

Práctico 2: Git y GitHub

Alumno: Facundo Gaston Vazquez

DNI: 40.717.820

Comisión: 22

1)

¿Qué es GitHub?

Github es como una red social para desarrolladores donde se comparten proyectos mediante repositorios, los cuales podemos descargar de forma local para ver y modificar. Además, también funciona en conjunto con Git para poder realizar colaboraciones entre equipos que trabajan en un mismo proyecto.

¿Cómo crear un repositorio en GitHub?

Para poder crear un repositorio en Github primero debemos de verificar nuestros datos mediante nuestro editor de código, en este caso al estar utilizando VSC, debemos ingresar a la terminal, iniciar un repositorio con el comando `git init` y entrar a Github para que nos dé el link para que cuando agreguemos los cambios se suban directamente mediante un `push`.

¿Cómo crear una rama en Git?

Para crear una rama nueva dentro de nuestro proyecto, primero debemos pararnos sobre la dirección correspondiente a nuestro proyecto. Luego de eso debemos de ingresar por consola el comando `Git branch "nombre de la nueva rama"` y oprimir enter.

¿Cómo cambiar a una rama en Git?

Para cambiar a otra rama del proyecto debemos ingresar por consola el comando `git checkout "nombre de la rama a la que queremos ingresar"`.

¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git, primero debemos fijarnos que estemos parados en la rama principal del proyecto o en su defecto en la rama donde queremos que se una la otra. Para ello tenemos que utilizar por consola el comando `git merge "nombre-de-la-rama"`

¿Cómo crear un commit en Git?

Para crear un nuevo commit en Git, primero debemos asegurarnos de guardar los cambios con “git add .” una vez guardados los cambios podemos hacer un git commit – m “comentarios” . Los comentarios dentro del commit suelen reflejar como buena práctica una especificación clara del cambio o agregado que realizamos en esta instancia.

¿Cómo enviar un commit a GitHub?

Para que los commits realizado en git figuren en github, como mencionamos anteriormente, debemos de tener linkeado nuestro repositorio local a github de esta forma cada vez que hagamos un git push, todos los cambios incluidos los commits se verán de forma online.

¿Qué es un repositorio remoto?

Un repositorio remoto es la copia de un proyecto pero que se aloja en un servidor externos, el cual nos permite ingresar desde otros dispositivos y facilita la colaboración entre equipos.

¿Cómo agregar un repositorio remoto a Git?

Para esto el primer paso, es tener una cuenta de github. Luego debemos de autenticar nuestras credenciales en git para que podamos hacer un git remote para linkear el proyecto local con el servidor de github.

¿Cómo empujar cambios a un repositorio remoto?

Para empujar cambios, como la misma palabra lo indica debemos de realizar por consola el comando git push.

¿Cómo tirar de cambios de un repositorio remoto?

Para ello debemos de realizar desde nuestro repositorio local un git fetch, de esta forma se descarga los últimos cambios realizado al repositorio, luego de esto los cambios no se aplican directamente si no que tendremos que hacer un git merge de la nueva actualización que descargamos para que se unan al proyecto local.

¿Qué es un fork de repositorio?

Un fork de un repositorio es básicamente poder descargarnos mediante la plataforma Github un repositorio público o “open source” para que quede alojado en nuestra cuenta y así en caso de necesitarlo hacer modificaciones y enviarlas como revisión al creador del repositorio original.

¿Cómo crear un fork de un repositorio?

Basta con ingresar al link del repositorio en github y luego apretar en el botón del margen derecho arriba para descargarlo a nuestra cuenta, también podemos cambiar el nombre del repositorio y agregar comentarios.

¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

Este sería el paso a paso para enviar un pull request:

1. Preparación de los cambios:

- Crear una rama
- Realizar los cambios
- Agregar y confirmar los cambios
- Enviar la rama al repositorio remoto

2. Crear la solicitud de extracción (pull request)

- Ir al repositorio remoto
- Encontrar la rama
- Crear la solicitud de extracción (Añadir información)
- Enviar la solicitud de extracción

3. Revisión y fusión

¿Cómo aceptar una solicitud de extracción?

Inicialmente, se accede a la solicitud en la pestaña correspondiente del repositorio y se examinan los cambios en "Files changed". Una vez que los cambios cumplen con los estándares del proyecto, se aprueba la solicitud con el botón "Approve". Luego, se fusionan los cambios en la rama principal con el botón "Merge pull request", eligiendo la opción adecuada.

¿Qué es una etiqueta en Git?

En Git, una etiqueta (tag) es una marca que apunta a un punto específico en la historia del repositorio. Se usa para marcar versiones de software (ej., "v1.0") o hitos importantes.

¿Cómo crear una etiqueta en Git?

Para crear una etiqueta ligera, se utiliza el comando `git tag <nombre_etiqueta>`, mientras que, para una etiqueta anotada, se emplea `git tag -a <nombre_etiqueta> -m "<mensaje>"`.

¿Cómo enviar una etiqueta a GitHub?

Las etiquetas no se envían automáticamente al repositorio remoto con `git push`, por lo que se debe utilizar `git push origin --tags` para enviarlas todas.

¿Qué es un historial de Git?

El historial de git es un vistazo a todos los cambios que realizamos durante el proyecto y que hemos ido guardando.

¿Cómo ver el historial de Git?

Para ver el historial de git podemos ingresar por consola el comando: `git log`

¿Cómo buscar en el historial de Git?

Hay varios comandos de búsqueda que nos permiten buscar algo en el historial de git de forma más específica como, por ejemplo:

- `git log --grep="corrección de error"`: buscará todos los commits que contengan la frase "corrección de error".
- `git log -S"funciónX"`: buscará commits que hayan añadido o eliminado la palabra "funciónX".
- `git log archivo.txt`: mostrará los commits que han cambiado el archivo "archivo.txt".

Así entre otros más que permiten añadir especificación a nuestras búsquedas.

¿Cómo borrar el historial de Git?

Podemos borrar el historial con `"git reset"`

¿Qué es un repositorio privado en GitHub?

Un repositorio privado en Github es un repositorio el cual no es de acceso público para todos los usuarios si no que solamente pueden acceder los colaboradores que tengas el acceso permitido por el dueño del repositorio.

¿Cómo crear un repositorio privado en GitHub?

Desde la pestaña inicial del inicio de github es igual que crear un repositorio público, solo que antes de conectar con el repositorio local debemos seleccionar privado.

¿Cómo invitar a alguien a un repositorio privado en GitHub?

1. Ir a la página principal de tu repositorio privado en GitHub.
2. Haz clic en la pestaña "Settings" (Configuración).
3. En la barra lateral izquierda, busca la sección "Access" (Acceso) y haz clic en "Collaborators & teams" (Colaboradores y equipos).
4. Haz clic en el botón "Add people" (Agregar personas).
5. Haz clic en el botón "Add [nombre de usuario] to [nombre del repositorio]" (Agregar [nombre de usuario] a [nombre del repositorio]).

¿Qué es un repositorio público en GitHub?

El repositorio público de Github es básicamente un repositorio que está a la vista de todos los usuarios para poder ver y descargar en caso de revisar o editar el código de dicho proyecto.

¿Cómo crear un repositorio público en GitHub?

Para crear un repositorio público en github es lo mismo que el privado, solamente que antes de dar a crear repositorio, debemos cambiar la opción de privado a público.

¿Cómo compartir un repositorio público en GitHub?

Para poder compartir el repositorio publico basta con enviar el link del repositorio de la página de Github para que otro pueda verlo.

Ejercicio 2: <https://github.com/facuvazquez1/TrabajoColaborativo-Ej-2>

Ejercicio 3: <https://github.com/facuvazquez1/conflict-exercise>