

Trabajo Practico

Nº2 Git y GitHub

Nombre y apellido: Juan cruz Ferraresi

1)

. Es un software especializado en llevar un control eficiente del ciclo de vida de todo proyecto además de los colaboradores que trabajan en él.

. Para crear un repositorio en GitHub hay que ingresar a la página web, crear una cuenta y una vez adentro hay que darle clic a un botón “+” (signo de más), el cual esta en la esquina superior derecha, y luego presionar el botón que dice “New repository”. Esto te va a abrir una pagina “Create a new repository”, donde hay que completar diferentes campos, por ejemplo: El nombre – La descripción (opcional) – Poner el repositorio publico o privado. Y por último darle clic al botón verde que dice “Create repository”

. Para crear una rama nueva en Git hay que poner el comando “git branch “nombre de la rama”

. Para cambiar a una rama hay que poner el comando “git checkout “nombre de la rama”

. Para fusionar ramas primero tienes que cambiar a la rama donde quieras integrar los cambios y ahí poner el comando “git merge “nombre de la rama a fusionar”. En el debido caso también se deberá solucionar los conflictos si es que hay.

. Antes de crear un commit, se necesita indicar a Git que archivos y modificaciones quieres incluir en ese commit, para eso se usa el comando “git add . “. Luego de haber agregado los cambios se puede crear el commit con el comando “Git commit”. Es importante dejar un mensaje descriptivo con tu commit para explicar que cambios se realizaron, el comando completo seria “git commit -m “descripción”

. Para enviar un commit a GitHub se necesita usar el comando “git push (nombre del remoto) (nombre de la rama). Un ejemplo común seria “git push origin main”. Pero si es la primera vez que se envia una rama nueva es recomendable usar el flag “-u”. Esto establece una conexión de seguimiento entre tu rama local y la remota. Ej: git push -u origin (nombre de la rama nueva)

. Un repositorio remoto es la versión en línea de tu proyecto Git, esencial para la colaboración, el backup y la gestión del ciclo de vida del desarrollo de software

. Se tiene que abrir la terminal o línea de comandos y se busca el directorio raíz de tu proyecto Git local. Se busca la URL desde la plataforma correspondiente y se se clickea donde indica “Clone” o “Code”. Se ejecuta el comando “git remote add (nombre del remoto) y (URL del repositorio remoto).

. Para empujar cambios a un repositorio remoto se tiene que haber commiteado los cambios locales, identificar el nombre del remoto (originalmente “origin”), se especifica la rama local que quieras enviar y se ejecuta el código “git push origin (nombre de tu rama local).

. Los cambios desde un repositorio remoto a tu repositorio local en Git, utilizas el comando git pull. Este comando esencialmente realiza dos acciones en una: primero, descarga los commits y las referencias del repositorio remoto. Se cambia la rama local donde quieres integrar los cambios

remotos “git checkout (tu rama local)”. Se ejecuta git pull (nombre del remoto) (nombre de la rama remota)”.

. Un **fork** de un repositorio en el contexto de plataformas como GitHub, GitLab o Bitbucket es esencialmente una **copia personal** del repositorio de otra persona (o de una organización) que reside en tu propia cuenta.

. Para crear un Fork de un repositorio se realiza desde la interfaz web, tiene varios pasos para seguir:

Se entra a la pagina del repositorio, se busca el botón Fork, que generalmente se encuentra en la parte superior derecha. Se hace click en el botón Fork y selecciona donde quieres forkar el repositorio, se configura los detalles (un paso opcional) y por ultimo se confirma la creación del Fork.

. Antes de enviar una solicitud de extracción hay pasos que se deben haber realizado antes:

-Haber forkeado el repositorio -Haber clonado tu Fork localmente -Se ha realizado cambios y commits -Y por último se envia los commits desde el local a la plataforma

Luego de haber hecho estos pasos se puede empezar a realizar la Pull request. Generalmente la plataforma te mostrara una notificación o un botón prominente indicando que la rama recién subida esta lista para ser comparada y enviada como Pull Request al repositorio original.

Se selecciona las ramas a comparar, se escribe un título y una descripción para el Pull Request. Se crea la Pull Request, se espera la revisión y se realizan cambios adicionales (si es necesario) y por ultimo se fusiona los cambios aprobados.

. Aceptar una Solicitud de Extracción (Pull Request o PR) es la acción que realizan los mantenedores o colaboradores con permisos de escritura en el repositorio original para integrar los cambios propuestos desde un fork. Los pasos serian:

Se recibe la notificación del Pull Request, se revisa y evalua la calidad y el impacto de los cambios. Si es necesario se comunica tus comentarios, se solicitan cambios y se realizan pruebas. Se aprueba el Pull Request, se fusiona la rama del Pull a la rama base, luego se cierra el Pull Request y se elimina la rama fusionada.

. Es una forma de marcar un punto específico en la historia de tu repositorio como importante. Hay dos tipos de etiquetas las ligeras y las anotadas

Las etiquetas ligeras son simplemente un puntero a un commit específico. Son fáciles de crear, pero no contienen información adicional como el autor o la fecha de la etiqueta, ni permiten un mensaje.

Las etiquetas anotadas son objetos completos en la base de datos de Git. Contienen más información, como el nombre del etiquetador, la dirección de correo electrónico, la fecha de la etiqueta y un mensaje. Se recomienda usar etiquetas anotadas para lanzamientos, ya que proporcionan más contexto.

. Para crear una etiqueta Ligera se usa el comando “git tag (nombre de la etiqueta)” y para crear una etiqueta Anotada el cual usa los siguientes comandos -a (annotated) y -m (message), el código seria: “git tag -a (nombre de la etiqueta) -m (mensaje de la etiqueta)”

. Para enviar una etiqueta que has creado localmente a tu repositorio remoto en GitHub (o cualquier otro repositorio Git remoto), necesitas usar el comando git push con opciones

específicas para las etiquetas. Si quieres enviar todas las etiquetas locales de una vez tienes que usar el comando “git push (nombre del remoto a enviar las etiquetas) –tags

En cambio, si quieres enviar una etiqueta en particular puedes especificar su nombre directamente en el comando. Ej: git push (nombre del remoto) (nombre de la etiqueta)

. El historial de Git es un registro cronológico de todos los cambios que se han realizado en un repositorio desde su inicio. Es como una línea de tiempo detallada de la evolución de tu proyecto.

. Para ver el historial de tu repositorio Git, el comando fundamental es “git log”. Este comando tiene muchas opciones para personalizar la forma en que se muestra el historial.

-Ver el historial básico “git log” mostrara el historial de commits en orden cronológico inverso.

-Ver el historial de forma más compacta “git log –oneline” muestra cada commit en una sola línea

-Ver el historial con un grafico de ramas y fusiones “git log --graph --oneline –decorate” este comando dibuja una representación visual de como las ramas se han bifurcado y fusionado

Hay muchos más comandos para ver de diferentes formas el historial.

. Buscar en el historial de Git es una habilidad muy útil para encontrar commits específicos basados en diferentes criterios, en este caso también tiene muchas formas de realizar las búsquedas, voy a destacar alguna.

-Buscar por mensaje del commit “git log --grep=(“palabra clave”)”. Esta es la forma más común de buscar

-Buscar por autor, se puede filtrar el historial por el autor del commit: git log --author= “nombre del autor”-

. Para borrar el historial hay varias formas de hacerlo:

-Eliminar el ultimo commit (si no lo has pusheado): git reset --soft HEAD^. Esto moverá el puntero “HEAD” al commit anterior, pero mantendrá los cambios.

-Modificar commits anteriores (si no los has pusheado), puedes usar el código “git rebase -i HEAD~n (la n es el numero de commits que quieres revisar.

. Un repositorio privado en GitHub es un repositorio de Git que solo puede ser visto y accedido por el propietario de la cuenta que lo creó y por las personas específicas a las que el propietario les ha concedido permiso de acceso.

. La forma de crear un repositorio privado es similar a crear uno público, se accede a la página web y se sigue los pasos como para crear un repositorio nuevo, pero en vez de dejar marcada la opción de “public” se selecciona la opción “private”

. Para invitar a alguien al repositorio privado se tiene que entrar en la configuración, navegar por la sección de “Collaborator and Teams”, se entra a la sección llamada “manage Access” y se hace click en donde dice “add people”. Se busca al usuario en el campo de búsqueda por el nombre de usuario, se lo selecciona, se puede revisar los permisos que se le otorgaran antes de añadirlo y por ultimo se hace clic en “add “nombre de usuario” to the repository”. Ahí le va a llegar una notificación por correo electrónico al invitado y deberá aceptar.

. Un repositorio público en GitHub es un repositorio de Git cuyo contenido es visible para cualquier persona en internet. Esto significa que cualquiera puede navegar por el código fuente, ver el

historial de commits, leer las issues, los pull requests y el resto del contenido del repositorio sin necesidad de tener una cuenta de GitHub o de iniciar sesión.

. Para crear un repositorio público, hay que seguir los mismos pasos para crear un nuevo repositorio y la opción genérica que te elige es la pública, así que mientras crees uno nuevo sin modificar nada se tendría que dejar publico

. La forma de compartir un repositorio publico en GitHub es compartiendo la URL ya sea directamente a alguien o por redes sociales.