

Programación 1 Práctico 2:

Valentín Piñeyro

1)

- a. GitHub es una comunidad donde se pueden compartir repositorios de forma pública o privada. Funciona como una red social, con lo cual cada usuario tiene un perfil en el cual puede subir sus repositorios y además acceder a otros perfiles para ver los repositorios de otros usuarios y, por ejemplo, clonarlos o ponerlos en nuestros favoritos.
- b. Para crear un repositorio en GitHub, se despliega el botón de “+” en la página de GitHub y se selecciona la opción de nuevo repositorio (o en GitHub desktop: Add -> Create new repository). A partir de eso se puede añadir nuestro repositorio local a través del comando `git remote add origin` y la dirección de nuestro repositorio remoto.
- c. Git proporciona una rama por default llamada master o main. Luego si se quiere agregar una nueva rama se debe hacer a través del comando `git branch <nombre de la rama>` o `git checkout -b <nombre de la rama>` (este segundo comando no solo crea la rama sino que nos chequea inmediatamente en esa nueva rama).
- d. Para cambiar a una rama en Git se debe usar el comando `git checkout <nombre de la rama>`.
Nuevamente, si se quiere cambiar a una rama que no existe se puede simplificar la creación de la rama y el checkout a la misma con el comando `git checkout -b <nombre de la rama>`.
- e. Para fusionar ramas en Git, se puede usar el comando `git merge <rama a fusionar>`, el cuál combina la rama a fusionar con la rama en la cual se está trabajando actualmente.
- f. Para crear un commit en Git, se usa el comando `git commit`, el cuál actualiza nuestros cambios en el repositorio local.
- g. Para enviar un commit a GitHub, se debe usar el comando `git push`, el cual actualiza el repositorio remoto con las modificaciones que hayamos incorporado a nuestro repositorio local.
- h. Un repositorio remoto es un repositorio almacenado en GitHub, al cuál se puede acceder a través de internet desde cualquier computadora. Esto permite el trabajo colaborativo entre personas, además de la posibilidad de trabajar desde distintas computadoras.

- i. Para empujar cambios a un repositorio remoto se debe usar el comando git push. Este comando actualiza el repositorio remoto con el repositorio local según las ramas en las que se esté trabajando. El comando puede ser más preciso si se le agrega git push origin(repositorio remoto por default) <nombre de la rama> (Cabe aclarar que el <nombre de la rama> se actualizará tanto en el repositorio local como el remoto). Para más precisión en el manejo de ramas se puede descomponer el campo <nombre de la rama> en <origen>:<destino> (usando :).
- j. Para tirar de cambios de un repositorio remoto se puede usar el comando git fetch o git pull. La diferencia entre los comando es que git pull copia los archivos a nuestro directorio (ya que en realidad está ejecutando dos comandos en un: un fetch al origen y un merge con nuestro repositorio local), mientras que git fetch solo copia los cambios a nuestro repositorio local.
- k. Un fork de repositorio es una copia exacta de un repositorio. Es de mucha utilidad ya que permite clonar repositorios enteros de otras personas y viceversa.
- l. Para crear un fork de un repositorio se ingresa a dicho repositorio y en la opción de fork se selecciona crear nuevo fork.
- m. Para enviar una solicitud de extracción, se accede al repositorio y, dentro de la pestaña pull requests, se crea una nueva solicitud de extracción. Esta sirve para fusionar determinados cambios a una rama.
- n. Para aceptar una solicitud de extracción, se accede al repositorio y, dentro de la pestaña pull requests, donde se muestran las solicitudes activas, se puede revisar y aceptar o rechazar la solicitud.
- o. Una etiqueta en Git es una marca que se le puede asignar a un determinado commit para diferenciarlo de los demás. Esto sirve para mantener ordenado el repositorio y suele usarse cuando grandes cambios son implementados.
- p. Para crear una etiqueta en Git se debe usar el comando git tag <nombre de la etiqueta>. De no especificarse un punto en concreto, este comando crea la etiqueta donde se encuentre el HEAD actualmente. Si se quiere, se puede especificar la ubicación de la etiqueta con el comando git tag <nombre de la etiqueta> <ubicación>.
- q. Para enviar una etiqueta a GitHub, una vez creada la etiqueta se puede usar el comando git push tag <nombre del tag> para enviar la etiqueta al repositorio remoto.
- r. El historial de Git es el registro de todos los cambios realizados al repositorio.
- s. Para buscar en el historial de git se puede usar el comando git log, el cual muestra la lista de todos los commits realizados.

- t. Para borrar el historial de Git se puede usar el comando `git rebase`, el cuál mueve todos los cambios a una nueva rama. Luego de esto se puede borrar la rama anterior y todo su historial. Otra forma posible sería crear una nueva rama, importar todo el contenido de la rama anterior a la nueva y, finalmente, borrar la rama de la cual se desea borrar el historial.
- u. Un repositorio privado en GitHub es un repositorio al cual sólo pueden acceder las personas que estén aprobadas por la persona que creó el repositorio. De la misma manera, nadie podrá ver el repositorio a menos que tengan acceso al mismo.
- v. Para crear un repositorio privado en GitHub se debe especificar que este es privado a la hora de su creación.
- w. Para invitar a alguien a un repositorio privado en GitHub, se debe ir a la sección de Acceso a colaboradores del repositorio. Desde ahí se pueden gestionar los permisos de acceso al repositorio.
- x. Un repositorio público en GitHub es un repositorio al cual puede acceder cualquier persona, ya sea a través de un link al mismo o a través del perfil de la persona que lo haya creado.
- y. Para crear un repositorio público en GitHub, de la misma manera que con uno privado, se debe especificar que este será público a la hora de su creación.
- z. Para compartir un repositorio público en GitHub, se puede hacer desde la sección de Acceso a colaboradores o, simplemente, compartiendo el link del mismo.

2) Repositorio de GitHub: https://github.com/i4nkvrtjs/Practico_2_Ejercicio_2

3) Repositorio de GitHub: <https://github.com/i4nkvrtjs/conflict-exercise>