone seguido del código de SSH que copiaste.

```
$mkdir temp // Crear la carpeta temporal
$cd temp // Cambiar directorio a temp
$git clone git@github.com:LizBarrios/ParadigmasP.git // Clonar repositorio
$cd ParadigmasP // Cambiar directorio
```

Crear branch

- Crea un nuevo branch usando el comando git checkout -b seguido del nombre del branch.
- Verifica que estás en el branch correcto usando git status.

```
$git checkout -b temp // Crear branch
$git status // Verificar branch
```

- Realiza cambios en tu proyecto y haz un commit.

```
$code . // Abrir proyecto en VSCode
$git add README.md // Añadir archivo a los archivos rastreados
$git commit -m "Modificar README" // Crear captura de cambios
```

- Sube tu branch a Github usando el comando git push seguido de -u origin y el nombre del branch. (-u origin solo se usa la primera vez, después solo se usa git push)

```
$git push -u origin temp // Subir branch a Github
```

Restaurar cambios

- Haz un cambio pero no lo añadas al stage.
- Usa el comando git restore seguido de un punto para eliminar los cambios.

Práctica 0: Gestión de Repositorios con Git y GitHub.

```
$git restore . // Eliminar cambios
```

- Si añadiste el cambio, debes ejecutar lo siguiente:

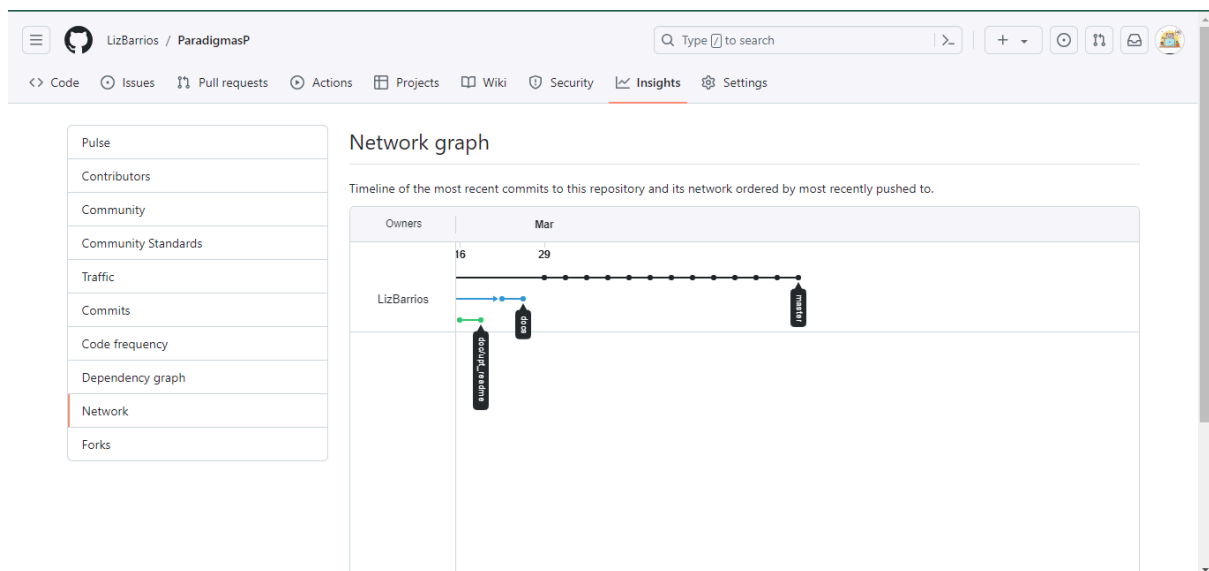
```
$git restore --staged README.md
```

Fusionar branch

- Cambia al branch que deseas fusionar.
- Ejecuta git merge seguido del nombre del branch que deseas fusionar.
-

```
$git checkout main
```

```
$git merge temp
```



Repositorio de GitHub

<https://github.com/LizBarrios/ParadigmasP.git>

Práctica 0: Gestión de Repositorios con Git y GitHub.

Conclusión:

En conclusión, la práctica 0 me ha proporcionado una comprensión más clara de las herramientas fundamentales para el desarrollo de software colaborativo. Desde la creación de una cuenta en GitHub hasta la gestión de ramas y la integración de cambios, hemos explorado paso a paso el proceso completo. Estas herramientas no solo facilitan el seguimiento y la organización de versiones de nuestro código, sino que también fomentan la colaboración efectiva entre desarrolladores, permitiendo un flujo de trabajo fluido y transparente. Al dominar estas habilidades, estamos mejor equipados para contribuir de manera significativa a proyectos de software y trabajar de manera eficiente en equipo en un entorno profesional de desarrollo de software. El manejar los cambios creo que depende mucho de la persona, ya que puede que a algunos se les haga más fácil desde el programa en el cual están realizando los cambios, pero debo admitir que para mi fue más eficiente el manejo de todos estos procesos desde la línea de comandos.