

Práctico 2: Git y GitHub

Ignacio Denunciato

1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :

- ¿Qué es GitHub?

GitHub es una comunidad donde nosotros podemos compartir nuestros repositorios de forma privada o publica.

- ¿Cómo crear un repositorio en GitHub?

para crear un nuevo repositorio solo hay que ir a la opción de “create new repository” y llenar los datos necesarios, una vez creado, hay que hacer un clone del repositorio a nuestra computadora para poder empezar a trabajar en el localmente.

- ¿Cómo crear una rama en Git?

para crear una nueva rama se utiliza el comando “git branch nombre-de-la-rama”, lo que crea la rama localmente

- ¿Cómo cambiar a una rama en Git?

para cambiar a una rama en git se utiliza el comando “git checkout nombre-de-la-rama”

- ¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git, primero hay que cambiar a la rama principal con “git checkout master” y una vez ahí, tendríamos que utilizar el comando “git merge nombre-de-la-rama” para integrar los cambios de otra rama

- ¿Cómo crear un commit en Git?

para crear un commit lo mejor es utilizar el comando “git commit -m (mensaje del commit)”, esto guarda los cambios con el mensaje proporcionado

- ¿Cómo enviar un commit a GitHub?

para guardar un commit en un repositorio remoto se utiliza el comando “git push origin nombre-de-la-rama”, esto enviará el commit a la rama correspondiente en GitHub

- ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión de tu repositorio Git que está alojada en un servidor, como GitHub. Permite a los usuarios colaborar en un proyecto de manera centralizada, compartir y sincronizar cambios entre diferentes máquinas

- ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a Git, utiliza el comando “git remote add” seguido del nombre del remoto (por ejemplo, “origin”) y la URL del repositorio

- ¿Cómo empujar cambios a un repositorio remoto?

para empujar o enviar cambios a un repositorio remoto se utiliza el comando “git push origin nombre-de-la-rama”, esto envía los cambios de la rama local al repositorio remoto

- ¿Cómo tirar de cambios de un repositorio remoto?

para tirar o traer cambios de un repositorio remoto se utiliza el comando “git pull origin nombre-de-la-rama”, esto descargará y fusionará los cambios de la rama especificada del repositorio remoto a nuestra rama local.

- ¿Qué es un fork de repositorio?

Un fork de repositorio es una copia personal de un repositorio que se encuentra en un servidor remoto (como GitHub). Al hacer un fork, puedes realizar cambios en el repositorio sin afectar el proyecto original.

- ¿Cómo crear un fork de un repositorio?

para crear un fork, primero debemos ir al repositorio que queremos copiar y hacer click en el botón “fork” en la parte superior derecha de la pagina, GitHub creará una copia del repositorio en nuestra cuenta, permitiéndonos trabajar de manera independiente en nuestra propia versión sin afectar el proyecto original. Luego, podremos clonar el fork a nuestra máquina local para empezar a hacer cambios.

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

Para enviar una solicitud de extracción (pull request) a un repositorio, primero hacemos los cambios en nuestro fork del repositorio. Luego, sigo estos pasos:

1. Hacemos un push de nuestros cambios a la rama correspondiente en nuestro repositorio remoto.
2. Vamos al repositorio original al que queremos contribuir.
3. En la página del repositorio original, hacemos clic en el botón "New pull request"
4. GitHub nos muestra una comparación entre nuestra rama y la rama principal del repositorio original. Revisamos los cambios y, si todo está bien, hacemos clic en "Create pull request".
5. Escribimos una descripción de los cambios realizados y finalmente hacemos clic en "Create pull request" para enviarla.

Ahora los dueños del repositorio original pueden revisar los cambios y decidir si los fusionan en el proyecto principal.

- ¿Cómo aceptar una solicitud de extracción?

Para aceptar una solicitud de extracción (pull request), primero revisamos los cambios propuestos en GitHub. Si todo está correcto y no hay conflictos, podemos hacer clic en el botón "Merge pull request". Luego, confirmamos la fusión haciendo clic en "Confirm merge". Esto integrará los cambios de la solicitud de extracción en nuestra rama principal o en la rama que hayamos seleccionado.

- ¿Qué es un etiqueta en Git?

Una etiqueta en Git es una referencia que se utiliza para marcar puntos específicos en la historia del proyecto, como versiones o hitos importantes. A diferencia de las ramas, las etiquetas no se actualizan, lo que las convierte en una manera ideal de marcar versiones estables del proyecto. Usualmente, se usan para identificar lanzamientos (por ejemplo, **v1.0.0**) y permiten a los desarrolladores volver fácilmente a esos puntos en el historial del repositorio.

- ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta en Git, usamos el comando `git tag` seguido del nombre de la etiqueta. Por ejemplo `git tag v1.0.0`, esto creará una etiqueta llamada `v1.0.0` en el último commit.

- ¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta a GitHub, primero creamos la etiqueta localmente usando el comando `git tag`. Luego, usamos el comando `"git push origin v1.0.0"` para empujar la etiqueta al repositorio remoto. Esto enviará la etiqueta v1.0.0 a GitHub. Si queremos enviar todas las etiquetas locales de una vez, usamos `"git push origin --tags"`, Esto sube todas las etiquetas que hemos creado al repositorio remoto en GitHub.

- ¿Qué es un historial de Git?

El historial de Git es un registro de todos los cambios realizados en un repositorio, incluyendo los commits, las ramas, las fusiones y las etiquetas. Cada vez que realizamos un commit, Git guarda un registro de los cambios realizados, junto con un identificador único (hash), la fecha y el autor del cambio. Este historial permite a los desarrolladores ver cómo ha evolucionado el proyecto a lo largo del tiempo, realizar un seguimiento de los cambios, revertir a versiones anteriores y colaborar de manera más eficiente en el desarrollo del código. El historial de Git es un registro de todos los cambios realizados en un repositorio, incluyendo los commits, las ramas, las fusiones y las etiquetas. Cada vez que realizamos un commit, Git guarda un registro de los cambios realizados, junto con un identificador único (hash), la fecha y el autor del cambio. Este historial permite a los desarrolladores ver cómo ha evolucionado el proyecto a lo largo del tiempo, realizar un seguimiento de los cambios, revertir a versiones anteriores y colaborar de manera más eficiente en el desarrollo del código.

- ¿Cómo ver el historial de Git?

para ver el historial de Git, usamos el comando `"git log"`. Este comando muestra una lista de todos los commits realizados en el repositorio, mostrando el identificador único de cada commit (hash), el autor, la fecha y el mensaje del commit

- ¿Cómo buscar en el historial de Git?

Para buscar en el historial de Git, usamos el comando `"git log"` junto con la opción `"--grep"` si estamos buscando un término específico en los mensajes de commit. Por ejemplo `"git log --grep=(mensaje a buscar)"`. Esto nos mostrará todos los commits cuyos mensajes contengan la palabra o frase que estamos buscando. Si queremos buscar por un cambio específico en el contenido de los archivos, podemos usar `"git log -S (texto a buscar)"`. Este comando nos mostrará los commits donde se añadió o eliminó el texto que buscamos. También podemos agregar opciones como `"--oneline"` para ver un historial más compacto.

- ¿Cómo borrar el historial de Git?

Si queremos borrar todo el historial y empezar desde cero, podemos eliminar el directorio `.git`, reiniciar el repositorio e insertar un nuevo commit vacío. Para hacerlo, usamos los siguientes comandos:

```
rm -rf .git
git init
git add .
git commit -m "Nuevo comienzo"
git remote add origin <URL-del-repositorio>
git push --force origin master
```

Este método eliminará todo el historial y comenzaremos con un repositorio limpio. Sin embargo, debemos tener en cuenta que sobrescribirá todo el historial y puede causar conflictos si hay otros colaboradores en el proyecto.

- ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un repositorio al que solo tienen acceso las personas que hemos invitado explícitamente. Los repositorios privados protegen el contenido del proyecto, manteniéndolo accesible únicamente para los colaboradores autorizados.

- ¿Cómo crear un repositorio privado en GitHub?

Para crear un repositorio privado en GitHub, primero accedemos a nuestra cuenta y hacemos clic en el ícono "+" en la esquina superior derecha, seleccionando "New repository". Luego, completamos los campos como el nombre y la descripción del repositorio, y en la sección "Visibility", elegimos la opción "Private" para que solo las personas invitadas puedan acceder. Finalmente, hacemos clic en "Create repository" para crear el repositorio privado.

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

Para invitar a alguien a un repositorio privado en GitHub, primero vamos al repositorio en cuestión. Luego, hacemos clic en la pestaña "Settings" (Configuración), en la barra de menú superior. Dentro de la configuración, seleccionamos "Manage access" (Administrar acceso) en el menú de la izquierda. Hacemos clic en "Invite a collaborator" (Invitar a un colaborador) e ingresamos el nombre de usuario o correo electrónico de la persona que queremos invitar. Después, seleccionamos el rol adecuado (como "Write", "Read", o "Admin") y finalmente hacemos clic en "Add" para enviar la invitación. La persona recibirá una notificación y podrá acceder al repositorio privado una vez acepte la invitación.

- ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un repositorio cuyo contenido está accesible para cualquier persona en la web. Esto significa que cualquier usuario de GitHub, o incluso personas que no tengan cuenta, pueden ver, clonar, bifurcar (hacer un fork) y contribuir al código si tienen permisos.

- ¿Cómo crear un repositorio público en GitHub?

Para crear un repositorio público en GitHub, primero accedemos a nuestra cuenta y hacemos clic en el ícono "+" en la esquina superior derecha, seleccionando "New repository". Luego, llenamos los campos como el nombre y la descripción del repositorio. En la sección "Visibility", seleccionamos la opción "Public" para que el repositorio sea accesible a cualquier persona. Finalmente, hacemos clic en "Create repository" para crear el repositorio público.

- ¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio público en GitHub, simplemente copiamos la URL del repositorio desde la barra de direcciones de nuestro navegador mientras estamos en la página principal del repositorio. Esta URL es pública, por lo que cualquier persona puede acceder a ella. Luego, podemos compartir esta URL a través de correo electrónico, redes sociales, o cualquier otro medio de comunicación para que otros puedan ver, clonar o contribuir al repositorio.az<<