

Trabajo Práctico Nº2 “Git y GitHub”

Nelson, Cristhian Alejandro

Actividades

- 1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada:
 - a) GitHub es una plataforma de alojamiento de código basada en la nube que utiliza el sistema de control de versiones **Git**. Permite a los desarrolladores colaborar en proyectos, gestionar repositorios, revisar código, realizar seguimiento de issues y desplegar aplicaciones
 - b) Hay que hacerse una cuenta e iniciar sesión en GitHub, ir a la esquina derecha donde figura “New Repository”. Darle un nombre, dar una descripción (la cual es opcional) y elegir si el repositorio va a ser público o privado. Se puede agregar también un archivo README.md. Y para finalizar clicar en “Create Repository”.
 - c) Se utiliza el comando *“git branch nombre-de-la-rama”*
 - d) Se utiliza el comando *“git checkout nombre-de-la-rama”*
 - e) Cambiamos a la rama de destino (*git checkout main*) y fusionamos la rama usando *git merge feature-x*
 - f) Utilizamos *git add .* y luego *git add miarchivo.txt*. Luego usamos el commit *git commit -m “Cambio”* (lo que está en comillas significa la descripción de los cambios que se hicieron).
 - g) Para enviar un commit a GitHub usamos *git push origin nombre-de-la-rama*
 - h) Un repositorio remoto es una versión del repositorio alojada en un servidor. Permite la colaboración y sincronización de cambios entre varios desarrolladores.
 - i) Para agregar un repositorio remoto a GitHub usamos *git remote add origin* (luego de *origin* iría la dirección del repositorio que queremos agregar)
 - j) Para empujar cambios a un repositorio remoto usamos *git push -u origin nombre-de-la-rama*
 - k) Para tirar de cambios de un repositorio remoto usamos *git pull origin nombre-de-la-rama*
 - l) Un *fork* es una copia personal de un repositorio de otro usuario en tu cuenta de GitHub. Permite modificar el código sin afectar el proyecto original. Usado comúnmente para contribuir a proyectos open source.

- m) Un *fork* de repositorio se crea dirigiéndose al repositorio original, luego clickeamos en la esquina superior derecha donde dice *Fork* y elegimos nuestra cuenta como destino.
- n) Para enviar una solicitud de *pull request* a un repositorio debemos hacer *fork* del repositorio original, clonar el fork usando *git clone url*, crear una rama para los cambios ejecutando *git checkout -b nombre-rama*, realizar los *commits* correspondientes y luego enviar los cambios al fork ejecutando *git push origin nombre-rama*. Una vez hecho todo esto, entrar al fork y clicar en “Compare and pull request”. Finalmente describir los cambios y clicar “Create pull request”.
- o) Para aceptar una solicitud de extracción vamos a la pestaña “Pull requests”, seleccionamos la solicitud y revisamos los cambios. Si está todo bien, hacemos click en “Merge pull request”.
- p) Una etiqueta (tag) en Git es una marca estática en un commit específico (la cual no cambia como las ramas).
- q) Para crear una etiqueta ejecutamos el comando *git tag*.
- r) Para enviar una etiqueta a GitHub usamos el comando *git push origin* para una etiqueta específica o *git push origin --tags* para todas las etiquetas
- s) El historial de Git es el registro de todos los commits realizados en un repositorio, mostrando autor, fecha, mensaje y cambios.
- t) Para ver el historial ejecutamos el comando *git log*.
- u) Podemos buscar en el historial de Git usando *git log --oneline* o *git log --follow archivo*.
- v) Borramos historial usando *git reset*.
- w) Un repositorio privado es un repositorio **solo accesible** para colaboradores explícitos. Ideal para proyectos cerrados o empresariales.
- x) Creamos un repositorio privado yendo a “New Repository”, elegimos “Private” en visibilidad, completamos campos faltantes y damos click a “Create repository”.
- y) Para invitar a alguien a un repositorio privado vamos a “Settings -> Collaborators” y hacemos click en “Add people”. Ingresamos el nombre de usuario o mail y elegimos permisos de admin, lectura y/o escritura.
- z) Un repositorio público es visible para **cualquier persona**, pero solo los colaboradores pueden editarlo (a menos que sea un fork). Ideal para open source.
- aa) Crear un repositorio público es el mismo proceso que un privado, pero seleccionamos “Public” en la visibilidad al crearlo.
- bb) Podemos compartir un repositorio público mediante URL directa, redes sociales o incluso mediante archivo README.md

2) Adjunto el link donde está resuelto el ejercicio:

<https://github.com/CristhianNelson/act2-mi-archivo>

3) Adjunto el link donde está resuelto el ejercicio:

<https://github.com/CristhianNelson/conflict-exercise>