**Resumen**

* **1**

¿Qué puedes hacer con gitk?

* **2**

Por defecto, GitHub usa la rama master como la rama principal. ¿Podemos cambiar la rama principal? **Verdadero, es una rama mas**

* **3**

¿Cómo llaman GitHub y GitLab a los "merges" (propuestas de cambios desde otra rama o repositorio)?

* **4**

Cuando hago un *fork* de un proyecto en GitHub, ¿lo que logro es? Un fork significa bifurcación, y en programación supone copiar un proyecto y partir de este,  hacerle modificaciones. Cuando trabajamos con repositorios Git, supone hacer una copia exacta del proyecto, generando dos URL distintas.

* **5**

Si usas:

git config --global alias.platzi "shortlog"

¿Cómo invocar ese comando? Git Platzi

* **6**

Cuando usas git stash, los cambios que “guardas” temporalmente se guardan en:

* **7**

¿Qué crean los tags en Git? En Git, una etiqueta o tag sirve básicamente como una rama firmada que **no permuta**, es decir, siempre se mantiene inalterable. Sencillamente es una cadena arbitraria que **apunta a un commit** específico. Puede decirse que un tag es un nombre que puedes usar para marcar un **punto específico** en la historia de un repositorio.

* **8**

Para transmitir cambios seguros entre tu repositorio local y GitHub, ¿qué se recomienda?

* **9**

Cuando hay un conflicto entre archivos lo mejor es:

* **10**

¿La terminal ideal para usar Git en Windows es?

* **11**

¿Con amend puedo?

* **12**

¿Para qué sirve GitHub Pages?

* **13**

Se crean dos archivos cuando creas una llave pública y privada para SSH, ¿cuál es la extensión de la llave privada?

* **14**

¿El HEAD en una rama es? El concepto de **HEAD** es muy simple: se refiere al commit en el que está tu repositorio posicionado en cada momento. Por regla general **HEAD** suele coincidir con el último commit de la rama en la que estés, ya que habitualmente estás trabajando en lo último.

* **15**

Con checkout puedo:

* **16**

En un repositorio público en GitHub, ¿qué puede hacer los colaboradores?

* **17**

¿Para qué sirve git grep?

* **18**

¿Cuál es la diferencia entre git rm y git reset HEAD?

* **19**

¿Las llaves públicas son?

* **20**

¿Qué guarda Git?

* **21**

Estás en la rama master y quieres mandar los últimos cambios que guardaste con git stash a la rama retomando-los-cambios (la rama no ha sido creada aún). ¿Cuál de las siguientes soluciones es **incorrecta**?

* **22**

¿Puedes conectar tu repositorio local con más de un repositorio remoto?

* **23**

¿GitHub es?

* **24**

Un repositorio remoto y un repositorio local:

* **25**

¿Cómo instalas Gitk?

* **26**

El dueño de un repositorio al ver un pull request puede:

* **27**

¿Cuándo debería hacer un merge?

* **28**

Los cambios a un archivo que están en “staging” están en:

* **29**

Si en el .gitignore agrego esto: images/\*.js. ¿Cuál de los siguientes archivos sería ignorado?

* **30**

1. Si ya hiciste el desafío de “Hazme un pull request,” ¿dentro de cuál etiqueta tenías que agregar el cambio? Dentro del ID “post” luego de “suscribete y dale like” agrega otra línea o párrafo con tu nombre o tu nombre de usuario en Platzi.

¡Suerte! Y #NuncaParesDeAprender

**Resumen**

* **1**

¿Qué puedes hacer con gitk?

* **2**

En un commit, los mensajes del commit son:

* **3**

Por defecto, GitHub usa la rama master como la rama principal. ¿Podemos cambiar la rama principal?

* **4**

¿Cómo llaman GitHub y GitLab a los "merges" (propuestas de cambios desde otra rama o repositorio)?

* **5**

Es mejor aprender a manejar Git con la terminal antes de hacerlo con herramientas visuales como Gitk porque:

* **6**

Cuando hago un *fork* de un proyecto en GitHub, ¿lo que logro es?

* **7**

¿Qué crean los tags en Git?

* **8**

Para transmitir cambios seguros entre tu repositorio local y GitHub, ¿qué se recomienda?

* **9**

Cuando hay un conflicto entre archivos lo mejor es:

* **10**

¿Con amend puedo?

* **11**

¿Para qué sirve GitHub Pages?

* **12**

Se crean dos archivos cuando creas una llave pública y privada para SSH, ¿cuál es la extensión de la llave privada?

* **13**

¿El HEAD en una rama es?

* **14**

Con checkout puedo:

* **15**

¿Cuándo deberías usar cherry-pick?

* **16**

¿Para qué sirve git grep?

* **17**

¿Cuál es la diferencia entre git rm y git reset HEAD?

* **18**

Un pull request es:

* **19**

¿Las llaves públicas son?

* **20**

¿Qué guarda Git?

* **21**

Estás en la rama master y quieres mandar los últimos cambios que guardaste con git stash a la rama retomando-los-cambios (la rama no ha sido creada aún). ¿Cuál de las siguientes soluciones es **incorrecta**?

* **22**

¿Puedes conectar tu repositorio local con más de un repositorio remoto?

* **23**

Un repositorio remoto y un repositorio local:

* **24**

¿Cómo instalas Gitk?

* **25**

Para que dos personas trabajen en paralelo sobre el mismo archivo se recomienda:

* **26**

El dueño de un repositorio al ver un pull request puede:

* **27**

¿Cuándo debería hacer un merge?

* **28**

Los cambios a un archivo que están en “staging” están en:

* **29**

Si en el .gitignore agrego esto: images/\*.js. ¿Cuál de los siguientes archivos sería ignorado?

* **30**

Si ya hiciste el desafío de “Hazme un pull request,” ¿dentro de cuál etiqueta tenías que agregar el cambio?