

## Respuestas TP-2

Marcos Rios

A)

### 1) ¿Qué es GitHub?

GitHub es una comunidad donde nosotros podemos compartir nuestros repositorios de forma pública o privada. Esa cuenta se asemeja a un perfil de redes sociales, donde vamos a poder guardar los repositorios de las demás personas o incluso clonarlas, además de que también es de código abierto lo que permite el trabajo en equipo, y el intercambio de códigos.

### 2) ¿Cómo crear un repositorio en GitHub?

Para empezar, debemos de crearnos una cuenta de GitHub, lo podemos hacer en la página web de GitHub. Una vez creamos la cuenta, podemos configurar el perfil y comenzar a crear los repositorios. Para poder crear nuestro primer repositorio lo primero que hay que hacer es clickear en new, una vez allí colocar el nombre alguna descripción si lo desea, colocar si queremos que sea público o privado, etc.

### 3) ¿Cómo crear una rama en Git?

Ni bien inicialicemos un repositorio de git este nos va a crear por defecto una rama llamada master. Para crear una nueva rama, se utilizará el comando git branch: `$ git branch nuevarama`.

Git sabe en qué rama estas mediante un apuntador especial llamado head. Que hallamos creado una nueva rama no quiere decir que estemos en ella.

### 4) ¿Cómo cambiar a una rama en Git?

Para poder cambiar de una rama a otra, tenemos que utilizar el comando: git checkout nuevarama esto si nos mueve el apuntador a la rama nuevarama.

### 5) ¿Cómo fusionar ramas en Git?

Para poder fusionar ramas en git, sigue un proceso que combina los cambios de una rama a otra. Primero, debemos estar en la rama a la que queremos fusionar los cambios. Por lo general, es a la rama principal (master o main) o cualquier otra rama en la que estés integrando los cambios. Por ejemplo, supongamos que queremos fusionar la nuevarama a la rama master, entonces vamos a pararnos en la principal: `$ git checkout master` una vez en la rama de destino, usa el comando git merge para fusionar la rama de origen en la rama actual: `$ git merge nuevarama` este comando incorpora los cambios de la nuevarama en master.

#### 6) ¿Cómo crear un commit en Git?

Para poder crear un commit en git primero hay que realizar los cambios en los archivos del proyecto que deseamos guardar en el commit. Después debemos agregar los cambios al área de preparación, una vez en el área de preparación podemos hacer el commit con el comando: `git commit -m '' ''` entre las comillas podemos poner un mensaje de lo que hemos hecho, aunque esto es opcional.

#### 7) ¿Cómo enviar un commit a GitHub?

Para poder hacerlo primero debemos clonar el repositorio de github a nuestra maquina local: `git clone https://github.com/tu\_usuario/tu\_repositorio.git` cd tu repositorio. Hacemos los cambios en los archivos del repositorio y agregarlos: `git add .` creamos un commit con la descripción de los cambios entre comillas, para poder enviar los commit al repositorio de github lo debemos empujar con: `git push origin nombre de la rama`.

#### 8) ¿Qué es un repositorio remoto?

Los repositorios remotos son versiones de tu proyecto que estén hospedadas en internet o en cualquier otra red. Podemos tener varios de ellos, y en cada uno tendrás generalmente permisos de solo lectura o de lectura y escritura.

#### 9) ¿Cómo agregar un repositorio remoto a Git?

Utiliza el comando `git remote add` para vincular tu repositorio local con un repositorio remoto y asignar un nombre a ese remoto: `$ git remote add (nombre) (url)` ponerle un nombre te ayuda a referenciarlo de manera más fácil en lugar de usar URL completa. Para asegurarnos de que el remoto se haya agregado correctamente y verificar el nombre que le has dado.

#### 10) ¿Cómo empujar cambios a un repositorio remoto?

Para poder empujar los cambios al repositorio remoto: `git push origin nombre de la rama`

#### 11) ¿Cómo tirar de cambios de un repositorio remoto?

Para poder tirar los cambios del repositorio remoto se usa el comando `git pull` para descargar y fusionar los cambios del repositorio remoto con tu rama local: `git pull origin nombre de la rama`.

#### 12) ¿Qué es un fork de repositorio?

En muchas ocasiones podemos ver en github repositorios que son de nuestro agrado y donde nosotros queremos tener una copia y hacerla algunas modificaciones, para ello GitHub nos ofrece la implementación de fork.

#### 13) ¿Cómo crear un fork de un repositorio?

Para poder crear un fork, debemos clonar en alguna carpeta y en algún archivo que nosotros notamos que podría mejorarse algo, procedemos a realizar algún cambio, lo agregamos al stage y lo commiteamos (todo esto desde nuestro repositorio local) y luego mandamos los cambios a el forking que hicimos con un git push origin master.

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

Para poder hacer un pull request nos dirigiremos a la solapa de pull request allí daremos click en new pull request, veremos una ventana a modo de resumen en donde se reflejarán los cambios que hemos hecho nosotros en comparación al repositorio original.

#### 15) ¿Cómo aceptar una solicitud de extracción?

La persona que creo el repositorio podrá ver en sus pull request el mensaje que le hemos enviado, para que lo pueda observar y si lo considera realizar el cambio sugerido. Lo mejor es que si el creador original del repositorio considera como buena la modificación y no genera conflictos con la rama maestra de su repositorio local, puede clicar en merge pull request y de esa manera sumaría a su repositorio los cambios sugeridos por el usuario.

#### 16) ¿Qué es una etiqueta en Git?

Git posibilita el uso de etiquetas para poder etiquetar puntos específicos del historial como importantes. Esta función se usa para marcar versiones de lanzamiento.

#### 17) ¿Cómo crear una etiqueta en Git?

Para empezar, hay que aclarar que git utiliza 2 tipos de etiquetas: ligeras y anotadas. Una etiqueta ligera es parecida a una rama que no cambia, sin embargo, las etiquetas anotadas se guardan en la base de datos de git como objetos enteros. Normalmente se recomienda que crees etiquetas anotadas, de manera que tengas toda esta información; pero si quieres una etiqueta temporal o por alguna razón no estas interesado en esa información, entonces puedes usar las etiquetas ligeras.

#### 18) ¿Cómo enviar una etiqueta a GitHub?

Lo que hay que hacer para empezar es crear una etiqueta en nuestro repositorio local, podemos crear una etiqueta anotada (Cómo puede ser información por ej: una fecha) O una etiqueta ligera (por ej: un puntero a un commit). Una vez hallamos creado la etiqueta en el repositorio local, necesitaremos empujarla al repositorio remoto en GitHub, lo podemos hacer con el comando: `git push origin`. Y para empujar todas las etiquetas creadas podemos usar: `git push origin --tags`.

#### 19) ¿Qué es un historial de Git?

El historial en git es una secuencia de todos los cambios realizados en un repositorio de git. Cada cambio en el repositorio se guarda como un commit, y cada commit contiene información sobre el estado del proyecto en el momento específico.

#### 20) ¿Cómo ver el historial de Git?

Para poder ver el historial de git, lo podremos conseguir con el comando: `git log` con tipear este comando en el bash de git podremos apreciar el historial de commits, estando situados en la carpeta de nuestro proyecto. El listado de commits estará invertido, es decir, los últimos realizados aparecen los primeros.

#### 21) ¿Cómo buscar en el historial de git?

Para buscar en el historial de commits de git, podemos utilizar algunos comandos y opciones que te permiten filtrar y localizar commits específicos.

Para buscar un commits que contenga una palabra o frase, podemos utilizar el comando: `git log --grep="la palabra clave"`

Para buscar en un rango de fechas específicas el comando utiliza: `--since` y `--until`.

Si buscamos de acuerdo con una persona específica ponemos: `git log --author`.

#### 22) ¿Cómo borrar el historial de git?

El comando `git reset` se utiliza para deshacer las cosas, hay algunos tipos de reset:

- `Git reset --modo head` (sirve para volver al commit anterior)
- `Git reset --modo head N` (volver hacia el N° anterior commit)
- `Git reset --modo hash-commit` (volver hacia un commit específico)
- `Git revert hash-commit` (volver hacia un commit especificado y generar uno Nuevo con su estado)

### 23) ¿Qué es un repositorio privado?

Un repositorio privado en github es un tipo de repositorio en el que el contenido solo es accesible para usuarios específicos que han sido autorizados. A diferencia de los repositorios públicos, donde cualquier persona puede ver y clonar contenido, un repositorio privado limita el acceso a los colaboradores que nosotros elijamos.

### 24) ¿Cómo crear un repositorio privado en github?

Para poder crear un repositorio privado, primero debemos, iniciar sesión en github, luego debemos dirigirnos a la pagina de creación de repositorios. En la esquina superior derecha de la pagina principal, debemos hacer clic en el botón "+" y hacemos click en "new". Y luego completamos la información (nombre del repositorio y descripción, aunque esta última es opcional), y seleccionamos que sea privada.

### 25) ¿Cómo invitar a alguien a un repositorio privado en GitHub?

Primero accedemos al repositorio, luego hacemos clic en la pestaña "settings", está en la parte superior del repositorio. Seleccionamos "collaborators" en el menú de la izquierda, esto nos llevara a la página donde podemos administrar colaboradores. Una vez allí hacemos clic en el botón "add people" e ingresa el nombre de usuario de GitHub de la persona que deseamos invitar, y para finalizar presionamos el botón "add" para enviarle la invitación.

### 26) ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un proyecto cuyo contenido es accesible para cualquier persona en internet, a diferencia de un repositorio privado, que está restringido y solo un grupo de colaboradores tiene acceso a él. Un repositorio publico permite que una persona pueda ver, clonar e incluso ayudar con correcciones, que el autor luego vera si las incluye o no.

### 27) ¿Cómo crear un repositorio público en GitHub?

Para poder crear un repositorio público, primero debemos, iniciar sesión en GitHub, luego debemos dirigirnos a la página de creación de repositorios. En la esquina superior derecha de la página principal, debemos hacer clic en el botón "+" y hacemos click en "new". Y luego completamos la información (nombre del repositorio y descripción, aunque esta última es opcional), y seleccionamos que sea público.

### 28) ¿Cómo compartir un repositorio público en GitHub?

Para compartir nuestro repositorio público, debemos acceder al mismo, copiar su URL (se encuentra en un cuadro con un texto "<>code").

B) URL: <https://github.com/MarcosRios547/TP-de-git-y-github.git>

C) URL: <https://github.com/MarcosRios547/conflict-exercise.git>