

## Trabajo Práctico N.º 2: Git y GitHub hecho por Shai Kohn

1A) GitHub es una plataforma para alojar repositorios remotamente, utilizada globalmente por los programadores para compartir su código y trabajar colaborativamente, entre otras cosas.

1B) Para crear un repositorio en GitHub debes ingresar a la página e ingresar a la sección de repositorios, donde aparece un botón que te llevará al menú de creación. En este deberás seleccionar el nombre del repositorio, la visibilidad del mismo (público o privado) y también tendrás otras opciones secundarias como si crear un readme o no.

1C) Para crear una rama en Git se utiliza el comando: **git branch**, seguido por el nombre que quieras darle a la rama. Ej: `git branch rama2`

1D) Para cambiar a una rama en Git se utiliza el comando: **git checkout**, seguido por el nombre de la rama. Ej: `git checkout rama2`

1E) Para fusionar ramas en Git primero debes asegurarte de que estes ubicado en la rama que quieras que reciba los cambios. Cuando estes seguro debes utilizar el comando **git merge**, seguido por el nombre de la rama que quieras fusionarle. En el caso de que haya errores (código que no coincida, por ejemplo) Git te lo indicará y podrás corregirlo en el momento, y si no hay la fusión se realizará automáticamente. Ej: `git merge rama2`

1F) En Git se crea un commit utilizando el comando **git commit -m**, seguido por el mensaje rodeado por comillas dobles. Ej: `git commit -m "Primer commit: Hola mundo"`.

1G) Se envía un commit a GitHub utilizando el comando **git push**. La primera vez que se ejecuta el comando se debe realizar la conexión con el repositorio remoto de la siguiente manera: `git push -u origin`, seguido por el nombre de la rama principal, que suele ser master o main.

1H) Un repositorio remoto es un contenedor de un proyecto que se encuentra hospedado en el internet para que su creador, u otras personas ajenas al proyecto, puedan acceder al mismo de forma remota, pudiendo copiarlo, editarlo o simplemente visualizarlo, dependiendo de la configuración del mismo.

1I) Para agregar un repositorio remoto a Git se utiliza el comando **git remote add origin**, seguida por el link del repositorio. Luego debes utilizar **git branch -M main**, para ubicarte en la rama principal y **git push -u origin main** para llevar los archivos del repositorio remoto al entorno local.

1J) Para empujar cambios a un repositorio remoto se utiliza el comando **git push**, pero antes se debe haber utilizado **git add** . (agrega todos los

archivos al stake) y **git commit -m "mensaje"** (para guardar los cambios).

1K) Para tirar cambios de un repositorio remoto se utiliza el **comando git pull origin**, seguido por el nombre de la rama. Ej: git pull origin main.

1L) Un fork de un repositorio es una copia de un repositorio en el que puedes trabajar libremente, teniendo como base los archivos del original, pero teniendo la posibilidad de agregar, editar y borrar archivos.

1M) Para crear un fork de un repositorio debes ingresar a la sección de código del mismo y seleccionar el botón fork, lo que te llevará a un menú donde deberás ponerle un nombre a la copia que estás creando, tras lo que luego solo basta con apretar en el botón de crear y ya tendrás tu fork.

1N) Para enviar una pull request a un repositorio remoto debes ingresar a la sección de pull requests del repositorio y seleccionar el botón de New pull request, donde deberás seleccionar que ramas quieres fusionar y realizar un commit que explique los cambios realizados y el por qué crees que se debería fusionar las ramas.

1Ñ) Para aceptar una pull request debes ingresar a la sección de pull requests, donde deberás buscar el que quieras aprobar y apretar en el mismo, lo que te llevará a visualizar con mayor profundidad la información del mismo: que ramas se quieren fusionar, los cambios realizados, el commit, etc. Luego de analizar que todo este correcto puedes apretar en el botón de aprobar pull request y listo. Dependiendo de la configuración del proyecto se realizarán los cambios automáticamente o no (los pull request pueden necesitar más de una aprobación según lo que especifique la configuración).

1O) Una etiqueta en Git es una referencia que apunta a un punto en concreto en el historial de Git.

1P) Para crear una etiqueta en Git se utiliza el comando **git tag**, seguido por el nombre que quieras darle.

1Q) Para enviar una etiqueta a GitHub se utiliza el comando **git push origin**, seguido por el nombre de la etiqueta.

1R) El historial de Git es un registro de los commits que se enviaron al repositorio.

1S) Para ver el historial de Git se utiliza el comando **git log**.

1T) No existe un comando para borrar el historial de Git, lo que se podría hacer es crear una nueva rama (que tendrá el historial vacío), agregarle el código de la rama anterior, para luego eliminar esta y convertir la nueva rama en la principal.

1U) Un repositorio privado en GitHub es un repositorio que por defecto es solo accesible a su creador, quien a su vez podrá otorgarles permisos a otras personas para poder visualizarlo o editarlo.

1V) Para crear un repositorio privado en GitHub debes ingresar a la página e ingresar a la sección de repositorios, donde aparece un botón que te llevará al menú de creación en el que tendrás la opción de elegir si el repositorio a crear será público o privado, por lo que bastaría con elegir la opción de privado.

1W) Para invitar a alguien a un repositorio privado en GitHub debes acceder a la sección de configuración del repositorio y luego acceder en el menú al panel de colaboradores, aunque primero al apretar en él tendrás que volver a introducir tu contraseña para confirmar que eres el dueño del repositorio. Ya en el panel tendrás un botón que abrirá un buscador por el que podrás buscar usuarios por su nombre de usuario, su nombre real o su email. A las personas que decidas invitar les llegará un email con la invitación a colaborar en el repositorio, la que deberán aceptar para poder tener acceso al mismo.

1X) Un repositorio público es un repositorio que por defecto es accesible para todo el mundo, tanto para visualizarlo, como editarlo o realizar forks del mismo.

1Y) Para crear un repositorio público en GitHub debes ingresar a la página e ingresar a la sección de repositorios, donde aparece un botón que te llevará al menú de creación en el que tendrás la opción de elegir si el repositorio a crear será público o privado por lo que bastaría con elegir la opción de público.

1Z) Para compartir un repositorio público en GitHub tienes la posibilidad de enviar el link del repositorio a las personas que quieras que accedan al mismo o sino puedes directamente invitarlos desde la sección de colaboradores.

2) <https://github.com/Shaikohn/Github-Ejercicio2-Shai>

3) <https://github.com/Shaikohn/GitHub-Ejercicio3-Shai>