

# PROGRAMACIÓN 1

## PRÁCTICO 2: GIT y GITHUB

### Actividades

#### 1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :

- ¿Qué es GitHub?

GitHub es una plataforma que permite alojar, compartir, guardar repositorios y colaborar en proyectos de código.

- ¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en GitHub necesitamos de tener una cuenta creada e iniciar sesión. Una vez que tenemos creada una cuenta en la plataforma de GitHub se encuentra la sección “New” o “Nuevo”, la cual nos permite crear un Repositorio colocándole el nombre que queramos para el mismo, si queremos agregarle una descripción breve del proyecto o si deseamos que el proyecto sea público o privado, entre otras opciones adicionales.

- ¿Cómo crear una rama en Git?

Para crear una rama en Git debemos estar posicionados en la carpeta del proyecto y utilizar el siguiente comando:

```
> git branch nombre-de-la-rama
```

donde en *nombre-de-la-rama* colocaríamos el nombre que queramos para nuestra rama, el cual debe de ser lo más conciso y descriptivo de lo que haremos en ella para evitar futuras confusiones.

- ¿Cómo cambiar a una rama en Git?

Para cambiarnos a una rama de Git, primero podemos utilizar el siguiente comando para ver todas las ramas disponibles en el proyecto:

```
> git branch
```

El cual mostrará todas las ramas que tenemos creadas en el repositorio local, la rama en la que te encuentres al momento de usar el comando aparecerá con un asterisco (\*).

Y para movernos entre ramas utilizamos el comando:

```
> git checkout nombre-de-la-rama
```

donde *nombre-de-la-rama* lo reemplazamos por el nombre de la rama a la que queramos ir.

### • ¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git debemos de colocarnos en la rama a la que querramos llevar los cambios hechos, normalmente se hace en la rama main que es la rama principal por defecto en todos los proyectos.

A continuacion utilizamos el siguiente comando:

```
> git merge nombre-de-la-rama
```

Donde *nombre-de-la-rama* se cambia por el nombre de la rama a la que querramos copiar los cambios hechos, lo que hara es traer todos los cambios realizados en la rama *nombre-de-la-rama* a la rama en la que nos encontrabamos parados.

### • ¿Cómo crear un commit en Git?

Crear un commit nos sirve para realizar un “punto de control” en el historial del proyecto y guardar los cambios realizados en los archivos.

Para crearlo utilizamos el siguiente comando:

```
> git commit -m "texto de ejemplo"
```

Donde explicaremos que cambios o ajustes se realizaron al momento de realizar el commit.

### • ¿Cómo enviar un commit a GitHub?

Si es la primera vez que vamos a subir un commit a GitHub se debe de hacer con el siguiente comando:

```
> git push -u origin nombre-de-la-rama
```

El cual establece la conexión entre la rama local y GitHub. Si ya subimos la rama anteriormente a GitHub y queremos guardar los cambios luego de hacer el commit podemos usar el siguiente comando:

```
> git push
```

Como ya esta creada la conexión de la rama local y GitHub, Git ya entiende que estamos subiendo los cambios a esa rama, la cual normalmente es la rama main.

### • ¿Qué es un repositorio remoto?

Es una versión de tu proyecto la cual está alojada en internet o en una red, los más comunes son GitHub y GitLab.

### • ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a Git se utiliza el siguiente comando:

```
> git remote add origin https://github.com/usuario/nombre-del-repo.git
```

Donde remplazamos usuario por nuestro nombre de usuario de GitHub y nombre-del-repo por el nombre de nuestro repositorio en GitHub.

- **¿Cómo empujar cambios a un repositorio remoto?**

Para empujar los cambios a un repositorio remoto, primero debemos de guardarlos en nuestro repositorio local, agregando los cambios con:

```
> git add .
```

```
> git commit -m "Descripcion cambios realizados"
```

El cual agrega todos los cambios realizados a nuestro commit.

Luego podemos utilizar el comando:

```
> git push origin nombre-de-la-rama
```

Donde subimos los cambios del repositorio local a el repositorio remoto.

- **¿Cómo tirar de cambios de un repositorio remoto?**

Para descargar los ultimos empujes desde un repositorio remoto utilizamos el siguiente comando:

```
> git pull origin nombre-de-la-rama
```

El cual trae fusiona todo lo que esta subido al repositorio remoto con nuestro repositorio local.

- **¿Qué es un fork de repositorio?**

Fork de repositorio es una copia de un repositorio remoto, puede ser tuyo o no, el cual permite realizar cambios/modificar código sin afectar el repositorio original.

- **¿Cómo crear un fork de un repositorio?**

En GitHub buscamos un repositorio que querramos copiar, puede ser uno nuestro, y buscamos un boton llamado "Fork". Le hacemos click y GitHub nos dejara guardarlo como si fuera un repositorio nuestro, agregandole un nombre, descripcion y hasta ajustar algunas opciones extras.

- **¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?**

Para hacer un pull request debemos de tener un fork en nuestro repositorio, luego debemos de clonar el fork, hacer algun cambio en el código, confirmarlos y subirlos a el repositorio remoto.

En GitHub dentro del fork va a haber una opcion que dice "Compare & pull request", daremos una explicacion de los cambios que realizamos y daremos click a "Create pull Request".

- **¿Cómo aceptar una solicitud de extracción?**

Debemos ingresar al repositorio donde se recibio la solicitud, hacer click en la parte de "Pull requests", podemos revisar los cambios que se realizo en la seccion de "Files changed" y si queremos aceptarla hacemos click en el boton "Merge pull request" y como ultimo "Confirm merge".

- **¿Qué es un etiqueta en Git?**

Es una marca especial que se usa para señalar momentos importantes en el historial de commits, sirve para poder volver a una version especifica, publicar lanzamientos en GitHub, etc.

- **¿Cómo crear una etiqueta en Git?**

Hay 2 tipos de etiquetas en Git.

Etiquetas ligeras: Son como un marcador a un commit específico. Para realizarla se utiliza el siguiente comando:

```
> git tag nombre-de-la-etiqueta
```

Etiquetas anotadas: Se guardan como objetos completos en la base de datos de Git, incluyen nombre, correo del autor, fecha y un mensaje. Para realizarla se utiliza el siguiente comando:

```
> git tag -a nombre-de-la-etiqueta -m "Mensaje de la etiqueta"
```

- **¿Cómo enviar una etiqueta a GitHub?**

Para subir las etiquetas a GitHub existen dos formas de hacerlo:

```
> git push origin nombre-de-la-etiqueta
```

Este comando sube a GitHub la etiqueta expecifica que indicamos.

```
> git push --tags
```

Este comando sube a GitHub todas las etiquetas que tengamos creadas y no subidas al repositorio remoto.

- **¿Qué es un historial de Git?**

El historial de Git es el registro completo de todos los cambios que se han hecho en un repositorio a lo largo del tiempo.

- **¿Cómo ver el historial de Git?**

Para ver el historial de Git se utiliza el siguiente comando, el cual mostrara una lista detallada de los commits:

```
> git log
```

- **¿Cómo buscar en el historial de Git?**

En Git existen varias formas para buscar en el historial algun commit.

Por mensaje del commit:

```
> git log --grep="palabra clave"
```

Por autor:

```
> git log --author="nombre o email"
```

Por fecha:

```
> git log --since="2024-04-01" --until="2024-04-03"
```

Por cambio específico en un archivo:

```
> git log ruta/del/archivo
```

Por contenido modificado:

```
> git log -S"texto buscado"
```

• **¿Cómo borrar el historial de Git?**

Para borrar todo el historial de Git se utiliza el siguiente comando:

```
> rm -rf .git
```

el cual borra el repositorio de Git y luego debemos de crear nuevamente otro.

• **¿Qué es un repositorio privado en GitHub?**

Un repositorio privado en GitHub es un proyecto de código que solo vos y las personas que autorices pueden ver o acceder.

• **¿Cómo crear un repositorio privado en GitHub?**

Para crear un repositorio privado debemos de entrar a GitHub, hacer click en crear un repositorio nuevo, agregarle nombre y descripción del repositorio y seleccionar la opción "Private".

• **¿Cómo invitar a alguien a un repositorio privado en GitHub?**

Dentro del repositorio privado, vamos a la sección "settings" dentro buscamos la sección "Collaborators" y le damos click. Agregamos el mail o nombre de usuario de la persona que queramos invitar y listo, la otra persona deberá de aceptar la invitación al repositorio.

• **¿Qué es un repositorio público en GitHub?**

Un repositorio público en GitHub es un proyecto de código que puede ser visto por cualquier persona, sin necesidad de iniciar sesión o tener permisos especiales.

• **¿Cómo crear un repositorio público en GitHub?**

En GitHub nos dirigimos a la parte que dice "New Repository" o solamente "New", configuramos con nombre y descripción del repositorio, como también si queremos modificar alguna otra opción, elegimos la opción "Public" y listo.

- **¿Cómo compartir un repositorio público en GitHub?**

Para compartir un repositorio público solamente necesitamos ubicarnos dentro del repositorio en GitHub y copiar la URL que se encuentra en la barra del navegador, les aparecera un link parecido a este:

[tahielheinze/UTN-TUPaD-Programacion1: Repositorio General de los trabajos realizados en la materia de Programación 1 de la UTN:TUP a Distancia](#)

## **2) Realizar la siguiente actividad:**

- **Crear un repositorio.**

- Dale un nombre al repositorio.
- Elije el repositorio sea público.
- Inicializa el repositorio con un archivo.

- **Agregando un Archivo**

- Crea un archivo simple, por ejemplo, "mi-archivo.txt".
- Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la línea de comandos.
- Sube los cambios al repositorio en GitHub con git push origin main (o el nombre de la rama correspondiente).

[tahielheinze/UTN-TUPaD-Programacion1: Repositorio General de los trabajos realizados en la materia de Programación 1 de la UTN:TUP a Distancia](#)