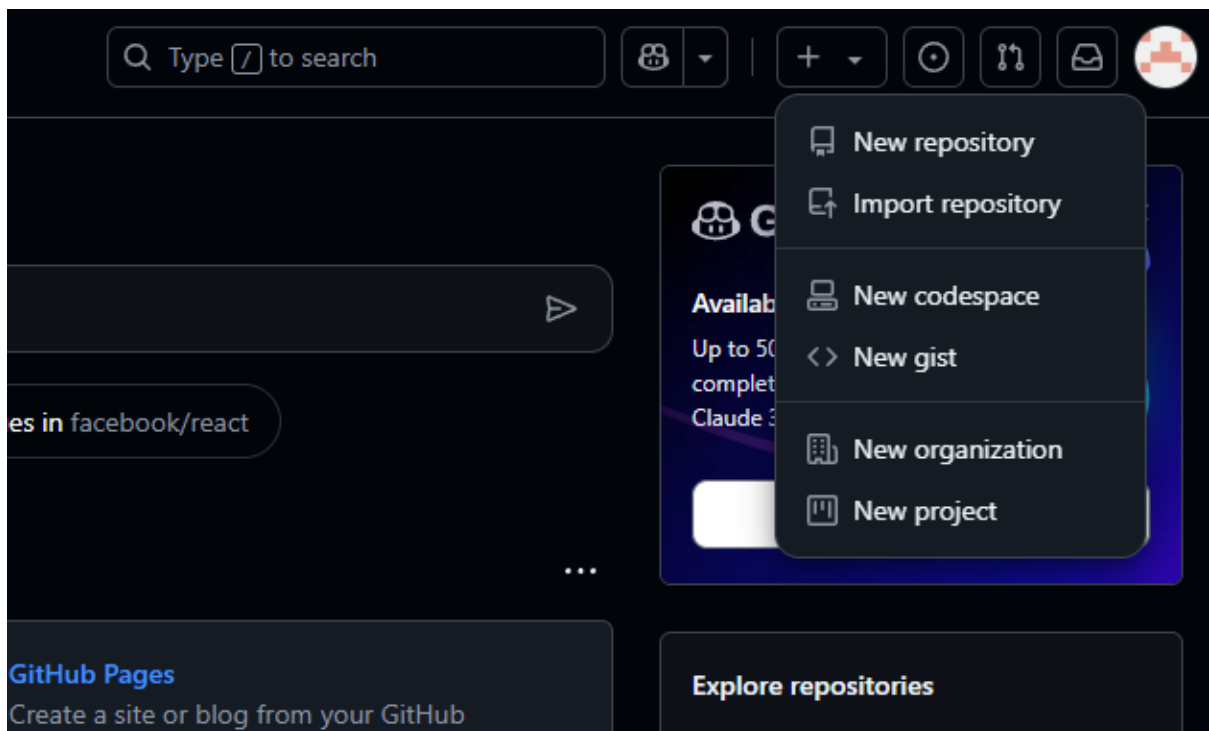


Trabajo practico 2- Programacion 1

1- Git hub es una plataforma y/o página que ofrece alojamiento de repositorios de control de versiones. El cual permite almacenar y gestionar proyectos de software.

2- Para empezar a utilizar GitHub primeramente es necesario crearnos una cuenta desde su página (<https://github.com/>), una vez hecha y logueada la cuenta debemos dirigirnos hacia arriba a la derecha donde se encuentra el icono "+", hacer clic y al desplegarse otro menú elegir "New repository"



Una vez creado debemos elegir el nombre, añadir una descripción y opcionalmente elegir si va a ser un repositorio privado o publico. Luego ir al botón verde de abajo y clickear "Create repository"

3- Para crear una rama en git primero debemos tener descargado Git en nuestra pc y loguearnos. Una vez hechos estos pasos también tenemos que ingresar a nuestro repositorio y ahí recién crear una nueva rama con el comando:

\$ Git branch NuevaRama (elegir nombre de nuestra conveniencia).

4- Para cambiar a una rama en git se debe utilizar el comando:

\$git checkout otraRama (nombre de ejemplo)

5- Primero debemos estar situado en la rama a la que se quiere fusionar.
Por ejemplo queremos fusionar la rama NuevaRama a la master, entonces nos posicionamos en la rama principal:

\$git checkout master

Una vez allí usar el comando marge para fusionar las ramas:

\$git marge NuevaRama

6- Un commit es un proceso que nos permite guardar y documentar los cambios realizados en el código a nuestro proyecto. Se realiza con el comando commit, por ejemplo: supongamos que tenemos un archivo que se llama ivan.py y dentro cambiamos el valor de la variable Nombre = Ivan a Nombre = Ivan Diaz, una vez realizado y guardado el cambio en el código, nos dirigimos a la consola de git y primeramente enviamos los cambios hechos en el archivo

\$ git add ivan.py o git add . (para agregar todos los archivos)

Una vez agregado o agregados los archivos procedemos a realizar el commit y escribir un comentario para saber que cambiamos y en el caso de que estemos trabajando con mas personas, ellos tambien vean el cambio:

\$ git commit m- "Se cambió el valor de la variable nombre"

Esto guarda el estado actual del código en el repositorio.

7- Primero debemos clonar el repositorio de GitHub, esto se hace copiando el link del repositorio a clonar y pegandolo en git con los comandos:

git clone <https://github.com/usuario/repositorio>

cd tu_repositorio

Despues hacer los cambios correspondientes y realizar los pasos del punto anterior (6) y luego para enviar el commit al repositorio en github utilizamos el comando:

```
git push origin nombreDelArchivo
```

8- Los repositorios remotos son versiones de un proyecto que están hospedadas en Internet o en una red local de una empresa, por ejemplo. Al cual pueden acceder las personas que esten trabajando en dicho proyecyo y aportar o editar las versiones del mismo, para mejorar la calidad y legibilidad del codigo o reportar y corregir errores que otros programadores hayan hecho.

9- Para añadir un repositorio remoto a git se usa el comando "git remote add" por ejemplo:

```
$ git remote add repolvan https://www.github/lvaanDiaz/UTN-TUPaD-P1
```

10- Una vez realizados los cambios que queremos en el repositorio usamos el comando push:

```
$ git push origin nombre_del_cambio
```

11- Para tirar los cambios del repositorio remoto usamos el comando git pull, por ejemplo:

```
$ git pull origin nombre_del_cambio
```

12- El fork es simplemente una copia de un repositorio que nos haya gustado, en la cual podemos hacerle los cambios y modificaciones que creamos convenientes, pero sin afectar al repositorio original y que tambien vamos a poder subir a nuestro github con los cambios realizados. Aclaremos y recordemos que estas modificaciones no se subiran al repositorio del autor.

13- Para crear un fork de un repositorio en GitHub, primero se debe iniciar sesión en la web y despues acceder al repositorio que se desea copiar. Una vez ahi, se hace click en el botón "Fork" ubicado arriba a la derecha, lo que generara una copia del

repositorio en nuestra cuenta personal. Esto permite modificar el proyecto de forma personal sin afectar el original.

14- Para hacer el pull request nos dirigiremos a la solapa de Pull requests del repositorio, ahí damos click en new pull request. Damos click en Create pull request y colocamos el asunto y abajo tenemos espacio para poder explicar el cambio que vamos a realizar nosotros.

15- El autor del repositorio verá en sus pull requests el mensaje que le enviamos, para que lo pueda observar y si lo considera realizar el cambio pertinente (además de poder responderle al usuario que le ha propuesto ese cambio).

16- Una etiqueta en git es como una **marca** que se coloca en un commit dentro del historial de un proyecto. Se usa sobre todo para señalar versiones importantes.

17- Para crear una etiqueta en Git, se utiliza el comando `git tag`, que permite marcar un commit específico dentro del historial del proyecto. Existen dos tipos: las etiquetas ligeras, que solo apuntan al commit, y las anotadas, que incluyen información adicional como el autor, la fecha y un mensaje.

18- Primero hay que crear una etiqueta en nuestro repositorio local. Podemos crear una etiqueta anotada o una etiqueta ligera: Ejemplo de etiqueta ligera:

```
$ git tag v1.0
```

Puedes verificar las etiquetas que has creado localmente con: `git tag`

Podemos verificar las etiquetas que creamos con:

```
$ git tag
```

Una vez creado la etiqueta, necesitamos empujarla al repositorio remoto en GitHub.

Puedes hacer esto con el siguiente comando:

```
$ git push origin v1.0
```

(origin es el nombre del repositorio remoto (por defecto suele ser origin) y v1.0 es el nombre de la etiqueta.)

Para empujar todas las etiquetas creadas usamos:

```
$ git push origin --tags
```

19- El historial de Git es una secuencia de todos los cambios realizados en un repositorio, cada cambio en el repositorio se guarda como un commit y cada commit contiene información sobre el estado del proyecto en un momento específico.

20- Para ver el historial de git usamos el comando:

```
$ git log
```

Con este comando de Git vamos a poder ver el historial de commits. Vale aclarar que vamos a ver primero los ultimos cambios realizados.

21- Para buscar en el historial de commits de Git, podemos utilizar varios comandos que nos permiten filtrar y localizar commits específicos.

Para buscar un commit que contenga una palabra o frase especifica en el mensaje del commit se utiliza git log -grep, por ejemplo:

```
$ git log --grep="palabra a buscar"
```

Para buscar un commit que ha modificado un archivo específico, usa git log seguido del nombre del archivo:

```
$ git log -- nombre_del_archivo
```

Para buscar un commit de una fecha especifica se usa las opciones --since y --until:

```
$ git log --since="2024-01-01" --until="2024-01-31"
```

23- Es un repositorio al que solo el creador del mismo puede acceder o los usuarios y/o colaboradores seleccionados personalmente por la cuenta dueña del repositorio para que estos puedan realizar los cambios que crean necesarios.

24- Para crear un repositorio privado debemos ir y clicar en el boton con el signo "+" y elegir "New repository".

Luego de darle un nombre y descripcion, abajo tenemos de descripcion elegimos la opcion private (el cual tiene un icono de candado), una vez seleccionado el modo privado, bajamos al boton verde que dice "Create repository" y listo, tenemos nuestro repositorio privado.

25- Para invitar a alguien a nuestro repositorio debemos ingresar al repositorio, luego a la pestaña "Settings", luego "collaborators" en el menu de la izquierda, una

vez allí dentro de Manage Accés clickeamos en "Add People" e ingresamos el nombre del usuario al cual daremos acceso.

26- Un repositorio público de git es aquel al que cualquier persona puede acceder para ver, clonar o contribuir.

27- Para crear un repositorio público en github debemos primero iniciar sesión y luego hacer clic en el icono "+" y en las opciones que se despliegan seleccionar "New repository", una vez allí le damos un nombre y descripción. Debajo la sección descripción elegimos la opción "Public" y bajamos para darle clic al botón verde de "Create repository" y listo.

28- Para compartir nuestro repositorio debemos ir al mismo y eleccionar el botón verde que dice "<>Code", al clickear se nos desplegará un pequeño menú y en ese debemos elegir en la sección "HTTPS", donde veremos un link (el link de nuestro repositorio), un botoncito cuadrado que nos va a copiar directamente con un clic, o simplemente sombrear el link y con clic derecho desplegar el menú del mouse y seleccionar copiar. Y ya tenemos el link de nuestro repositorio copiado para enviar a quien deseemos.

2) <https://github.com/IvaanDiaz/repositorioTP2>

3) <https://github.com/IvaanDiaz/repositorio2TP2>