

## Actividad 1:

1\_ GitHub es una herramienta de desarrollo colaborativo muy importante para los desarrolladores de software, permite crear, guardar, editar y copiar códigos o proyectos entre otras cosas, también sirve para estudiar y entregar trabajos prácticos.

2\_ Para crear un repositorio en GitHub, debemos entrar a la pagina web de GitHub (<https://github.com/>), registrarnos con un eMail y una contraseña, luego tocar el botón verde que dice new ubicado arriba a la izquierda en la pantalla. Tenemos que ponerle un nombre y decidir si va a ser público o privado. Hay varias opciones adicionales como crear un archivo README o elegir una licencia, pero estas, no son necesarias para la creación del repositorio.

3\_ Para crear una rama en Git, debemos escribir el código `'git branch [nombre]'`. Por ejemplo, si queremos crear una branch (rama) llamada nuevaRama, en la consola debemos escribir: `'git branch nuevaRama'`.

4\_ Para cambiar de rama en Git, debemos escribir el código `'git checkout [nombre]'`. Por ejemplo, si estamos en la branch Master y queremos pasar a la branch nuevaRama, en la consola debemos escribir: `'git checkout nuevaRama'`.

5\_ Para fusionar branches en Git, debemos escribir el código `'git merge [nombre]'`. Por ejemplo, si estamos en la branch Master y queremos fusionarla con la branch nuevaRama, en la consola debemos escribir: `'git merge nuevaRama'`.

6\_ Para crear un commit en Git, primero debemos inicializar Git en la carpeta donde estamos trabajando con el código `'git init'`, luego debemos escribir `'git add .'`, para agregar todos los archivos y carpetas a Git para poder registrar sus cambio. Y el último paso es hacer el commit, el cual se hace escribiendo `'git commit '`, esto nos permite registrar los cambios para volver a versiones anteriores, o solo para tener un historial de cambios. También puede hacerse `'git commit m- [texto explicativo]'`, para dejar una nota sobre los cambios. Por ejemplo `'git commit m- cambio de nombre en variables principales'`.

7\_ Para enviar un commit a GitHub debemos ejecutar el comando `'git push origin [nombre]'`. Si queremos subir un commit que hicimos en la rama Master, escribimos: `'git push origin Master'`.

8\_ Un repositorio remoto es una versión o una copia de tu repositorio guardada en internet o en otra red. Sirve de copia de seguridad, como medio para compartir el proyecto, etc.

9\_ Para agregar un repositorio remoto a Git debemos ingresar el comando `'git remote add origin [link del repositorio]'`.

10\_ Para empujar cambios a un repositorio remoto se usa el comando `'git push [nombre del repositorio] [nombre de la rama]'`. Se usa principalmente en repositorios de otras personas para enviar una corrección.

11\_ Un fork de un repositorio es una copia de un repositorio creada en una cuenta diferente permitiendo desarrollar cambios sin afectar al original. Es distinto del clonado, que descarga el repositorio localmente, el fork se realiza generando una copia en la cuenta del usuario.

12\_ Para crear un fork, vamos al repositorio, tocamos el botón que dice fork, nos da la opción de elegir con qué nombre nos va a aparecer en nuestro perfil, también nos da la opción de agregarle una descripción. Por último tocamos donde dice 'create fork' y está listo. Tendremos un fork del repositorio en nuestro perfil.

13\_ Para enviar un 'pull request', primero debemos ir al repositorio al cual queremos mandarle, luego le hacemos un fork para trabajar el proyecto desde ahí. Realizamos los

cambios que nos parezcan necesarios, 'hacemos git add' ., 'git commit' y por último 'git pull [nombre del repositorio] [nombre de la rama]' para subir los cambios a nuestro fork, refrescamos la pestaña de GitHub y nos va a aparecer la opción para hacer el pull request. La presionamos, ponemos toda la información sobre los cambios que hicimos, lo más claro posible, y enviamos la solicitud.

14\_ Para aceptar un pull request, entramos al repositorio al cual nos la enviaron, vamos al apartado de pull requests, la revisamos y si nos gustan o nos sirven los cambios, la aceptamos haciendo clic en merge pull request, luego nos da la opción de escribir un mensaje a quien nos mando la solicitud y por último tocamos confirm merge.

15\_ Un tag o etiqueta es una forma para identificar commits, es decir, etiquetar puntos específicos del historial como importantes.

16\_ Un tag se crea escribiendo en la consola el comando 'git tag [nombre]' para etiquetar el último commit, o si queremos etiquetar un commit anterior, escribimos 'git tag [nombre] [código de commit(checksum o hash corto)]' el checksum del commit lo vemos accediendo al historial de confirmaciones 'git log --oneline'.

17\_ Para enviar las tags a GitHub, en la consola de Git escribimos 'git push tags'.

18\_ Un historial de git, es una lista con todas las confirmaciones (commits) de un repositorio.

19\_ Para ver el historial, se usa el comando 'git log'.

20\_ Para buscar en el historial de Git, debemos ingresar el comando 'git grep [palabra clave]', y esto buscará en el historial de commits esa palabra clave. Por ejemplo, si yo necesito encontrar un commit y mi repositorio tiene miles, para no estar buscando uno por uno, escribo 'git commit cambio', y me aparecerá una lista con todos los commits que en su comentario tenga la palabra cambio. Haciendo mucho más fácil el encontrar el que estamos buscando.

21\_ Para borrar el historial de Git, se usa el comando 'git clear'.

22\_ Un repositorio privado en GitHub es un repositorio al cual solo puede acceder el creador del mismo y otros usuarios habilitados por este. Una curiosidad sobre los repositorios privados es que el historial de confirmaciones no se puede eliminar permanentemente, a menos que se elimine todo el repositorio.

23\_ Para crear un repositorio privado en GitHub, se debe entrar a la página de GitHub, hacer clic en new, y cuando nos de a elegir si hacerlo público o privado, ponemos que privado y le damos a create repository.

24\_ Para invitar a alguien a un repositorio privado, entramos al repositorio, clickeamos en invite collaborators, add people y escribimos el mail, nombre de usuario o nombre completo, seleccionamos al usuario que queremos agregar y apretamos add [nombre de usuario].

25\_ Un repositorio público en GitHub, es un repositorio al cual todos los usuarios pueden acceder, pueden hacer un fork, pueden clonar el repositorio, pueden enviar pull requests, etc. sin necesidad previa de recibir permiso del dueño del repositorio. Se utilizan para compartir software de código abierto, modificar y distribuir el software.

26\_ Crear un repositorio público en GitHub es muy parecido a crear uno privado. Primero debemos entrar a la página de GitHub, luego apretar el botón de new, ponerle un nombre al repositorio y seleccionar la opción de public. Damos clic en create repository y listo.

27\_ La forma más sencilla de compartir un repositorio público de GitHub, es enviando el link a este. Por ejemplo, en este trabajo comparto mi fork del repositorio de la materia, <https://github.com/lautarillo/UTN-TUPaD-P1>.